



# The future of lab submission systems at Chalmers

An overview of implementation models and design

Bachelor thesis

Martin Ahlberger, Johan Berndtsson, Gustav Nilsson,  
Kamil Miller, Magnus Rönnerberg, Edvin Tobiasson

Department of Computer Science and Engineering  
Chalmers University of Technology  
University of Gothenburg  
Gothenburg, Sweden, 2018

# Abstract

Chalmers is about to change learning management system from PingPong to Canvas. The department of computer science and engineering are uncertain about the future of the current lab submission system, FIRE. There are different solutions that lets Canvas work as a lab submission system. This thesis tries to examine which of the solutions that will work best.

The result will be in the form of four solution propositions. These solutions propositions were made by following the Goal-Directed Design Process, stated by Alan Cooper. In the end a recommended solution proposition is presented which, based on the information gathered in this study, suit Chalmers' needs best.

# Sammanfattning

Chalmers kommer byta lärplattform från PingPong till Canvas. Institutionen för data- och informationsteknik är osäkra på vad som ska hända med det nuvarande labbhanteringssystemet, FIRE. Det finns olika lösningar som låter Canvas användas som ett labbhanteringssystem. Denna rapport försöker reda ut vilka lösningar som fungerar bäst.

Resultatet presenteras som fyra lösningsförslag. Dessa lösningsförslag är framtagna genom att följa Goal-Directed Design Process, definierad av Alan Cooper. I slutet presenteras ett rekommenderat lösningsförslag som, baserat på informationen framtagen i denna rapport, passar Chalmers behov bäst.

# Acknowledgements

We would like to thank our supervisor, Marco Fratarcangeli, for the help he has provided during this thesis. We would also like to thank the course responsible staff that agreed to help us by being interviewed and fill out our survey.

# Contents

<b>Terminology</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Aim . . . . .	1
1.3 Literature . . . . .	1
1.4 Problem/Task . . . . .	2
1.5 Scope . . . . .	2
1.6 Achieved Results . . . . .	3
<b>2 Theory</b>	<b>4</b>
2.1 User Experience . . . . .	4
2.2 Interaction Design . . . . .	5
2.3 Persona . . . . .	5
2.4 Goal-Directed Design Process . . . . .	6
2.4.1 Research: User and domain . . . . .	7
2.4.2 Modeling: Users and context of use . . . . .	7
2.4.3 Defining requirements . . . . .	8
2.4.4 Defining a framework of design structure and flow . . . . .	9
2.4.5 Refinement of framework . . . . .	10
<b>3 Method</b>	<b>11</b>
3.1 Workflow . . . . .	11
3.2 Research . . . . .	12
3.2.1 Domain . . . . .	12
3.2.2 Users . . . . .	13
3.3 Data Analysis . . . . .	14
3.3.1 Personas . . . . .	14
3.3.2 Requirements . . . . .	17
3.4 Prototypes . . . . .	19
3.4.1 Based on existing system . . . . .	20
3.4.2 Based on Custom Design Concept . . . . .	20
<b>4 Current State</b>	<b>22</b>
4.1 PingPong . . . . .	22
4.2 FIRE . . . . .	22
4.2.1 Usage . . . . .	22
4.2.2 Development . . . . .	23
4.3 Canvas . . . . .	23
4.3.1 Functionality . . . . .	24
4.3.2 Extending Canvas . . . . .	24

4.4	The users . . . . .	25
<b>5</b>	<b>Design Initiative</b>	<b>27</b>
5.1	Problem Statement in the Goal-Directed Design Process . . . . .	27
5.2	Vision statement in the Goal-Directed Design Process . . . . .	28
5.3	Matching the problem with the vision . . . . .	28
5.4	Requirements . . . . .	29
<b>6</b>	<b>Solution Propositions</b>	<b>31</b>
6.1	Using Canvas out of the box . . . . .	31
6.2	Canvas with custom made LTI extension . . . . .	32
6.3	Integrating FIRE into Canvas . . . . .	33
6.4	New, standalone submission app integrated in Canvas . . . . .	34
<b>7</b>	<b>Discussion</b>	<b>36</b>
7.1	Problem with data collection / Method . . . . .	36
7.1.1	Survey Review . . . . .	36
7.1.2	Interview Review . . . . .	37
7.2	Credibility Analysis . . . . .	38
7.3	Method Evaluation . . . . .	38
7.4	Discussion about solutions . . . . .	39
<b>8</b>	<b>Recommendations</b>	<b>41</b>
8.1	Canvas with dedicated LTI application . . . . .	41
8.2	Using Canvas out of the box . . . . .	41
	<b>References</b>	<b>43</b>
<b>A</b>	<b>Survey</b>	<b>44</b>
A.1	Original . . . . .	44
A.2	Improved . . . . .	45
<b>B</b>	<b>Persona - Partial Results</b>	<b>46</b>
B.1	Range Charts . . . . .	46
B.2	Common Variables in Groups . . . . .	48
B.3	Group Variables . . . . .	49
B.4	Assigned Persona Types . . . . .	50
<b>C</b>	<b>Personas</b>	<b>51</b>
C.1	Made using Quick Guide . . . . .	51
C.2	Made using GDDP . . . . .	53
<b>D</b>	<b>Requirements - Partial Results</b>	<b>61</b>
D.1	Persona Expectations . . . . .	61
D.2	Context Scenarios . . . . .	61
<b>E</b>	<b>Sketches</b>	<b>67</b>

<b>F Custom Design Concept</b>	<b>70</b>
F.1 Navigation . . . . .	70
F.2 Overview View . . . . .	73
F.3 Submissions View . . . . .	76
F.4 Settings View . . . . .	78
F.5 Grading View . . . . .	80
F.6 About the implementation . . . . .	83
<b>G Canvas Tutorial</b>	<b>84</b>

# Terminology

**FIRE** - Internal system currently in use at the Department of Computer Science and Engineering at Chalmers. An application that offers functionality for submitting and grading lab assignments as well as rich functionality in administering teaching assistants.

**LMS** - Learning Management System, a system meant to contain all online activities for a course.

**Canvas** - An LMS that will be up and running at Chalmers in the near future.

**PingPong** - The current LMS in use at Chalmers.

**LTI** - Learning Tools Interoperability. A standard designed to enable plugin software to communicate with any LMS that implements it. Thus providing a platform where plugins can be written that is not coupled to a certain LMS.



# 1 Introduction

## 1.1 Background

Canvas will replace PingPong as the learning management system at Chalmers. This will introduce changes for all teachers, students and teaching assistants at the university. This change will also introduce a choice to be made for staff at the Department of Computer Science and Engineering.

Many faculty members at the department has for a long time been using the FIRE system to handle submissions in their courses because they have not been comfortable with PingPong with its complicated structure: overwhelming teachers when setting up courses. These professors must now make the decision if they want to migrate over to Canvas or keep using FIRE.

For the department there are several benefits if people migrate to the new system. It will reduce the cost for Chalmers when it comes to maintain several separate systems, present a unified interface to all users as well as offer an improved user experience by providing all online activities from one system.

The decision to migrate needs to be considered carefully as Canvas might not support the workflow that is optimal for each teacher and the benefits of using only Canvas is not apparent at first glance.

## 1.2 Aim

The aim for this report is to provide a good foundation of knowledge about the different ways to provide a lab submission system at the department in the future. This is useful for teachers, future developers as well as the department at Chalmers.

Functionalities requested by the course responsible staff and how to best implement them will also be examined.

## 1.3 Literature

The theory of Goal-Directed Design Process (GDDP) will be followed because it is stated as one of the primarily used processes in the industry. The process is defined by Alan Cooper and is described in the book About Face. This book will be used in this project as the primarily theory source for this study.

A second motivation for choosing this book is to avoid spending too much time comparing different work methods. It was also a source that had been used in prior courses as such gained further credibility. The book was also recommended by teachers at Chalmers.

The process of GDDP is chosen because it claims to lead to user friendly software, focus on user needs and provides a repeatable process. It is also easy to follow and is recommended by a lot of sources online [1, 2, 3, 4].

## 1.4 Problem/Task

The problem is, in general, to find out what an optimal lab submission system would look like for the Department of Computer Science and Engineering.

- What functionalities are the course staff at the department expecting/requiring?
- What of that functionality already exists in Canvas?
- How would any additionally needed functionality best be integrated in Canvas?

## 1.5 Scope

The focus of the project is on the administrative part of handling submissions in mainly programming related courses. This delimitation was given by the stakeholder and proposer of this study.

Since the focus is on administration, the main user focus is on course responsible staff. As such, the students were not considered unless it directly related to course staff goals. The staff of focus is from the Department of Computer Science and Engineering since that is where FIRE is mostly used.

Implementation details of proposed systems are not part of this study, as focus is not on implementation. An overview of what is needed to be achieved is given. An implementation will be made only as a proof of concept where it is shown that necessary data can be fetched from the Canvas API.

The theory of GDDP suggests testing of the design proposals. These test should be applied by letting real end users try the designs and these tests were seen as too heavy to perform and was accomplished with more lightweight design validation (Key path scenarios, Validation scenarios) [5].

PingPong and its user base has not been studied since the system has seen very sporadic use at the department and most teachers have no real experience using it. Furthermore, it is decided that PingPong will be replaced, so it is not considered in this study.

## 1.6 Achieved Results

This thesis is made as a pre-study of how to handle lab submission system on Chalmers in the future. Therefore the report itself is the main result. The following is a list of what was achieved and is described in this report.

- Performed a study using the state of the art method within interaction design
- Performed interviews with the course responsible staff at the Department of Computer Science and Engineering
- Generated a list of required functionality in a lab submission system according to the interviewed staff
- Created several solution propositions for a future lab submission system at Chalmers
- Created a tutorial for using Canvas out of the box as a lab submissions system
- Created a complete design concept for a customized extension application for Canvas

## 2 Theory

*This chapter describes the theory that lies at the foundation of the study. First is a definition of user experience, followed by an explanation of interaction design. Presented subsequently are the Goal-Directed Design Process and a definition of proof of concept. Finally, there is a description of personas. Offered in all sections are an explanation or motivation on why the theory exists and its use, along with an understanding of why this theory is relevant for the study.*

### 2.1 User Experience

User experience is a central notion for interaction design, with several definitions of different nuances available from a wide range of sources. One of these definitions says that user experience should be something that meets the exact needs of the customer while being simple and even enjoyable to use [6].

Another definition, given by the Oxford dictionary, says that user experience should be something that encompasses the overall experience that a person using a product has [7].

A third definition says that user experience means how a product behaves and its usage by people. More precisely that every and any product that is used by someone, ranging from newspapers to services offered by a company, has a user experience. In particular, it is also about how people feel about the product, and the pleasure and satisfaction they gain from using it [8].

So as shown, user experience has lots of different definitions, but most of these only differ in how abstract they are. Moreover, in this study, the third definition will be used. It is the one with the right amount of abstraction while at the same time not being too vague. Also, it is indeed a more well-rounded definition, as it is a reformulation of the first definition [6, 8].

A further point to make is that one cannot design user experience, only for evoking a particular user experience. Meaning designers, cannot design for a stimulating experience. Instead, designers create design features that induce it [8].

Why the user experience notion is so important then, is because it gives a general concept to keep in mind while designing products. Without it, product development would spiral into a cluster of ideas with the only motivation of doing them for the sake of doing them, regardless of whether they are useful or even wanted by the products' intended users. More practically, caring about user experience in a software engineering context means that we produce software that is intended to attract users with well-defined interactions [5].

”It is not enough that we build products that function, that are understandable and usable, we also need to build joy and excitement, pleasure and fun, and yes, beauty to people’s lives” - Donald Norman [9].

## 2.2 Interaction Design

Interaction design is a new field when compared to others, such as System Architecture. In particular, it is a less disciplined field. In brief terms, interaction design is ”the practice of designing interactive digital products, environments, systems and services” [5]. Additionally, interaction design is ”designing interactive products to support the way people communicate and interact in their every day and working lives.” [8]. At its core, interaction design is a way of reasoning and developing system behavior and its design. Interaction design is also not about focusing on the aesthetic choices made for one’s product, but instead make decisions based on user understanding and cognitive principles [5].

The main aim of interaction design is to enhance the positive aspects of the user experience as much as possible while reducing the negative ones. In this context, the negative elements can include frustration and annoyance, while the positive aspects include enjoyment and engagement. In more abstract terms, the view of interaction design is one where it is the fundamental basis for the different fields that concern themselves with research and design of computer-based systems for people [8]. Interaction design is as much about the aesthetic choices one makes for the design, as it is about understanding the users and cognitive principles [5].

The need for a basis such as interaction design stems from the fact that many products are inefficient from an interactive point of view [5]. This inefficiency follows from industry practices where developers remain ignorant about the users, choosing to prioritize system construction over user needs. Interaction design methodologies are created to reduce these deficiencies in the development process [8].

There are five component activities defined within interaction design. The activities identified are the following: Understanding, Abstracting, Structuring, Representing, Detailing [10, 5].

”Interaction Design is not guesswork” - Cooper [5].

## 2.3 Persona

One of the core concepts used in interaction designs is a type of modeling called personas. The goal-directed design process (see chapter 2.4) itself puts much emphasis on defining personas. It is of importance to provide a suitable definition of what personas are and why they are used.

In the most straightforward definition, a persona is a "fictional, yet realistic, description of a typical or target user" [11]. The primary purpose of a persona is memorability, to make team members on a project remember the users for whom they are building the product. As such, personas should not have too many unnecessary details [11].

Personas should also act as an aid, giving a few sets of users to refer to when making design decisions about the system, instead of trying to make those decisions based on the potentially extensive data collected about real users. Questions like "Is this feature relevant?" are much easier to answer when designing for a specific user, i.e. a specific persona [5]

Personas come in several types, three of which are of importance in this study [5]:

- **The primary persona** represents the user the system should be designed for, allowing the design decisions to base on the opinion of a single (fictional) person that is an amalgamation of several interview subjects [5].
- **The secondary persona** represents a vastly different user which would still be reasonably satisfied by a system designed for the primary persona. Any extra needs the secondary persona might have must not be implemented in such a way that it interferes with the needs of the primary persona [5].
- **The supplemental persona** is not a regular user of the system, but with an important role that needs to be considered as well in the design process [5].

## 2.4 Goal-Directed Design Process

GDDP was initially developed by Alan Cooper, a veteran within the interaction design community [5, 12]. It was defined to be a way of working towards Goal-Directed Design, a methodology where products should be designed and constructed such that the users can achieve their goals [5]. The aim of GDDP is to provide a sequence of steps such that the purpose of Goal-Directed Design is fulfilled. At its foundation lie the five component activities of interaction design defined earlier (see chapter 2.2).

It was defined because the industry did not have reliable and/or complete processes for developing successful products. Typically programmers have to make design decisions which often boil down to ease of use or ease of coding. It is not hard to see which choice will have priority, given that programmers are often under extreme pressure to deliver. Essentially this means that any and all designers should broaden their horizons and take part in the user research [5].

The process itself can be divided into six phases (figure 2.1). In particular the process in its entirety is a more concrete specification of the activities specified by [10]. The difference and strength of GDDP is that there is a greater emphasis on modelling user behaviour and defining behaviour of systems [5]. The sixth phase

will be omitted, due to that it needs a separation between designers and developers in order to be performed, something which is outside of the scope of this study.

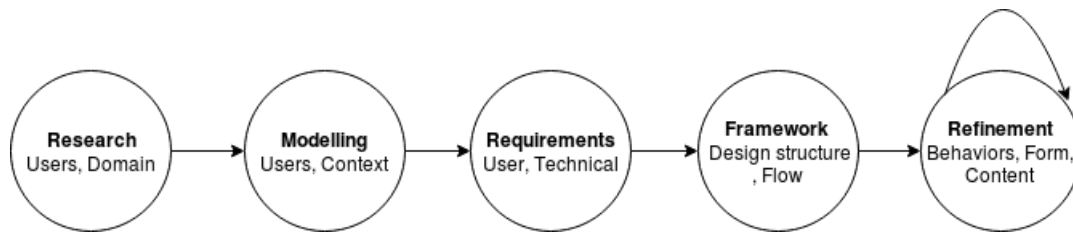


Figure 2.1: Interpretation of The Goal-Directed Design Process where, for this study, the sixth phase were chosen to be omitted. [5].

### 2.4.1 Research: User and domain

In order to improve a system or develop a new one, data must first be collected. The main aim of this step is to get a good foundation of knowledge upon what will be deemed necessary to fulfill the aim can be built [8].

The data itself will come in many different forms, but more importantly from a wide range of different sources. It is as such important to employ valid activities for collecting data relative to the source and to ensure that qualitative research is conducted. The sources themselves can be people, literature, documentation, websites or even other systems [5].

The activities that are the most appropriate for GDDP include Stakeholder interviews, Subject Matter Expert interviews, user interviews, user observation/field studies, literature reviews and product/prototype audits [5].

The strength of user and domain research comes from its ability to distribute the activities such that a lot of them can be made in parallel. For example literature reviews could be made at the same time as user interviews without any particular loss of data in between [5].

### 2.4.2 Modeling: Users and context of use

With data at hand, there is a need for molding and refining it into something that is useful. The need for this step comes from the principle of abstraction as defined by the core interaction design activities [10]. Meaning that, among other things, the motivations of the user needs to be derived, even if they themselves can not put it into words [5]. For capturing these, there is a useful concept of a user model, called personas.

Personas are a tool for specifying the more abstract things such as user behaviors or mindsets. For a persona to be truly useful, a considerable amount of work needs to be put down. Of significance is identifying the patterns in user behavior that are of meaning, instead of making decisions based on stereotypes or generalizations [5].

Once the personas are created, it is time to define the goals. A goal can imply a range of different things. For this study, two definitions are of focus. The first one defines these as the drivers behind the behaviors that personas provide the context of use for [5]. Another definition merely is that goals are what the persona wants or needs to fulfill, a motivation on why the persona is doing some action [8]. In figure 2.2 a brief overview of the relationship between personas and goals is presented, as well as a quick introduction to scenarios (see chapter 2.4.3).

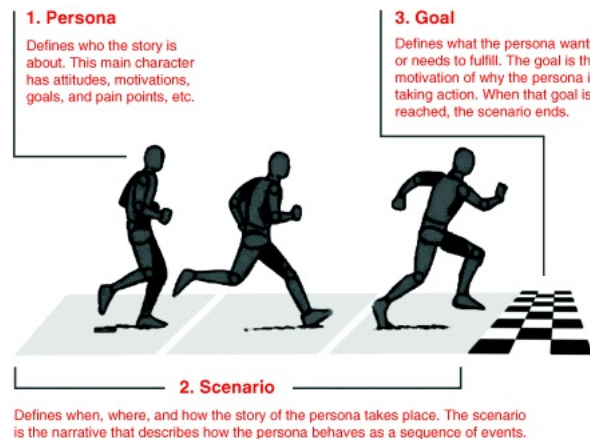


Figure 2.2: Relationship between personas, goals and scenarios [8].

### 2.4.3 Defining requirements

The requirements definition phase is where the connection between a user (or any other model) and the design concept is made. For generating these requirements, an analysis based on context scenarios is performed [5].

Context scenarios are a high-level description of how a product can best serve the needs of personas. They are stories describing how the personas would achieve their goals in an optimal system. The gain from developing these scenarios is that it gives a designer the power to imagine a user experience that is ideal. These scenarios have the side effect of getting the designers to start thinking about flows within the system under design and get the creative process started [5, 8].

From these context scenarios, a list of requirements is extracted. Requirements come in several forms, but for this study, only data and functional requirements are in focus. Data requirements are needs that demand representations in the system of objects and information. Functional requirements, on the other hand, are needs that facilitate functions that act upon the data in the system [5].

Given an user-centered approach, this activity as a whole is one of the most fundamental within interaction design. It requires one to get to know one's users, in aspects more than just knowing what they do, but also the motivations driving them [8]. Another detail defined here is also the problem and vision statement, a short document describing what the problems are and the counterpart on what an envisioned solution might entail [5].



## 2.4.4 Defining a framework of design structure and flow

The primary focus is to define the product concept, or in other words the framework. This concept includes the fundamental design decisions regarding the behavior and visual design of the product. It is worth to note that in this step, the focus is on the user interface structure and any behaviors, rather than exact positioning and measurements. The aim is to stay at a relatively high level of description, rather than hammering out all the gritty details. The gain of following these steps is that one can get started reasonably easy and instead deepen the grittier details through iteration. One can decide things in rough terms without getting too stuck in a particular design concept [5].

In more concrete terms, what needs to be defined:

- **Form factor**, essentially the size and shape of the product
- **Posture**, is a way of specifying the range of attention the product should get
- **Input methods**, describe what ways of interacting with the product exist
- **Data elements**, fundamentals of a product. Examples of these are photos, email messages and product items in a webshop
- **Functional elements**, operations performed on the data elements and any other representation in the interface
- **Functional groups and the hierarchy**, groupings of the functional and data elements, along with any hierarchy that can be defined

Once these details have been defined, the visual parts of the interface can be sketched. Here different techniques can be used to produce designs. Among these are Lo-fi and Hi-fi prototypes, which are merely different ranges of detail on the drawn interface [5].

In the end, two kinds of scenario types are defined, key path scenarios and validation scenarios. These key path scenarios describe how a persona interacts with the product. These scenarios are the concrete versions of the context scenarios defined earlier. Noteworthy is that the language used in the key path scenario should match the one that described in the interaction design concept. The validation scenarios, on the other hand, are less detailed, and typically consist of asking "what if..." questions. The value of doing this is to try and truly test all possible interactions, smoothing over any potential holes within the design [5].

In parallel, a visual design should be defined. Essentially these boil down to the style or styles used in a design. Of importance to keep in mind here are concepts such as how different colors interact with each other, font size, material properties of the interface. In situations where the design is an upgrade or extension of some existing system or product, it is also possible to use the already established style, as to not break the overall appearance of the system. This design concept is typically called the visual style guide. [5].

## 2.4.5 Refinement of framework

Having defined an acceptable design concept, the process enters the Refinement phase in which the final design is formed through iteration. Extra emphasis is made to follow defined interaction design patterns. These patterns are general solutions to previously analyzed problems, tried and tested by the industry rigorously [5]. In other terms, they are descriptions of what the best practices are within some design domain. It is worth to note that these patterns are not step-by-step guidelines on how to design some specific interface. Instead these patterns are suggestions to solve specific problems that can arise in the design process [13].

The primary activity performed in the refinement phase is to translate any sketches, that is to say, the Lo-fi / Hi-fi prototypes into full-screen depictions of the user interface. Note that no work is done to implement these interfaces. Staying at a high level gives the potential for recognizing problems that can still be squashed. Once the design is fully ironed out, only then are the designs mechanically translated into an implementation. The rest of the phase follows the same structure as defining a design concept (see chapter 2.4.4), with the difference that changing the form factor and input methods are unnecessary and will be skipped [5]. It is of importance to conform to the visual style guide defined in earlier steps (see chapter 2.4.4).

During the end of this phase, or after each iteration, the prototype or design concept forms a concrete solution. It is thus desirable to evaluate the solution or solutions produced. This evaluation can be done in a range of different ways, categorized in two overarching groups. Testing against the defined personas and scenarios or user testing. The former is mostly a way of stepping through our scenarios and checking the requirements to see if the solution conforms to the higher level specifications made [5]. The latter uses concepts like usability testing. In this form of test, a user is given a set of tasks which they should try to complete. During this test different types of data is collected such as time to complete a task, number and type of errors per task, number of users completing a task successfully. These types of data (along with others) form a measure of usability [8].

There also exists a concept of walkthroughs, which can be performed on designers and users alike, provided that the subjects walked through are new, meaning they have had little to none previous exposure to the solution. The walkthrough itself is similar to usability testing, namely that the subject walks through a task or series of tasks. A variant of these is called cognitive walkthrough, in which effort is made to answer a trio of questions [8]:

- Does the user know what to do to achieve the task?
- Are the correct buttons/menu items visible and apparent?
- Does the user get accurate feedback for some action?

## 3 Method

*Within this chapter the process and how the work was conducted is described. It is starting with how the work was structured and then sequentially going more in-depth for each phase of the study. It starts by going through how research was conducted within the different relevant fields, then transitioning in to how the research data was analyzed and evaluated. Finally it is described how the research and data was used to create prototypes and designs.*

### 3.1 Workflow

The way this study has been approached is tightly coupled to GDDP (see chapter 2.4). In figure 3.1 a diagram of the workflow is presented. During the start up weeks, a preliminary investigation was conducted and a planning document was written.

With the planning finished, the project entered the first phase of the GDDP. Here three activities were performed in parallel. Firstly domain research, in which data was gathered regarding the domain. Secondly interviews, which were performed with the current users of FIRE, stakeholders and potential domain experts. Lastly user research using alternative data gathering methods, such as surveys/questionnaires. In parallel, a mid presentation was held where the preliminary findings were presented along with the planning.

With the data gathered the study entered the second phase, user modelling. In this step, personas and their goals were defined.

Once that was done, the third phase started. Here work was done to define the requirements and context scenarios. These were based on the previously defined personas and their goals.

The fourth phase encompassed defining the framework, the initial abstract design properties. In the fifth and final phase of the chosen instance of GDDP work was done to refine the framework into a solution proposition. In parallel with the fourth and fifth phase, some emphasis was put on trying to provide a proof of concept and prototyping with the available technologies.

In parallel with the different phases, report writing was performed. Finally, a conclusion was reached with the help of the report. This conclusion was worked into the final presentation.

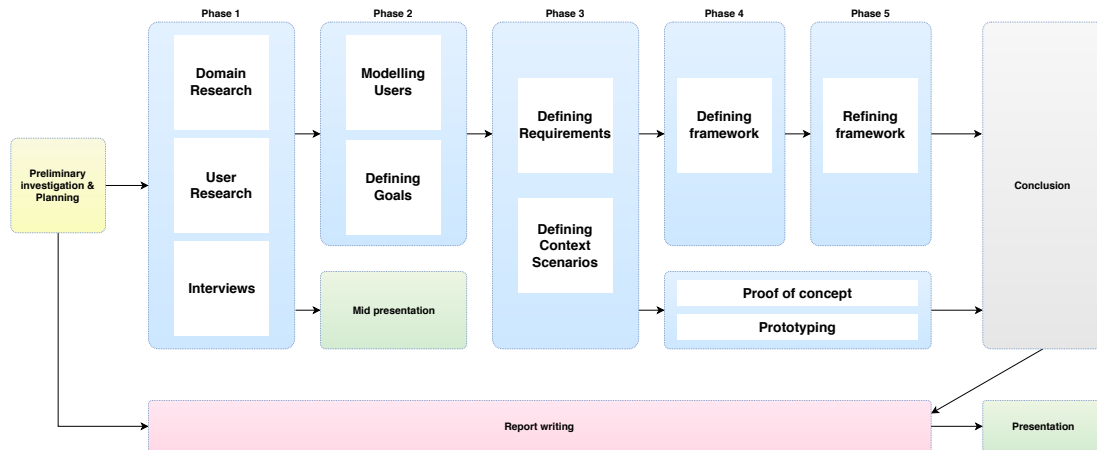


Figure 3.1: Diagram over the workflow.

## 3.2 Research

During the research phase of the study, information and knowledge was acquired on relevant topics. This ranged from information on specific systems to the fundamentals of interaction design. This was done primarily by reading in the case of interaction design, and exploring the actual systems in order to learn about them in addition to studying the documentation. Of particular note are the books About Face [5] and Interaction Design: Beyond human-computer interaction [8] that for the most part laid the foundation of the workflow for this study.

Another major part of the research was the users and the user analysis. Conducting interviews and surveys generated the data that was the backbone of this study. For research purposes this meant that a clear methodology on how to conduct the gathering and handling of the data was needed. The research phase generated a deeper understanding of the problem, knowledge about specific systems and methods and data consisting of interview and answers to the survey.

### 3.2.1 Domain

As a starting point it was important to learn what functionality Canvas offers out of the box and learn about the existing extensions. Through constructing informal use cases based on experience of old courses taken at the department, the functionality was explored. This served the purpose of examining how to accomplish certain tasks in Canvas. For example:

- Creating a course
- Creating and modifying assignments
- Find submissions made by students, that are not graded yet.
- Grade a submission sent in by a student/group.

- Distribute workload between members of the grading staff.
- Send messages to different students and student groups.

The primary purpose of this step was to learn what was possible in Canvas. This functionality was later checked against the requirements that was formulated as part of the requirements step. This to see if all requirements could be met without any implementation efforts.

For the technical side of extending Canvas, a lot of documentation was read. Also some very small prototypes of LTI-applications were made to make sure the documentation had been understood correctly. As a part to see whether there is an existing LTI extension, an investigation was made in the different LTI categories to see what exists and have the same functionality as FIRE.

An investigation of the FIRE system, as it exists today, was also part of the domain research. The investigation was made mostly to get to know FIRE better and to have a better understanding of the current workflow as the interviews where conducted.

### **3.2.2 Users**

The most important source of information when working with user experience and interaction design are the intended users of the system. In this case, there is a somewhat homogeneous user base consisting exclusively of faculty members at the Department of Computer Science and Engineering at Chalmers. This in theory would mean that the data gathering would yield somewhat cohesive and uniform results.

#### **Surveys**

To start collecting data, a survey was produced (see Appendix A). This was done by evaluating what information was needed and then designing questions that would provide those answers. An effort was made to design questions that would produce more elaborate answers than just for example yes or no. When the survey was deemed passable it was sent out to various individuals that uses or have used FIRE. These individuals where mostly faculty members at the Department of Computer Science and Engineering or connected to the department in some way. The survey also served as the baseline to build the interviews on and it contained a question asking if someone answering were interested in participating in an interview. This inquiry is what the interview selection was later based on.

#### **Interviews**

A template of questions to be asked during the user interviews was formed. This template included the same questions as in the survey in order to get more elaborate

answers to them. It also included several new questions, including questions seeking feedback about different ideas for a future design.

Before each interview, this template was copied and extended with questions related to the answers by this specific individual in the survey. This resulted in a final set of questions for the interview.

The interviews followed a semi-structured format which entailed that follow-up questions were allowed, as well as elaborating on topics that were discovered during the interview itself. This meant that the interview subjects were free to elaborate on the answers they had already provided and, if needed, answer any follow-up questions. This format also allowed the interviewee a possibility to expand on ideas and concepts that the prepared questions did not deal with.

The interviews were booked in unison with the interview subject and conducted with two interviewers, one to ask questions and the other to take notes. In addition to the notes, the interviews were also recorded and later transcribed for review, in order to prevent any loss of information.

### **3.3 Data Analysis**

In order to have a clear way to make design decisions, the collected data was analyzed in several sequential steps, as defined in the Goal-Oriented Design process. The effort is mainly to group different users and create an abstraction to work with, instead of having to go back all the way to user interviews every time a decision needs to be made.

#### **3.3.1 Personas**

It is difficult to make decisions about the design of a system when there are multiple users and opinions to take into account, since they differ and will ask for different solutions to the same problem. Therefore, the data collected about the users was distilled into a set of personas. Each persona represents a fictional person, based on the personalities and behaviours of the users analyzed.

First of all, two general personas were generated using the survey data and a quick guide [14] just to get a feeling for the users, and get acquainted with personas as a tool. The personas resulting from that process is available in Appendix C.1.

When they had been completed, the process of creating personas following the steps from GDDP started. The data used to generate those personas was mainly from the interviews, since they provided the most in-depth data about each individual user. Having the most information about each user is important to get the most detailed view in which the users differentiate, allowing them to be properly merged. The process to generate the personas using GDDP was as follows:

## Roles and Variables

The five acquired interview objects were all teachers, which meant that no grouping of them were needed. Instead, they can all be analyzed as a single group.

By reading through the interviews, a list was assembled of every point where they had a difference in opinion, or ways of working. This list of items, called behavioural variables, represents the information that makes the users different from each other.

*Five different types of variables were assembled:*

- Activities, what they do
- Attitudes, what they think
- Aptitudes, their ability to learn
- Motivations, what they want
- Skills, what they can or know

The activity variables simply state a single activity that can be done in different amounts, while the other variables consists of two extremes that are the opposites of each other. A total of 35 behavioural variables were collected. They are available as headlines for the charts in Appendix B.1.

## Mapping variables

To get a picture of who each user is, a value was assigned to them for each behaviour variable based on the interviews. This gives each user a set of values that completely describes them in relation to the other users.

The values chosen were 1-5. In the activity variables, a one means that the subject never does the activity, while a five means that it does it a lot. As for the variables that has two opposing ends of the spectrum, a one represents a complete fit of the left end, and a five a complete fit with the right end. In binary variables where it either fits or not, a five was assigned as yes, and one as no.

The five steps of granularity in assigning values was chosen because it allowed for a middle point for neutral values, as well as a point between the middle and the extreme, allowing to see a difference between those with the strongest association, and those who are just leaning in that direction.

This resulted in a spreadsheet, with the users as rows, the variables as columns, and their corresponding assigned value in the cells. The table holds all information relevant to the interview subjects usage of the system, but it was hard to get an overview of how different people relate to each other in different questions. Therefore, based on this table, range charts were created.

For each variable, the different subjects were placed along a line, with their placement corresponding to their value assigned on that variable. Furthest to the left

represent a one, and furthest to the right represents a five. Subjects with the same value were placed next to each other to indicate a grouping around that point. The complete generated range charts can be found in Appendix B.1.

## **Identify Patterns**

With the range charts available, patterns were searched for in regards to the groupings of the users. To find which users had most correlation in their placement with each other (indicating that they are a similar type of person), a count of how many variables each grouping had in common was performed. Having the same value as a different set of users in as many variables as possible indicates a close relationship.

The range chart was passed through looking for every occurring grouping of users. Each variable in the range chart was then analyzed, noting in each group if they had a common value in that variable. This gave a table of all common variables in each of the groups.

The more common variables a single group has, the more it indicates they are similar as users. The table was then analyzed to see which groups stood out in regards to the number of common variables, indicating a larger correlation between those users. A line was drawn where only the groups over a certain number of common variables were deemed interesting, and picked for further investigation. The complete table as well as the line drawn is available in Appendix B.2.

The resulting set of groups were then analyzed for redundancy. Groups that were already part of a larger group were removed as they provided no extra information. The resulting groups were then non-overlapping and distinctly different in their behaviour, and can therefore be viewed as distinctly different types of users of the system.

## **Synthesizing**

In order to make each group act like a single individual with a united personality and behaviour, the variable values of all members in each group were merged into a single value. This resulted in a new value for each variable that represented the entire group. When a large discrepancy existed between the values of the different group members, either the average value was selected, or a certain value was hand-picked to make sure necessary information was not lost.

When an interview subject is occurring in several groups, the opinions of the others were primarily chosen whenever the values within the group differed. Ensuring that the unique opinions did not get lost, differentiating the groups as much as possible by downplaying their common part.

The resulting group variable values in text form can be seen in Appendix B.3



## Types

Each of the resulting groups were assigned a different persona types, indicating their relevance to the project.

The personas was assigned in such a way that designing the system for the primary persona would not infuriate the secondary persona as much as the other way around.

Three different persona types were used: Primary, Secondary, Supplemental. The roles assigned to which groups can be found in Appendix B.4

## Expanded description

Each of the groups generated and the information collected about them were used to create descriptions of them as fictional individuals. Each group were assigned a name, and a portrait photo that was considered to capture the type of people they are. A background text was written with their imagined life status to further enhance the picture of them as real people, and who they are.

The synthesized information from each group in the "Synthesizing"-step was rewritten as fluid text, where each of the key points occurs. This was done by writing a sentence/paragraph per behaviour variable. Each of these parts were then reorganized and combined into a fluid text.

Lastly, a single quote was created for each of the personas with the point of giving the personas a more living character.

The final result after this step was a set of complete personas, describing a set of different individuals.

The finished personas are shown in chapter 4.4.

### 3.3.2 Requirements

A brief description of the theory on how to work with requirements is available in the theoretical framework. More concretely the Goal-Directed Design methodology was followed, which defines 5 steps in the process of defining requirements. These steps were followed but sometimes with some minor exceptions.

With that in mind, the steps are not explicitly defined by the process and left a lot of room for interpretation. Some gaps had to be filled by creativity and discussion within the group.

#### Problem and vision statement

The first step in defining requirements is creating a problem and vision statement.

The problem statement defines the problems that exists in a concise manner to get a clear picture of the actual problems and what they entail. This is helpful to determine what it is that actually needs to be solved. The vision statement then concisely defines the vision for how the stated problems should be solved in the finished system, resulting in a clear design initiative moving forward.

The problem statement was created by first having a group discussion to get a rough idea of what the problems were and what was required from the product. Several drafts where made, discussed and dismissed until reaching the finished version.

The vision statement was discussed from the perspective of how the product could alter the current situation by giving the users what they need and how that creates value for the university. Each specific problem was given a specific vision for what a solution should look like.

The resulting problem and vision statements are presented in chapter 5.

## **Brainstorming**

The brainstorming process was performed in groups of two to three members, discussing all thinkable functionality and requirements that could be useful in the system. All ideas were written down, including ideas that was thought of as not so promising. Even some ideas that was found really bad at the time, or impossible to implement was included. The key point was to not miss any ideas that could be practical when the thinking about the system becomes different later in the process.

## **Identify persona expectations**

This is the first step in the process of defining requirements that is using information directly from the previously constructed personas. The personas biographies where analyzed in combination with their goals and pain points as well as their synthesized requirements . This analysis was used to create the personas expectations on the system which will be used in the upcoming context scenarios.

This step was made rather small since the personas soaked a lot of the time for this phase and as such could also be directly translated to persona expectations.

The complete list of identified expectations can be found in Appendix D.1.

## **Constructing context scenarios**

Context scenarios serves the purpose of showcasing a story that highlights how users actually would use the system. To achieve this a creative process is required. Different scenarios that the personas would most likely encounter were constructed. Through the personas expectations in combination with the personas background and mental model a story gets told. A greater emphasis was put on creating context

scenarios for the primary and secondary personas since they would carry greater weight in the future design decisions.

The scenarios were subject to change since additional plausible scenarios arose from working on the subsequent step in the designing of the requirements. As such the step of constructing context scenarios was revisited to some degree.

The complete list of constructed scenarios can be found in Appendix D.2.

## Identifying design requirements

Design requirements were constructed using two approaches because it was hard to determine satisfactory requirements in the first attempt. Initially the requirements were extracted directly by reading the context scenarios. Through reading the story, information about which data would be necessary in the system, what methods the user would like to perform and in which context these data and functions would appear. This created a long list of data, methods and contexts but it was difficult to really understand them in how they would perform in the system. They were also incomplete as the context scenarios were not as covering as would have been needed.

In the second approach, the focus started with views instead. The views that could be extracted directly from the first approach were used and the methods and data fields were mapped to each view. From that perspective more views were looked at and a more complete interpretation of the system emerged. In every view there were data and functions that would be useful for the user. The context scenarios were walked through once again to make sure nothing was missing and in the end there was a clear picture of the system. To easier discuss the layout of the views some of them were sketched, all of them but one in very low detail level.

The complete list of requirements can be seen in chapter 5.

## 3.4 Prototypes

With the requirements assembled, two different prototypes were created. Prototypes are necessary to visualize what should be made in order to fulfill the requirements.

The first prototype was based solely on the existing Canvas system and functionality already available to the users of the system. The method of creating it is available in the following section 3.4.1. However, not all requirements were able to be satisfied in this solution proposition.

Another prototype was designed with a custom UI and necessary functions. The focus was solely on meeting the requirements that had not already been met satisfyingly in the previous prototype. As a result, it was designed as an addition to the first prototype instead of a standalone solution, making the combination of the two

fulfill all requirements. The method of creating it, using the prototyping steps in the Goal-Directed Design Process, are presented in the sub-chapter 3.4.2.

### **3.4.1 Based on existing system**

The requirements, defined in chapter 5, were mapped with the functionality in Canvas collected from the domain research. This led to a list of requirements grouped by either: "Working well in Canvas", "Existing in Canvas but ease of use can be improved" or "Not existing in Canvas".

The requirements listed in "Not existing in Canvas" were put aside. The functionality in Canvas that was used to perform the remaining requirements were described in a document. As this document describes how the requirements can be reached step by step, it was converted to a tutorial for a clear view of how to perform complex tasks in default Canvas. This tutorial can be found in Appendix F.6.

### **3.4.2 Based on Custom Design Concept**

A custom UI prototype was developed as an addition to the first prototype. Doing so allows the designed system to only address the requirements not already satisfyingly met in default Canvas, instead of all of them. Following are the steps based on GDDP that were followed in order to create it:

#### **Define form factor**

The first step was to define the form factor of the resulting system. On which devices would it be used? With which input methods? How would it relate to the current systems in use? These decisions were taken based on the personas and their preferences in system usage, and assembled in a document for reference.

#### **Define elements**

The requirements that had been deemed necessary for this custom solution were analyzed so that functional and data elements could be extracted from them. Functional elements are actions the user should be able to do while data elements are the data that needs to be stored and manipulated in the system by those actions. This resulted in a list defining what the user should be able to do in the system.

#### **Functional groups**

The next step was to group the actions that has to be performed into a hierarchy. That is, organize the needed functionality in different places in the system, in dif-

ferent views and panes. Note that no designing is actually taking place here, only a grouping of elements to make sure everything is logically organized in the system.

### **Sketching the framework**

From the grouping in the previous step, a set of different views within the system had already been defined. The next step was to design them graphically and decide on the best way to present the needed elements and functionality on screen.

The views were designed in groups of two to three persons. First they were roughly sketched on a white board and thereafter reviewed in larger group. After some revisions had been made on the whiteboard, they were transferred to paper sketches with a pencil, this time with a bit more accuracy and a higher level of detail. One of the sketches was drawn from the white board directly in a digital environment to early get a feeling for the visual style guide of Canvas and how it could be applied to the sketches.

The initial design prototypes can be found in Appendix E.

### **Key path scenarios**

In order to validate that the created designs did indeed hold for the intended users, they were checked against the context scenarios of the personas. The scenarios were walked through step by step, checking where the persona would have clicked and where it would take them, making sure the scenarios are possible in the designed system.

Several faults were found during this process, in which the sketches were appropriately adjusted to accommodate the missing elements.

### **Validation scenarios**

Before the final design, validation scenarios were carried out. First a brainstorming session was held where all possible things you could want to do in the system were written down. After collecting these ideas, they were checked against the design to see if they were all possible. Any found mistakes were adjusted in the sketches, as in the previous step.

### **Final design**

Lastly the validated sketches were transformed to digital designs showing how they would look integrated in Canvas with the Canvas style guides applied. Even at this step some validation was made which caused these designs being refined a few times.

## 4 Current State

*This chapter contains a summary of the current state based on the information collected during the research phase. This includes information about the current and future systems in use at Chalmers, as well as a summary of the types of users that are currently using FIRE.*

### 4.1 PingPong

The current LMS of Chalmers, PingPong is rarely used for courses at the Department of Computer Science and Engineering. There is a negative preconception about the complexity of the program. None of the interviewees have actually tried using it in an actual course, mostly dismissing it out of hand because of colleagues advising against it or getting deterred by the amount of options when initially trying it. It should be stated that there are many teachers at other departments who are using PingPong. It should also be noted that the decision of changing LMS from PingPong to Canvas is already taken which renders PingPong obsolete no matter the case. Therefore PingPong has not been a focus in this study.

### 4.2 FIRE

From the interviews performed, it is clear that most participants like to use FIRE since it offers a streamlined experience without the need for much configuration or knowledge about the system. Primarily because the system is only used for handing in and correcting submissions, without any additional functionality that an LMS typically contains.

Since FIRE only can be used to administer submissions, extra effort need to be expended to bring the students all the information that they require. For example, it is not possible to present the assignment details in FIRE, so lab instructions will have to be hosted somewhere else. Because of this, all teachers interviewed have a separate, open, course page where students and other interested people can find information about the course.

#### 4.2.1 Usage

When a new course is started, the examiner must contact the FIRE development team, so they can set up a new course. Every study period a new virtual server is then started that hosts that instance of the course. This instance must then still be

hosted for quite a long time, since all records of the students progress are stored in the system, and might need to be accessed if there is any problem with accreditation.

Students register an account in FIRE for each course and teachers adds teacher assistants, that gets a separate account for each course as well. As a result of this, any student can register for a course instance in FIRE, even if they are not registered for the course itself, making them ineligible to receive a grade in the course.

The system offer functionality for students to submit their solutions to assignments as well as for course staff to download submitted files and pass/reject those submissions. Optionally a point score can be given for the submission. Each assignment needs two deadlines set, a first and a final deadline. The two deadlines can however be set to the same time if only one is needed.

The only additional usage for FIRE is that it contains email lists for all registered students.

In an interview with the developer currently responsible for FIRE, it was clear that there are several problems concerning the hosting of FIRE, which is done in-house. To begin with, the developers do not have administrative privileges on the host server, which has made it quite tricky to maintain the server. It has mostly been solved with administrator scripts and such, but it has never been an adequate solution. Since all course instances are hosted on a separate server, each instance must also have its own security certification, which is a finite resource in many cases. There have been times in the past where a new course instance could not be started because the system ran out of certificates.

## 4.2.2 Development

According to our interview with the project initiator, there has been some difficulty finding developers to continue developing FIRE. This is also shown from our interviews with teachers, as some has expressed dismay over missing functionality that they have requested, but there has been no time to address these requests. To further complicate things the current developer tasked with maintaining FIRE is scheduled to leave Chalmers after the summer (2018).

FIRE is developed in Python, with the web framework Pyramid. There have been instances, again according to the current developer, where Python has broken backwards compatibility when upgrading the run-time to incorporate security updates.

## 4.3 Canvas

Unlike FIRE, Canvas is a complete LMS designed for teachers and students to participate and communicate in a course. There are also tools that aims to help teachers administrate, grade and keep track of their students. A large part of this study have been to investigate what functionality exist and how it could be used

to implement a lab submission system that best suits the Department of Computer Science and Engineering. Therefore, the focus will be on the part of the system that relates to assignments and submissions.

### 4.3.1 Functionality

If a user has the rights to start a course, it is done directly from the dashboard. The choice exist to copy an old course, for reusing course material. Several options for graded course components exists, including quizzes, tests and submitted assignments. They can be divided into modules, so that students can get an easy overview of coming events. It also gives teachers a greater ability to structure their material since an early module can be a prerequisite to start a later module.

Assignments are written in a rich text editor built into Canvas, where embedding of several media types are possible. This lab information is presented to the student in the same view as submissions are done. Several grading schemes are available from the beginning, as well as possibility to define new grading formulas. There is only one deadline for each lab, although extensions are possible for a specific group, or a general extension of the deadline. There are also options for making it a group assignment, peer reviewed and other.

For each assignment, a group set is selected or created. These groups can either be filled with students automatically, filled manually by the course staff or students can be allowed to form their own groups. A group set can span several assignments, but if different sizes of groups is needed for some labs, specific group sets can be created for them. It should be noted that submissions can be made by students even when they are part of a group that has not been filled or even not part of a group at all.

Grading is done either through the grade book application or the SpeedGrader application. Grading through the grade book is done per student basis. If a grade is given for a student on a group assignment, the grade is passed to all students in their group. In SpeedGrader, groups are shown directly for group assignments. Each possible grader can grade any submission as they are not assigned to a specific grader.

Rich functionality for student statistics are available, where instructors can see missed deadlines, submitted assignments, points for assignments and much more.

There are also a lot of functionality to encourage communication. This includes, but is not limited to private messages, messages concerning a specific submission and broadcasts to an entire class.

### 4.3.2 Extending Canvas

Even though Canvas is an open source project, modifying the source code is not an option for Chalmers since, according to interviews with the stake holder, Canvas will be hosted on a cloud server and thus not open for modified source code.



There are other means to customize Canvas, in the form of LTI-integrated applications. Using this standard one can create a stand-alone web application, hosted on any server, and integrate it into Canvas. To support the standard, each Canvas instance has an API that any authenticated app can use to add additional views to Canvas, as well as extract data from the system.

As an additional bonus, there are a lot of existing LTI-apps that can be introduced to any course hosted in Canvas. The difficulty in this area has been to find ones with the correct functionality. Among this list of LTI-apps, none have been found mimicking the behavior of FIRE.

## 4.4 The users

The information from the interviews were distilled into three different personas that describes the different type of people that are currently using FIRE. Below is a summary of the personas. The complete versions can be found in Appendix C.2.

**Primary persona, Eva Lindh:** A 43 year old teacher who loves to help students and see them grow. She enjoys motivating her students within the field and often tries new interesting assignment structures. She communicates a lot with the students, helping them and giving feedback, which she feels is not very convenient in the current system.

### Goals

- Make the course go as smoothly as possible for everyone involved
- Have a high ratio of students motivated within the field
- Establish interesting assignment structures

### Pain Points

- The students has to be cross-checked manually with the official course registration
- Lacking support for communication with student and staff within the system
- Hard to get an overview of what needs to be done
- Lackluster tools for providing feedback to the students
- Difficult to manage the student groups when people leave/switch groups
- Hard to create different assignment structures

**Secondary Persona, Urban Anderson:** A 62 year old teacher who is very comfortable in the way he teaches his subject. He wants to spend as little time himself as possible, and instead manages the teaching assistants. He enjoys supervising everything, but he feels like the ability to do so is lacking

## Goals

- Ease of use
- Efficiently manage the distribution of work between teaching assistants
- Reduce his own workload
- High pass ration in his curse

## Pain Points

- There is no user hierarchy, allowing teacher assistants to do everything he can do
- The type of grades available is very inflexible
- There is no good overview of what has happened in the system
- Need for creating a fake student account to check how it look for students
- No connection between grades in current assignment system and the LMS
- System allows for invalid file extensions to be uploaded
- Group Management overall

**Supplemental Persona, Alexander Popov:** A 29 year old developer. Since he is the one creating FIRE, he feels like he knows what it should be, and therefor decides what and how something is implemented. He thinks the management of the hosting is troublesome.

## Goals

- Keep his previous workflow
- Optimize own workflow

## Pain Points

- No administrative rights to hosting
- No backup options
- Certificate maintenance

# 5 Design Initiative

*The following chapter will present the issues that were found based on the findings in the previous chapter, current state. Problem and vision statements are key in GDDP and are tools to keep focus on the problems to solve throughout the design process. Afterwards a section on the requirements themselves is presented, representing the technical level details.*

## 5.1 Problem Statement in the Goal-Directed Design Process

The old LMS PingPong is not considered useful by many teachers. At the Department of Computer Science and Engineering it is considered complicated to set up. Many also avoid contact with it due to the reputation it has among other teachers. Therefore, some of them found or created other platforms. Teachers that will not use Canvas will cause an inconsistency between courses which is a problem for students and a problem for Chalmers because teachers will not achieve some of the benefits that an LMS can provide. A need to see what is best for Chalmers has to be taken into account.

FIRE in its current state is a functional lab submission system, but it takes too much time and effort to maintain, leading to some desirable functionality not getting implemented. There is also a stated problem finding resources to maintain it and to keep it functional.

FIRE is entirely separate from other systems and does not share student data and authentication. This increases the administrative workload for both teachers and students.

The following is a list of FIRE's problems:

1. FIRE does not fulfill all the requirements that has been produced by this project.
2. FIRE is sub-optimal from a maintenance point of view.
3. The usage of FIRE instead of the Chalmers' main LMS leads to inconsistency in used software between courses on the school.
4. FIRE is completely separate from other systems, meaning that student data and logins are not shared. This increases the administrative workload for both teachers and students.

## 5.2 Vision statement in the Goal-Directed Design Process

The vision is to create a system that all students, registered to any course on Chalmers, can use. The system should also let the teachers and teachers assistants, who will be using it, find it supportive for their tasks. This system should have an integration with Canvas so that its users will not be bothered with more credentials than necessary for Chalmers systems, as well as make it easy for teachers to report results from the system. The vision is also to develop a system that is possible to maintain with less resources than the current system, freeing resources for developing extended functionality instead.

The following is a list of the vision statements:

1. Design a system that suits the teachers needs better than the current FIRE system.
2. Design a system that is possible to maintain with fewer resources than the current FIRE system.
3. Design a system that is visibly present in some capacity within Canvas, while being consistent with the Canvas design.
4. Design a system that integrates the student data and logins from Canvas.

## 5.3 Matching the problem with the vision

It is of worth to see whether the problems that has been stated match up with the visions. As such this section provides a brief explanation of each given problem statement and what vision counterpart it has.

The first problem statement specified that FIRE currently lacks features desired by teachers, which would be beneficial for them. This statement means that any potential solution should incorporate these desired features, which is what the first vision statement dictates.

The second problem statement specifies that FIRE has issues on the maintenance side of things. This statement means that for any solution, for that solution to be valid, the maintenance should among other things use fewer resources. This statement is just the inversion of the second vision statement defined.

The third problem statement formulates that usage of different platforms/systems leads to inconsistencies. What this means for a solution are two things. First, the solution should be visible/present in some way within Canvas. Second, the visual style guide of the solution must adhere to the style guide of Canvas, as to not break the flow. Once again, this is the very definition of the third vision statement.

The fourth and final problem statement says that the administrative workload of FIRE is high, this because it is a separate isolated system. A solution then should aim to decrease this workload, which is just another way to formulate the fourth and final vision statement.

## 5.4 Requirements

An important result from any study of a potential computer system is the requirements that potential users have. These were extracted from context scenarios which is written to get a certain empathy for the personas. Following is a list of the requirements extracted this way.

### Data requirements

Teachers can...

- ... see done/scheduled submissions on course level (1).
- ... see student history. Which assignments did he/she pass as part of which group (2).
- ... see the progress of submissions handled by teaching assistants (3).
- ... see submissions at risk of being corrected too late (4).
- ... see an overview of tasks to be completed in course (a TODO list) (5).

### Functional requirements

Teachers can...

- ... change deadline on assignment (6).
- ... create a new course in Canvas (7).
- ... create a copy of an old course in Canvas (reuse old course material) (8).
- ... add a new assignment to a course (9).
- ... configure teaching assistant workload from settings (10).
- ... configure Grade formula for specific assignment (11).
- ... configure group size for specific assignment (12).
- ... configure accepted file types for specific assignment (13).
- ... add and remove teaching assistants to course (14).
- ... grade submissions according to a workload setting distribution (15).
- ... only see submissions they have been assigned to handle, according to the workload settings (16).

- ... override deadline for a specific group from assignment/group (deadline extension) (17).
- ... Communicate about a specific submission (only seen by course staff) (18).
- ... send a message to a selection of students that are registered on the course (19).
- ... download submitted files from submission (20).
- ... set deadlines for grading submissions (3 work days after deadline, for example) (21).
- ... see the course page as a student would (22).
- ... see all submissions for a specific assignments (23).
- ... delegate course responsibilities to teacher assistants (24).

# 6 Solution Propositions

*This chapter will present four different alternatives for lab submissions systems. These will vary in how well they meet the previously presented requirements as well as how much effort will be needed to realize them.*

## 6.1 Using Canvas out of the box

One solution is to simply use Canvas out of the box. Canvas is already a widely used system at numerous universities and other educational institutions. This opens up for easy collaboration between schools and departments within Chalmers. For an introductory tutorial in how to use Canvas as a lab submission system, see Appendix F.6.

### Advantages

- No development time needed.
- No maintenance or hosting needed.
- Adaptable on course level.
- Possible to add external extensions.
- Same system across departments/institutions, enabling knowledge exchange between teachers.
- Access to all online material in one place.

### Disadvantages

- The extension environment is difficult to navigate.
- Finding out details about an LTI App can be time consuming.
- Placement of some functionality is not always intuitive.
- Statistics and information is almost always shown on student level, not group level.
- Different roles in the system must be changed by an administrator on the institution level. The rights of a role can not be adjusted on course level.
- Course staff must learn a new system.

## Requirement fulfillment

The advantage of having no development time of course means there are some compromises with this solution, in that not all requirements are met. Therefore, this solution proposition does not meet the following requirements: 3, 4, 10, 15, 16, 18, 21. For some requirements, there is functionality that support it, but the implementation is not very intuitive: 1, 2, 19. For details about these requirements see chapter 5.4.

## 6.2 Canvas with custom made LTI extension

There are requirements that Canvas does not meet at all, or does not meet at a passable level. By building a customized LTI, the best of both the available Canvas functionality and custom functionality could be combined. This solution would need extra maintenance outside of Canvas to make sure the integration does not break in the future. Some hosting for the LTI-app would also be needed, and possibly a database for data not available in Canvas.

By doing extra development over time, functionality that works out of the box in Canvas but is lacking in some way could be replaced by functionality developed in the LTI extension. This makes this solution dynamic, as a working system could be set up quickly and more of the functionality could be moved from Canvas to the extension over time if needed to improve the system.

For more details regarding the proposed design for a custom LTI, see Appendix F.

### Advantages

- Focused functionality, grouped the way course staff at the Department of Computer Science and Engineering wants it.
- Easier to maintain compared to FIRE due to smaller code base, since a lot of the functionality from Canvas is used.
- Can be implemented incrementally, customizing it to teacher needs and according to their continuous feedback.
- Access to all online material in one place.

### Disadvantages

- Requires maintenance and active development.
- Since maintenance is needed, there might still be a problem finding developers, just as with FIRE.
- Requires a hosting solution.



- If Canvas is switched to a new LMS, the new LMS would need LTI integration to support the built extension.
- Course staff would still need to learn a new system.

### **Requirement fulfillment**

By extending Canvas, the application designed as part of this study would improve on several areas. A new grading view would be implemented, enabling the use of automatic assigning of teaching assistants to specific submissions. This in turn enables the statistics of the TA work to be calculated and shown. It also makes the communication between groups and grading TA more intuitive and placed in closer proximity to the submission itself. An overview of submissions and grading would also be added. This would not be possible to do without a new grading view, since the standard SpeedGrader application can not be extended to take these considerations into account. The only requirement that is left unimproved with our suggestion for this alternative is the student history, which already exists in Canvas but does not show what group a student was part of when receiving a grade for a submission.

## **6.3 Integrating FIRE into Canvas**

FIRE as it is today have been under development for some years and is used in many ways that is appreciated by both teachers, teacher assistants and students. This system could be integrated into Canvas by replacing the current registration model by authentication through Canvas. Grades and/or points from labs could also be transferred into Canvas directly from FIRE.

### **Advantages**

- Take advantage of the code base that already exists.
- Familiar interface for all users.
- Dedicated system focused on one thing.
- Authentication problems can be fixed using Canvas API.
- Grades from FIRE could be exported to Canvas to take advantage of any future automatic integration with Ladok.

### **Disadvantages**

- Problems with finding developers for FIRE will continue.

- If Canvas is switched to a new LMS, the new LMS would need LTI integration to support the built extension.
- Maintenance problems will continue.
- Old pain points still exist.

### **Requirement fulfillment**

As this would be a continuation of the current environment, the only improvement would be the introduction of a more unified authentication scheme. However, any further development time that can be spared can be spent on improving FIRE to cover more requirements. The main requirements that would not be fulfilled with this solution is the functionality to easily reuse an earlier iteration of a course. Also several overview functions, such as student history, that would make work easier for teachers would not exist with this solution.

## **6.4 New, standalone submission app integrated in Canvas**

This solution is mainly the same as the previous but could be seen as a total remake of FIRE. The difference is that with a new take on a lab submission system the problems of FIRE could possibly vanish.

### **Advantages**

- Potentially a perfect system for the users' needs.
- Potentially easier to maintain than FIRE.

### **Disadvantages**

- Time consuming and expensive to develop.
- Hard to define a perfect system.
- Might still be difficult to find developers.
- The advantages could probably be achieved with less resources if these resources are put into developing FIRE.

## **Requirement fulfillment**

As this would be a complete new system, all requirements could be handled. The assumption made in this study has been to use Canvas functionality, so this alternative has not been explored further.

# 7 Discussion

*In this chapter different problems are discussed, as well as results, possible improvements and reflections.*

## 7.1 Problem with data collection / Method

There has been an overarching issue with collecting data. This issue comes in two forms. First, the number of respondents in the questionnaire was lower than expected. Second, the number of performed interviews was also lower than would have been preferred

### 7.1.1 Survey Review

Taking a closer look at the questionnaire statistics, shown in figure 7.1, the number of total responses is presented on the Y-axis. The questions are presented along the X-axis in the same order of appearance as in the survey. This along with three measurements: answered, did not answer and irrelevant answer.

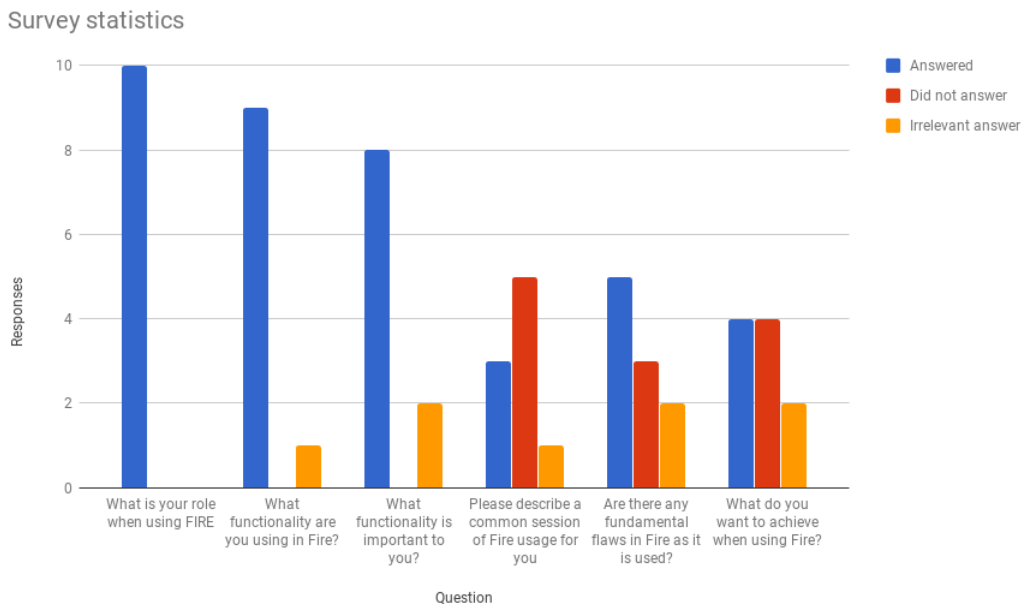


Figure 7.1: Column chart of the survey statistics

Looking at the staples, it is clear that as the survey progresses, a rapid drop of answers is present. This drop correlates with the questions becoming more elaborate.

At the same time, it is also apparent that the number of answers that are not relevant to the question asked increases. This increase hints at a problem about the depth of the questions. And indeed, during the post analysis of the survey it was concluded that the questions themselves might be easy to understand but only if the respondent is more proficient with interaction design concepts or knows the underlying purpose of the question.

Several conclusions about the survey could thus be made. The survey gave some basic insight about the usage of FIRE. There would however be more benefits in the early part of the study to plan a much smaller and directed survey instead. This direct survey should be both smaller in the number of questions but also in scope. As example, instead of asking "What functionality are you using in FIRE?", a simpler question would be "Name a good feature in FIRE". This gives a lot more freedom and does not put the respondent on the spot to think hard about the answer. In turn, the grittier details is saved for the interviews. This would give quick and relevant data without the ambiguity in what would constitute as an answer.

Doing a direct survey would also benefit the project, as thinking harder about what kind of information was actually wanted from the questionnaire would have further enforced the interaction design mindset. Also the possible domain problems would have to be considered from start.

To review the survey, as well as a version revised with respect to the newly found knowledge of it, see Appendix A.

## 7.1.2 Interview Review

A different set of problems occurred in the interview process. The most pressing issue is the questions asked, and the resulting data. It was not matched good enough for what the GDDP expected. GDDP expects data to come in the form of a measure of how often some problem comes up or how often something is done. The interviewers answers should be used to model a persona with clear, prioritized, pain points. As it turned out in the study, a list of the different problems teachers experience today was assembled, but no clear image emerged depicting how often these problems affected the day-to-day work. This image was cleared up somewhat, however, due to the groups collected experiences. Having taken several courses at the department, providing prior experience on how many teachers likes to run their courses.

The questions themselves were not bad, nor the semi-structured format for the interviews. Instead, extra steps should have been taken to ensure that it was clear what kind of information the interviews should provide beforehand. This would probably have prolonged the interviews a bit but would give an easier process afterwards. These extra steps could result in clearer follow-up questions to capture the less conspicuous things like motivation.

In connection with the interviews, there was also an opportunity to observe how many of the teachers work with FIRE, giving much insight into the kind of workflows

that exists today. All of this acted as a counterpoint for the other data points that was not obtained through the interviews themselves.

The number of interviews raises another question. Would increased quantity of interviews give more diverse data? As seen in the interviews, even when the answers were of the same vein, the only real differences was wording and some of the motivations behind them. When their answers differ the respondents can be sorted into two types of groups: those who put effort in the system to make more interesting labs and those who put more focus on other parts of the course.

those who wants to minimize their administrative workload and those who enjoy working with the system.

Even though the pool of answers was low, along with the number of interviews, the data is uniform enough to a degree where its reasonable to apply the following assumption:

A more substantial volume of data would probably not affect the result in a significant way, along with its relevance within GDDP.

## 7.2 Credibility Analysis

The results produced in this study are credible for a number of reasons, some of which have already been discussed in the previous chapter. The information gathered has been taken from first hand sources, interviews, which further strengthen the credibility of the study. The results are also based on a process accepted within the interaction design community, which aids credibility. Of focus are the persona and scenario generating processes which have gained credibility as fluid, workflow enhancing activities.

## 7.3 Method Evaluation

During the initial research regarding interaction design it was quickly realized that some kind of design principle needed to be followed. The team settled on the Goal-Directed Design Process and went from there. There might be other processes to follow but since GDDP is well used in the UX industry it was decided to just stick with it. This process gave a very clear sequential order in which to generate a robust design which is very suitable when conducting a study since each step can be well documented. The idea of following a process also makes it easy to trace back to find reasoning of design decisions. This all resulted in clear objectives and suitable goals, a good thing in group based work.

Since the process is very sequential where each step builds on the previous, it was very difficult to split up the workload efficiently compared to when doing the initial research. Since it was not an option working with different steps in parallel the

process could at some time prohibit efficient work. On the flip side, many steps was well served by working in group, as discussions and brainstorming was a big part of the work to be done.

Some steps in the process were just briefly described in the literature which resulted in a lot of time being spent getting a common understanding of concepts and ideas. An example is the theory of constructing the context scenarios where it is briefly explained what their purpose is and only one generic example is given. Designing context scenarios was viewed as an important step and took longer than expected to finalize due to lack of information.

So in summary, there were good aspects of the chosen design process that made the workflow clear and cohesive. There were also bad aspects in that the workflow became very rigid and the work became harder to split up in some stages. The GDDP in regards to this study would probably be chosen again if the study were to be redone because of the clearly defined and traceable steps, making very clear that the design decisions are not based on guesswork but actual data and analysis work. It is clear from some of the less thoroughly described steps that the process is well suited to be repeated in other projects. To have earlier experience with the process and interaction design would be very beneficial when trying to fill out the less defined steps, as in many other fields, much can be achieved through repetition.

## 7.4 Discussion about solutions

The four proposed solutions all have their advantages and disadvantages. The new, standalone submission app alternative will not be examined any further because that if Chalmers would have the resources to develop that system, they would probably be better invested if they focused on improving the existing FIRE application. If these improvements were made in order to minimize future maintenance, improved integration with Canvas and add extra functionality it would end up with the same advantages and disadvantages as a new, standalone submission app.

The first proposed solution to use Canvas right out of the box could be a serious alternative. There is functionality for handling lab submissions and group management meaning it would look and work pretty well for students, but the grading features is complicated and time consuming. The requirement to be able to split workload between many teacher assistants is not possible to achieve in the system, but could of course be done manually. Even though some functionality is not especially user-friendly it would still have the basic functionality for testing and grading students performing lab submissions. The problem to set up a course to use the right functionality could be solved by making a tutorial on how to adapt a Canvas course in the best way to handle lab submissions. The main advantage of this solution is that it requires neither developing resources or maintenance over time.

The idea of using one of the existing Canvas LTI extensions developed by others would be great if there existed one that would have functionality to handle all

requirements. There are some LTI extensions that do some of the work but they do not support all, or nearly all, requirements.

There are two more alternatives that both require development and maintenance, but on the other hand would deliver the superior experience for the users. One way is to use the possibility of extending Canvas with a custom made LTI integrated application. This solution would require some development resources, both for creating the application in the first place as well as maintaining it. These resources would not necessarily have to be spent up front, since Canvas has functionality to cover most cases. Extensions could be developed continuously, adding streamlined functionality over time. This would gradually specialize Canvas for its use at Chalmers and enable a quick feedback loop between developers and course staff. Functions that are appreciated in Canvas can be kept, minimizing maintenance work, and the parts that are not can be replaced by specialized functions.

The second development option is to use FIRE as a basis for a very small LTI integrated application. FIRE functionality would largely be the same but authentication and course registration could easily be connected to Canvas, removing the need for students to register for each course in FIRE and for teachers to handle students that are not registered to the course. This proposed solution would not solve the problem of maintaining FIRE, which has been a big one according to our interviews with the stakeholder of the project. However, if effort is put on FIRE, its maintenance problem could probably be reduced.



## 8 Recommendations

In chapter 6 several possible ways forward was described. Because there has not been an LTI extension found that can properly mimic FIRE functionality, presented are the following recommendations. Depending on priorities within the Department of Computer Science and Engineering, each one of them can be a good next step in the introduction of Canvas.

The two main propositions to consider, produced in this study, is using Canvas out of the box and developing a small and dedicated application to streamline the experience.

### 8.1 Canvas with dedicated LTI application

A Chalmers LTI app would be the best way to go, assuming any development time can be dedicated. The reason for this is that it will offer a more streamlined experience when handling submissions, something that has become an expectation after working with FIRE.

This solution offers a good compromise between minimizing maintenance work and offering a good user experience for all course responsables. It minimizes maintenance because the code base is kept low since a lot of functionality from Canvas can be used. The LTI application can be designed to meet all requirements that the course staffs have.

It is also important to not fall into the trap of PingPong, which many of the teachers interviewed had a quite strong dislike of, despite not having properly tested it. PingPong has a reputation of being hard to use and time consuming. Having the future promise of an easier to use system might get the staff responsible for the courses to invest early into Canvas.

The complete generated custom design for a dedicated LTI is described in Appendix F.

### 8.2 Using Canvas out of the box

Another possible solution is to utilize Canvas as it is. Canvas offers a fair interface and most of the requirements of the course responsible staff can be satisfied without spending any development time. As an additional bonus, there would be no code base to maintain, making it a cheaper recommendation.

As an aid, a tutorial was produced for completing the necessary basic tasks within Canvas, see Appendix F.6.

# References

- [1] Simon Pan. *The only UX reading list ever*. Date Accessed: 2018-03-24. 2015. URL: <https://medium.com/interactive-mind/the-only-ux-reading-list-ever-d420edb3f4ff>.
- [2] Guy Ligertwood. Date Accessed: 2018-04-01. 2017. URL: <https://uxplanet.org/5-design-books-every-ux-designer-should-read-b4f81361d02c>.
- [3] Julie Blitzer. Date Accessed: 2018-04-01. 2014. URL: <http://julieblitzer.com/UX-Reading-List>.
- [4] Tomas Laurinavicius. *The Ultimate UX Reading List to Take You to the Next Level*. Date Accessed: 2018-04-02. 2017. URL: <https://theblog.adobe.com/the-ultimate-ux-reading-list-to-take-you-to-the-next-level>.
- [5] Alan Cooper, Robert Reimann, and David Cronin. *About Face 3: The essentials of Interaction Design*. Wiley Publishing, 2007. ISBN: 978-0-470-08411-3.
- [6] Don Norman and Jakob Nielsen. *The Definition of User Experience (UX)*. Date Accessed: 2018-04-02. 2018. URL: <https://www.nngroup.com/articles/definition-user-experience/>.
- [7] Oxford. *User Experience*. Date Accessed: 2018-05-02. 2018. URL: [https://en.oxforddictionaries.com/definition/user\\_experience](https://en.oxforddictionaries.com/definition/user_experience).
- [8] Jenny Preece, Yvonne Rogers, and Helen Sharp. *Interaction Design: beyond human-computer interaction*. Permission: 2018-05-02. Wiley, 2015. ISBN: 978-1-119-02075-2.
- [9] Donald Norman. "Introduction to This Special Section on Beauty, Goodness, and Usability". In: *Human-Computer Interaction Volume 19* (2004).
- [10] Gillian Smith and Philip Tabor. *The Role of the Artist-Designer*. Addison-Wesley, 1996. ISBN: 0-201-85491-0.
- [11] Aurora Harley. *Personas make users memorable for product team members*. Date Accessed: 2018-04-20. 2015. URL: <https://www.nngroup.com/articles/persona/>.
- [12] Computer History Museum. *Alan Cooper*. 2018. URL: <http://www.computerhistory.org/fellowawards/hall/alan-cooper/>.
- [13] Jenifer Tidwell. *Designing Interfaces*. O'Reilly, 2010. ISBN: 978-1-449-37970-4.
- [14] usability gov. *Personas*. Date Accessed: 2018-04-20. 2018. URL: <https://www.usability.gov/how-to-and-tools/methods/personas.html>.

# A Survey

## A.1 Original

### FIRE Survey

In the process of creating a lab submission system that's integrated in Canvas, Chalmers new LMS, we would like to have some feedback on the current system used today, Fire. Your answers will be important to us and probably to you since Canvas will probably be introduced on Chalmers in the fall 2018.

The foundation of creating a new lab submission system is made by a group of students as a bachelor thesis. The group consists of Martin Ahlberger, Johan Berndtsson, Gustav Nilsson, Kamil Miller, Magnus Rönnerberg and Edvin Tobiasson, supervised by Arne Linde.

Answers in Swedish, English or Pig Latin will be good!

\* Required

1. Name

---

2. What is your role when using FIRE? \*

*Mark only one oval.*

- Course owner  
 Lab supervisor

3. What functionality are you using in Fire? \*

---

4. What functionality is important to you? \*

---

5. Please describe a common session of Fire usage for you.

---

6. Are there any fundamental flaws in Fire as it is used?

---

7. What do you want to achieve when using Fire?

---

8. Additional comments:

---

9. Would you like to participate in a short interview with us? \*

*Mark only one oval.*

- Yes  
 No

10. If yes, please write your email address below

---

## A.2 Improved

### FIRE Survey

In the process of creating a lab submission system that's integrated in Canvas, Chalmers new LMS, we would like to have some feedback on the current system used today, Fire. Your answers will be important to us and probably to you since Canvas will probably be introduced on Chalmers in the fall 2018.

The foundation of creating a new lab submission system is made by a group of students as a bachelor thesis. The group consists of Martin Ahlberger, Johan Berndtsson, Gustav Nilsson, Kamil Miller, Magnus Rönnberg and Edvin Tobiasson, supervised by Arne Linde.

**1. Name at least one thing you like about FIRE**

---

---

---

---

---

**2. Name something you dislike about FIRE**

---

---

---

---

---

**3. If you would like to participate in a short interview with us, please fill in your email address**

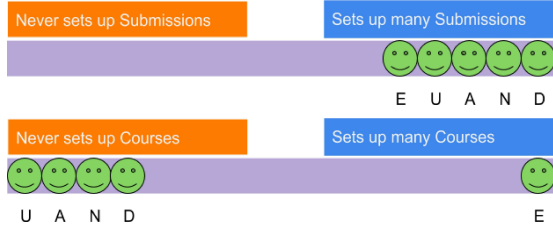
---

# B Persona - Partial Results

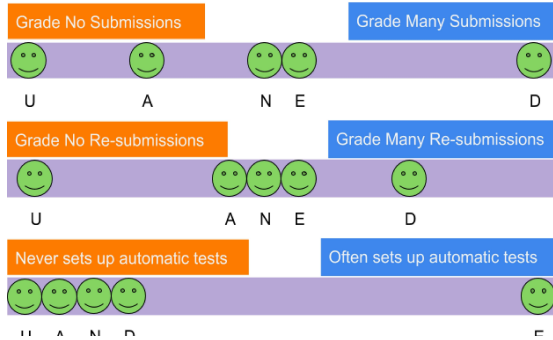
## B.1 Range Charts

### Activities

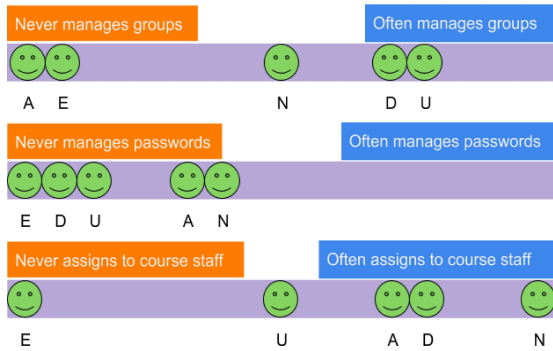
#### Setup



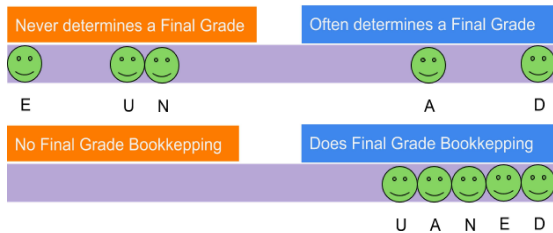
#### Submissions



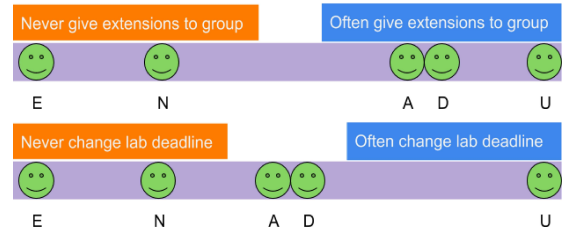
#### Management



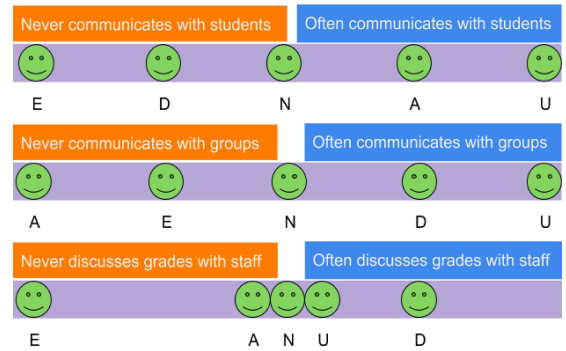
#### Working with the result



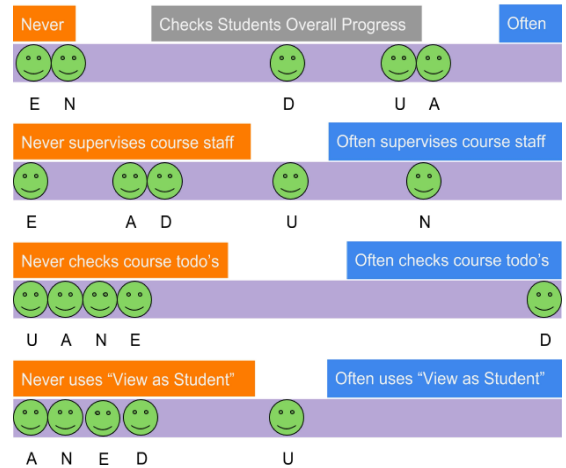
### Time



### Communication

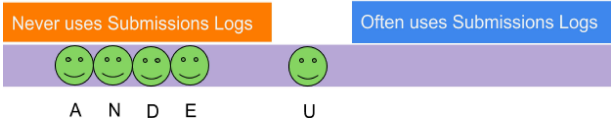


### Checking

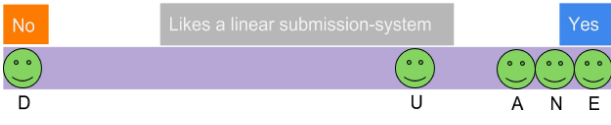
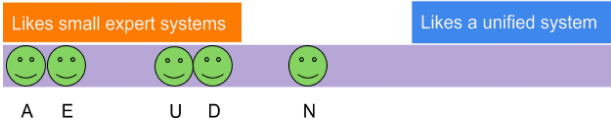


### Other

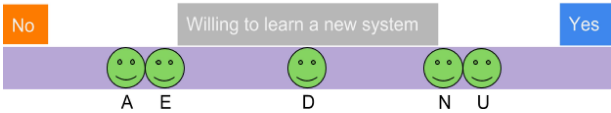




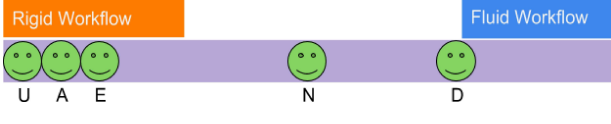
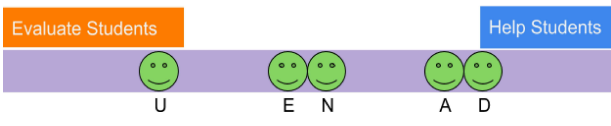
**Attitudes**



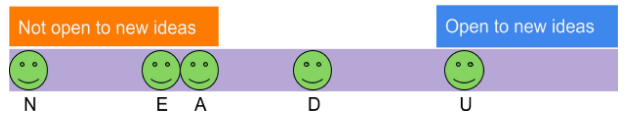
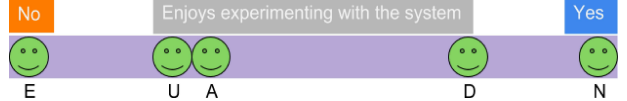
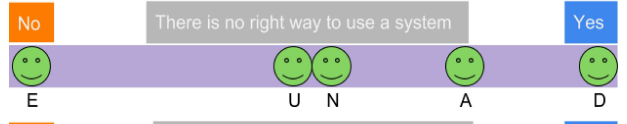
**Aptitudes**



**Motivations**



**Skills**



## B.2 Common Variables in Groups

The green line represents the delimiter where only groups below the line were deemed interesting for further work due to their large amount of common variables.

UAND	ANE	AD	UAN	END	EDU	AN	UN
Setup Submissions	Grade Submissions	Setup submissions	Setup Submissions	Setup Submissions	Setup Submissions	Setup Submissions	Setup Submissions
Setup Course	Grade re-submissions	Setup Courses	Setup Courses	Grade re submissions	Manage Passwords	Setup Courses	Setup Courses
Automatic Tests	Change lab deadline	Automatic Test	Automatic Tests	Communicate with groups	Final Grade Bookkeeping	Grade Submissions	Automatic tests
Assign course staff	Communicates with group	Extension to group	Communicate with Students	Final grade bookkeeping	View as Student	Grade resubmissions	Discusses Grades
Final Grade	Final Grade Bookkeeping	Change deadline	Discuss grade with staff	View as Student		Automatic tests	Manges groups
View as Student	Course To Do	Assign to course staff	Assign to course staff	Submissions Logs		Change lab deadline	Determine final grade
Hates Fire	View as student	Determines final grade	Final Grade Bookkeeping			Communicate with students	Final grade bookkeeping
Computer Novice		Final bookkeeping	CCheck to do's			Discuss grades with staff	Supervise course staff
Teacher long time		Checks student progress	View as student			Manage Passwords	Check to do's
		Supervise course staff	Hates Fire			Assign course staff	Hates Fire
		View as student	Viewable for everyone			Final grade bookkeeping	Same system
		Hates Fire	Right way to use			Check to do's	Viewable for everyone
		Right way to use system	Rigid Workflow			View as student	No right way
		Enjoys experimenting	Computer Novice			Hates Fire	Willing to learn
		Open to new ideas	Teacher			Same system	Computer Expert
		Evaluate Students	Submission notes			Right way	Teacher long time
		Computer novice				Computer Expert	Submissions notes
		New teacher				Teacher long time	
		Submission logs				Submission notes	

UAD	UANE	ANDE	EA	EN	UAE	AND
Setup Submissions	Setup Submissions	Setup Submissions	Setup Submissions	Setup Submissions	Setup Submissions	Setup Submissions
Setup Courses	Final grade bookkeeping	Grade re submissions	Grade submissions	Grade submissions	Final grade bookkeeping	Setup Courses
Automatic tests	Check to do's	Final grade bookkeeping	Grade re submissions	Grade re submissions	Check to do's	Automatic Tests
Assigns to course staff	Viewable for everyone	View as Student	Communicate with groups	Extensions to group	Viewable for everyone	Change lab deadline
Final grade bookkeeping	Rigid Workflow	Submissions Logs	Manage groups	Change deadline	Rigid Workflow	Assigns to course staff
Check student progress			Final grade bookkeeping	Communicate with groups		Final grade bookkeeping
Hates Fire			Supervise course staff	Final grade bookkeeping		View as Student
Enjoys experimenting			Check to do's	Overall progress		Hates Fire
Computer Expert			View as Student	Check to do's		Computer Novice
Teacher long time			Open to new ideas	View as Student		Teacher Long Time
			Willing to learn	Evaluate students		Submission Logs
			Rigid workflow			
			Computer Expert			
			Teacher for a long time			
			Submissions logs			



## B.3 Group Variables

### Group AD

#### Activities

Sets up many submission (course administration)

Submissions notes

Grades some submissions (help staff, course administration, evaluate students)

Grades some resubmissions (help staff, course administration, evaluate students)

Gives extensions to groups (help students)

Often commuincates with students/groups (help students and staff)

Discuss grades with staff (help staff, course administration)

Sometimes manages groups (help students, course administration)

Assigns course staff (course administration)

Determines final grade (course administration)

Checking students overall progress (help students and staff)

Sometime checks course todos

#### Attitudes

All courses should not use the same system

Course information should be open to the public

There is no "right" way to use the system

Open to new ideas

#### Aptitudes

Somewhat willing to learn new systems

#### Motivations

Evaluating students

Non-linear workflow

Help students

#### Skills

Computer savvy

Longtime teacher

#### Pain points

Not appropriate to non.linear workflow (accept/reject)

Registration of students

Lacking support for communication with student, staff and self (notes)

Hard to get an overview of what need to be done

Group management chaos (svårt att se historik/vem som gjort vad. leaving groups but not formally)

Hardcoded deadline rules

Lackluster tools for providing feedback

#### Skills, experience and abilities

Administration of grades

Communicative

Thinking outside the box

Pedagogical

Analytical mindset to progress overview

#### Goals

Make the course go as smoothly as possible for everyone involved

High ratio of students motivated within the field

Establish interesting assignments/assignment structures

### Group UAN

Sets up many submissions (course administration, efficient time management)

View as student

-iew history (user, group, TA)

Often sets up automatic tests (efficient time management, workload reduction)

Occasionally commuincates with students/groups (course administration)

To some extent discuss grades with staff (course administration, workload reduction)

Assign course staff (course administration, workload reduction)

Occasionally check todos

Manage course staff - Workload reduction

Somewhat dislikes Fire

Open information

Right way to use a system

Open to experimentation with systems

Willing to learn new systems

Rigid workflow

Evaluate students/grade students

Computer savvy

Teacher experience

Too much authority for TA's in current system/unclear hierarchy

Wants more flexible grade rules /define grades themselves

Lack of a good overview of what has been done

Need for creating fake student accounts to check how something looks like for a student

No connection between the current system and registered students.

No connection between grades in current system and LADOK

System allows for invalid file extensions to be uploaded

Group Management overall

The need for submitting everything at once

Structured mindset

Steady course experience (same course for several years with same structure)

Statistical mindset to progress overview

Ease of use/ Effective course administration

-Efficient Time management

-Workload reduction

High pass ratio

### Group EA

Setup Submissions (Course administration, sticks to his principles)

Grade some submissions (Evaluate students)

Grade re some submissions (Evaluate students)

Rarely communicate with groups (Course administration)

Final grade bookkeeping (Course administration)

Sets up courses (System administration)

Password management

Likes Fire

Thinks Fire works well

Unmotivated

Unwilling to learn new systems

Rigid workflow

Grade students

Sticks to his principles

Computer Expert

Teacher for a long time

Developer of Fire

No administrative rights to hosting

No backup options

Certificate maintenance

Course administrative skills

Set up courses

Keep his previous workflow

Optimise own workflow

## B.4 Assigned Persona Types

The three different relevant groupings found were assigned different persona types:

- **Group AD** - Primary persona
- **Group UAN** - Secondary persona
- **Group EA** - Supplemental persona

# C Personas

*This appendix contains the collection of personas that were produced during the study. The first two personas were produced by lifting information from the questionnaires only. The final three, namely Eva Lindh, Alexander Popov, Urban Andersson were generated by following GDDP rigorously.*

## C.1 Made using Quick Guide

**Persona:** Teacher Assistant

**Name:** Alex Persson

**Goals and Tasks:** Alex would like to perform quick grading or at least get the grading/feedback done with as little overhead as possible. Manipulating deadlines easily if necessary is also a necessity.

During work time Alex often spends time grading submissions in today's system. Some of the tasks in this activity consist of downloading the different assignments submitted. Accepting or rejecting them, leaving feedback. Sometimes Alex also has to modify a deadline for some submissions in case of some situation. As such Alex would like the system to be used to allow for this to be done in an intuitive and quick way.

Alex would also like there to be support for contacting students in some fashion, something the system today does not do so well. Some functionality missing today is that there is no real safeguard against accepting or rejecting a submission. If by some accident one of these buttons are pushed then the process needs to be redone and feedback needs to be rewritten from the start. Not to speak of the extra mails the students get from the system. Alex also feels like there should be an easier way to overview which student is in which group per assignment. Currently the system only allows for checking the current group the student is in.

Generally Alex feels that the current systems support for groups and students working alone could be better. As it stands now there is a lot of hiccups happening when some students leaves a group due to some issue which requires extensive time to resolve.

**Environment:** Alex is an experienced computer user that also has user experience of other systems such as the current one. And while the current one is good and a lot better than the ones used before. It still feels like there is room for improvement.

**Persona:** Teacher at D/IT Section

**Name:** Kim Svensson

**Goals and Tasks:** Wants to use a submission system that is easy for all parties involved. During work times Kim spends the time contacting students, organising the course and assignments, managing grades and holds lectures.

As such Kim does not like to spend too much time on administrative tasks and would like to spend as much time as possible on lectures and helping the students. When Kim uses the system today there are no common usage session, and this is something that Kim does not feel needs to change, as you use your tools to whatever situation is at hand.

It is important for Kim to be able to have a system where students can submit assignments in an easy and intuitive way. They want to be able to communicate with their students or fellow teachers / TAs through the system. On some assignments Kim wants to be able to give point grades on others an alphabetical one, and in other situations just a simple pass/fail. Synonymous with this functionality there should also be support for leaving feedback to the students. There should also be some easy way to overview the grade information on a larger group of students. In particular there should be some easy way to export grades into the official grade database, as opposed to having to manually check each student as is done today. Kim also spends unnecessary time today on having to cross check whether someone registered in the system used today is registered to the course, as these two are separated.

Some of Kims courses are more focused on feedback as opposed to pass/reject loops. As such Kim feels it would be nice if the system could also support more non-linear workflows as this. For example more dynamic capabilities of changing deadlines. There should of course be support for resubmission of an assignment if the deadline allows.

Kim sometimes also has some straightforward assignments, for these some support for automatic grading would be nice. And sometimes it would also be nice to be able to leave small personal comments on assignment incase Kim wants to look at it again.

A fundamental function Kim wants to have is to be able to create group / solo assignments without too much hassle, something todays system does not support.

**Environment:** Experienced computer user. But due to a lot of administrative work puts either the work of going through submissions on the teacher assistants or does not have enough time to help the students themselves. Kim also feels that the current overall system does not allow for optimal situations in some of the courses.

## C.2 Made using GDDP



### Eva Lindh - Primary Persona

Age: 43

Occupation: Teacher

**”It is so nice to see young and creative minds.”**

#### Goals

- Make the course go as smoothly as possible for everyone involved
- Have a high ratio of students motivated within the field
- Establish interesting assignment structures

#### Pain Points

- The students has to be cross-checked manually with the official course registration
- Lacking support for communication with student and staff within the system
- Hard to get an overview of what needs to be done
- Lackluster tools for providing feedback to the students
- Difficult to manage the student groups when people leave/switch groups
- Hard to create different assignment structures

#### Background

Two children Lisa and Klara, twins 8 years old, husband who is IT developer and he wants to complete a “Klassiker” during this year. Eva is an educated teacher and has a personal interest in Computer science. She is highly motivated to create meaningful and interesting labs for her students. She is sometime a bit stressed by taking care of her kids when her husband does a lot of training, but she really thinks he should try to reach his goals in the sports field. Two times a week, or three if she finds the time, she is doing yoga with her friends to get some gossip and relax.

## Bio

Eva Lindh has been a teacher for several years, is comfortable with teaching her course and the computer skills needed in the process. She uses labs in her course both to help the students understand the subject better, as well as a basis for grading them at the end of the course.

At the start of each study period, she asks the system developer to start a new instance of her course, as well as let the students know that they can register. She manually checks so that everyone that should be doing the labs are registered in the lab system.

She sets up all of the assignments herself (which are mostly a copy of last year's assignments), as well as grade some of the submissions from the students, while leaving the bulk to her teaching assistants. It is she who decides how much of the work each assistant should do each week, depending on their time available, and how much she takes on herself.

Eva is social and likes to have a lot of contact with the people around her, both the assistants, as well as the students. A lot of this communication happens outside the system, through e-mail and other means.

Most of the communication with the students, either as groups or individuals, is to discuss different aspects of the assignments, and explain the task and subject the best she can. Sometimes a group has not finished an assignment in time, in which she might give an extension to the group so they get another chance.

Whenever she grades a submission, she leaves a text for the student with feedback in the system. However, she often finds it hard to make herself clear, since all she can do is to send the student a single text, completely separately from the files submitted.

When it is hard to know how a group should be graded, she thinks it is comforting to discuss the matter with the other staff. Likewise, she sometimes get contacted by the assistants who need help grading certain submissions.

Eva also manages the grouping of students, making sure that everyone in the group is still actively a part of it, as well as allow groups to have a different number of members. When people have switched groups during the course, she has to check so that people who according to the system has not finished some lab, hasn't finished it while being part of another group earlier.

Every now and then, she likes to check how the students are doing in general and how everything is progressing, to see if there is anything that should be done. Sometimes this can be making some change to the course, helping the other staff, a certain student, or simply grade any new submissions from groups she has undertaken.

Since she finds it a bit hard to get an overview of the big picture in the current system, she periodically transfers all data to a custom Excel-sheet where she can do different checks on how everything is going. At the end of the course, she calculates

a final grade for each student based on how the different labs went using the same Excel-sheet, which she then reports to Ladok.

Eva thinks it is fun to try new things in the way the course is built up, optimizing the way it is taught, but she finds it hard to do so since the labs can only be formed in a certain way in the current system. There is a list of labs, all with exactly two deadlines, that can only be either accepted or rejected. She is open to new ideas, although she finds it a bit daunting to start using a new system when the old one actually works okay.

Eva thinks teachers should be able to use whichever systems they want, in any way they want, to fit the needs of the course the best it can. At least as long as all course information is publicly open so everyone that is interested about the subject can read about it, no matter if they are actually taking the course.

**Image source:**

<https://www.flickr.com/photos/139030176@N03/23778514499>



## **Urban Andersson - Secondary Persona**

**Age:** 62

**Occupation:** Teacher

”The students call me ‘The Urb’, but I don’t know what it means”

### **Goals**

- Ease of use
- Efficiently manage the distribution of work between teaching assistants
- Reduce his own workload
- High pass ration in his curse

### **Pain Points**

- There is no user hierarchy, allowing teacher assistants to do everything he can do
- The type of grades available is very inflexible
- There is no good overview of what has happened in the system
- Need for creating a fake student account to check how it look for students
- No connection between grades in current assignment system and the LMS
- System allows for invalid file extensions to be uploaded
- Group Management overall

### **Background**

Urban has been a multitasker in his earlier life. When he was young he worked as a programmer at Forsmark nuclear power plant doing a lot of security maintenance. He also had a great hobby in photography as well as wind surfing a few times a year. He was also active in orienteering when he was younger, while now still being active as a coach for the local club. 20 years ago he had a break down due to his



stressful job at Forsmark. After he recovered he had to stop with all his hobbies and decided to work as a teacher instead to try to teach the new generation how to program. Unfortunately the teacher job was harder than he thought. He doesn't think he really understand the younger people nowadays, maybe because he doesn't have any kids himself. Now he is only looking forward to retire so he once again will have time with his hobbies.

## **Bio**

Urban has been a teacher for several years, and he is comfortable using computers for a many different tasks, although he finds the analog equivalents to be superior in certain areas. He uses labs to let students practice the field, and to eventually check which succeeded to learn it.

Urban is very comfortable with the course he is teaching. He has found exactly how he likes it to be, and it has been the same way for 8 years now. He believes in a strict system, where a submission is either accepted or rejected, and you have to pass all labs to pass the course.

When a new new study period start, he asks the developer to create a new instance of this old course. After it has been set up, or anything been changed, he likes to verify so it looks good as a student. Therefor he always creates a fake student account in which he can log into his own course to see how he looks. He then just ignores the test user during the rest of the course.

He takes on some of the correction work himself, since he likes to be a part of it, but assigns most of the work to the current teaching assistants. He often checks how they are doing when correcting submissions. His goal is that they should always correct them within 72 hours so students have a chance to fix their submissions. Any misbehaving teaching assistants needs to be talked some sense to.

Since he is very keen about deadlines, he very rarely allows deadline extensions. He is also keen to make sure that he himself does everything he should do as fast as possible. He does this by, keeping track of his to do's, always making sure no one is needing him at the moment.

Most students make the same mistakes every year, so in order to reduce the amount of correcting he has to do, he has set up automatic tests that can reject assignments directly on submissions.

Urban talks with the students whenever they contact him, which mostly happen over email. He likes to try to go back in the history of the system and try to see what has happened earlier whenever a situation occurs. This could be to make sure that a submission was uploaded before the deadline even when a group forgot to submit, or to evaluate potential cheating.

At the end of the course, he manually transfers which students passed from the lab system to ladok, which is pretty error prone. Last year he accidentally passed a

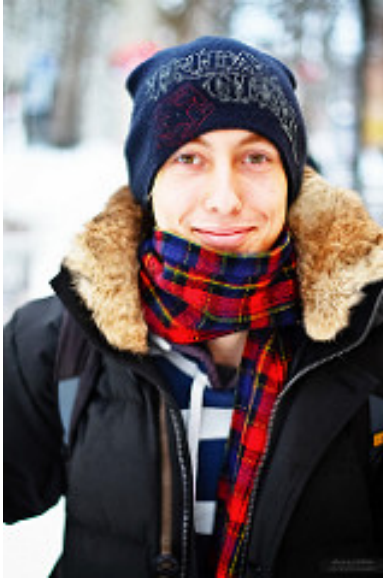
student that shouldn't have passed, but the damage was already done, so now he is extra careful when he is doing it.

He somewhat dislikes the current lab submission system, due to several, as he sees it, flaws and irritations. He has several ideas for features he would like, but he would rather be irritated at things missing than having too many options though. He hates when things are hard to find.

He would be open to learn a different system if it provides a better experience. However, he sees a point in all courses using the same system, both for teachers and students. It is easier to collaborate and students knows how things work so he doesn't have to explain it to them.

**Image source:**

<https://www.flickr.com/photos/lannigan/38560415760>



## Alexander Popov - Supplemental Persona

Age: 29

Occupation: Developer

”I created it, so I know how it should be.”

### Goals

- Keep his previous workflow
- Optimize own workflow

### Pain Points

- No administrative rights to hosting
- No backup options
- Certificate maintenance

### Background

Alexander Popov had an early interest in computers and since he had a single mother who worked more or less all the time he got most of his information on the world through the internet. This led him to a good understanding of English and a somewhat cold attitude towards his own country which he felt tried to fool him with its propaganda. So when the opportunity to move to Sweden for a position as a PhD-student presented himself he didn't hesitate. Having a rather bleak childhood where most of his time was spent with computers left him with a logical and calculating mindset together with somewhat lacking social skills. This has been improved over time and he now has a small circle of friends and is in regular contact with his mother who now lives a more relaxing life where she doesn't have to work nearly as much.

## Bio

Alexander is the go-to guy to get your technical problems sorted out, he is the system administrator and helps teachers with setting up their courses. He is efficient and experienced in this but is not too happy to accommodate to specific or intricate task since this disrupts his workflow and what he thinks FIRE should accomplish.

Alexander has been helping out in courses within his field ever since he started his PhD but does so reluctantly. It's not that its beneath him but more that its not why he is at the university. He grades submissions and helps students out when required but he is well aware of that he has no passion for it. At times when feels especially uninterested he will just fix the errors and provide the answers. He keeps his communication with the students to a minimum since the ones who have problems will reach out to him anyways.

He is happy with FIRE and how it works but is not interested in working on it more than necessary. The maintaining of the system mostly fell into his lap since he helped develop it and the reasons he is still doing it is because he doesn't trust anyone else to do it and it gives him a sense of pride and power to be the person to make things work.

With this said he admits that problems arise but since they are quickly solved he hardly counts them as actual problems. But he is frustrated with the university's apparent lack of interest in providing resources, he knows that what he does is deemed somewhat important and he can't fathom why he can't be granted more serverspace to run instances and store backups on.

It annoys him when teachers want something special or new for their courses since it's not really his job to fix it, but he begrudgingly carries on because somehow he feels like it's his duty to assist these poor souls that can't fend for themselves.

Working a long time with courses within the field has made Alexander better at communicating and teaching, even tho he himself is not particularly aware of this its has been noted by others. He has always been meticulous about most thing but this innate ability has been improved by working as course administrator.

When faced with the prospect of a new system that would replace FIRE he feels uneasy. The current situation is according to him stable and works as intended and there is no reason to meddle with something that works. But if it comes to it, he wants no part of developing or maintaining the new system so he can focus on his areas of interest.

### Image source:

<https://www.flickr.com/photos/139030176@N03/23778514499>

# D Requirements - Partial Results

## D.1 Persona Expectations

### **Eva Lindh**

Open to new ideas but somewhat unwilling to learn new systems.

Wants the ability to choose a lab submission system if it suits her needs but doesn't want to be forced into it.

Likes to work with software that fits her needs, not to be locked into one specific system.

Wants a flexible workflow.

Focus on evaluating and help students learn and pass the course.

Wants to set up a course as easy as possible and then just want it to work.

An easy way to communicate with both students and staff is important.

Wants the system to support the creative progress for the students.

### **Urban Andersson**

Could use a manual to learn a system if it could learn him more.

One of the main things Urban does is the actual course administration.

Would like more functions from FIRE than the actual system. Wants a simple and straightforward workflow but wants the possibility to customize it depending on the course.

Wants to be able to optimize the workflow for a course once so the time spent later can be minimized. Wants to be able to copy the course settings to the next time he sets up the same course.

An easy and effective way to communicate with students and TA's in the course.

Wants to be able to see an overview of the progress for the students in the course.

Urban is using teacher assistants, who help him with the submissions, which he also wants to administrate.

## D.2 Context Scenarios

### **Eva Lindh**

#### **Starting a course**

In the start of every study period Eva, starts a new course and she also sets up the lab tasks for the course. Eva thinks it's fun to start a new course and therefore she takes her time to perform this task. When she is starting, the system gives her

options to get started with a copy of the same course from last year. The deadlines used in the course before are moved to this year with the same time between labs. Eva needs to change the deadline dates to better fit the school schedule, but the prefilled dates helps her.

She finds that all students are automatically fetched from the LMS system at her school. She realizes that there is an odd number of students and she wants them to work in pairs. However, there is always someone who wants to work in a group of 3 instead, so that won't be a problem as the system has the ability to change group size for some groups.

Eva has been experimenting with the thought to change the course so the students will receive a grade, and not just approved or not which was the case before. There are a few ways to set up how grades will be handled and Eva chooses to have a number grade on each assignment, which then sum to a total score providing the final grade on the lab moment.

Eva already knows who the teacher assistants will be, so she can set up the workload for the entire course. Eva divides the workload on every assignment so each TA get 20% of the correction, and takes 10% herself. She realizes that she will be too busy on the last one, so she gives herself 0%, and the TA's gets assigned 25% each automatically.

Finally she writes a lab PM instructing the students what to do for each lab since the tasks has been adjusted a bit as well. She publish the document and reminds herself to put the text for each specific lab next to the corresponding lab page making it easier for the students to see what's expected.

### **Message by student**

A few weeks into the course Eva receives a message from Carlo. She has been taking a few hours off to spend the afternoon with her daughters entering the Malmian Whale, since it's only open a few days a year. Eva reads the message very quick and realizes that she made a mistake describing the third lab that could lead to a misunderstanding.

Eva looks at the overall progress for the lab and finds that many of the student groups will soon be entering the third lab. Therefore she asks her daughters to have a quick look at the room of big mammals while she composes a message to her students. It's easy to send the message to all students. She also sends an extra message to Carlo, thanking him for pointing out the error.

Eva had also received a request from a group of three that wanted to lab together, instead of two as the normal group size is. In the request they have written a message explaining why and since Eva had a problem of an uneven student count in the course, she tells them that it is okay in a message, and adds an exception for their group size in the system by setting it to accept 3 students.

## **Correcting labs**

Eva just arrived at work. She has an hour before her lecture, so she thought she would take a look at everything that needs to be done. She starts her laptop and opens up the overview. She can see that she has 2 things to do concerning grading student submissions. The oldest one is a comment. She opens it and reads it. A student had a question about something, so she writes a short answer.

She opens the next submission in line. She sees that there is an unread comment. The student is asking why they were rejected, when they actually had submitted the requested file. Eva reads the feedback she had left earlier, and opens the submitted files again. There it indeed was, she had missed it, and the contents looks good as well. She clicks on the submission result and clicks on new result. She selects the correct grade, and writes a short message stating that it was her mistake, and that they had done a good job. She saves it, and the new result replaces the old one, accepting the lab, while the new feedback is appended as an update to the old one.

She open the next submission in line. It is the student she just answered. She has written “Thank you for the help!”. She marks the comment as read since she doesn’t need to answer to that. End of course

When the final deadline has passed, Eva looks at the overview and sees that there are still two lab groups that haven’t finished one of the assignments. She looks at the submissions to see if she can find out why. She can not, so she contacts them and asks why. When they respond, she extends the deadline by a few days, but starts to grade the other groups anyway. Since the system has calculated a final score for the lab moment already automatically for everyone, she just have to take a look and make sure it all looks okay.

Some groups are just below the point for a grade, Eva looks a bit on their code and if she finds it to deserve the higher grade, she adjust the grade manually. The grades could be found by the students as well, and they will also have the opportunity to ask Eva to have an extra look on the grade, if it possibly could be raised.

## **Urban Andersson**

### **Start a course**

Urban had a good day the first day of the course period. He prepared the last of the course: publishing course PM, lab PM and speaking to his teacher assistants. Everything goes smooth and he leaves office a few hours earlier because he knows there will be a lot of work answering questions from the students the next day after the first lecture. On his way to the tram stop there is something he forgot: he never set up the labs.

Urban had earlier decided that he should use exactly the same labs as the year before since they were good, and he doesn’t want to do more work than necessary. Therefore he just selects to import last years settings as they are. When Urban takes

a look at the TA's on the course, he remembers that one of them has left Chalmer's. He therefore removes him and let fewer TA's do a larger portion the grading, until a new one is found.

Next day he looks into the lab system once more. He looks at the lab overview noticing that nearly all students have found groups and that a few of them have already made their first submission. He needs to finalize and change the workload distribution between his assistants so the students can start getting feedback. He has found a new TA which he enters into the system. He looks at his notes from the meeting with his assistants yesterday, and splits the workload to 40% for two of them and 20% for the last one who has another programming lab course to support. He also knows that the system tries to let the same TA correct the same group over all labs, so he doesn't have to worry about that.

### **Question from student**

A few weeks into the course, Urban receives a message in the lab submission system from a student right after a lecture. She is asking why she can't find any information about the next lab, which she had been told by Urban would be visible after the lecture. Urban thought there was something wrong because he activated the lab just before the lecture started.

Urban clicks Rebecca's profile and then the button "Show as Rebecca". As a course responsible Urban can reach exactly the same views as the students. When he shows the system he finds out that the assignment is invisible for Rebecca as she pointed out. He therefore have a look at her profile again and notices that she hasn't been accepted on the second lab which is a prerequisite for the fourth lab she is asking about.

Urban think back to the lecture and realizes that he didn't mention it to the students even though it's stated in the lab PM. He also had forgot to write it next to the lab. Urban has been used to that students these days sometimes misses important information because of sloppiness.

He messages Rebecca back and tells her to fix the second lab and then the fourth lab will be ready. Urban receives an instant message back where she points out she have been waiting for a teacher assistant to correct the lab for some time now and that it's frustrating to wait. Urban notices that the system had notified him about a very delayed correction, which he had missed. He sends a message to the assigned grader asking what is going on. He then messages Rebecca, telling her he has contacted the TA, and that he send her the instructions for the fourth lab so she can get started while they wait for an answer.

### **Supervising course/Correcting labs**

Urban starts his computer to see how everything is progressing. He is presented with an overview where he can see that almost all students has completed the first



lab, which is reassuring. However, 60% has not started on the second lab yet, even though the first deadline is in 4 days. He clicks on reminders, writes a message that the “deadline is in 4 days, so make sure you have started”, and selects every group that has yet to submit their first submission for lab 2.

When returning to the overview, he also notices that the 5 groups has waited more than 3 days for their submission to be graded, which is his goal time for correcting submissions. He opens the detailed information and sees that all 5 has the same responsible teaching assistant. He immediately contacts the TA to ask what is going on.

Returning to the overview, he can easily see how his own grading is going. The groups he has assigned to himself has two new submission and one new comment. Clicking on it takes him to the grading page where he can see all that needs to be done. As he already knew, there are two new submissions to grade. The second one has a comment attached where the student explains that the file doesn't work on Windows for some reason. It was received just two hours ago, so he can wait a bit with that one.

Instead he opens the top submission, the oldest thing yet to be done, which was received 12 hours ago. He sees that this is their fourth submission attempt. He scrolls up for a quick overview of what has happened earlier, before clicking the download button, downloading all files to his computer in a zip.

After looking at the files, he is ready to provide feedback. The work is almost done, but it fails a simple specific case. He clicks the “Grade” button on the submission where he is prompted for the result. He looks at the available result labels, and feels like none of them really captures how little is left to do for them. He therefore creates a custom label “One test fails”. He writes in the text box which case it is failing, and submits.

He returns to the grading page again, and now notices that there are three new comments on different labs. He takes a peak at the comments to see what they are about. After the deadline reminder he sent, the students apparently started asking about the problems that has stopped them from submitting. One question is very simple, so he opens it and answers it directly. The other two would take quite a bit more time to complete, so he leaves them for later since it is time for lunch.

## **End of Course**

When the final deadline for the last lab has passed in the course, it is time to report the final result to the grade system. He opens up his overview and sees the few groups that didn't succeed with at least one lab highlighted. Since he reports the grades of students and not groups, he switches to the student view. There he can see that 12 students were part of a lab that didn't get accepted, meaning that they failed the lab part of the course. One student in particular had passed all labs it did within its group of two, but the single lab within a group of 5 was rejected.

Since the final course date has passed, he can click on a button that exports the final grade to submit to the grade system. At the same time, the students gets a message of their final grade and that it has been reported to the grade system so he doesn't have to answer any questions about the resulting grades.

## **Alexander Popov**

### **Correcting labs**

Alexander is a teacher assistant which means that he and a few others skilled developers are correcting labs. Because the students who have made these labs wants feedback as fast as possible Alexander is usually correcting labs once a day. The course administrator has given Alexander 33% of the groups to correct, the same amount as the two other teacher assistants.

When he starts working he gets a list of all submissions that he has to correct and he can easily download all of them in a single click. Alexander corrects the labs in the order prioritized by the system by providing feedback for them one by one.

While he is working he gets notified that some groups has handed in more submissions. For most of them he just let them be for now, as he will handle them when his is done with the groups he is currently working with. But if it is a group that he just sent feedback too, or if he knows that the group probably just made a very little change, he takes a quick look at it in between the grading of the others.

# E Sketches

This appendix contains the initial sketches from the design process.

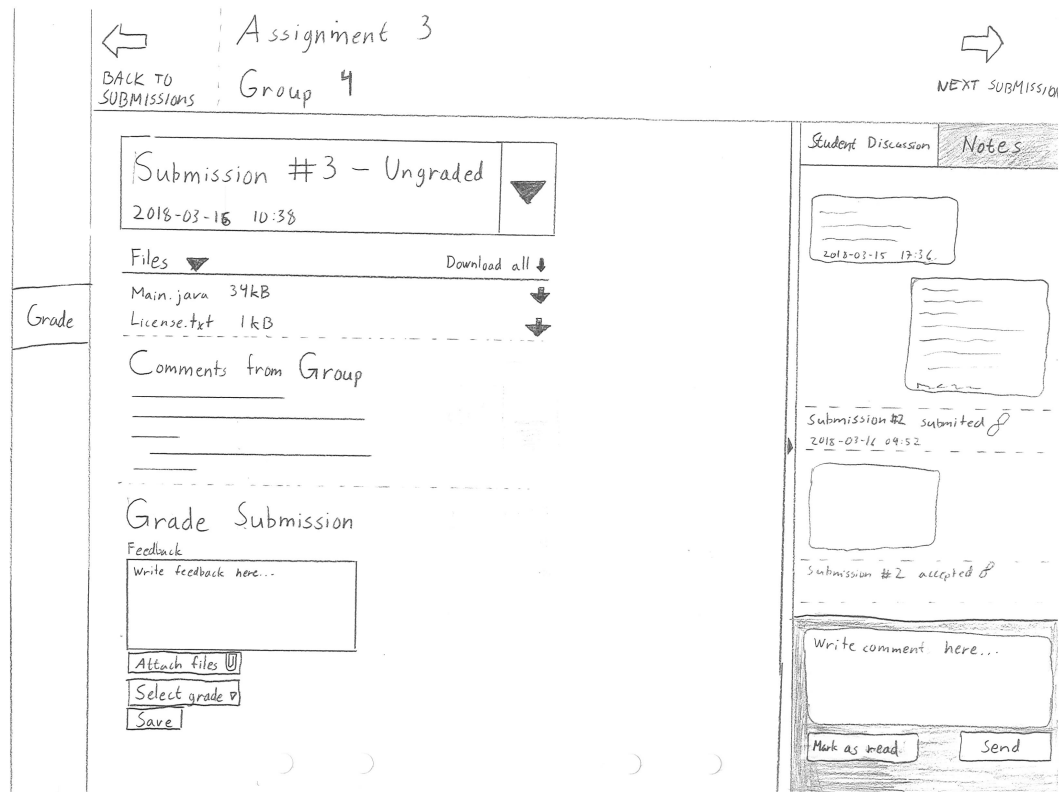


Figure E.1: Grading View

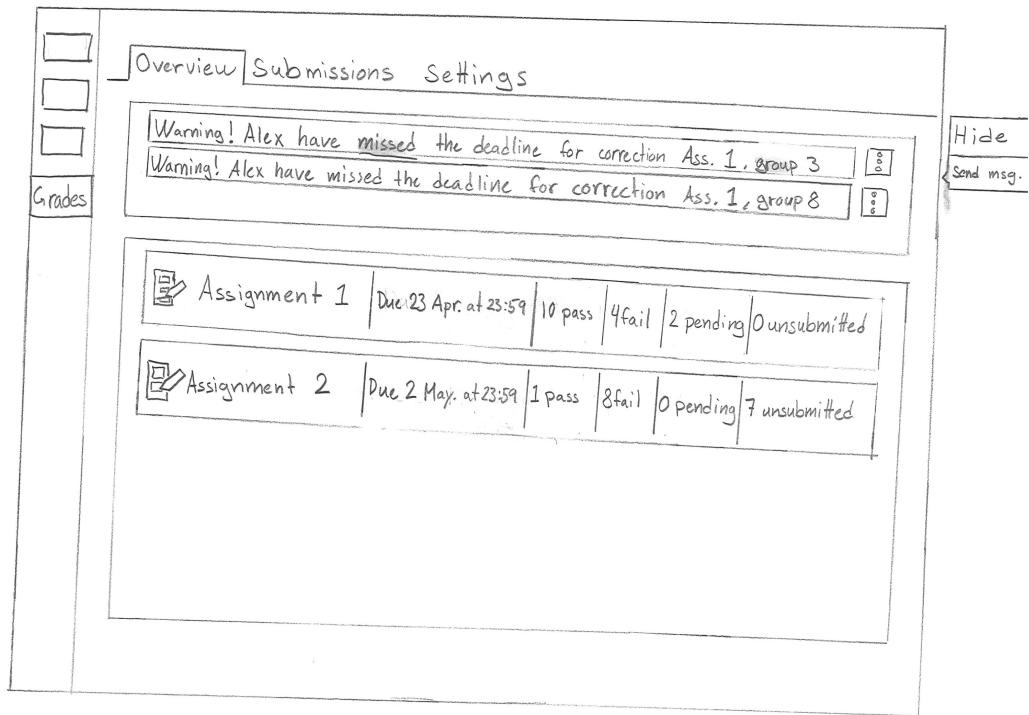


Figure E.2: Overview View

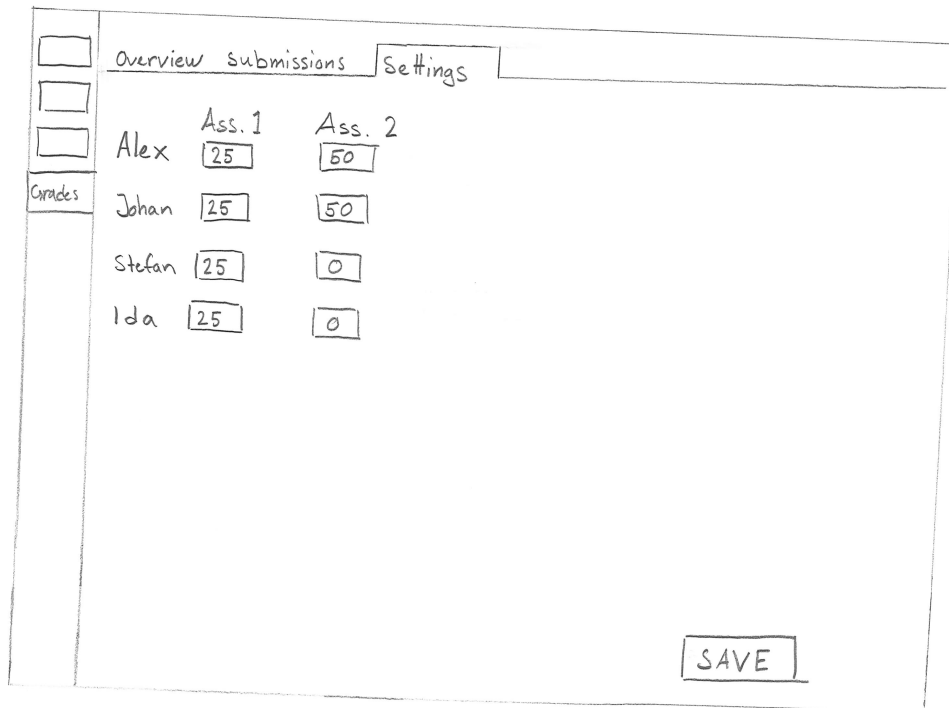


Figure E.3: Settings View

← → C SÅker | https://nordunet.instructure.com/courses/738/assignments/3883

☰ johanTest > Assignments > Grade view

NORDUnet

- Account
- Dashboard
- Courses
- Groups
- Calendar
- Inbox
- Commons
- Help

Home

Announcements

Discussions

Assignments

**Grading**

People

Modules

Chat

Fire-local



Fire-server

## Submissions

Only show group assignments with TODO's

Your submissions

Sort by: Priority

<input type="checkbox"/>	Assignment	Group	Status	Submission time	Comment time	TODO	Content
<input type="checkbox"/>	<b>Assignment 2</b> Download files (2)	DAT255 - 14	Ungraded	49 hours ago (1 submissions)	9 hours ago (4 comments)	Read comments, Grade submission	 
<input type="checkbox"/>	<b>Assignment 2</b> Download files (2)	DAT255 - 13	Accepted	12 hours ago (3 submissions)	6 hours ago (2 comments)	Read comments	
<input checked="" type="checkbox"/>	<b>Assignment 1</b> Download files (4)	DAT255 - 6	Rejected	35 hours ago (2 submissions)	(0 comments)		

Download (1) selected submissions

## Other TA's submissions

<input checked="" type="checkbox"/>	Assignment	Group	Status	Submission time	Comment time	TODO
<input checked="" type="checkbox"/>	<b>Assignment 2</b> Download files (2)	DAT255 - 25	Rejected	2 hours ago (1 submissions)	2 hours ago (1 comment)	
<input checked="" type="checkbox"/>	<b>Assignment 2</b> Download files (2)	DAT255 - 30	Accepted	12 hours ago (3 submissions)	6 hours ago (2 comments)	
<input checked="" type="checkbox"/>	<b>Assignment 1</b> Download files (4)	DAT255 - 36	Ungraded	20 hours ago (2 submissions)	(0 comments)	
<input checked="" type="checkbox"/>	<b>Assignment 1</b> Download files (4)	DAT255 - 36	Ungraded	17 hours ago (2 submissions)	(0 comments)	

Download (4) selected submissions

Figure E.4: Submissions View

# F Custom Design Concept

*This appendix shows the complete design for a custom design concept that was created using the method in chapter 3.4.2. The design extends and improves the functionality already available in Canvas by adding an LTI with custom views.*

## F.1 Navigation

This part describes the overall structure of the custom LTI, both within itself and in relation to Canvas.

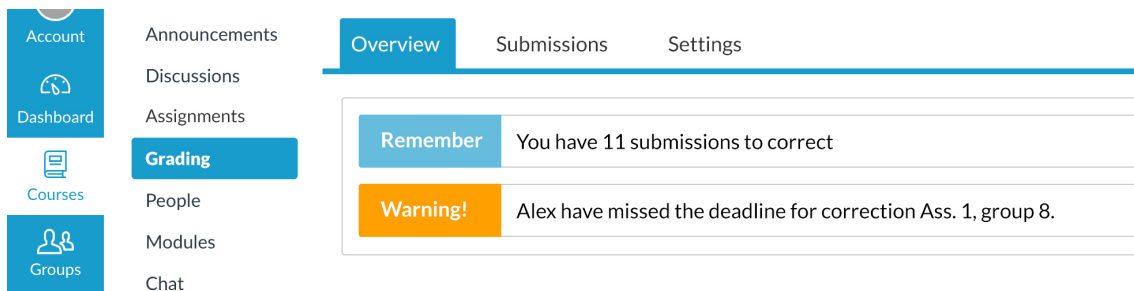


Figure F.1: The Navigation of the Custom LTI

### F.1.1 Placement in Canvas

When the custom LTI is loaded in Canvas, a new menu item is available in the course menu called "Grading". It is visible at the left side in figure F.1. The already existing menu item "Grades" will still be available and relevant to the user as it provides a view of the final result of every individual student. The new "Grading" item will instead have a focus on groups, which is more relevant when grading submissions.

### F.1.2 The different tabs

Within this new section, three different tabs are available, "Overview", "Submissions" and "Settings". The tab-selection is visible in the top right of figure F.1.

By only adding one menu option in Canvas and instead have the three options as tabs within, all the new UI is collected in one place. It also makes it easier to display different functionality conditionally.

The three tabs are designed to be relevant to different users. This means that the customization of the UI for different types of users can be done exclusively by

choosing which tabs should be visible, without having to change the content of any tabs.

Students will not have access to anything since they can not do any grading, preferably hiding the menu item in Canvas altogether. Teacher assistants will only have access to the "Submissions"-tab as it holds everything they need in order to correct submissions. Since they only need one tab, the actual tab bar at the top should be removed.

For the course responsible, all three tabs are relevant and should be displayed.

### F.1.3 The flow

In total, the new designed UI within the LTI holds 4 different views. A chart of the flow between them is presented in figure F.2 (Note that this chart only applies to course responsables which has access to all different views).

Also please note that all of these views are exclusive to graders. Students will submit assignments, send comments, etc, through the standard Canvas interface.

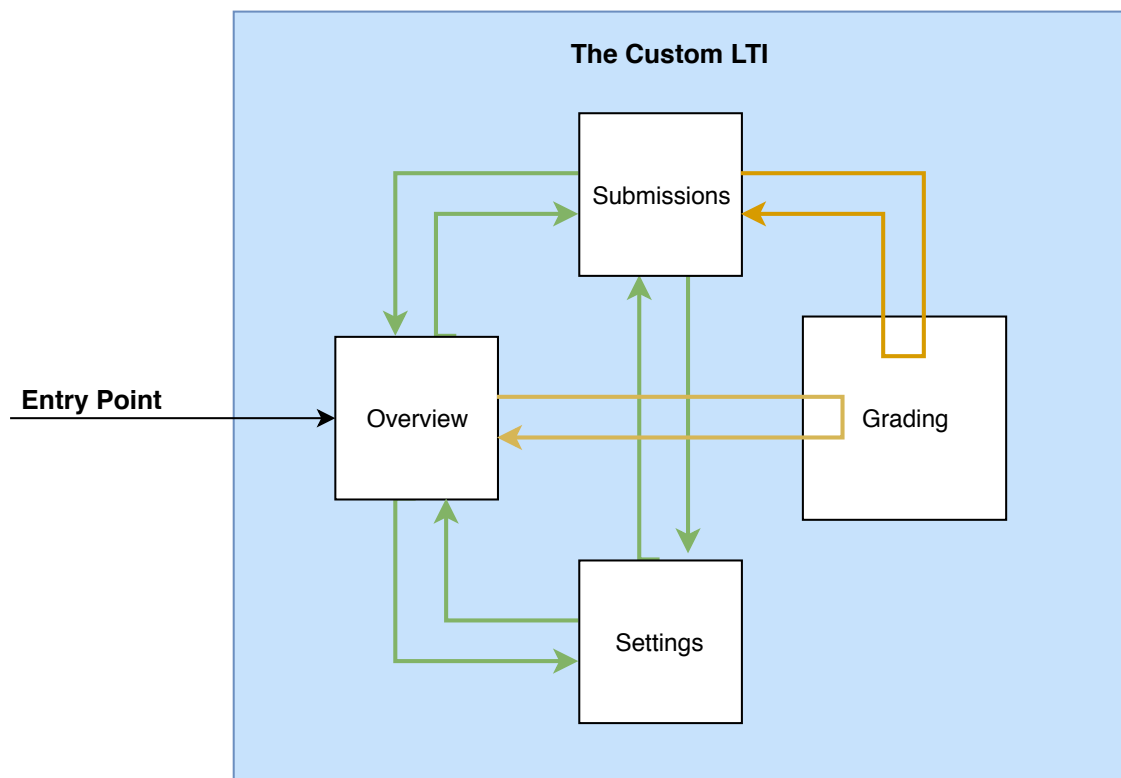


Figure F.2: The flow of the custom LTI for course responsables

The entry point is when clicking the "Grading" menu option in Canvas. The "Overview View", "Submissions View" and "Settings View" are all displayed in their respective tabs. The "Grading View" does not have its own tab, but is instead opened within one of the existing tabs. This means that upon entering the "Grading

View”, the list of tabs in the top is not available anymore. This to make sure the user understands that it is not actually in the selected tab anymore. Instead a click on the back button is required in order to return (see in figure F.5 how the tabs are not visible, but instead a large back button is available in the top left corner).

Note that both the Overview tab and Submission tab shows all submissions to the users, allowing them to enter the ”Grading View” for any submission. However, their purpose, and therefor layout, differs drastically. The Overview is supposed to give a high-level look at how different parts of the course are going, if any assignment is particularly hard, which groups are falling behind, how a certain grader is doing, etc.

The Submission View instead focuses on what needs to be done by you from a grading standpoint. Emptying the list in the ”Submissions View” means that you have completed your grading work, and can focus on other things.



## F.2 Overview View

The screenshot shows the 'Overview' tab in the Canvas LMS interface. The left sidebar contains navigation options: Home, Announcements, Discussions, Assignments, Grading (selected), People, Modules, Chat, Fire-local, and Fire-server. The main content area is titled 'johanTest > Assignments > Grade view' and has three tabs: 'Overview' (selected), 'Submissions', and 'Settings'. Under the 'Overview' tab, there are two information boxes: a blue 'Remember' box stating 'You have 11 submissions to correct' and an orange 'Warning!' box stating 'Alex have missed the deadline for correction Ass. 1, group 8.' Below these is a 'Status' section with a table showing assignment statistics and a list of groups with their submission status.

Status						
By Assignment / By Group / By Grader						
<b>Assignment 1</b>	Due 2 may 23:59	15 pass <a href="#">More info &gt;</a>	3 fail <a href="#">More info &gt;</a>	3 pending <a href="#">More info &gt;</a>	1 unsubmitted <a href="#">More info &gt;</a>	All Groups (22) <a href="#">More info &gt;</a>
<b>Assignment 2</b>	Due 2 may 23:59	1 pass <a href="#">More info &gt;</a>	8 fail <a href="#">More info &gt;</a>	3 pending <a href="#">More info &gt;</a>	7 unsubmitted <a href="#">More info &gt;</a>	All Groups (19) <a href="#">More info &gt;</a>

Group #2	Group members (3) > Anders Södergren Tobias Angerer Johan Olsson	Graded by: Artem Popov <a href="#">Change grader</a>	Pending submitted: 6 hours ago
Group #4	Group members (2) >	Graded by: Artem Popov <a href="#">Change grader</a>	Pending submitted: 18 hours ago
Group #5	Group members (3) >	Graded by: Artem Popov <a href="#">Change grader</a>	Pending submitted: 15 hours ago

Figure F.3: The Overview tab

The Overview tab (presented in figure F.3) is pre-selected for course responsables, making it the first thing they are presented with when they select "Grading" in the Canvas course menu. This tab is supposed to give an overview of how both the correction work is going and how the student groups are doing at the assignments.

### F.2.1 Information boxes

At the top of the tab, a list of information boxes is shown to draw attention to urgent matters. Note that the number of information boxes will depend on the situation, where it is common with no boxes displayed at all. There are two different types of boxes that can appear:

#### Missed correction deadline

When a correction deadline has been set up in the "Settings"-tab (F.4), a warning is shown whenever a staff member has not corrected a submission within the set time frame. This allows the course responsible to take action if necessary. Clicking on the information box takes the user to the particular submission.

The information box will disappear whenever the submission has been graded.

## Correction reminder

A reminder box is shown whenever the course responsible has correction work to do, with a reminder to switch to the submission tab. Clicking it takes them to the submissions tab directly as well. The box displays both the number of comments and ungraded submissions by counting the items in the "TODO"-column available in the Submissions tab (F.3).

## F.2.2 Status list

Below the information boxes is a list with the purpose of displaying how different parts of the course are going. At the top there is a selection between "By Assignment", "By group" and "By Grader". This changes the way the list is displayed. "By assignment" is default when entering the Overview.

Please note that the three different options display the same information, only organized in different hierarchies.

### By Assignment

"By Assignments" shows the different assignments at the top level in the list, as seen in figure F.3. Clicking on the assignment name takes the user to the corresponding assignment in the Canvas interface to change any of its Canvas-related settings, like its instructions or deadline.

Next to the assignment name is a breakdown of the distribution of groups and their current status within the assignment. Clicking on either of the different statuses ("Pass", "Fail", "All groups", etc) expands the box to show a list with all groups with the selected status. Note the example selection of "3 Pending" on Assignment 2 in figure F.3, which then displays the three groups with a pending status.

Each of the groups in the expanded list holds several pieces of information, like who is grading the submission, the time waiting and its current status. A button is provided for the course responsible to manually change the automatically assigned grader.

Clicking on the group name takes the user to the relevant submission in the "Grading View" (F.5). Clicking on the group members extends a list showing all the members.

### By Group

"By Group" shows the different assignments at the top level in the list. Clicking on the group name takes the user to the group settings within the Canvas UI.

Next to the group name is a breakdown of the different statuses. Clicking on either of them expands the box to show a list with all submissions with the selected status.

Each of the submissions in the expanded list holds the same information as the other modes. Clicking on the assignment name takes the user to the relevant submission in the "Grading View" (F.5).

### **By Grader**

"By Grader" presents the different graders at the top level of the list, with their assigned submissions being expandable by clicking on the different statuses. The expandable items holds the same information as the other modes, with the ability to enter the "Grading View" (F.5) for a certain submission.

## F.3 Submissions View

Assignment	Group	Status	Submission time	Comment time	TODO	Notes
<input type="checkbox"/> Assignment 2 Download files (2)	#14	Ungraded	49 hours ago (1 attempts)	9 hours ago (4 comments)	Read comments, Grade submission	
<input type="checkbox"/> Assignment 2 Download files (2)	#13	Accepted	12 hours ago (3 attempts)	6 hours ago (2 comments)	Read comments	
<input checked="" type="checkbox"/> Assignment 1 Download files (4)	#6	Rejected	35 hours ago (2 attempts)	(0 comments)		

Download (1) selected submissions

Assignment	Group	Status	Submission time	Comment time	TODO
<input checked="" type="checkbox"/> Assignment 2 Download files (2)	#25	Rejected	2 hours ago (1 attempts)	2 hours ago (1 comment)	
<input checked="" type="checkbox"/> Assignment 2 Download files (2)	#30	Accepted	12 hours ago (3 attempts)	6 hours ago (2 comments)	
<input checked="" type="checkbox"/> Assignment 1 Download files (4)	#36	Ungraded	20 hours ago (2 attempts)	(0 comments)	
<input checked="" type="checkbox"/> Assignment 1 Download files (4)	#36	Ungraded	17 hours ago (2 attempts)	(0 comments)	

Download (4) selected submissions

Figure F.4: The Submissions Tab

This submissions tab (presented in figure F.4) shows all the different submissions in the course in a way that is optimized for focusing on what needs to be done by the current user.

### F.3.1 The Lists

The page holds two separate lists of submissions. The one at the top only shows the submissions that has been assigned to the current user for correction, making sure the users clearly know what they are responsible for.

The list below shows the submissions the other staff member has been assigned to correct. This allows the user to help its colleagues and correct submissions it is not assigned to (the user still has to correct all of its own assigned submissions). The submissions assigned to other staff is collapsed by default and can be expanded by the arrow next to the headline (it is expanded in the figure).

In the top of the tab, two settings for how the lists are displayed is available. First is the ability to select in what way the submissions are sorted. By default it is

sorting on priority which sorts based on the TODO's as well as the time the submission/comment was submitted. A new comment places itself high so student's gets the answers they need as fast as possible. Other sorting modes are available as well, for example based on the type of assignment, allowing the user to only focus on correcting labs of the same type.

The second setting, which is turned on by default, is the "Only show submissions with TODO's". This hides all assignments where nothing new has happened. This allows the user to only focus on the submissions that needs attention.

### **F.3.2 Submission Item**

Each of the submissions in the list contains certain necessary information; which assignment the submission relates to, which group submitted it, its current status, when it was submitted, when it was commented last and the TODO's. To the far right there is access to the submitted files through the file icon. Also, whenever notes exists on group assignment, a notes icon will appear, which when clicked shows a small preview of the notes (the notes are further explained in F.5.3). This allows the staff to communicate either with themselves or other by being able to leave any information of their choice.

Clicking on the assignment text takes the user to the "Grading View" (F.5).

Below each of the lists of submission is a download all button which downloads the submitted files for all submissions at once. Boxes to the left of the submission also allows you to only download a certain set of submissions. This reduces the manual labour for staff who likes to correct a set of submissions at the same time.

Whenever the correction deadline has passed, the submission time is highlighted in red to alert the user that it has taken too long.

Whenever a group submits a new submission, the TODO "Grade submission" appears on that submission. It stays there until it has been graded. Whenever a new comment is made on any group assignment, the TODO "Read comments" appears on that group assignment. It stays there until the user either marks it has read or writes an answer in the "Grading View" (F.5).

## F.4 Settings View

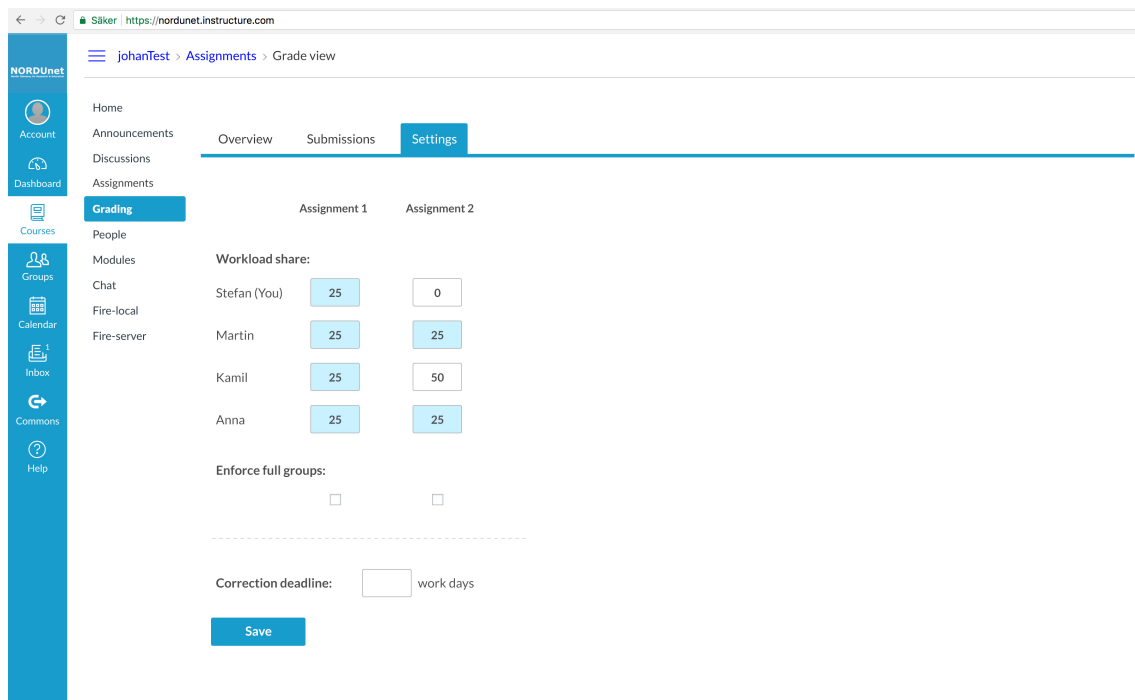


Figure F.5: The Settings Tab

The settings tab (presented in figure F.5) holds only the settings related to the LTI. Other settings related to the course or the assignments themselves will still only be accessible through the Canvas UI. Most of the settings are divided based on assignment in order to allow control over the assignments independently.

### F.4.1 Workload distribution

First at the top there are the settings regarding the workload distribution. By default, the workload is split equally between the course responsible and all added TA's. A blue box means that the value is automatically assigned. As shown in the first column (showing the workload for assignment 1), there are 4 people correcting, so each of them gets 25%. In the second column (assignment 2), a manual value of 0 and 50 has been set to Stefan and Anna respectively. To show that the values has been set manually, they are colored white. The rest of the workload, 50%, is equally split between the remaining automatic (blue) boxes, assigning each with 25

### F.4.2 Enforce full groups

Below is the setting for enforcing full groups. In Canvas it is only possible to set an upper limit in the number of group members. Even if all groups have a set size of 5, it is possible to be alone and still submit. Therefore this checkbox is needed

for course responsible who would like the groups to be the number they set exactly. Then if someone needs an exception, like allowing 4 or 6, the course responsible can easily change the size of that group in Canvas. As long the enforce group setting is checked, they can only be the number picked by the teacher. For teachers who allow any size up to a certain limit, the functionality can be left unchecked, and Canvas will behave as default.

Whenever the enforce full groups is activated, the LTI will create an automatic reject on any submission if the submitted group does not hold the required number of members. The rejection message explains why it was rejected and prompting the group to either change their number of members to the required number, or contact the course responsible.

### **F.4.3 Correction deadline**

The Correction Deadline setting allows the course responsible to set a goal for how fast submissions should be corrected. Whenever this deadline time is exceeded, the course responsible will be notified with an information box in the "Overview View" (E.2. The assigned grader will see if the correction deadline has been passed in the "Submissions View" (F.3 as well, knowing the course responsible has been notified.

By default the correction deadline is not set, meaning no information boxes or missed deadlines will be displayed for any user regarding the correction time.

## F.5 Grading View

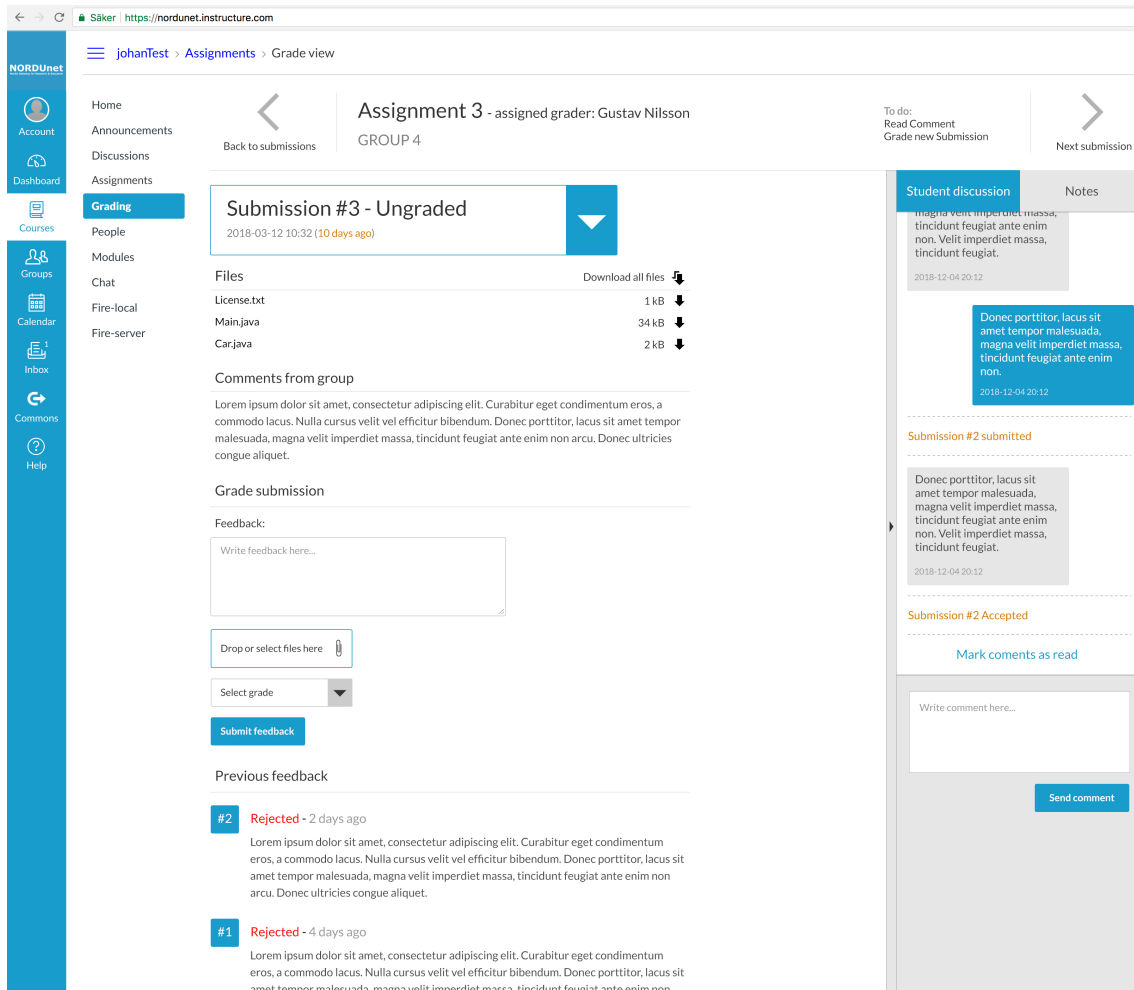


Figure F.6: Grading view

The grading view is where the actual submissions and all related information is shown. The screen estate is split into three distinct parts.

### F.5.1 Top bar

The Top Bar presents the user with key navigation and information. To the left is the back button, returning the user to the previous view. The text below the arrow is contextual stating where the user came from (either to the Overview, or the Submissions).

Next to the back button is information about which group and assignment the submission relates to, as well as who is responsible for the grading. Next to it is an overview of the TODO, the same as the TODO-column in the Submissions View (F.3). This gives the user the ability to, at a quick glance, see what needs to be done regarding this particular submission. This is of extra importance when the



user clicks the "next submission"-arrow to the far right. It takes the user to the next submission in line based on the selected sorting in the previous view. Since the user will be presented with a new submission without having selected it manually from the list, the TODO allows it to see what needs to be done.

Note that the next submission arrow becomes disabled when the end of the list is reached.

## **F.5.2 Center Part**

The center part is where the actual submission and grading is shown. At the top there is a large dropdown allowing the user to look at different submissions for the same assignment. The dropdown button holds key information as which attempt it is, as well as submission time. The latest submission is selected when entering the Grading View.

### **The Submission**

Below the dropdown is the selected submission displayed. First are the submitted files presented. They can be downloaded individually or all at once. Below them is the comment written by the group during submission.

### **The Grading**

Below the submission is where the actual grading is performed. Provided is a feedback box where the grader can write its feedback. Below the textbox is also the ability to upload feedback files, which can consist of a sketch or other helpful information for the students.

Under the feedback, the grader chooses a grade (based on the selected grade type for the assignment in Canvas), with a "Submit"-button below. When clicking the "Submit"-button, the system asks the user for confirmation before submitting, to make sure it was not by accident. This warning mentions the lack of feedback if there is no text in the feedback textbox. Note that the user is unable to submit at all without selecting a grade, and is instead prompted to provide one when trying to submit.

Below the grading tools is the feedback from previous attempts shown. This to help the grader get a quick overview of the history of the submissions and why it was previously rejected. By having all feedback collected in one place, the user doesn't have to switch between submissions in the top dropdown. Note that only the actual feedback from previous attempts is shown. No extra information, such as time, files, etc is provided.

### F.5.3 Right Pane

The right pane holds two separate functions, located in two different tabs at the top.

#### Student discussion

The student discussion panel is for allowing the students and staff to ask questions to each other regarding the particular assignment or submission. Every comment made in this group assignment is available in chronological order regardless of during which attempt the comment was made.

The list of comments always starts at the bottom showing the newest comment. Scrolling upwards reveals older comments. The event of new submissions are shown in the list among the comments to clearly show during which attempt the comments were made. Keeping all comments in one big flow allows conversations to be taken over the course of several attempts without any separation.

Below the list is a textbox for writing a new comment, as well as a button for sending the comment. Whenever someone other than the assigned grader writes a comment in the student discussion, a "Read comments" TODO is added to the submission.

When a submission has a "Read Comment"-TODO, another button dubbed "Mark as Read" becomes available next to the "Submit Comment"-button. Either clicking "Mark as Read"-button or writing a comment removes the "Read Comment" from the TODO, and therefore also removes the "Mark as Read"-button.

Note that only the assigned grader can mark the comments as read in order to remove it from the TODO-list. A different staff member can not, making sure the assigned grader gets notified of what is written.

#### Notes

In the second tab there is a place for staff members to leave notes on the submission. These notes are only visible for other staff members, not students.

The notes is a single block of text. This means that it can be utilized in any way that fits the workflow of the staff using the system. It can be used for reminders, general instructions, or any other piece of information. Since the notes are available in both the "Overview View" and the "Submissions View," it provides easy access information about the submission for all staff.

The tab itself consists of a large textbox with a "Save"-button. The textbox holds the current notes for the group assignment. The notes can be edited directly in the textbox, and saved by clicking "Save". Note that this note is the same regardless of the selected submissions in the submission dropdown.

## F.6 About the implementation

The LTI needs to be hosted on a server. The LTI is then loaded whenever the new "Grading"-section is opened or directly interacted with by a user.

In order to automatically reject submissions from groups with unaccepted group sizes, a web hook needs to be implemented where the LTI is notified whenever a new submission is sent. Other than that, the LTI needs no notifications in order to be run, and can simply be fetched whenever a teacher requests to use the UI by clicking the "Grading"-section in the course menu.

There are a few pieces of information that will have to be stored in a database outside of Canvas, data that is not already available.

- **Workload** - The workload for each staff member for each assignment (as decided in the "Settings View").
- **Group Size** - The enforcement of group size for all different assignments within the course (as decided in the "Settings View").
- **Assigned Grader** - Which group and assignment each staff member is responsible for grading (which is assigned automatically based on the selected Workload).
- **Correction Deadline** - If the course has correction deadline, and if so, how long it should be (as decided in the "Settings View").
- **Notes** - The notes on each group assignment.
- **TODO's** - If there are unread comments in each group assignment.

The rest of the data can be fetched from and updated directly to Canvas, such as submissions, assignments, comments, etc.

# G Canvas Tutorial

## Abstract

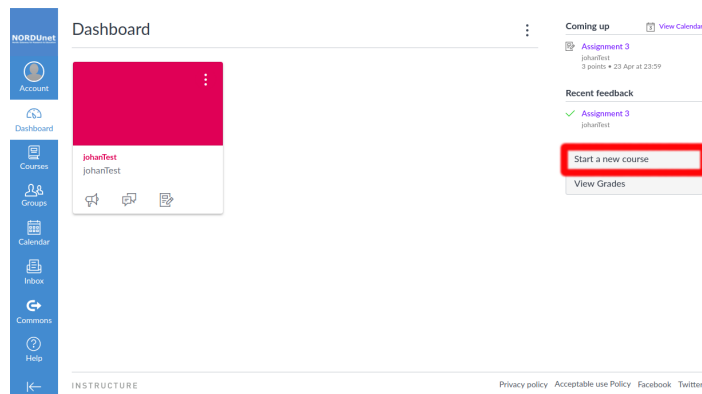
This selection of tutorials are meant to bring course responsables a quick introduction in using Canvas as a lab submissions system. Each chapter focus on a specific function that loosely correspond to a requirement that the course responsables at the Department of Computer Science and Engineering have expressed in interviews during this study.

## Contents

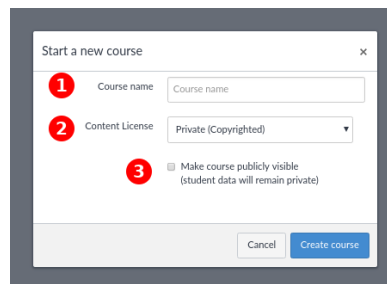
<b>1</b>	<b>Create new course</b>	<b>2</b>
<b>2</b>	<b>Create a copy of a course</b>	<b>4</b>
<b>3</b>	<b>Create new Assignment</b>	<b>7</b>
<b>4</b>	<b>Grading assignment</b>	<b>13</b>
4.1	Gradebook . . . . .	13
4.2	SpeedGrader . . . . .	16
<b>5</b>	<b>Modules</b>	<b>18</b>
<b>6</b>	<b>Add Teacher Assistant to course</b>	<b>21</b>
<b>7</b>	<b>Sections</b>	<b>23</b>
<b>8</b>	<b>Add Learning Tools Interoperability (LTI)</b>	<b>24</b>
<b>9</b>	<b>Send messages</b>	<b>25</b>
<b>10</b>	<b>Student view</b>	<b>28</b>
<b>11</b>	<b>Notes on a student</b>	<b>29</b>
<b>12</b>	<b>Navigation</b>	<b>31</b>

# 1 Create new course

You can create a new course from the dashboard. New courses are created as empty course shells and you will be added as course instructor automatically. As shown in figure 1 below, picturing the dashboard, the first step to create a new course is to press the button **Start a new Course** in the right sidebar.

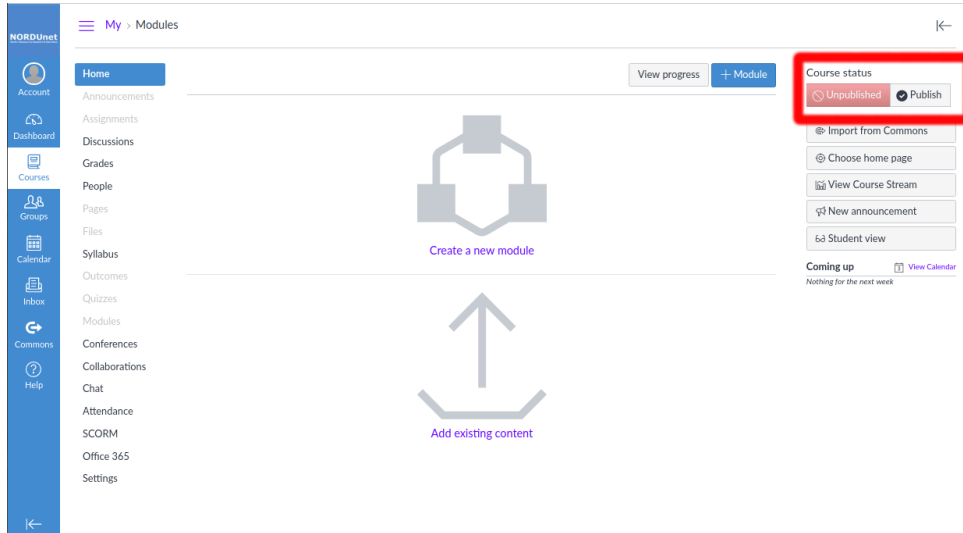


When this step is done, there will be a new pop-up window opened.



1. Enter the name of the course
2. Select content license. This will be private (copyrighted) by default. If you want your course content to be able to be reused, you can change it to **public**. You will be able to change this in the settings of the course later on.
3. Check this box if you want to make the course publicly visible. This means that everyone with the URL to the course site will have access to the course, student data will remain private. You will be able to change this whenever you want in the course settings.

Press **Create course**. This will create an empty course shell for you, then you will be able to add content and edit the course.

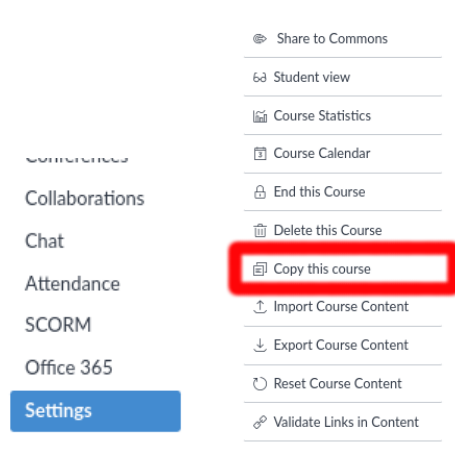


This is how your course site will look like when new. To make the course available you need to press **Publish** in the top right corner. When unpublished no one will be able to access the course.

## 2 Create a copy of a course

There is a few ways you can create a copy of an old course, two ways are explained below.

First navigate to the course you want to copy. Press the **Settings** button on the course menu on the left. This will take you to the settings page of the course. Next you need to press the **Copy this course** on the menu on the right side of the page.



## Copy My Course

Please enter the details for the new course.

Name  1

Course code  2

Start date  3

End date  3

Content  All content 4  
 Select specific content

Adjust events and due dates 5

Date adjustment  Shift dates 6  
 Remove dates

Beginning date  change to  7

Ending date  change to  7

Move from  to  8

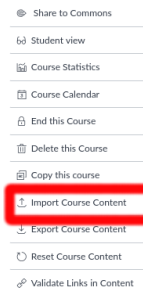
---

9

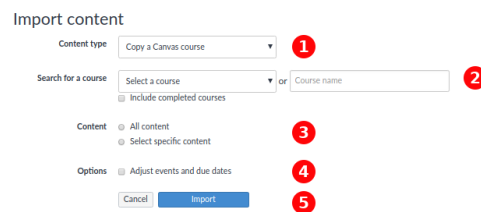
1. Enter the name of the new course
2. Enter a course code for the new course
3. Set start and end date
4. Choose if you want to copy all the content from the course or select some specific content
5. select if you want to adjust events and due dates. If this is not filled, step 6-8 will not be visible and can be ignored
6. Select if you want to shift or remove the dates
7. Adjust new dates
8. Move events or due dates from one weekday to another
9. Press **Create course**



The second way to create a copy of a course is to import content from another course. To do this you need to create an empty course shell as described in section 1 and navigate to the settings page of this course.



Press the **Import Course Content** on the right side menu. This will bring up a new window where you will adjust the settings for your imported content.



1. Select content type. In our case we choose **Copy a Canvas course**
2. Select either the course to copy in the dropdown menu or enter the name of the course in the textbox
3. Select if you want to copy all content from the course or just some specific content
4. Adjust events and due dates of the course
5. Press **Import**

### 3 Create new Assignment

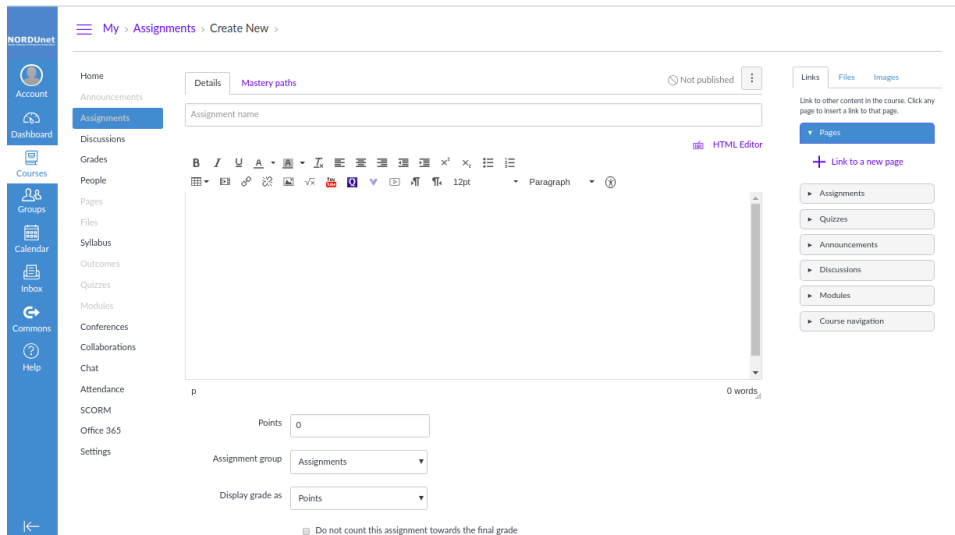
Creating and editing course assignments is a fundamental part of any LMS system. This is how it is done in Canvas.

- Home
- Announcements
- Assignments**
- Discussions
- Grades
- People

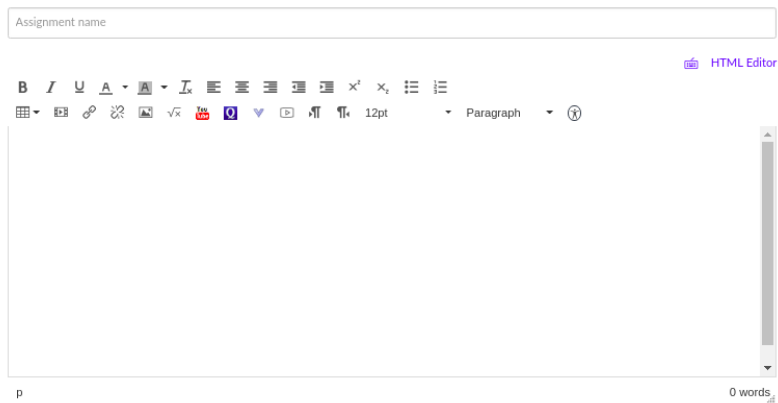
The first thing you need to do is to navigate to the course page, in which you want to create an assignment, and then press the **Assignment** button on the course menu on the left side of the page. After this press the **+Assignment** button on the top right of the page.

**+ Assignment**

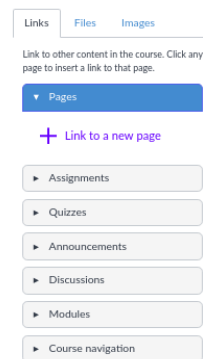
This will open up a new page with all settings for the assignment. Note that assignment settings are persistent to always remember and display the settings created or edited in the previous assignment on the course.



In the following section an explanation for the settings are presented.



On the top of the settings page you will enter the name of the assignment. This is the name of the assignment the students will see when visiting the course. Just below the assignment name field you will find something called the Rich Content Editor and this is where the assignment PM is created. You can add text, images, links, equations or media.



Use the sidebar to the right to link to or upload course content, including files and images. You can then include this in the assignment PM.

1 Points

2 Assignment group

3 Display grade as

4  Do not count this assignment towards the final grade

1. Enter the amount of points the assignment is worth
2. Add the assignment to an assignment group. This is to group together a number assignments for better organizing
3. Select how you want the grades to be showed. You can choose between for example complete/incomplete, points or letter grade
4. Select if you want this grade to be counted or not in the final grade of the course

Submission type

**Online entry options**

Text entry

Website URL

Media Recordings

File Uploads

Next thing to do is to select the submission type. In this field you choose the way students submit their solutions. No submission is when you don't want the students to submit an assignment in Canvas. Online is when the student will submit their assignment in Canvas, when this is chosen you will be able to select entry options as shown in the picture above. You can choose on paper if the student will submit the assignment to you but not in Canvas, this is more like the traditional face-to-face approach.

Group assignment

This is a group assignment

Next up is to select if this assignment is a group assignment, if selected a new window will pop up.

1. Enter the name of the set of groups
2. Select if the students can sign up for a group on their own. This way students can organize themselves into groups. Note that as long as this option is enabled, students can move between groups. When enabled you can tell Canvas how many groups it should create and also choose a limit to how many students each group can have
3. If 2 is not enabled you can automatically split the students into the amount of groups you want or create the groups manually
4. Select if a group leader is needed

When **Save** is pressed you will also be able to select if you want the grade to be set individually or as a group.

Peer reviews

Require peer reviews

**How to assign peer reviews**

Manually assign peer reviews

Automatically assign peer reviews

Reviews per user

Assign reviews

Must come after due date. If blank, uses due date.

**Anonymity**

Peer reviews appear anonymously

The next option is whether or not this assignment should be peer reviewed. When enabled you can select whether to manually or automatically assign peer reviews. If automatically is selected you can assign how many reviews per student, assign a due date and make the reviews to appear anonymously.

1  Sync to SIS  Include this assignment's grades when syncing to your school's Student Information System

2  Moderated grading  Allow a moderator to review multiple independent grades for selected submissions

3

Assign

**Assign to**

Everyone ×

**Due**

**Available from**  **Until**

+ Add

4  Notify users that this content has changed

5

1. Select whether or not the grade from this assignment will be included when syncing to your school's Student Information System
2. Select if you want moderators to be able to review the assignment. Moderators need to be added before you can enable this option
3. Assign due date for the assignment. You can choose to make the same due date for everyone or individual due dates for different groups. You can also assign the availability for the assignment
4. Since this is the same page as when you edit an assignment, you can notify all that the content have been changed
5. Lastly you need to **Save** the settings, or if this is a new assignment **Save & Publish** so the students are able to access the assignment

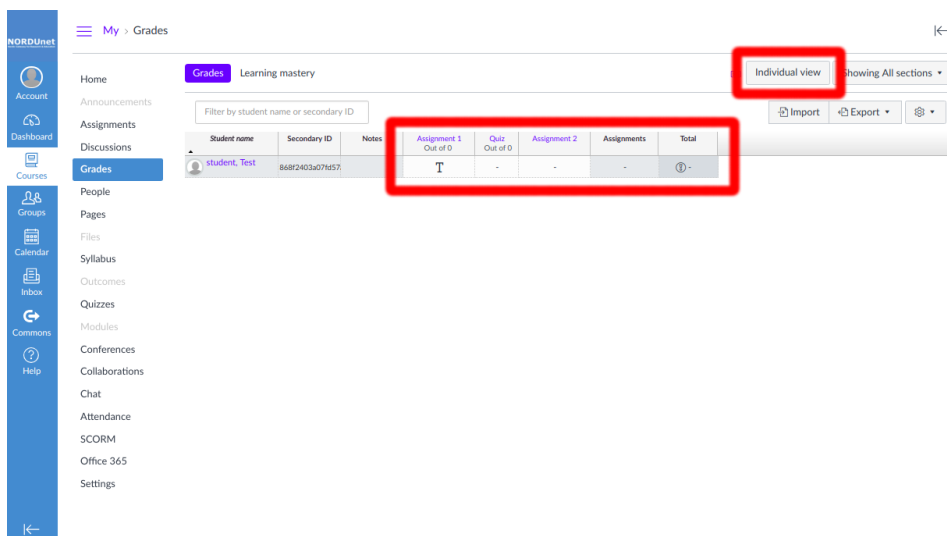
All of these settings are editable by pressing the **Edit** button on the assignment you want to edit.

## 4 Grading assignment

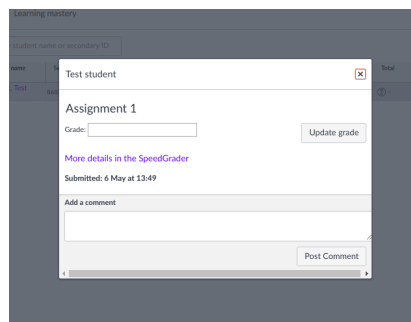
Just as important as creating new assignments is grading assignments. This can be done on different ways. It can be done through the gradebook or through the SpeedGrader. Below is a explanation for both ways.

### 4.1 Gradebook

To find the gradebook, press the **Grades** button on the course menu on the left hand side.

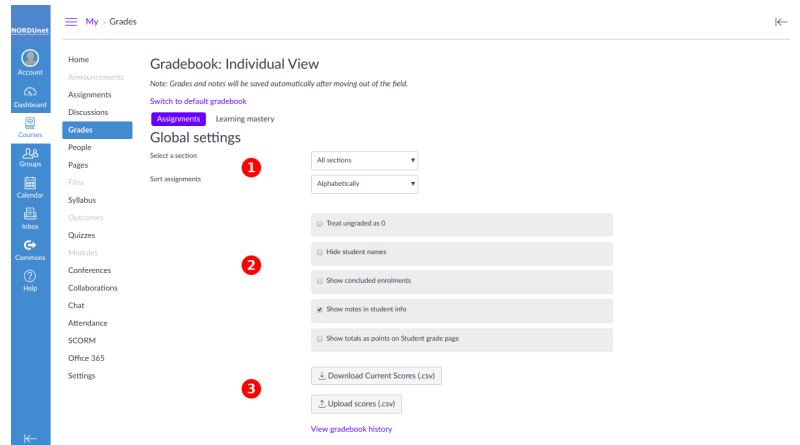


On the gradebook page registered students on the course are listed. The grade can be entered in the textfield of the assignments of a certain student or press the upper right corner of the field which will open up a new window. In this window the grade can be edited and a comment can be made.

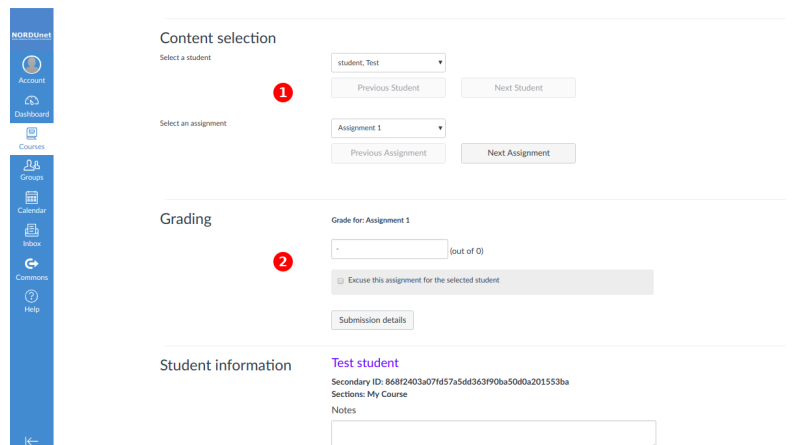




From the gradebook page there is also the possibility to go to the individual view of the gradebook, this is done by pressing the **Individual View** button on the top right of gradebook page.



1. Select section and how to sort the assignments
2. Some extra settings
3. Download or upload .csv files of the grades



1. Select student and assignment
2. Enter the grade, excuse the student from this assignment, **Submission details** will open up the edit grade window again where a comment can be added

Assignments

Final grade: -

Assignment information

**Assignment 1**  
See this assignment in speedgrader

Download all submissions

1 Submission types: Online text entry  
Graded submissions 0

Points possible	Average score	High score	Low score
No points possible	No graded submissions	No graded submissions	No graded submissions

2 Muted?

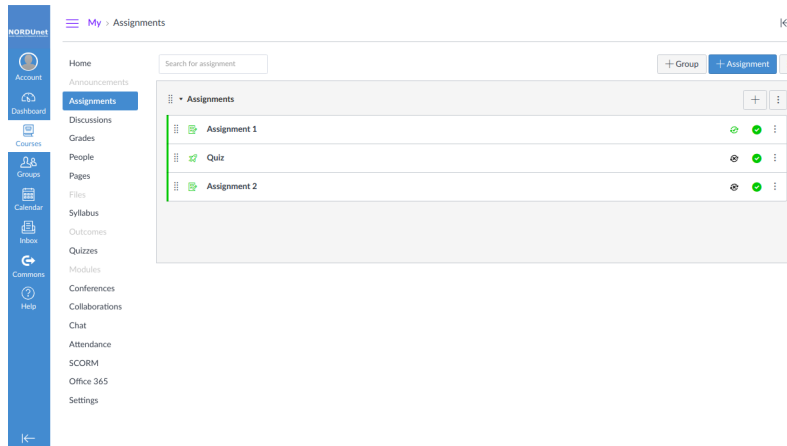
3 Message students who...

Set default grade

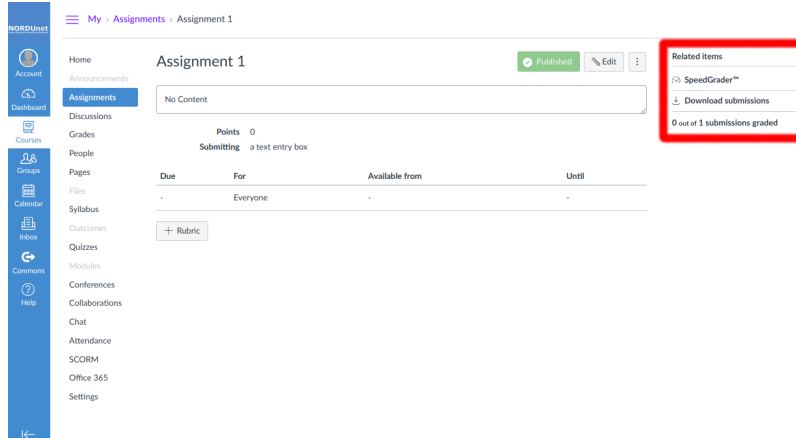
1. Assignment information, download all submissions
2. Send messages to students who for example are late with a submission
3. Set a default grade for the assignment, meaning all students get the same grade

## 4.2 SpeedGrader

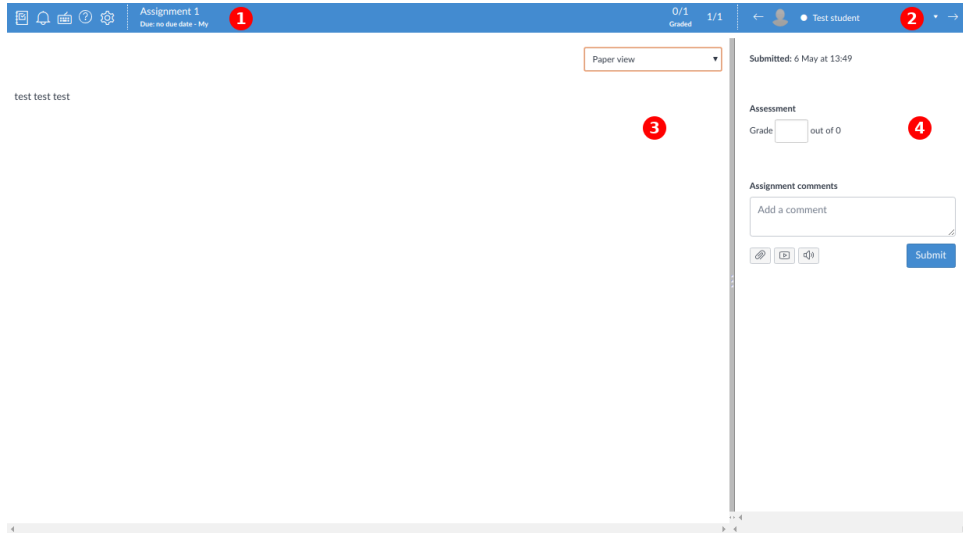
The second way to grade assignments are through an application called SpeedGrader, which will give a good overview when grading assignments. To access this application, navigate to the assignment page on the course menu.



Click on the name of the assignment to be graded.



Click the **SpeedGrader** button on the right sidebar to open up the application, SpeedGrader will be opened in a new window.



1. SpeedGrader menu
2. Name of the current student or group to be graded
3. A field where the submitted file will be seen, can enter comments directly on the file to be visible for students
4. Enter grade and comment, can also attach files or media content

Press **Submit** to save the grade.

## 5 Modules

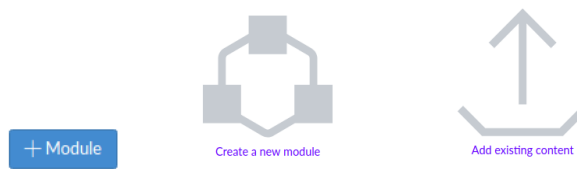
Modules are like chapters in a book. It is a way to organize the content in the course to be easier for both teachers and students to follow. For example a teacher can divide every week into a module where all assignments and lectures for a specific week are added to that module. Modules can be set as prerequisites for each other. The next section will explain how to use modules in a course.

Quizzes

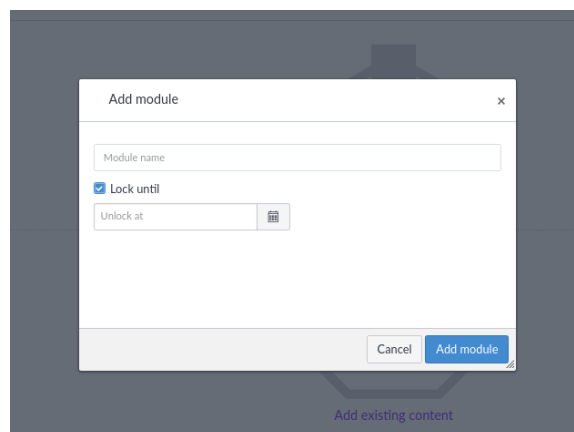
**Modules**

Conferences

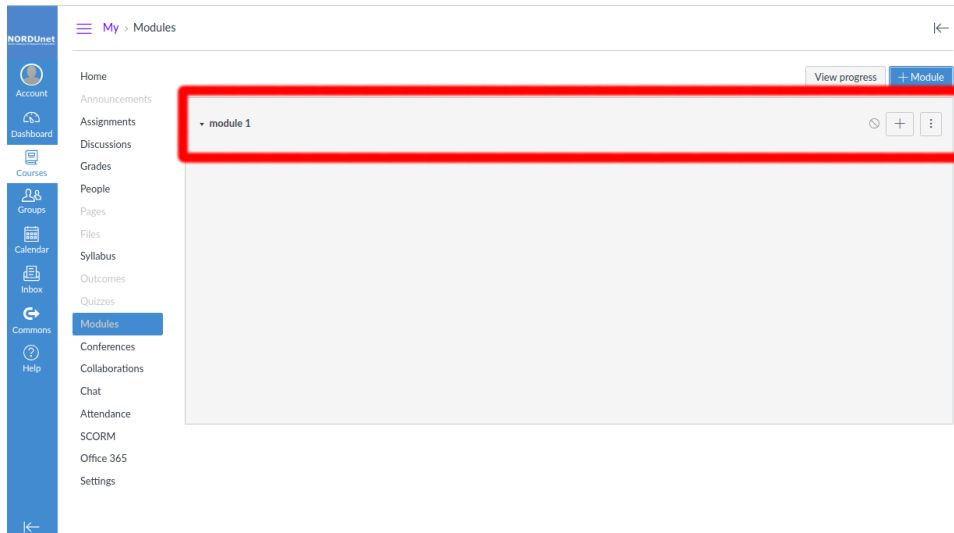
First navigate to the course page and press the **Modules** button. When this is done the modules page will be opened.



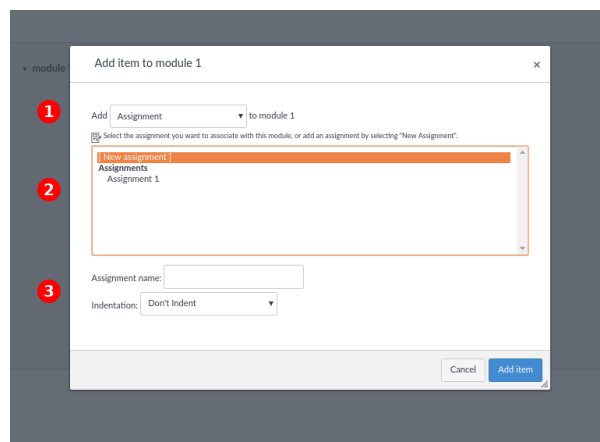
When opening this page for the first time there will be two buttons for creating a new module. The option import existing content, described in section 2, is available as well. Press on **+Module** to create a new module.



A new window will appear where you will enter the name of the module. Also, you will be able to **Lock** the module, meaning that the content will not be available until the date the teacher selects. Press **Add module** to continue.



When a new module is created, an empty module shell will appear on the module page. Press the + button to add content to the module.



1. Select what type of content to add, in this case assignment is selected
2. If an assignment is created, as in section 4, this will appear in the list and can be added. If not, choose **[New Assignment]** and the assignment can be edited later
3. If new assignment is chosen, enter the name of the assignment. Also, select if indentation is preferred

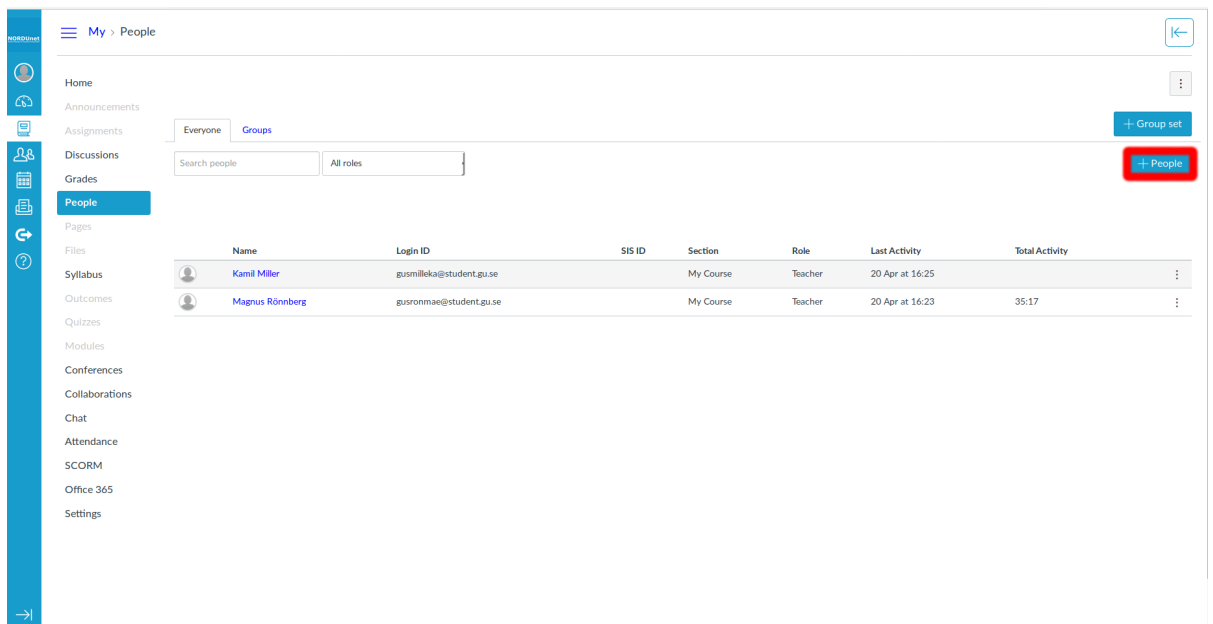
The screenshot displays the 'My Modules' interface in NORDUnet. On the left is a vertical sidebar with navigation icons for Account, Dashboard, Courses, Groups, Calendar, Inbox, Commons, and Help. The main content area is titled 'My > Modules' and includes a 'View progress' button and a '+ Module' button. The modules are listed as follows:

- module 1** (0 pts):
  - Assignment 1 (0 pts)
  - Quiz (0 pts)
  - Web page
- module 2** (Prerequisites: module 1):
  - Lecture notes
  - Assignment 2

There is a list of types of content to be added, for example assignments, quizzes, web pages and lecture notes. Above there are two modules added with different content. Module 2 have module 1 as prerequisite, meaning that students need to finish module 1 before they can start on module 2.

## 6 Add Teacher Assistant to course

A teacher assistant is a helping hand for teachers. One main task a teacher assistant is to grade assignments. To add a teacher assistant navigate to the course page and press the **People** button on the course menu on the left hand side of the page.



The screenshot shows the 'My > People' page in a learning management system. The page has a blue sidebar on the left with various navigation options. The main content area shows a list of people in the course. The table below is a representation of the data shown in the screenshot.

Name	Login ID	SIS ID	Section	Role	Last Activity	Total Activity
Kamil Miller	gusmilleka@student.gu.se		My Course	Teacher	20 Apr at 16:25	
Magnus Rönnberg	gusronmae@student.gu.se		My Course	Teacher	20 Apr at 16:23	35:17

On this page you will be able to see all registered people to this specific course with some extra information. Among this information are for example name, Login ID and role. The next step is to press the **+People** button on the upper left corner. This will open up a new window for adding people to the course.



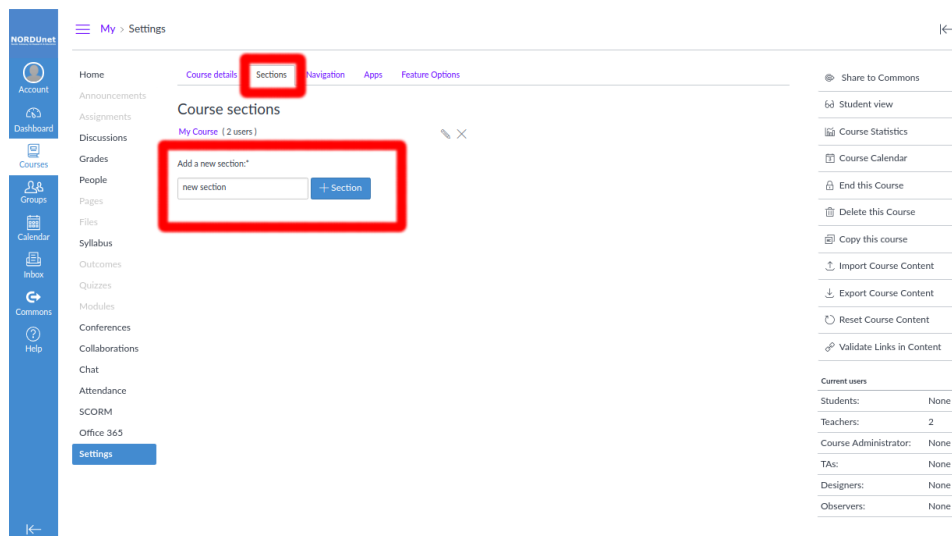
The image shows a 'Add people' dialog box with the following elements:

- 1** 'Add user(s) by' section with radio buttons for 'Email address' (selected), 'Login ID', and 'SIS ID'.
- 2** A text input field for user IDs, with an example: 'lsmith@myschool.edu, mfooster@myschool.edu'.
- 3** 'Role' dropdown menu set to 'Student' and 'Section' dropdown menu set to 'My Course'.
- 4** A checkbox labeled 'Can interact with users in their section only' which is currently unchecked.
- A user icon and the instruction: 'When adding multiple users, use a comma or line break to separate users.'
- 'Cancel' and 'Next' buttons at the bottom right.

1. Select what type of ID the user(s) will be added by
  2. Enter the ID of user(s) to be added
  3. Select role and section of the user(s)
  4. Lock the user(s) to only be able to interact with other users in their section
- Press **Next** to continue.

## 7 Sections

Sections are a way to divide the students into groups. This can be used as a way of handle the workload of the teacher assistants for example, to do this you simply add a teacher assistant to the section that assistant will handle. Either the teacher add the people manually to the sections or they can be added automatically by Canvas.



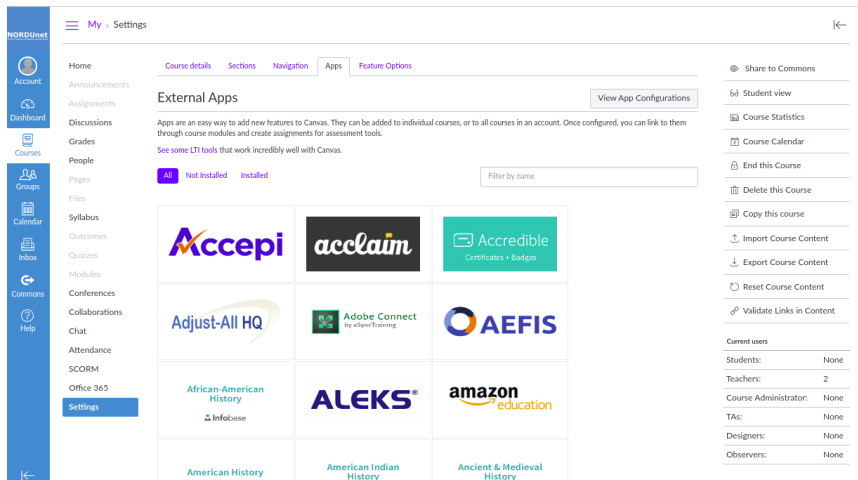
The screenshot shows the Canvas LMS interface for a course. The 'Sections' tab is highlighted in red. Below it, a red box highlights the 'Add a new section' form, which includes a text input field with 'new section' and a '+ Section' button.

Current users	
Students:	None
Teachers:	2
Course Administrator:	None
TAs:	None
Designers:	None
Observers:	None

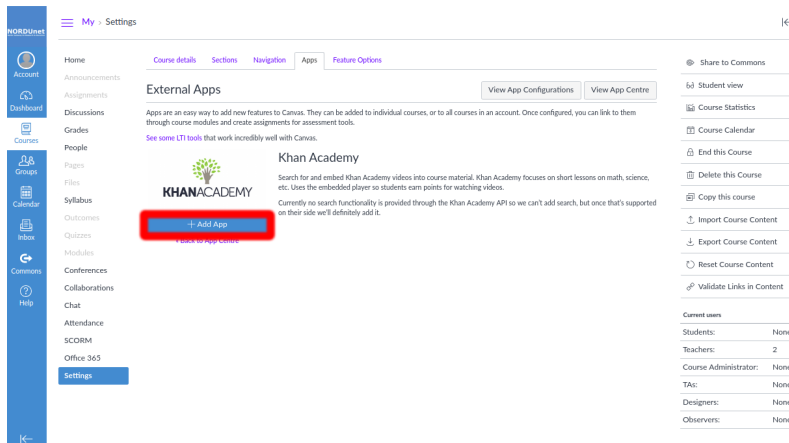
To add a new section you navigate to the course page, press **Settings** button on the course menu and press the **Sections** tab on the top of the page. When this is done you simply enter a name for the section in the text field and press **+Section** button.

## 8 Add Learning Tools Interoperability (LTI)

There is a rich content of external tools with the ability to be integrated in Canvas. These applications can be added through the Canvas App Center or the Edu App Center. Once added, the external application may be able to be added to the course menu, assignments or assignment PM. All this external tools follow a standard called LTI, so the applications are simply called LTI-apps.



To find the Canvas App Center, simply go to the settings page and select the **Apps** tab on the top of the page.

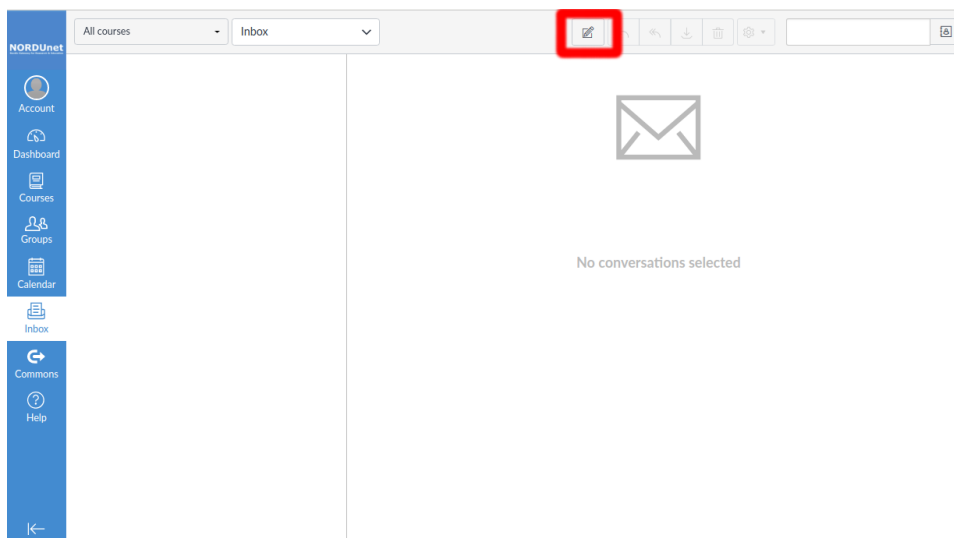


Find the desired application and click it, and then press **+Add App** button. This makes this application available to the course page. Depending on the type of app it will be available on different parts of the course page.

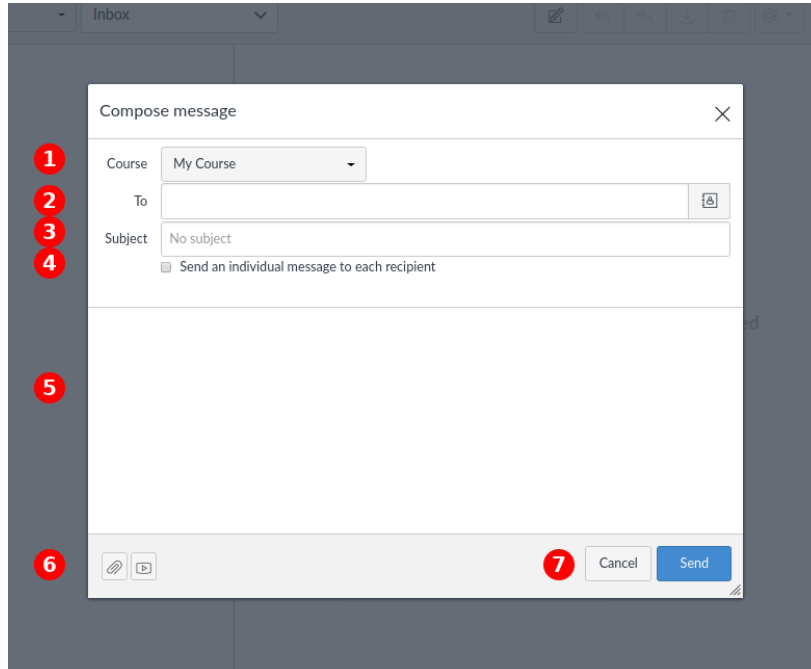
## 9 Send messages



To read and send messages between you and other users of Canvas, you need to navigate to your personal inbox. This is done by pressing the **Inbox** button on the Canvas menu on the far right, this will take you to the inbox page.



To send a new message, press the **Compose a new message** button on the top of the page. This will open up a new window.



1. Select what course or group to send the message
2. Select to whom to send the message. Can be entered manually in the text field or use the button to the right of the text field to choose in a list. Multiple or single people can be chosen
3. Enter the subject of the message
4. Check if the message should be send individually to all users selected, if not the message will be forwarded with all contacts included
5. Enter the message
6. Attach a file, record a video or audio to attach
7. Press send

There is also a way to send messages to students who for example are late with a submission of an assignment. This is done by navigating to the individual view of the gradebook, how to access this is explained in section 5, choose the specific assignment and scroll down to the bottom of the page and press the **Message students who...** button. This will open up a new window.

Assignment 1

Message students for Assignment 1

Message students who...

for Assignment 1

Haven't submitted yet

Subject:

No submission for Assignment 1

Message:

Cancel Send message

Set default grade

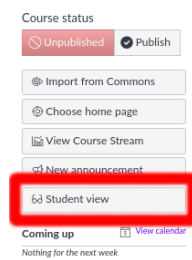
In this window there is a dropdown menu with a selection of parameters for sending messages to students who:

- Haven't submitted yet
- Haven't been graded
- Scored less than
- Scored more than

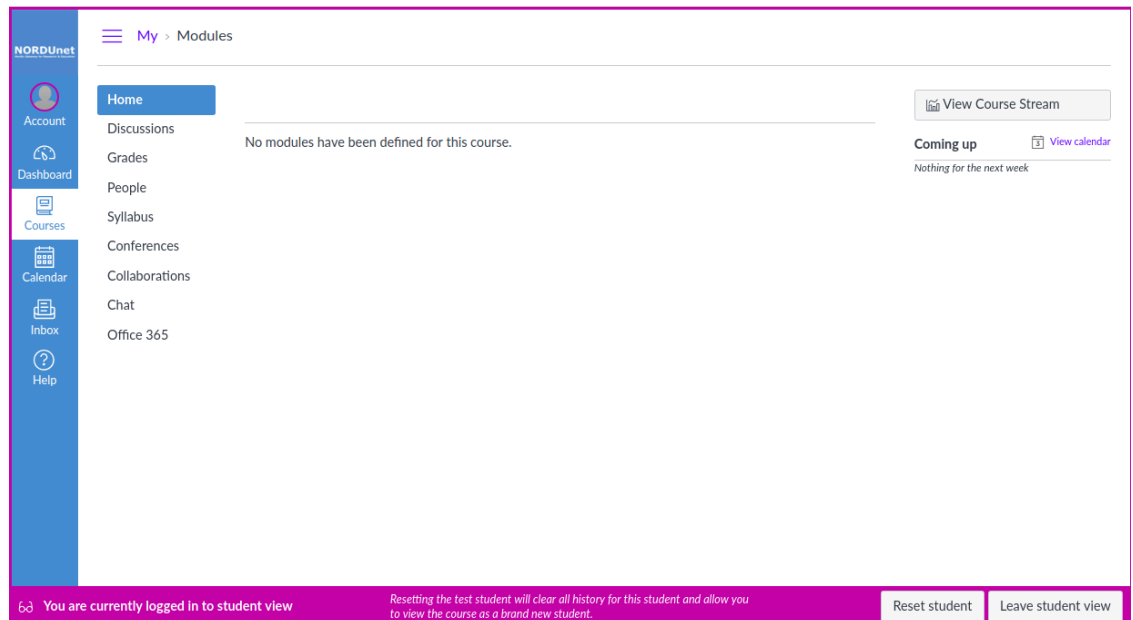
Select one of these and enter the subject and message in their respected field and then press send.

## 10 Student view

Student view is a way to enter the course page as a test student to see what the student will see. This is good to use before publishing the course to see so everything works as intended.



To enter the student view press the **Student view** button on the right side panel of the course. This will create a test student and enter the course page. The test student will act as a regular student and can for example do all the assignments. To reset the student, if the same assignment need to be tested again, just press the **Reset student** button on the bottom right. To exit the student view just press the **Leave student view** button on the right bottom corner.

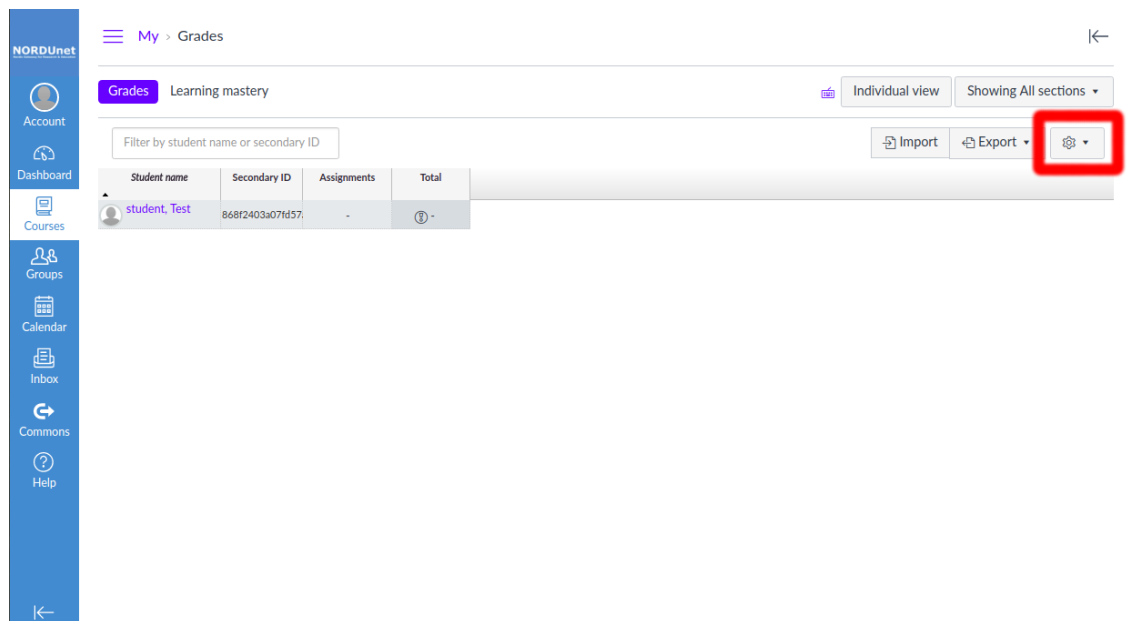


## 11 Notes on a student

A teacher and teacher assistant can enter notes on a specific student, for fellow teachers and TA's on that course to see only, the student will not be able to see this notes. This can be used for example if a certain student need extra help.

Assignments  
Discussions  
**Grades**  
People  
Pages  
Files

To be able to do this push the **Grades** button on the course menu on the left side. This will open up the grades page.

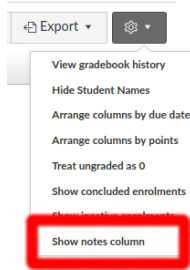


The screenshot shows the Canvas LMS interface. On the left sidebar, the 'Grades' button is highlighted with a red box. The main content area displays the 'Grades' page for 'Learning mastery'. At the top right, the 'Settings' button is highlighted with a red box. Below the settings are buttons for 'Individual view' and 'Showing All sections'. A search filter is present: 'Filter by student name or secondary ID'. Below this is a table with the following data:

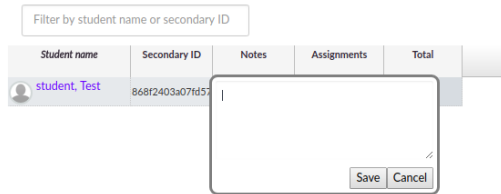
Student name	Secondary ID	Assignments	Total
student, Test	868f2403a07fd57	-	-

The notes are not visible as default and need to be enabled. Do this by press the **Settings** button on the top right.





When the **Settings** button are pressed, a drop down menu appears. Find the **Show notes column** and select it.



To be able to write a note on the student, click the **Notes** column on that specific student. This will make a textbox appear and this is where the note is written. The note can be up to 255 characters long. When done, simply press save. The **Notes** column can be hidden in the same way as making it showed. Hidden notes are still saved.

## 12 Navigation

The course menu is editable in an easy way. The order of the buttons can be changed, buttons can be removed or add new buttons. This is done by navigating to the settings page on the course.

The screenshot shows the Moodle course navigation settings page. The 'Navigation' tab is active, displaying a list of items to be reordered. The 'Settings' menu item is highlighted in the left sidebar. The right sidebar shows course statistics and user counts.

Item	Order
Home	1
Announcements	2
Assignments	3
Discussions	4
Grades	5
People	6
Pages	7
Files	8
Syllabus	9
Outcomes	10
Quizzes	11
Modules	12
Conferences	13
Collaborations	14
Chat	15
Attendance	16
SCORM	17
Office 365	18

Current users:

Students:	None
Teachers:	2
Course Administrator:	None
TAs:	None
Designers:	None
Observers:	None

Press the **Navigation** tab on the top of the page. All the buttons on the course menu will appear and simply click and drag the buttons into wanted order, or drag them to the bottom of the page to remove them from the list. Press **Save** when done.