



Norwegian University of
Science and Technology

Improving Peer Assessment Quality in a Web Development Course

Håkon Larsen Solbjørg

Master of Science in Informatics

Submission date: June 2018

Supervisor: Trond Aalberg, IDI

Norwegian University of Science and Technology
Department of Computer Science

dedication

I would like to extend a thank-you to my supervisor, Trond Aalberg, for helping me through this thesis.

Furthermore, I would like to thank all the people who in some way have been part of my studies at NTNU, for helping to make these years some of the best years of my life.

Abstract

Academia has used peer reviews to ensure high quality of scholarly work for decades, and the way it allows the reviewers to gain an insight into the work of other people has proven beneficial to learners, particularly through the process of peer assessment. Peer assessment is an educational method where students assess the work of peer students, and state of the art research have found it a reliable method for assessing a significant number of students while simultaneously improving the learning experience of the assessors.

Due to lack of quality in some peer assessments done in a web development course at NTNU, work started to investigate ways to improve the quality of the assessments. After a literature review, the two primary research questions for the thesis were ‘How can the use of peer assessments be improved with regards to the usefulness and learning experience for students?’ and ‘How can course staff spend less time evaluating peer assessments while still maintaining legitimacy?’.

Following the research questions came the conceptualization and prototyping of an information system which implements a usefulness-rating of peer assessments, where the receiving students rate the assessments.

The results from a survey of students in the course mentioned above suggest that a rating system will increase the effort students are willing to put into the assessments. If the usefulness-rating of the assessments makes part of their grade for the course, more than 90% were willing to produce high-quality assessments.

Due to lack of interest from contacted course staff at the university, a full-scale test of the system was not possible. Therefore, the conclusion of the thesis is currently purely theoretical, and further work is required to address the research questions thoroughly.

Sammendrag

Akademia har brukt fagfellellevurderinger ('peer review') for å sikre høy kvalitet på forskningsarbeid i hundrevis av år, og måten prosessen tillater folk å få innsikt i andres arbeid er bevist nyttig for folk i læring. En 'medstudentvurdering' er en vurderingsform hvor studentene vurderer og karaktersetter medstudenters arbeid, og moderne forskning har funnet ut at prosessen er pålitelig samtidig som den øker læringsutbyttet til studentene.

På grunn av lav kvalitet i noen øvingsoppgaver i et webutviklingsemne ved NTNU begynte et studie på hvordan man kan øke kvaliteten på medstudentvurderinger. En gjennomgang av relevant litteratur på fagområdet definerte to hovedspørsmål for forskningen: 'Hvordan kan kvaliteten på bruken av medstudentvurderinger økes med tanke på nyttigheten og læringsutbytte for studentene?' og 'Hvordan kan fagstaben bruke mindre tid på å evaluere medstudentvurderinger samtidig som de opprettholder legitimitet i prosessen?'.

Etter forskningsspørsmålene fulgte konseptualisering og utvikling av en prototype for et informasjonssystem som implementerer nyttighetsvurdering av medstudentvurderingene, hvor mottakerene av medstudentvurderingene skal vurdere nyttigheten av dem.

Resultater fra en spørreundersøkelse rettet mot studenter i det ovennevnte emnet peker mot at en nyttighetsvurdering kan øke innsatsen studenter velger å legge i medstudentvurderingene. Hvis nyttighetsvurderingen blir brukt som del av karakteren i emnet mente over 90% at de ville gjort en god innsats.

På grunn av mangel på interesse fra fagstabene på universitetet som ble spurt var det ikke mulig å gjennomføre en test i stor skala. Derfor vil konklusjonen av denne oppgaven kun være begrunnet i teorien bak den, og videre arbeid behøves for å svare fullt ut på forskningsspørsmålene.

Table of Contents

Abstract	i
Sammendrag	ii
Table of Contents	vi
List of Tables	vii
List of Figures	ix
Acronyms	xi
1 Introduction	1
1.1 Background and Motivation	1
1.2 Approach	2
1.3 Thesis Outline	2
2 Background	5
2.1 Peer Review	5
2.1.1 Usage of Peer Review	6
2.2 Peer Review Information Systems	7
2.2.1 EasyChair	7
2.2.2 ConfTool	7
2.2.3 TurnItIn	7
2.2.4 Moodle	7
2.2.5 Gerrit	8
2.3 Peer Assessment	8
2.3.1 Increase Learning Outcome by Using Peer Assessments	8
2.3.2 Improving Peer Assessment Quality	10
2.3.3 Reducing Administrative Efforts	11
2.3.4 Is Student Peer Grading Viable?	11

2.3.5	Peer Assessments with Multiple Rounds of Feedback	12
2.3.6	Motivation	12
2.3.7	Massive Open Online Courses	13
3	Methodology	15
3.1	Research Questions	15
3.2	Literature Review	16
3.3	Survey Design	16
3.3.1	IT2810 – A Web Development Course	16
4	Survey Results	19
4.1	Survey responses	19
4.1.1	Perceived Usefulness of the Current Peer Assessment Process . . .	19
4.1.2	Rating Helpfulness of Assessments	23
4.1.3	Change of Behavior if Assessments are Rated	23
4.1.4	Respondents Thoughts About the Concept	23
4.2	Interpretation of Survey Results	25
4.2.1	Perceived Usefulness of the Current Peer Assessment Process . . .	25
4.2.2	Rating Helpfulness of Assessments	25
4.2.3	Change of Behavior if Assessments are Rated	26
4.2.4	Respondents Thoughts About the Concept	26
4.3	Findings	27
5	System Design	29
5.1	Stakeholders	29
5.1.1	End Users	30
5.1.2	Developers	30
5.1.3	System Administrators	31
5.2	Identified Problems in State of the Art	31
5.2.1	Improving Assessment Quality	31
5.2.2	High Workload for Course Staff	31
5.2.3	Reliability of Students as Evaluators	32
5.2.4	Multiple Rounds of Feedback	32
5.2.5	Lack of Motivation from Students	33
5.3	Concept	33
5.3.1	Feature Parity with Existing System	33
5.3.2	Avoid Data Duplication	34
5.3.3	Automate Distribution of Assessments	34
5.3.4	Facilitate Handing in Peer Assessments	35
5.3.5	Allow for Rating of Received Peer Assessments	35
5.3.6	Facilitate the Evaluation Process	35
5.3.7	Export Data	36
5.4	Refined Peer Assessment Process	36
5.4.1	Create Assignment	36
5.4.2	Peer-Assess Assignment	37
5.4.3	Rate Assessment	37

5.4.4	Evaluate Assignments and Assessments	37
6	Implementation	39
6.1	System Architecture	39
6.1.1	Availability	40
6.1.2	Modifiability	40
6.1.3	Interoperability	41
6.2	Technologies	41
6.2.1	Backend	41
6.2.2	Frontend	41
6.2.3	API	42
6.2.4	Authentication and Authorization	43
6.2.5	Hosting	45
6.2.6	Kubernetes	45
6.2.7	Amazon Web Services	46
6.3	Third Party Integrations	46
6.3.1	Dataporten and FEIDE	46
6.3.2	Blackboard	47
6.4	Development	47
6.4.1	Unit Testing	47
6.4.2	API Versioning	48
6.4.3	Integrating with Third Party Services	49
6.5	Prototype	51
6.5.1	Create Assignment	54
6.5.2	Hand in Deliverable	54
6.5.3	Peer-Assessing Deliverable	55
6.5.4	Rate Received Assessment	55
6.5.5	Evaluation	57
7	Experiment	59
7.1	Usability Test	59
7.1.1	Usability Test Process	59
7.1.2	Usability Test Discussion	60
7.2	Suggested Experiment	60
7.2.1	Assignment Plan	60
7.2.2	Assignment Content	60
7.2.3	Assignment Delivery	61
7.2.4	Peer Assessment Process	61
7.2.5	Peer Assessment Rating Process	61
7.2.6	Evaluation Process	61
8	Discussion	63
8.1	How the Concept Solves the Identified Problems	63
8.1.1	Improving Quality	63
8.1.2	Reducing Administrative Efforts	64
8.1.3	Reliability of Students as Evaluators	65

8.1.4	Multiple Rounds of Feedback	65
8.1.5	Lack of Motivation from Students	66
8.1.6	Large Courses	66
8.2	Discussion of Research Questions	66
8.2.1	Research Question 1	67
8.2.2	Research Question 2	67
8.3	Analysis of the Research Approach	67
8.4	Analysis of the Results and Discussion	68
8.4.1	Missing Field Test	68
8.5	Future Work	68
8.5.1	Expert Groups	69
8.5.2	Multiple Rounds of Feedback	69
8.5.3	Identifying Collusion	69
8.5.4	Increase Motivation	69
9	Conclusion	71
	Bibliography	73
	Appendix	77
A	Appendix	77
A.1	Questionnaire	77
A.1.1	Question 6	77
A.1.2	Question 7	78
A.1.3	Question 8	78
A.1.4	Question 9	78
A.1.5	Question 10	79
A.1.6	Question 11	79
A.2	Usability Test Stories and Scenarios	83
A.2.1	User Stories	83
A.2.2	Test Scenarios	83
A.3	Usability Test Consent Form	86
A.4	Usability Test Transcript	88
A.4.1	Test 1	88
A.4.2	Test 2	89
A.4.3	Test 3	89
A.5	Usability Test Findings	90

List of Tables

- 4.1 ‘What fits best with regards to how you felt receiving feedback from your peers?’ 22
- 4.2 ‘What fits best with regards to how you felt giving feedback to your peers?’ 22

- 5.1 Stakeholders 30

- A.1 ‘Identify with the following statements’ 77
- A.2 ‘What fits best with regards to how you felt receiving feedback from your peers?’ 78
- A.3 ‘What fits best with regards to how you felt giving feedback to your peers?’ 78
- A.4 ‘Thoughts about being able to rate the review’ 78
- A.5 ‘Thoughts about time spent revieweing’ 79
- A.6 Overview of scenarios tested 88

List of Figures

2.1	Model of a generic peer review process	6
2.2	User Interface of the TurnItIn Feedback Studio (Turnitin LLC, 2018a) . .	8
2.3	Example of rubrics from the TurnItIn Feedback Studio (Turnitin LLC, 2018a)	9
4.1	‘I was satisfied with the thoroughness of the assessments I received.’ . . .	20
4.2	‘Those who assessed me were interested in me doing better next assignment.’	21
4.3	‘I was interested in making those who I assessed better for next assignment’	21
5.1	Model of the proposed peer assessment process	37
6.1	Visualization of the HTTP ‘request-response’ cycle.	40
6.2	The OAuth2 Authorization Grant flow, as per Figure 3 in Hardt (2012). .	44
6.3	OAuth2 Authorization Prompt, by Parecki (2014)	45
6.4	Two git commits which have their commit status set in GitHub.	48
6.5	The OAuth2 Dashboard in Dataporten.	49
6.6	The OAuth2 Authorization prompt in Dataporten.	50
6.7	The login form in Dataporten.	50
6.8	View of assessment creation	52
6.9	View of rubrics creation	53
6.10	View of assessment tasks	54
6.11	View of a single assessment	54
6.12	View of received assessments	55
6.13	View of a single received assessment	55
6.14	View of the assessments ready to be evaluated	56
6.15	View of one assessments ready to be evaluated	56
7.1	Example of a peer assessment rubric	61
7.2	Example of a peer assessment rating	62
7.3	Example of a peer assessment evaluation view	62

Acronyms

API Application Programming Interface. 40–42, 48, 49, 51

AWS Amazon Web Services. 45

CDN Content Distribution Network. 46

FEIDE Felles Elektronisk IDEntitet (‘Common electronic identity’). 46

HTML HyperText Markup Language. 42

HTTP HyperText Transfer Protocol. 40, 41, 43, 45

ID Identifier. 47

JSON JavaScript Object Notation. 42

LMS Learning Management System. 7, 9, 13, 29, 31, 32, 38, 41, 43, 47, 54, 61, 64, 68, 91

MOOC Massive Open Online Course. 13

NTNU Norwegian University of Science and Technology. i, ii, 1, 2, 16, 43, 46, 47, 51, 67, 71

REST Representational State Transfer. 40, 41, 51

SSO Single Sign-On. 44

TCP Transmission Control Protocol. 45

UUID Universally Unique Identifier. 51

Chapter 1

Introduction

Academia has used peer reviews to ensure high quality of scholarly work for centuries. ‘Peer review’ is a process where your peers verify the work you have done, where the main focus lies on maintaining high-quality work based on sound scientific principles. While reviewing, the reviewers get a unique look at the work of other people, and the insight gained from this process has been proven beneficial for learners. Education has started using this process, ‘peer assessment’, as an evaluation method, where students assess the work of their peers.

Peer assessment is a time-consuming process, and it often heavily burdens course staff with sifting through all the assessments and responses to make sure they are valid and bring meaning to the students. This thesis studies the use of peer assessment in a web development course at the Norwegian University of Science and Technology (NTNU), where participating students completed group projects and assessed their peers. Furthermore, the thesis studies previous work related to the use of peer assessments in education and combines this research with survey responses from students enrolled in the web development course.

Following the research of difficulties related to the field of peer assessment commences the development of a concept to solve these issues. An information system implements the concept, and a prototype is ready for testing.

1.1 Background and Motivation

Scholarly peer-reviewing and educational assignments have comparable qualities, particularly with regards to quality and adhering to scientific principles. Peer assessments seem to be a viable alternative for use in education, however; the course coordinator of a web development course has experienced problems with the thoroughness of students’ assessments, which lowers the learning experience for everyone involved. The course coordinator wanted help to study and solve these issues, especially with regards to the administrative process and the lack of thoroughness from the enrolled students.

To lower the administrative work with evaluating assignments, or in this case, evaluating assignments and assessments, it is sometimes viable to employ multiple choice question-assignments instead, to survey the direct knowledge of students. However, such assignments enforce strict boundaries on what kind of assignments are viable, and it mostly removes the possibility of students to do creative work.

Therefore, to ensure that students are capable of creating work in which they explore and come up with solutions themselves, open-ended assignments are essential. However, one of the significant problems with open-ended assignments is that students produce content which is time-consuming to assess. By exploiting peer students to facilitate the assessment process, course staff can lower their efforts with regards to assessing each student.

While the peer assessment method has potential to reduce work efforts required by course staff, it also enhances the learning process of students by allowing them to review the work of other students. Therefore, it is interesting to improve the process of peer assessments so that it can be more widely applied.

1.2 Approach

The course coordinator and lecturer of a web development course at NTNU supervised the work of this thesis. He used peer assessment for his course and presented some problems with the current state of peer assessments in the course. Following these findings, a literature review was conducted to find a potential cause of the problems, as well as identifying other potential issues with regards to peer assessment. The findings from the literature review and the initial problems from the course coordinator were the basis for the research questions used through this research.

The research questions are:

RQ1 How can the use of peer assessments be improved with regards to the usefulness and learning experience for students?

RQ2 How can course staff spend less time evaluating peer assessments while still maintaining legitimacy?

Students of the course responded to a survey which aimed to find potential issues the students faced during the course, both to see if they aligned with the presented problems as well as to identify further problems the course staff might not have observed. Furthermore, the survey aimed to find out if the students experienced the same problems as state of the art research on peer assessment had found.

Chapter 2 contains the literature review, and Chapter 3 goes into further detail regarding the survey.

1.3 Thesis Outline

The outline of this thesis is as follows:

Chapter 2 presents the theoretical background for the study including a literature review which surveys state of the art peer assessment.

Chapter 3 describes the research methodology used throughout this thesis.

Chapter 4 presents the results and discussion of a conducted survey.

Chapter 5 develops the concept for a prototype based on the works from the literature review in Chapter 2.

Chapter 6 describes the implementation process with a focus on the technical implementation of the system as well as the technologies used and how they solve some of the identified problems from Chapter 5.

Chapter 7 contains the conducted experiment as well as a suggestion for a more extensive experiment.

Chapter 8 discusses the work done throughout the thesis, mainly regarding the identified problems, research questions, and future work.

Chapter 9 concludes the thesis.

Background

Peer review is a process which verifies that scholarly work meets the expected standards of academia, which often works by submitting some work for review after having done research. Peers in the same field will review this work to make sure it meets the criteria for scholarly work, such as being based on previous work in the same field as well as logical assumptions present in the data. Scientific journals require this process to make sure the published work is valid so that others can rely on it for further studies.

For the rest of this thesis, the term ‘peer review’ will be used regarding the process of scholarly peer reviewing, and the term ‘peer assessment’ will refer to the use of peer review for educational purposes where students assess the work of each other, unless otherwise mentioned.

This chapter will present theory regarding the peer reviewing process, present some state of the art information systems for facilitating working with peer reviews, and previous work in the field of peer assessments. Section 2.1 introduces a generic peer-reviewing process as well as some variations of it; Section 2.2 presents state of the art information systems regarding peer reviews and peer assessments, and Section 2.3 contains a literature review of previous work in the field of peer assessments.

2.1 Peer Review

The core peer-reviewing process is quite simple. When publishing to a conference or journal, some publications tend to have different processes based on pre-processing such as considering the topic at hand and adhering to style guides for the journal. However, the primary steps regarding the reviewing process are standardized. These steps are the ones which include the author or their peers in the process. Figure 2.1 shows the generic peer review process which consists of the following three steps:

1. A ‘call for papers’ is hosted
2. An author hands in work

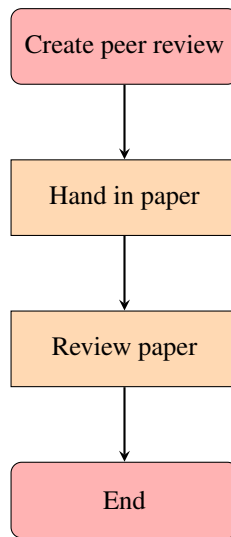


Figure 2.1: Model of a generic peer review process

3. A peer reviews the work

Completion of the third step concludes the peer review and decides what happens next. For journal or conference submissions, the submissions usually get accepted or rejected. However, some processes add further actions required by the author.

In case the reviewer has misunderstood some of the work, the author may respond to the review with a rebuttal. A rebuttal is a letter where the author may correct the assumptions or findings of a reviewer, in case they are incorrect. The rebuttal often happens before a decision to accept or reject has been made, and can be used to allow the author to revise the paper, which might flip a potential rejected submission if their work is acceptable, but not in its current state.

2.1.1 Usage of Peer Review

While peer review is in active use in academia, the benefits of peer reviews have shown useful for other disciplines too. With a grounding in pedagogy, students assessing their peers' assignments have proven beneficial impact on their learning experience (Topping, 1998).

The computer science discipline also knows about the potential benefits from peer reviews and has adopted it as 'code review'. Code review is quite similar to the process of peer review, as in that someone develops some source code which he or she puts up for review. Peers can verify the correctness of the code, whether it is regarding logical correctness or with regards to business requirements, and accept or reject the code. Usually, the code will be updated and re-submitted if the initial code review fails.

2.2 Peer Review Information Systems

With the various relevant use-cases described, this section aims to introduce some ways of organizing peer reviews using state of the art information systems to facilitate the process.

The systems presented range from being developed for academia, such as journals or conferences, through peer assessment systems designed for educational use, to code review systems meant for doing a code review in software projects.

One of the processes used in academic peer reviewing systems is a process called ‘call for paper’. This process tells paper authors that a conference, or journal, is ready for submissions regarding its topic. Following this callout, authors submit their work and the peer review process can begin. Two of the systems presented in this section facilitates this process.

2.2.1 EasyChair

EasyChair is a conference management tool, which supports the call for papers process (EasyChair Ltd, 2018). Its intended use is as a validation tool for submitted papers to a conference where the reviewers use a scale to rate the submissions from ‘strong reject’ through ‘neutral’ to ‘strong accept’. Authors use these ratings to know the relevancy and quality of their paper.

2.2.2 ConfTool

ConfTool is another conference management tool (ConfTool GmbH, 2018). For the sake of this dissertation, the features of this system are identical to the ones of EasyChair, hosting call for papers where the reviewers accept or reject the submissions.

2.2.3 TurnItIn

TurnItIn (previously PeerMark) is a system focused on educational use. It is a platform where students hand in an assignment and peers or instructors review it (Turnitin LLC, 2018b).

The tools provided focus on simplicity of grading an assignment, with reusable rubrics for evaluation, automatically generated feedback suggestions and a system for detecting plagiarism. Figure 2.2 shows one of the views in the user interface of TurnItIn.

The primary difference from the previously mentioned conference tools and this educational system is that this system focuses on helping students to assess their peers so that they can improve their deliverable.

2.2.4 Moodle

Moodle is a complete Learning Management System, and it contains a module for doing peer assessments (Moodle, 2018). The source code of the system is open source, so people can easily extend its functionality with custom modules or improvements.

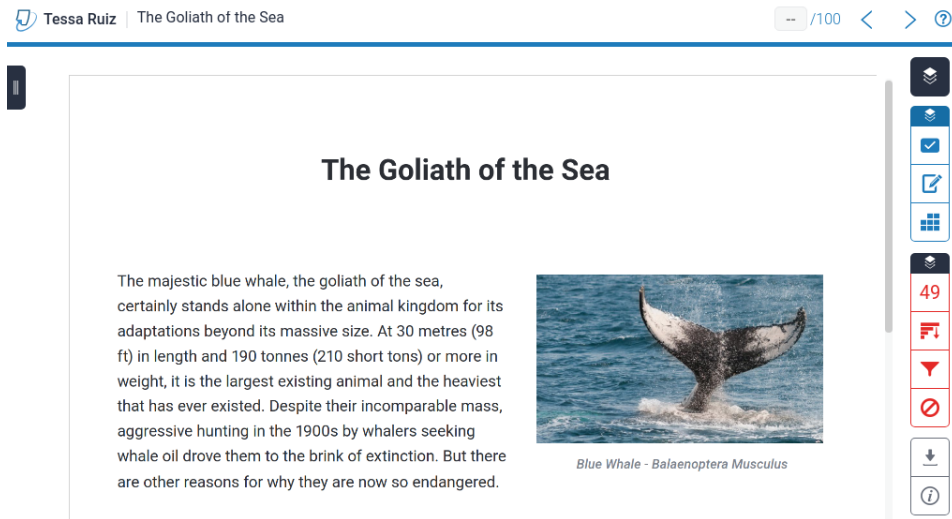


Figure 2.2: User Interface of the TurnItIn Feedback Studio (Turnitin LLC, 2018a)

2.2.5 Gerrit

Gerrit is a code review tool open sourced by Google, which initially was closed source. It now supports some of the most widely used version control systems, such as git (Gerrit, 2018). It was initially created to do code review on the web, and since its release, numerous other tools have followed suit.

2.3 Peer Assessment

Peer assessment is the use of peer reviewing in an educational setting, such as to have peers grade each other's assignments. Often, teachers provide the assessors with some rubrics which contain criteria which the students should use to grade an assignment. Figure 2.3 showcases an example of such rubrics.

This section contains a literature review of previous work done in this domain, with interest in identifying potential problems which an information system can solve.

Peer assessment has its benefits and disadvantages, as widely known in its previous research (Topping, 1998; Mostert and Snowball, 2013). While the enhanced learning experience for students is excellent, the practical drawbacks such as organizing the assessments is a showstopper for some teachers.

2.3.1 Increase Learning Outcome by Using Peer Assessments

By using peer assessments as an evaluation technique, students have to view other possible solutions to the same problem as the one they have solved themselves. This process opens their mind to other ways to approach problems and can be a positive influence (Sadler and Good, 2006).

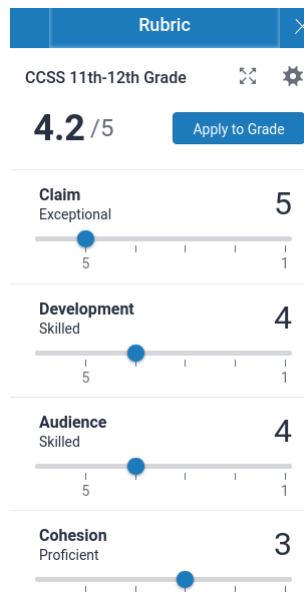


Figure 2.3: Example of rubrics from the TurnItIn Feedback Studio (Turnitin LLC, 2018a)

A questionnaire by Yanqing et al. (2011) discovered that students are open to a cycle in which they improve their programming assignment after given constructive remarks. The same research found that students participating in the peer code review process would improve their code after having others look at it. Additionally, they found that following code standards¹ made it simpler both for students and reviewers to have something to look at, and most participants wrote better code after the process. The primary learning outcomes observed were:

- Increased programming ability
- Higher conformance to coding standards
- Ability to make suggestions and accept criticism
- Ability to learn in a collaborative setting

Furthermore, Mostert and Snowball (2013) studied the use of formative peer assessments in a macroeconomics course which heavily focused on English writing. The course used a peer assessment module in an LMS, and surveying students taking the course found that 58% of the students suggested that the peer assessment process enriched their learning.

¹Code standards, or conventions, are conventional ways of writing lines or blocks of code. The standards can be with regards to newlines, whitespaces, indentation, and so on. They exist to make the structure of code familiar across programs.

2.3.2 Improving Peer Assessment Quality

Gehring (2014) studied how the quality of peer reviews can be improved. When using peer assessments in education it is crucial to quality control the assessments, because the grades are given by other students who might not have enough experience or expertise in the field. Five different quality control mechanisms were studied, with different strengths and weaknesses. The mechanisms were:

- Calibration
- Reputation Systems
- Meta-Reviewing, Manual and Automated
- Helpfulness Ratings
- Rating vs. Ranking

Calibration requires staff to spend time before the assessment process can begin, in the form of instructors having to do one, or more, ‘correct’ peer assessment(s). Students will assess the same assessment, and a comparison of the score from the student and teacher is carried out. The results are usually used to make sure the students are competent enough to be assessors.

Reputation Systems are used to award skilled assessors with a positive reputation and unskilled assessors with a negative reputation. Awarding of reputation happens after completing the assessments. The reputation will gain traction as more assessments complete, so the initial assessments will not have any reputation.

Meta Review is a method where a third-party assesses the original assessment. The third party can be either staff or other students. It is also possible to partially automate this process with the current state of the art natural language processing tools (Ramachandran, 2013); however, that is out of scope for this thesis.

Helpfulness Ratings is a method in which the author rates the helpfulness of the feedback in the assessment. Teachers could include this rating in the final evaluation of the students.

Rating vs. Rankings considers whether the method in use is a rating or a ranking method. The difference is whether the assessment is done using a uniform scale, where any assignments, in theory, can receive the same grade, contrary to methods where students rank the assignments from best to worst. One of the main advantages found for the rating approach is that only the work of the author is relevant; the quality of other students’ work will not impact the assessment. Furthermore, it is easier to define criteria for the assessments. Meanwhile, rankings avoid a problem which presents itself in rating settings; that assessments tend to have an overall high average rating (Gehring, 2014).

2.3.3 Reducing Administrative Efforts

Assessment of open-ended assignments is a time-consuming process (Topping, 1998). An information system can remedy the problems with regards to the substantial amount of tedious effort required by course staff to organize peer assessment, Davies (2009) has found. While the logistics of the process is only one side of the administrative efforts, the use of students to assess their peers requires some follow-up to make sure the grades are justified, which Section 2.3.4 will present.

The future of assessment might be entirely computerized with for example natural language processing as tested by Ramachandran (2013). However, research with regards to artificial intelligence assessing open-ended student deliverables is quite low in state of the art research and is out of scope for this thesis.

2.3.4 Is Student Peer Grading Viable?

Studies by Falchikov and Goldfinch (2000) and Sadler and Good (2006) show that students tend to align with the grading done by their teacher, so even though the peer assessors are in the process of learning the topic themselves and not being experts in the field, their knowledge can be applied to peer assessments.

Sadler and Good (2006) studied 7th graders and found that the grades from the peer assessments aligned with the ones from their teachers. Falchikov and Goldfinch (2000) did a meta-analysis on some previously conducted peer assessments in university-level courses and found that peer grading agrees with marks from the teacher. Topping (1998) found the average reliability of such peer assessments to be ‘acceptably high’ across 18 out of the 25 studies compared in the study. An example of such an ‘acceptably high’ reliability is 88% agreement with the grading of the teacher, according to Topping (1998). He also found that the grades tend to cluster around the median grade.

Bejdová et al. (2014) studied an informatics course and found that the quality of the assessment is not as good as the one from a teacher. The study did not define the difficulty level of the course taught, so if there is a discrepancy in the difficulty level of the previously mentioned courses and this one, that might have an impact. Based on this finding, Bejdová et al. (2014) acted cautiously regarding using students as sole assessors in a peer assessment setting. However, the study did acknowledge the positive effect peer assessments has as part of the learning process.

Expert Groups

Studies have found that university-level peer assessment has an agreement-rate of 69% (Falchikov and Goldfinch, 2000) with regards to the grade of the teacher, while the middle-school study had an agreement-rate of 90% Sadler and Good (2006).

The studies by Bejdová et al. (2014) found a similar agreement-rate across all their students. However, they also found that a group of ‘expert’ students achieved 94% agreement-rate, whereas the overall agreement-rate for all students was between 60% and 70%. If it is possible to establish groups of expert students, more trust can be placed on students to remedy the lack of course staff grading students.

Collusion

When allowing students to evaluate work created by themselves or their peers, they might have a bias towards giving themselves higher grades because it might end up giving them a higher grade. Therefore, collusion is something to keep in mind when doing peer assessments. In a paper by Song et al. (2017), two types of collusion were identified, named ‘Pervasive’ and ‘Small-Circle’ collusion. ‘Pervasive collusion’ is when students give high scores to any assignment they assess, while the ‘small circle’ colluders were an organized group of students who gave each other higher scores.

To identify the colluders some algorithms were constructed, and the findings were that colluders had little impact on the overall scores in the experiments. The average inflation across all the experiments was found to be 1 – 2%, and less than 10% of assessors partook in collusion for more than 70% of the studied assignments.

Colluders are definitively in the minority as of now, Song et al. (2017) claims. However, if they gain the majority, they would be the ‘norm’ while the legitimate ratings will be the outliers. Until this happens, though, collusion should be viewed as a trivial problem with regards to the legitimacy of peer assessments, and implementing some algorithms as defined in Song et al. (2017) can remedy its impact.

2.3.5 Peer Assessments with Multiple Rounds of Feedback

The use of peer review in academia consists of the classic ‘submit-review’ cycle as demonstrated in Figure 2.1. This process does not necessarily allow for feedback out-of-the-box. In education, however, such feedback provides enhanced learning and allows students to improve.

In an experiment by Song et al. (2016), students assessed an assignment in two rounds. One round focused on formative feedback, while the next one focused on summative grading. Both of the rounds included rubrics for assessment, which makes it easier for the assessors to know where to focus.

The formative rubrics were found to be of good help, measured by the fact that the reviews had higher volume compared to previous years and a lower rate of empty comments. The assessments were also found to be more constructive and were composed better. Students were expected to revise their deliverable with regards to the comments from the formative assessment and hand it back in again for round two.

The second round used summative rubrics, where the point was for the students to assign a grade to the revised assignment. Song et al. (2016) found that the reliability and validity of the grades were higher in the summative round. A discussion regarding this is missing; however, the authors suggested that it might have to do with the variance in initial submissions. Furthermore, they found that the reliability was higher in the evaluations which separated the formative and summative assessments than in a baseline-case which mixed these.

2.3.6 Motivation

Motivation is a significant factor in the learning experience and effectiveness of learning (Ames, 1990). Its use regards both a feeling of achievement, which happens after

having studied, as well as a method to motivate students to study in the first place.

A study by Simionescu et al. (2017) researched the use of gamification to improve motivation for students and simplicity of grading for the teachers in a peer assessment environment. Gamification is ‘the use of game design elements in non-game contexts’ and is often used to improve user experiences (Deterding et al., 2011). Based on research by Abramovich et al. (2013), its use in education has proven positive effects. However, the findings are not solely positive, and the usefulness depends on the previous knowledge of the student.

Abramovich et al. (2013) created a plug-in for the Moodle LMS to allow students to give ‘badges’ to each other to track progression, just like rubrics do in other peer review systems. The impact of the study is a bit uncertain because no test has been conducted to verify increased motivation yet. However, when appropriately implemented, with carefully designed badges, gamification has potential to be a good motivator (Abramovich et al., 2013).

2.3.7 Massive Open Online Courses

While this thesis focuses on a conventional course taught in-person at a university; problems regarding grading of open-ended assignments are present in state of the art research of MOOCs as well.

A Massive Open Online Course (MOOC) is an educational course which anyone in the world can attend, using the internet as a bridge. The sheer number of attending students cause much work for the course staff to grade everyone, especially if they require open-ended assignments.

Automatic grading tools, notably for open-ended assignments, are not good enough to deploy on such a scale at the current time according to Fang et al. (2017). Meanwhile, to be able to provide students with a productive and useful learning process, being able to address open-ended assignments is of interest. By implementing peer assessments, it is possible to move work from course staff to students while also improving their learning experience.

Chapter 3

Methodology

This chapter presents the methodology used for carrying through this thesis. After having corresponded with the project supervisor regarding a web development course which used peer assessments for its assignments and his experience, work carried over to a literature review to further study related problems and issues.

Section 3.1 describes the research questions, while Section 3.2 describes the literature review, and Section 3.3 describes the survey and its design. Section 3.3.1 presents the surveyed university course.

3.1 Research Questions

Based on the findings of the literature review in Section 2.3, the following research questions will be the primary concerns for research through this thesis:

RQ1 How can the use of peer assessments be improved with regards to the usefulness and learning experience for students?

RQ2 How can course staff spend less time evaluating peer assessments while still maintaining legitimacy?

To see if students of a web development course felt familiar with the issues presented by the course coordinator, the design of a survey and questionnaire ensued, to assess the students' perception of the evaluation method.

To answer the research questions, conceptualization of an information system to solve the issues started after having identified the issues, and implementation of an information system followed shortly after.

3.2 Literature Review

Section 2.3 contains a literature review of work related to peer assessments. Relevant articles were found using the search functions of IEEEExplore¹ as well as some articles suggested by the project supervisor which ended up being available through Harvard and Springer. The primary search terms were regarding peer review, peer assessment, peer review in education, quality of peer assessments, and information systems facilitating peer reviews or assessments.

3.3 Survey Design

A survey helped to identify how students in a web development course experienced the peer assessment process they were part of during their course, as well as to find out if a revised peer assessment process could improve the process.

The survey consisted of a questionnaire which focused on the helpfulness of the assessments the students received for each assignment. Most of the questions aimed to identify the agreement or disagreement with various statements regarding the assessments, such as how much they felt they learned from the feedback they received. These questions used the ‘Likert-scale’ for agreement and disagreement as defined in Oates (2006). The final question was an optional open-ended question where respondents could voice any thoughts they had on the topic.

Oates (2006) describes some measures to ensure quality in a questionnaire, particularly with regards to making sure that the questions will receive the answers the author expects. The measures are ‘construct validity’ and ‘reliability’. Construct validity regards making sure that the questionnaire asks questions which generates answers that are useful for the intended purpose, and reliability regards that the results are reproducible by surveying the same group of people once more with the same questionnaire.

As a measure to verify these qualities in the questionnaire, two students from the course did a pre-test of the survey. They provided some feedback with regards to some of the questions, mainly towards which ones were relevant and how well the provided answers covered all the potential answers so that respondents found a suitable alternative. After revising some questions after the pre-test, a third party with experience in survey design reviewed it before it was published to the students enrolled in the course.

3.3.1 IT2810 – A Web Development Course

IT2810² a course at NTNU which focuses on web development for people experienced in fundamental web technologies. By shifting the focus from the underlying web technologies onto the current state of the web, enrolled students create rich and interactive web applications based on state of the art JavaScript frameworks such as React and Angular. These technologies are out of the scope of this research.

IT2810 is a course where its assignments are heavily project-based, and the projects range from creating simple programs to more complex systems. Peer assessments were

¹ IEEEExplore: <https://ieeexplore.ieee.org/search/searchresult.jsp>

² Course information: <https://www.ntnu.edu/studies/courses/IT2810>

used to evaluate students and their projects in this course, but they require substantial effort. Students submitted their peer assessments through surveys set up in Google Forms. Course staff manually created the surveys for each project assignment and had to map the answers back to the initial students manually, which created much work.

The students from this course responded to a survey about the experiences they had using peer assessment in this course. Chapter 4 presents the data gathered from this survey.

Chapter 4

Survey Results

This chapter presents the results of the survey conducted on students in a web development course. Section 3.3 describes the survey design, and Section 3.3.1 presents the course surveyed.

Section 4.1 presents the results of the survey, and Section 4.2 discusses the findings. The findings are discussed by themselves as well as with relation to previous work. Chapter 5 describes the process of conceptualizing an information system, which uses the discussion of previous work as the groundwork.

4.1 Survey responses

This section will present the survey responses from the survey defined in Section 3.3. For a complete reference to the questions, answers, and responses, see Appendix A.1. Central questions discussed will have their data in-line where appropriate. For a discussion of the results, see Section 4.2.

The survey first established a baseline of the respondents, after which it turned to more specific questions. The baseline consisted of asking the users how much experience they had with technologies such as the ones used in the course. In total, 64 out of the 184 who attended the course chose to respond to the survey.

This section divides the various questions into subsections where related questions are combined.

4.1.1 Perceived Usefulness of the Current Peer Assessment Process

The first section of the survey asked about how helpful the students felt the peer assessments were. Students answered using a Likert-scale ranging from 1 (strongly disagree) to 5 (strongly agree).

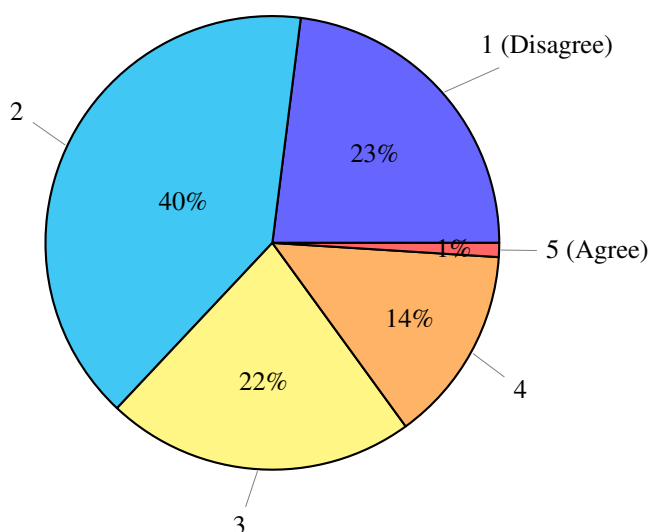


Figure 4.1: 'I was satisfied with the thoroughness of the assessments I received.'

Thoroughness of Assessments

Figure 4.1 shows the distribution of answers. From the recorded answers, 15% agreed or strongly agreed with the statement that their peers thoroughly assessed their assignments, 22% were indifferent, and 63% disagreed or strongly disagreed with the statement.

'Interest in my improvement'

After having asked about the thoroughness of the assessments, students responded regarding how much they felt that their peers wanted them to improve based on the feedback in the peer assessments. Figure 4.2 shows the distribution of answers. 8% agreed or strongly agreed that their peers were interested in them doing better for the next assignment, 23% were indifferent, and 69% disagreed with the statement.

'Interest in my peers' improvement'

The next question surveyed how the students felt assessing others by asking them about their interest in their peers improving for the next assignment. Figure 4.3 illustrates the distribution of responses, where 42% of the respondents agreed or strongly agreed with the statement, 29% responded being indifferent, and 29% disagreed or strongly disagreed with the statement.

The following questions dig a bit deeper to try to understand the reasoning for the initial responses towards helpfulness of the peer assessments. Respondents were asked to select one out of four statements which they identify with regarding the feedback they received from each peer assessment. The two questions ask the same question, but one bases its

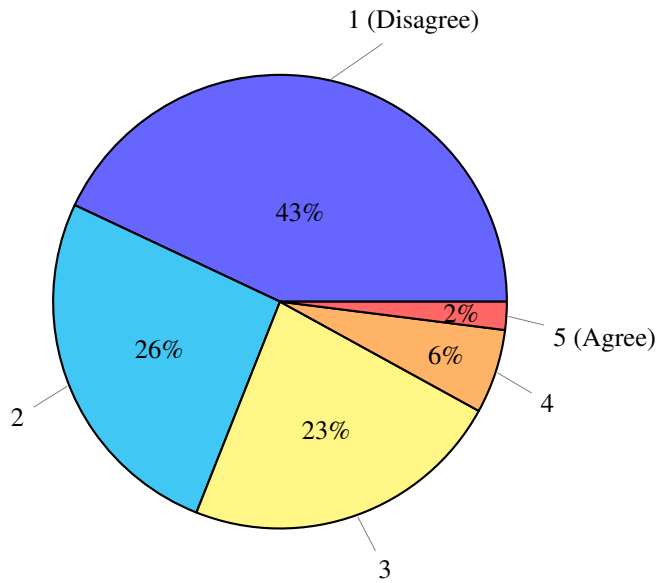


Figure 4.2: 'Those who assessed me were interested in me doing better next assignment.'

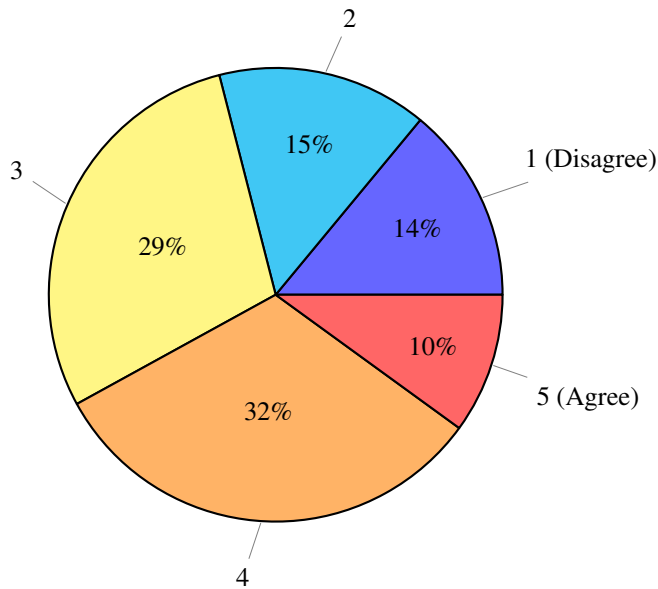


Figure 4.3: 'I was interested in making those who I assessed better for next assignment'

Table 4.1: ‘What fits best with regards to how you felt receiving feedback from your peers?’

Question	Survey response
The feedback helped me a little bit, but I am pretty experienced in the field already.	37%
The feedback gave me some help about things I did not know about, and other minor improvements.	62%
The feedback gave me quite a few tips about things I did not know about.	0%
I learned a lot from the feedback.	1%

Table 4.2: ‘What fits best with regards to how you felt giving feedback to your peers?’

Question	Survey response
I seldomly found something to comment upon, because I am pretty inexperienced in the field.	17%
I could comment on a thing here and there.	35%
I could comment on quite a bit for most of the reviews.	28%
I always found something to comment on.	20%

context on the received peer assessment while the other focuses on assessing their peers’ assignments.

Perceived Benefit of Peer Assessment Feedback

This question was a follow-up question regarding how the students tried to improve based on the feedback from the assessments, to try to find out the reasoning for why students agreed or disagreed with the statements from the previous section. Table 4.1 shows the possible answers as well as the distribution of responses by the survey responders.

1% of the respondents felt that they learned a lot from the feedback, 37% felt that the feedback helped a little bit; however, they were quite experienced already, and 62% felt that the feedback helped a little bit, pointing towards small issues. In total, 99% of the survey responders meant that the feedback helped a little bit or nothing at all.

Assumed Percept from Assignment Feedback

To find out what students mostly commented on and helped their peers with, responders identified with a statement regarding what they commented on in their peer assessments.

Table 4.2 shows the distribution of responses. In short, 17% meant that they did not find much to comment on because they were pretty inexperienced in the field, 35% meant they could comment on a thing here and there, 28% found things to comment on for most assessments, and 20% always found something worth commenting.

4.1.2 Rating Helpfulness of Assessments

The survey introduced a concept like the one presented in Chapter 5, after which students responded to some questions regarding the presented concept, its potential and how they would have utilized it.

The following four questions had closed-ended answers, either yes or no. The responses will identify how interested students are in such a concept, based on what kind of assessments, feedbacks, and scores they receive. The questions surveys if there is an interest in the concept if the received feedback is either good or bad, as well as if the received score is good or bad. Appendix A.1.4 contains the detailed responses to the questions.

For all four of the questions, respondents were interested in the features added by the presented concept, namely the ability to rate the assessments based on their helpfulness. 88% were interested in being able to rate the assessment if the score was low and 85% were interested in having the ability to rate the assessment if the feedback was terrible. Furthermore, 73% were interested in having the ability to rate the assessment if the received a high score, and 71% were interested in being able to rate the assessment if the feedback was excellent.

4.1.3 Change of Behavior if Assessments are Rated

After querying the interest of the features in the presented concept, a follow-up question asked how they would change their behavior with regards to doing the peer assessments if their assessments were to be rated. Appendix A.1.5 details the responses for these questions.

37% responded that they would spend more time on the peer assessments if the receiver could rate the assessment. The majority, with 54%, responded that they already spent much time assessing, and would not change their behavior. 9% responded that they would not change their behavior no matter what.

The next question asked if the students would change their behavior if the grading process for the assignment would include the rating of the assessments. The majority with 57% responded that they would spend more time assessing if that was the case, while 35% responded that they already spent much time on the process. 8% responded that they would not change their behavior either way.

4.1.4 Respondents Thoughts About the Concept

The final question of the survey was an open-ended question which asked if the respondents had any thoughts, ideas or feedback. This section will present some of these, while Appendix A.1.6 shows them all in their original typing. Out of the 24 responses to the question, three responses were empty and therefore removed. The results include the remaining 21 responses.

Some of the responses list the same concerns. For conciseness, these responses are grouped and addressed collectively. Some responses mention concern for the fact that the assessors are not experts and some express concern regarding the assessors not being experienced in the field. For this presentation, and the discussion in Section 4.2.4, the two concerns are separated.

Non-Expert Assessors

Four of the responses were critical regarding allowing their peers to evaluate their work due to them not being experts. An example of such a comment is this one (response number 3 in Appendix A.1.6):

‘I think it might be better to have people who are experienced in the field to evaluate the projects which account for the course grade.’

Inexperienced Assessors

Six respondents were concerned about their peers not being experienced enough in the field to evaluate them. An excerpt of response 17 is an example of such a statement:

‘The students are in the process of learning the curriculum and are therefore not qualified to grade their peers.’

Lack of Effort from Assessors

Five responses issued concerns regarding the lack of effort put into the assessments, where their assessors could potentially only have used two minutes to complete the assessment process. One of the respondents (response 14) described this as such:

‘There was always one group not taking the peer assessment seriously, and we got the lowest possible score on all the questions.’

General Comments, Feedback and Thoughts

Apart from these shared concerns, some respondents suggested improvements for the current process or the described concept. For example, response 13 suggested that the assessments should go through quality assurance. Response 15 suggested organizing multiple rounds of peer assessment, where the students could grade their peers’ improvement from one round to the next. The same responder suggested that course staff could be part of the peer assessment process as well, as one of the peer assessors per group. He or she also suggested identifying anomalous scores.

One respondent raised concerns regarding collusion, where they mentioned that since the students knew one another, they scored each other higher. Another one mentioned that he or she learned something from the peer assessment process, i.e., by assessing their peers’ solutions. He or she also mentioned, along with one other, that the process took a substantial amount of time, and sometimes felt demotivating.

4.2 Interpretation of Survey Results

This section discusses the findings of the survey presented in Section 4.1 and tries to identify what the students experienced during the peer assessment process as well as the problems they faced. The findings are relevant for the concept development presented in Section 5.3.

The discussion follows the same structure as Section 4.1 did when presenting the results, as in; the discussion regarding questions relevant to each other happen collectively.

4.2.1 Perceived Usefulness of the Current Peer Assessment Process

Out of the 64 respondents of the survey, 63% were unsatisfied with the thoroughness of feedback in peer assessments. 15% were to some degree satisfied with the thoroughness, while 22% were indifferent; and a comparison with the results from the question regarding how much respondents felt their peers wanted them to improve finds similar numbers, particularly with regards to the disagreement. 69% responded that they disagreed or strongly disagreed with the statement that their peers wanted them to improve for the next assignment. Only 8% responded that they agreed with the statement to some degree, while 23% were indifferent.

Furthermore, 29% of the respondents answered that they disagreed to some degree with the statement of them wanting their peers to improve for the next assignment, while 42% responded that they to some degree agreed with the statement.

While 42% of the students wanted their peers to improve for the next assignment, only 8% felt that their peers gave them feedback worthy of improvement. Close to none of the respondents perceived a noticeable gain from the feedback, with 99% of the respondents only receiving tips about minor improvements in their project. While this is an extremely high percentage, it might be possible that the question missed an alternative, so that respondents chose the answer both for low-quality assessments and if the respondents found themselves experienced. 17% of the respondents answered that they were inexperienced in the field and rarely found something worth commenting.

These findings can point towards the current approach not being used correctly, or at the very least that it could be improved.

4.2.2 Rating Helpfulness of Assessments

More than two-thirds of the respondents would make use of the concept presented in the survey to rate the peer assessments they received, no matter if the feedback or score was good or bad.

Based on the responses regarding the thoroughness of the peer assessments, it seems like the respondents would utilize such a rating system to mark the quality of the assessments. Section 2.3.2 presents some research regarding the improvement of peer assessment quality, and the chosen strategy of rating peer assessments seem to be of interest to the responders.

The process of rating the helpfulness of an assessment requires low effort while still providing useful feedback for the course staff, without them having to dig deep into the

assessment. By requiring students to complete this low effort action, course staff can receive a quick overview of the assessments.

4.2.3 Change of Behavior if Assessments are Rated

37% of the students responded that they would change their behavior to put more effort into the assessment process if their peers could rate the conducted assessments. Furthermore, 57% responded that they would change their behavior to spend more time assessing if the course staff used the rating and assessment as part of the grading process.

63% of the respondents experienced that they did not receive helpful feedback from the peer assessments and nearly 60% did not show a specific interest in their peers to improve. These numbers suggest that 37% and 57% would put more effort into assessing if they were rated, or rated and graded, respectively. More effort could lead to making them higher quality and possibly providing increased learning benefits to their peers.

It seems that giving incentives to do a good job could improve the quality of the peer assessments. Students responded that they would put more effort into the assessments if they could rate the assessments as well as if the assessments comprised part of the grade.

4.2.4 Respondents Thoughts About the Concept

This section discusses the comments responders brought up in the open-ended question, as presented in Section 4.1.4.

Non-expert Assessors

Out of the 21 responses accounted for in Section 4.1, four were critical regarding having their peers assess them due to them not being experts.

While the developed concept does not address this problem, previous work by Bejdová et al. (2014) has identified the same issue and suggested solving it by defining experts group. They found that expert groups of students have potential as experts in the field, such as for student assessment. Section 8.5.1 discusses this topic further.

Inexperienced Assessors

Furthermore, six respondents were openly concerned about their peers not having enough experience to grade them.

While the point mentioned in the previous paragraph, regarding students being qualified to grade their peers, might hold true, the students are not directly grading their peers. They are suggesting a grade and the course staff will determine the final grade. However, the point still bears some weight. Having inexperienced peers grade the more experienced peers' work could be unfair, for example, if the inexperienced peer did not know about the theory behind the way their more experienced peer had done something, they might end up treating it as incorrect. Section 2.3.4 and Section 5.3.6 discusses this concern further.

If assignments are very open with regards to their grading criteria, and students get complete freedom to be creative with how they solve the assignment, and it can be hard to define if the deliverable passes its criteria. Assignment criteria are defined by course staff,

and therefore out of scope for this thesis. Introducing expert groups, as discussed in the previous section, might remedy this problem. See Section 8.5.1 which discusses this topic further.

Lack of Effort from Assessors

Another repeating point brought up through the responses was concerns towards their peers not spending enough time and effort on the assessments. By implementing quality assurance in the form of recipients rating the received assessments, students have some motivation to put more effort into the assessment. Taking the rating of the assessments into account when grading the students might further improve the quality, as the results presented in Section 4.1.3 suggests.

General Comments, Feedback and Thoughts

Some of the other points brought up in the open-ended section of the survey were interesting. This thesis addresses some of these points; such as doing quality assurance of the assessments, identifying outliers in the assessment scores, and collusion schemes where students score each other higher.

Other points were a bit more innovative; such as including course staff as one of the peer assessors for each assignment. Course staff are experts in the field and could work together with expert groups to ensure higher quality assessments, as well as providing a baseline for what the score really should be.

Someone suggested doing multiple rounds of assessments, for example, one initial assessment to assess the assignment and identifying problems, and another round at a later time to grade the assignment, at which it is possible to see how the deliverable has improved since its initial delivery. Song et al. (2016) tried this suggestion in practice, as Section 2.3.5 presents. The idea is exciting and can be tried using the system developed, however, identifying what has changed from one delivery to the next is out of the scope of this thesis.

One respondent said he or she sometimes felt demotivated during the process. Without knowing why this was the case, it might be worth surveying how the peer assessment process can be more motivating. Section 8.5.4 presents this as potential future work.

4.3 Findings

The findings from this survey mention some measures on how to improve the peer assessment process in IT2810. Furthermore, it backs up some findings from previous research, namely that the use of helpfulness ratings has potential to improve quality of the assessments. This section will focus on the areas relevant for further research, and implementation as part of a prototype.

Therefore, the concept to be developed as part of this thesis will focus on quality assurance. Previous research has found that quality assurance can increase the quality of peer assessments, as presented in Section 2.3.2, and the findings from the survey suggest

that implementing quality assurance through a rating system can be beneficial. Chapter 5 describes the design of a concept for the prototype.

System Design

This chapter focuses on the design and conceptualization phase of the thesis, and uses the presented data from Chapter 2 as well as Section 4.1 as its groundwork. Section 5.1 identifies the relevant stakeholders and their interests before Section 5.2 delves into state of the art related works to identify potential problems as well as suggesting solutions. The project aims to create an information system, so the problem-finding phase focuses on solutions that can be facilitated by such a system.

Following the identification of potential concerns comes the design of a concept. Chapter 6 details the implementation of this concept as an information system.

Lunde and Sjøvoll (2017) developed a plug-in for the Blackboard LMS to facilitate peer assessments, but experienced that support from Blackboard was severely lacking, which made the development experience sub-par. Therefore, a decision was made to create an entirely stand-alone system, which should integrate with an LMS to achieve the same benefits as if the system was part of the LMS. Having a stand-alone system also removes the constraint on one single system, which allows the institution to change the underlying LMS as well as allowing other institutions to use the system. Section 5.3.2 goes in-depth with regards to the topic of lessening data duplication as well as making it possible to use the system by following some simple standards.

The concept, therefore, puts forward the requirement of being stand-alone to any LMS. Section 5.2 identifies concerns in state of the art peer assessment systems, which creates the groundwork for further requirements for the implementation of a system described in Chapter 6.

5.1 Stakeholders

Stakeholders are people who have some gain from the system being developed. They can be anything from developers to end users, including server administrators, maintainers and testers.

The stakeholders in this system are mainly professors who manage a course and some assignment in a course, teaching assistants and other course staff with the same tasks and

Table 5.1: Stakeholders

Stakeholder	Concern
Course staff (end users)	Usability, Availability, Security
Students (end users)	Usability, Availability, Security
Developers	Maintainability
System administrators	Maintainability, Interoperability

students in such a course. Table 5.1 concisely lists the stakeholders and their concerns. This section discusses their need in greater detail.

5.1.1 End Users

The end users of the system are interested in it being available when they need it as well as usable, with regards to being easy to use. In addition to these factors, knowing that the system is secure and therefore does not leak information to people who should not have access to it is crucial.

Course Staff

The course staff, composed of professors and teaching assistants, are the ones who administer and manage assignments. Their job is to create assignments which can create value for the students participating in the course, while also making sure it stays relevant to the topic.

Their primary concern for the system is its usability so that it does not add more time to use than any current system does. Secondary to that is its availability so that it is available at any time work is to be done.

Students

Students are the ones having to hand in a review through the system, so their concern is high usability. People from different backgrounds are all potential users, so how to hand in an assignment and how to find assignments are essential factors.

5.1.2 Developers

Developers are future maintainers as well as the original developers of the information system, and their interest lies in that the system is easily maintainable and extendable if need be.

5.1.3 System Administrators

The ones making sure the system runs and is available are interested in that the system can interoperate with other systems, is easily maintainable as well as having high availability.

5.2 Identified Problems in State of the Art

Section 2.3 presents a literature review of related work in the field of peer assessments, wherein common themes are structured together. This section will discuss these findings and identify problems therein, followed by some suggested solutions, which may lay the groundwork for the concept presented in Section 5.3.

5.2.1 Improving Assessment Quality

One of the primary goals of this thesis is to improve the usefulness of peer assessments in student evaluation, especially with regards to the helpfulness of the assessments and learning experience of the students. Research presented in Section 2.3.2 goes into detail towards some specific measures to be implemented to improve the quality of peer assessments.

Suggested Solution

For this thesis, a rating system seems the easiest to implement with regards to the organization and logistics of courses work. Therefore, further discussion will base itself on improving peer assessment quality by adding ratings to the process. Other methods are also possible, but they intervene with the course in other ways, such as meta-reviews which require students, or course staff, to dive into yet another unfamiliar project, which is time-consuming, or calibration which require course staff to lay down a significant effort into creating calibration assessments. Reputation systems seem unfit in a course which has few assignments and only runs for one semester.

5.2.2 High Workload for Course Staff

Evaluating open-ended assignments is a time-consuming process. However, it is sometimes required to do so to make sure students can think for themselves and come up with open-ended answers, or if the assignment itself requires creative work from the students.

As described in Section 2.1, the way to set up a peer assessment is time-consuming, as is the way to distribute which student should review which other student's work. In the course described in Section 3.3.1, it is all done manually. While some systems support the ability to assign people to review others, they are entirely stand-alone to large LMS-es. A stand-alone system means it requires its users to create new accounts and its administrators to set up courses and assignments. The user accounts, courses, and assignments are already set up in the LMS, so why should they double up on the effort?

Suggested Solution

An information system which could connect to the current LMS to fetch students, groups and assignments could reduce the set-up time and effort.

The system should strive to achieve feature parity with systems in the current state of the art so that it provides the same features as other peer assessment systems. Some of the requirements for these features would include automatically assigning people to assess their peers, authenticating users through a central user directory, and allowing import of assignments from a pre-existing LMS.

It is essential to keep the connection to an LMS generic so that any other LMS, or even a simple user directory, can replace it.

5.2.3 Reliability of Students as Evaluators

Section 2.3.4 goes into detail regarding how students might not be reliable evaluators due to not being competent enough to assess their peers. Overall, though, state of the art research considers peer assessment to be a reliable method for assessment. Furthermore, the benefit of students assessing their peers' work is definite. Making teachers go through both the assignments and the reviews would double their work, some of which they are already in over their head.

While state of the art research defines peer assessment as reliable, the reliability rates are around 60% on the low end. These rates might not allow to fully trust peer assessments as evaluation, so further measures should be taken to improve the agreement rate.

Suggested Solution

The problem with potentially unreliable student evaluators has some research behind it already, as presented in Section 2.3.4. One suggestion to improve the reliability can be by aggregating the scores given to each student through the peer assessments and then present and compare them. By comparing the scores from multiple students on one assignment, it is possible to identify anomalous assessments or assignment scores, which identifies potential problems regarding reliability. By implementing this on a per-user basis, separate users and assessments have different boundaries for what constitutes an anomalous score.

Another potential solution is to identify a group of experts students. Expert groups have increased reliability and confidence in their assessments, as found by Bejdová et al. (2014). It is possible to place more trust in an expert group, which could alleviate the course staff of even more work. Section 8.5.1 discusses the use of expert groups as part of future work.

5.2.4 Multiple Rounds of Feedback

Section 2.3.5 presented some research where students could iteratively improve their deliverables by first receiving constructive feedback on their work and later receive a score. Iterative work is widespread, and being able to improve the deliverable before being graded can be interesting.

Suggested Solution

As presented in related work, doing multiple rounds of assessments which focus on different aspects of the work can improve the quality of the assessments. By allowing course staff to select what kind of rubrics their students should fill out during the peer assessments, the use of multiple rounds of feedback is possible. The use of multiple rounds of feedback is something that the course staff has to decide, due to the increase in time required. Section 8.5.2 discusses this as something which can be studied further.

5.2.5 Lack of Motivation from Students

Simionescu et al. (2017) found that students sometimes lacked the motivation to study. Lack of motivation can lead to less effort put into assignments, which in a peer-assessment setting would result in poor assessments. Low-quality assessments could work against the enhanced learning experience gained from peer assessment.

Suggested Solution

To motivate students to complete peer assessments, they could get some reward. In education, one of the central rewards is recognition of skill, usually handed out as a grade. It is possible to heighten the expectation towards the quality of the peer assessments by assessing the assessments. By handing out grades based on the quality of the assessments completed, motivation among the students could increase.

Students in the course described in Section 3.3.1 received no reward for doing their assessments, other than the opportunity to look at the work of other people. In itself, that opportunity is excellent; however, based on the survey results in Section 4.1, most students did not find the experience useful in itself.

5.3 Concept

This section describes the concept developed to resolve the issues presented in Section 5.2. The primary area of focus is the research questions, but the identified problems are a bit more specific towards how they can be solved.

Based on the previous work presented in Section 2.3, adding quality assurance to peer assessments could improve their quality, and several processes with the current system can be automated. In short, the concept pertains to creating an information system which requires low effort to operate while still facilitating an increase in peer assessment quality.

5.3.1 Feature Parity with Existing System

The course presented in Section 3.3.1 used Google Forms for its peer assessments. Google Forms is a simple tool for creating online forms, or surveys, and therefore does not have extensive tooling support for facilitating peer assessments compared to dedicated peer assessment tools such as TurnItIn. See Section 2.2 for a presentation of TurnItIn and other peer review and assessment tools.

Using Google Forms created extra work for the teacher who had to set up the surveys and collect the responses manually, but did not create more effort for the students. While the ease of use for the students was extraordinary, compared to state of the art tools, it lacked custom toolings like inline rubrics and automatic distribution of the assessments.

The use of Google Forms, or other survey tools, to do peer assessment did not present itself in state of the art studies. However, the problems identified by the course staff were present both in state of the art studies as well as in previous iterations of this course, so the issues in the course were in line with problems identified in state of the art research.

With that in mind, the system to be created needed to be on par with the used features from Google Forms, primarily with regards to the types of questions used. The course mostly used question which measured agreement or disagreement, where users rated some criteria of the project on a scale of one to five. For each of the agreement-questions, an open-ended question followed; where students gave a remark on why they gave that score. The surveys utilized these questions in a rubric style, which pointed the students toward specific criterions to rate.

Achieving feature-parity with the way the course mentioned above used Google Forms seemed feasible for a prototype.

5.3.2 Avoid Data Duplication

By developing a stand-alone information system, keeping track of different users require an authentication mechanism. However, to simplify the process of creating a user account in the system, the system should re-use user credentials by delegating the authentication process to an external user directory.

An authentication system makes sure the system knows which user tries to do an action, and it can use this to serve appropriate content. In the course using Google Forms, the students had to supply the names of all participants in the assessment while also supplying the project of which they assessed. By implementing an authentication system and automation of these processes, all of these steps end up being automated.

NTNU uses a standard protocol for authorizing applications, which allows anyone to set up an application which can request some data on behalf of a user, with his or her informed consent. Having a central user directory available simplifies the user creation process, so much that it happens automatically by just clicking a button to authorize the application. Section 6.2.4 describes this process in great detail.

5.3.3 Automate Distribution of Assessments

One of the primary functions in peer assessments systems is the automatic distribution of students to peer assessments. This feature should be available in the prototype as well.

The previously studied web development course manually completed this process, because Google Forms does not have such a feature. The manual process consisted of a random number generator to divide the assessments across the groups. While this process is quite simple, the division can become lopsided; some groups may receive many assessments while some other received only a few.

An information system could easily automate this process, while also making sure that the distribution happens evenly so that every group receives the same amount of assessors.

5.3.4 Facilitate Handing in Peer Assessments

One other primary task for the system to be implemented is to facilitate the process of peer assessments. The main features of this part of the system are to maintain and present the rubrics the course staff created for the assessment process and make sure handing in an assessment is simple.

The system should implement features such as the ones described previously in this concept, regarding feature parity, authentication, reducing the amount of data duplication and automating the manual tasks.

5.3.5 Allow for Rating of Received Peer Assessments

The system should implement a system for rating the assessments received through the assignments. A rating system is straightforward to implement regarding technicalities, ease of use as well as not adding more work. Furthermore, the system can also utilize the ratings during the evaluation process. Read more about this process in Section 5.3.6.

The rating system should allow students to rate the peer assessment they receive with regards to how helpful they found the assessment. The system should also allow the student to add some comments regarding the assessment if applicable, such as a rebuttal in case of an inadequate assessment.

5.3.6 Facilitate the Evaluation Process

The previous features combined allow the system to facilitate the evaluation process even more than state of the art peer assessment systems. By storing relevant information in the same system, while also connecting users and groups to each assessment, the logistics regarding evaluating and assessing each assignment becomes pretty simple. There is no need to map the various users, groups, and assignments together manually.

With the information from the ratings in the system, it is possible to find which assessments agree with each other and which does not. Based on this information it is possible to determine if there are any assessments which disagree with the norm, so-called anomalous assessments. Examples of such assessments might be incorrect assessments or assessments with low quality. The course staff could focus their efforts on these assessments to make sure the assessor receives feedback regarding the low-quality assessment, while also making sure that the recipient of the poor assessment gets a higher quality assessment with feedback to improve their work.

With that in mind, assessments can be outliers for multiple reasons. The student might have misunderstood the question or the solution in the deliverable, could have put low effort into the assessment, acted maliciously as part of a collusion scheme to gain higher grades and so on. It is crucial to keep that in mind if allowing the system to define outliers or grades automatically. Therefore, the system should allow evaluators to overview and intervene in the grading process. It should be able to function on a range from fully manual to fully automatic, with steps of manual intervention in between.

The system could also utilize the ratings to try to come up with a grade on the original assignment. If a student agrees with the feedback in the assessment and rates it highly, the assessment could be correct, and the initial assignment could be graded based on the scores

from the assessment. With multiple assessments available for each student, combining ratings can increase the accuracy of estimating a grade for the assignment.

5.3.7 Export Data

The system should be able to publish relevant information back to Blackboard, i.e., grades from the evaluation process. Furthermore, it should be possible to download an archive of all the data and information.

5.4 Refined Peer Assessment Process

This section aims to model the process of handing in a peer assessment based on the concept developed in the previous section. It will separate the tasks to be done by course staff and students and visualize how the work efforts are divided.

One of the primary goals of the system is, as mentioned previously, to increase the quality of peer assessments in a university course. With the improvements defined in its concept, this system can potentially reduce work efforts required by course staff as well as students, while simultaneously increasing the quality of the peer reviews; thereby making students learn more from the process.

In short, the tasks to be carried out to organize a peer assessment are (visualized in Figure 5.1):

1. Create an assignment (staff)
2. Hand in an assignment (student)
3. Peer assess assignments (student)
4. Rate received assessments (student)
5. Grade assignments (staff)

5.4.1 Create Assignment

The system should facilitate the process of creating a peer assessment by implementing the suggested features from the concept described in Section 5.3.

The course staff has to provide the required parameters when creating a peer assessment, such as assignment title, due date and so on. The parameters should be familiar to the ones used in similar systems as well as regular assignments. When creating an assignment for a specific course, only users assigned to that course can access it. The setup-process includes steps where course staff can add rubrics for the students to focus on through the assessments. Re-using these rubrics for future assessments is possible so that the work laid down in creating great rubrics is not lost.

Students can only access the assignment, either to view it or to hand in their assessments, before its due date and after its publication date. When the deadline passes, each student will receive assessments from their peers.

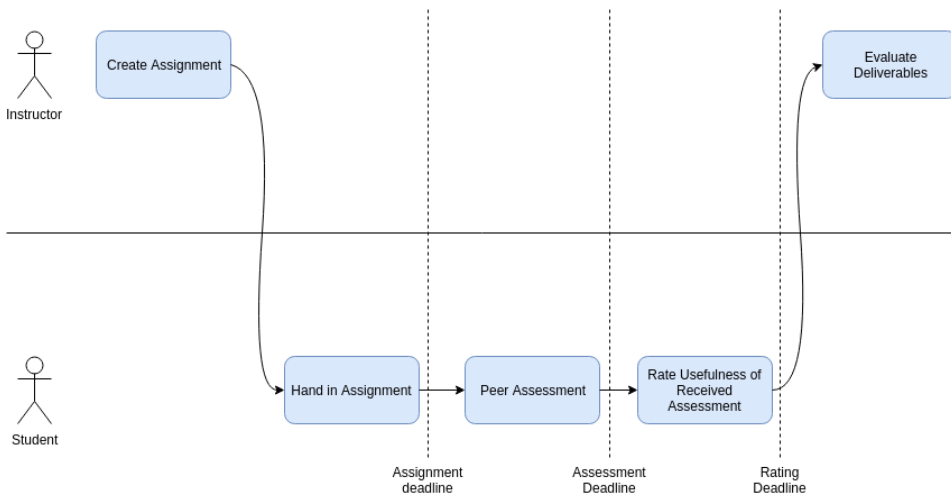


Figure 5.1: Model of the proposed peer assessment process

5.4.2 Peer-Assess Assignment

The assessment process is quite similar to the one found in similar systems, in which that they receive some deliverable from one of their peers, assess it, and fill out the rubrics.

The system should be easy to use, and it should allow for users to save their progress if they are unable to complete the full assignment in one sitting.

5.4.3 Rate Assessment

When the due date of an assessment has passed, everyone receives the assessments of his or her assignments. They can rate those assessments based on how usable they were, how much they agree with them, the quality of the feedback provided, as well as providing comments detailing these characteristics. Course staff can use these ratings to decide how accurate and helpful an assessment is. Furthermore, the use of ratings can help identify outliers in the assessment process, which the evaluation process can repercuss.

Based on how granular the course staff wants the rating process to be, the ratings can address anything from the assessment in general down to every single topic in the predefined rubrics.

5.4.4 Evaluate Assignments and Assessments

Students initially handed in an assignment; their peers assessed it, and then the student who submitted the assignment got to rate the assessment. The evaluation process is when course staff is to grade the assignments, and with the full peer assessment process completed, they have significant amounts of data to process; possibly even double the amount of a regular peer assessment process, due to the rating process.

An evaluation system should help course staff to identify outliers in the process, as well as potentially automatically assign a grade to the assignments based on their assessment and the assessment ratings. By comparing assessments done by different students for the same assignment, it is possible to identify outliers based on the score given. This information allows the evaluation system to find ways to lower the required effort, such as to assign a grade automatically for assignments where the assessments agree or when students agree with the assessments.

After completing the evaluation process, the system should be able to publish the grades for each student to an LMS to keep all student records in one place.

Implementation

This chapter presents the process of implementing an information system which addresses the concept developed in Section 5.3. The information system is a web application, where the central component is a web page which facilitates the peer assessment process. The web page communicates with a server to retrieve and store information regarding the assessments.

This presentation is a technical look at how to implement the concept, with regards to the identified problems as well as the technical aspects, integrations and architecture.

Section 6.1 describes the architecture and how it solves the needs of both the system and its stakeholders. Section 6.2 introduces the technologies in use, and Section 6.3 introduces essential third-party systems. Section 6.4 discusses the development process in general. Some of the technologies and systems have more of an impact in solving some of the identified problems, and will, therefore, receive more focus in the discussion. An example of such a discussion is the one regarding authentication and authorization in Section 6.2.4.

6.1 System Architecture

A system architecture is a way of mapping some ‘business’ goals to a set of well-known requirements (Bass et al., 2012). These requirements can be generalized down to some standard requirements which can help figure out what features are crucial for the system. With the requirements defined, testing that they accomplish what the business goal targets become quite easy.

Therefore, to satisfy the features defined in Section 5.2 and the stakeholders of the system (Section 5.1), an architecture facilitates the development process.

The end users wanted a system that was always available when it was needed and it should be straightforward to use. The system administrators and developers want a system that is easy to maintain. To satisfy these requirements, as well as the problems to be solved from Section 5.2, a client-server architectural pattern is the only relevant choice.

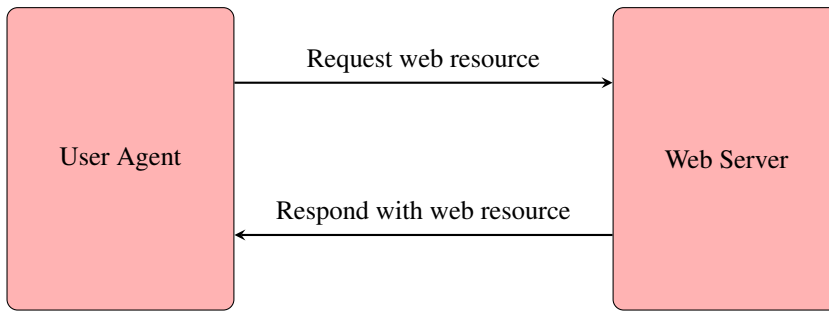


Figure 6.1: Visualization of the HTTP ‘request-response’ cycle.

The following subsections mention how the requirements of each of the stakeholders identified in Section 5.1 are solved.

6.1.1 Availability

By using the HTTP as its underlying communication protocol, the system can use the internet as its platform and therefore ‘always’ be accessible. The availability requirement will, therefore, mostly depend on the hosting of the system, which is out of the scope of this prototype. The technologies used to implement the information system does not void the availability requirement in any way.

Furthermore, the system supports ‘horizontal scaling’, which means that it is possible to start multiple instances of the service. By supporting this, it is possible to increase throughput in the application by starting multiple instances of it. This support also allows for rolling upgrades, in which the application will not be taken offline while an upgrade is in progress, allowing users to use the application even while upgrading the application.

6.1.2 Modifiability

The client and server communicate using HTTP and the standard ‘request-response’ cycle. The HTTP ‘request-response’ cycle works by having a User Agent request a web resource, for example a web page, from a web server. The server receives this request, and responds with the resource (Fielding et al., 1999). This process is visualized in Figure 6.1.

The server exposes an Application Programming Interface (API) which the client can query, and to standardize the requests and responses that could be asked and retrieved; the chosen architectural style for the API was Representational State Transfer (REST). Following pre-defined standards or styles makes it much simpler for other people to become acquainted with the work, and thus makes the system modifiable (Bass et al., 2012).

Development of the system follows the style guides defined for Python and Elm, which makes the code consistent across files and programs. Furthermore, the code base ships with some tests for the most vital processes which makes sure the functionality does not break when updating code. Section 6.4.1 details this topic further.

6.1.3 Interoperability

In itself, the system is interoperable with anything that can communicate using HTTP and follows the endpoints defined by its API, as long as it can authorize itself for those functions. For more detail on authorization, read Section 6.2.4.

With that in mind, there was one requirement for the system to automatically publish grades to an LMS. The system is required to be interoperable with such systems to achieve this. For this prototype, the LMS in question is Blackboard, and Blackboard exposes a REST API to do such communication, which makes the integration possible.

6.2 Technologies

Development of two separate systems ensued, to satisfy the architectural design choice. ‘Django’¹, a web framework in Python, was the baseline for the server (‘backend’). Read more about the backend in Section 6.2.1. Elm² was the programming language of choice for the client (‘frontend’). The API connecting the two projects was developed using the GraphQL³ library. Section 6.2.2 describes the frontend and Section 6.2.3 describes the API.

Section 6.2.5 explains how the system was hosted, while Section 6.3 goes into detail on which third-party integrations complemented the system.

6.2.1 Backend

Django was chosen for the project mainly for having something which easily scales with the project, meaning it is maintainable and easy to understand, while also being a robust framework. Having had much experience with the framework personally was also a driver; having a familiar environment is beneficial when developing a system.

Additionally, Django brings quite a bit of functionality ‘for free’. User administration, authentication, user and group management, registration and much of the security around these topics is entirely handled by the framework, which means developers can focus on the actual system and its features rather than re-inventing the wheel.

6.2.2 Frontend

Frontend web applications nowadays rely heavily on JavaScript to dynamically update a web page. JavaScript is a dynamically typed language, and with that comes its flaws. One of JavaScript’s huge flaws is the way values have multiple ways of being of null type. They can be null or undefined. Furthermore, the tooling around the current ecosystem is bizarre, requiring substantial set-up efforts to get something up and running. After having heard much praise about Elm, a new, functional, programming language for the web, as well as wanting to learn something new, Elm was to be the programming language for the frontend client.

¹<https://djangoproject.com>

²<http://elm-lang.org>

³<http://graphql.org>

Elm is a programming language which compiles to JavaScript. It ships with a compiler which requires statically typed programs while also being extremely helpful regarding any compile-time errors. This means it removes the possibility of ever experiencing runtime errors where there is a type mismatch. Because the language does not have a `null` type, it is not possible to experience a `null` value either. Elm can replace JavaScript without losing any functionality, but rather add type safety to save time hunting down bugs.

State Management

State management is how an application stores its internal state, which can consist of various data and information about the user, the web application and so on. Web applications usually store this data as structured data until visualized to the user as HyperText Markup Language (HTML).

State management usually happens server-side, maybe even in databases if not stored in the user session. When the client requests some data, the server presents it. However, web applications nowadays might request a vast amount of data for future presentation and store it in their local storage, which removes the need to query the server further when re-using data it has already fetched.

The client-side handling of such information has spawned new problems on how to keep track of all the data, how new data should enter, how data should be updated and so on. These topics are still discussed and relevant to this day, to make the development process easy to understand so that developers easily can manage the flow of data through the application.

6.2.3 API

An Application Programming Interface (API) is an interface a system exposes which allows others to communicate with it. Because this project consists of a separate client and server, allowing the client to communicate with the server is crucial. For this system, the API primarily handles serialization and deserialization of data to and from the database when the client communicates, through the GraphQL library. GraphQL is a standardized query language for APIs, and it was used in this project primarily because it seemed interesting, and there previously have not been much standardization in APIs.

Python has a library for working with GraphQL, called ‘Graphene’⁴. This library does the actual work of serializing and deserializing data, which are the processes of converting some data into or out of an internal data structure to a shared data structure, i.e., JavaScript Object Notation (JSON). When the client requests a resource from the backend through the API, the backend will request the resource from the database, serialize it into a Python type, before deserializing it into a JSON type before sending it back to the client.

The backend runs an access control system to make sure that only users with the appropriate authorization can access restricted content. This system hooks itself into the request chain before trying to fetch data from the database; making sure the user is authorized. Section 6.2.4 delves a bit further into the authorization and authentication part of the system.

⁴Graphene library: <http://graphene-python.org/>

6.2.4 Authentication and Authorization

Authorization is a process to determine if someone has permission to do something, for example, if he or she are allowed to access a specific course in an LMS. Meanwhile, authentication is a process which identifies a specific user, usually on subsequent visits. Furthermore, authentication removes the need for people to re-authorize themselves by storing information about them in a user account. The user accounts store details about authorization so that users do not have to authorize themselves each time they access a restricted function.

The developed system utilizes token-based authentication, which is like a session cookie. Users are expected to identify themselves with such a token every time they access the system.

To identify the user, each HTTP request to the server requires a token. The backend will validate this token upon every request to make sure it still is valid. If it has expired, users will be prompted to log in again. Clients are expected to send the token in the HTTP Authorization header. (This is something applications should implement, and not something end-users should be expected to do.)

For the developed prototype, obtaining such a token from the backend server requires authenticating with Dataporten using the OpenID Connect protocol. Dataporten is the authorization provider used at NTNU. The following paragraphs detail OAuth2 and OpenID Connect, while Section 6.3.1 describes Dataporten and Section 6.4.3 describes the implementation of authorization and authentication for this system.

OAuth2

OAuth2 is a process to implement authorization using tokens. Tokens are unique for each authorization, and can only be used to access the functions for which they received access. The implementation of OAuth2 is out of scope in this research; however, because it is used multiple times throughout the system, it warrants a short introduction.

Figure 6.2 shows a visual representation of the process to be carried out in the OAuth2 Authorization Grant flow. This flow allows external applications to access resources on behalf of a user, such as retrieving information about the user. To do so, the user has to authorize the application, allowing it to access data on their behalf. The flow looks a bit complex; however, it is quite seamless for the end user. The steps in the flow are:

1. Use the client application. If unauthorized ('not logged in'), the client will redirect the user to an authorization server.
2. Receive an HTTP redirect to authorize somewhere, and follow it.
3. User authorizes the request, by being logged in and answering a prompt about allowing the application to access the requested information (Figure 6.3 shows an example of an authorization prompt).
4. After validating the request and authorization from the end user, the authorization server generates an authorization code and sends it back to the user. The authorization server redirects the user back to the initial client with this code.

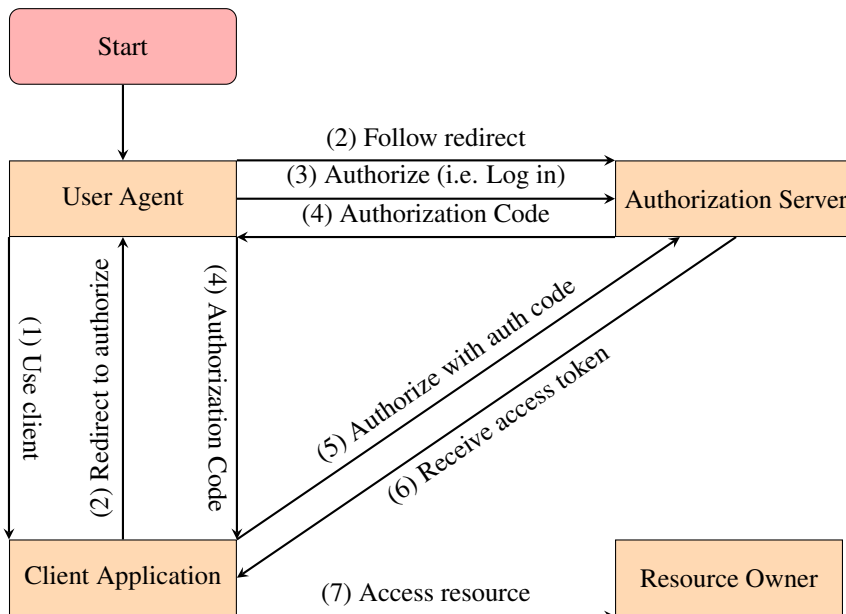


Figure 6.2: The OAuth2 Authorization Grant flow, as per Figure 3 in Hardt (2012).

5. The client extracts the authorization code and sends it to the authorization server, exchanging it for an access token.
6. Receive the access token from the authorization server, which contains the authorization to access the requested resources.
7. Access the requested resources by authorizing with the access code.

While the process seems complex, some of the steps happen in the background so that users will not notice them. The user will notice being asked to authorize the application, after which the system automatically completes the subsequent steps. Receiving the authorization code, exchanging it for an access token and finally requesting the original resource all happens in the background.

OpenID Connect

OpenID Connect is a protocol which extends OAuth2 to provide authentication since the specifications for OAuth2 does not describe that, which allows users to control their authentication by implementing measures for creating long-lived tokens which support being deactivated.

Single Sign-On

A system implementing Single Sign-On (SSO) is an information system which keeps track of a user's authentication session independently of other services, which removes the need

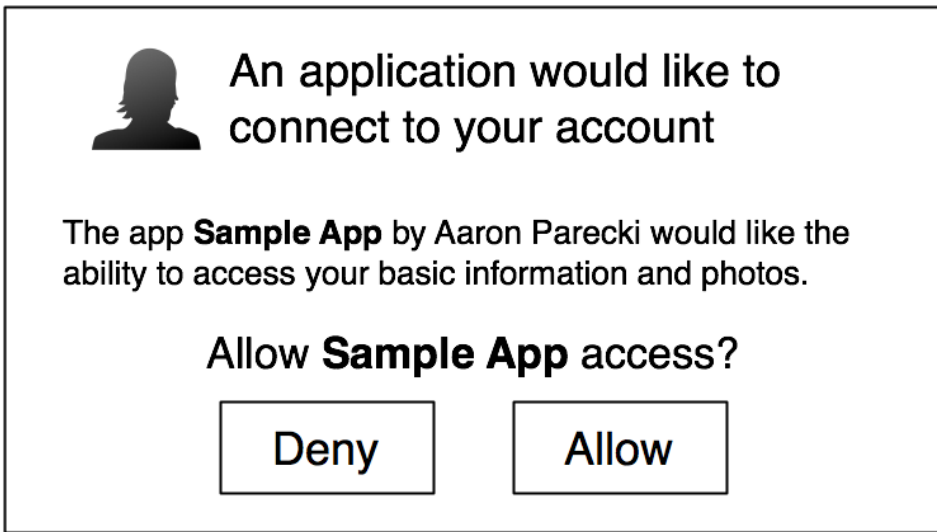


Figure 6.3: OAuth2 Authorization Prompt, by Parecki (2014)

for each of these systems to keep track of the authentication of the user. This system also removes the need for users to keep track of multiple user accounts for the various services and having to re-type their credentials when visiting different services.

6.2.5 Hosting

As briefly mentioned in the introduction of Section 6.2, the backend and frontend are different services. The backend is hosted as a standard web server, which simply requires a Transmission Control Protocol (TCP) socket listening for incoming HTTP requests. The frontend code is hosted as a static website in AWS while the backend code is hosted as a containerized web application in Kubernetes. The following sections present the hosting methods used: Section 6.2.6 present Kubernetes and Section 6.2.7 presents AWS.

6.2.6 Kubernetes

Kubernetes is a container orchestration tool. Container technology is a means of separating code from infrastructure, where the infrastructure comprises the physical components and operating system of a server. The containerization concept makes it possible to run an application anywhere rather than being restricted to a specific environment like Microsoft Windows, macOS or a Linux or UNIX flavor.

Kubernetes being a container orchestration tool means that it supports configuring an application and it will take care of starting, stopping and upgrading it upon request from the administrator.

End users will not experience any change from a traditionally hosted server on bare metal. However, due to container orchestration techniques, it is possible to release a new

version of the system it is hosting without having to shut down the application. By not ever shutting down the application, it is theoretically possible to achieve 100% uptime, even while upgrading the service. The technique of never shutting down a running server during an upgrade is called a ‘blue-green deployment’. This process is out of scope for this research.

6.2.7 Amazon Web Services

AWS is a suite of web services hosted by Amazon. Part of them is ‘Amazon S3’, which is an object storage service. This service stores any file and it is distributed around the world using the Content Distribution Network (CDN) of Amazon. Such a service is useful for hosting static files⁵, such as the frontend source code and styling of a web page, because having a full-scale web server host such a small file is a waste of resources.

Therefore, Amazon S3 takes care of hosting the frontend code as a ‘static webpage’. A ‘static website’ is a configuration option which makes Amazon S3 look for an `index.html` file and serve it as any web server would, rather than making the user download the file.

6.3 Third Party Integrations

Integration with third-party services makes it possible to reduce the need to duplicate data. By utilizing a central user directory and an LMS, there is no need to re-create user accounts, courses, course groups and their hierarchies in an external system. This section describes the process of integrating authentication with Dataporten, exporting students and students groups from Blackboard and publishing information back to Blackboard. Section 6.4.3 goes into detail with regards to the implementation of these third-party services.

6.3.1 Dataporten and FEIDE

Dataporten is a service which connects various information systems, allowing end users as well as the institutions using the service to control where the personal information ends up. It bases its authentication on top of the Felles Elektronisk IDEntitet (‘Common electronic identity’) (FEIDE) service, which is a solution developed for the Ministry of Education and Research in Norway to make authentication for educational institutions and services simple. Both these services make authentication and authorization simple, and they implement and expose standard interfaces for these processes.

Dataporten implements OAuth2 and OpenID Connect interfaces, which are protocols for authorizing and authenticating users using token-based methods. The protocols allow the user directory to be centralized while asking its users for permission to send data to external systems. To read more about these protocols, refer to Section 6.2.4.

Any student at NTNU has a user account which can log in through Dataporten, and in fact, Dataporten and FEIDE are used to log in to almost all university services. Users will, therefore, be familiar with this process. It also describes what information the application

⁵A static file is a file which seldom changes and is requested by many people.

will be able to access. In this case, the application asks for the general OpenID claim which contains a unique Identifier (ID) as well as the full name of the user. Nothing else is strictly required unless the course requests it.

Section 6.4.3 describes the implementation process of these integrations.

6.3.2 Blackboard

Blackboard is a Learning Management System, and for the time being, it is the LMS in use at NTNU, after having changed from 'It's Learning' during the summer of 2017.

As briefly mentioned in Chapter 5, Lunde and Sjøvoll (2017) researched developing a peer review module directly in Blackboard, but due to the lack of development support, the new prototype was to be developed stand-alone from Blackboard.

The Blackboard API provides access to which students attend which course, as well as if there were any course groups set up. It is also possible to query for assignments and whether the assignment expected an individual or group delivery. With this information, it is possible to set up the course and its assignments in Blackboard and fetch all the data from the course to a separate system, which keeps the assignment system familiar to its users.

6.4 Development

Development of the system started during the fall of 2017, and the following sections detail some of the essential tasks completed throughout the development. Section 6.5 details the finished solution.

6.4.1 Unit Testing

Django ships with an excellent framework for unit testing, which was helpful for creating tests to cover the critical functionality of the application. Test-driven development was one of the methods used for developing the app, where defining tests happens before writing code. By describing what the application is expected to do through unit tests, it is possible to think about the features as well as the implementation subconsciously. Furthermore, when implementing the feature, the tests are already there.

Continuous Integration

Continuous Integration is a process which runs automated tests, like unit tests, when code changes. Usually, this happens when new code is committed to the project. Utilizing such a process makes spotting errors much more straightforward, and it is used massively in open source communities.

While running the tests on a local machine would suffice, a continuous integration system was set up to run unit tests automatically when new code was committed to the code repository. While not strictly necessary, such a system is excellent to have to make sure that each new code check-in warrants a test run; it does so automatically, no user intervention required. Moreover, if something fails, it notifies the author of the code, for

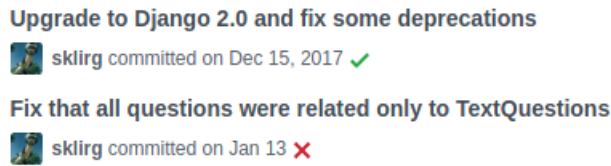


Figure 6.4: Two git commits which have their commit status set in GitHub.

example via GitHub Commit Statuses. Sometimes people forget to run tests for a simple little fix, and errors can occur. Figure 6.4 shows an example of the GitHub commit status icons, where the commit with the cross has failed a check and the checkmark symbol says it passed all checks.

Continuous Delivery

While it was handy to have a system set up to run tests on the system every time the code changed, the system could achieve other tasks as well. This particular system could automatically release a new version of the code after building a new artifact. Such a release scheme is called Continuous Delivery. After releasing a new version, the running back-end system could be updated by merely downloading the new artifact and restart. While releasing a new version of some code does not require significant effort, having a system doing it removes the possibility of human error, because the system does it the same way every time. It also removed the need to release a new version manually every time some code was updated. Moreover, it could create multiple versions which each contained some new code. If one of the versions had a bug in it, it would be easy to roll back to using a previous release until a new version fixed the bug.

6.4.2 API Versioning

A problem which can arise when developing applications that communicate together across an API is that one of the applications can be out of date compared to the other. One of the triggers of this happening can be that the data source updates the data type of an item it exposes, for example changing a user identifier from an integer field to a string field. If the consuming client believes the field always is an integer, but it suddenly exposes strings, the application may not be set up to handle that.

One of the pretty standard solutions is to create a version management system for the API. Versioned APIs allow people to choose which version of the API their systems use so that they can upgrade to the new API version when they see fit, and after doing that, they can switch the API version. For this to work, it is crucial that the developers of the API be careful when updating code, and make sure to release a new API version if something is to change. However, while API versioning solves the problem, it creates extra work by having to follow the development and upgrades of the API, making sure to upgrade the version for newer features and potential bug fixes.

Having worked with APIs a lot personally, GraphQLs vision of not using versioning, but rather creating new resources and deprecating old ones sounded smart. That, combined

UNINETT Dataporten Dashboard ▼ English

[Main overview personal](#) / [Client Scholar - Peer Assessment System](#)

Overview

[Basic info](#)

[Extended info](#)

[OAuth details](#)

[Auth Providers](#)

Scholar - Peer Assessment System

Client ID: `488944fb-514e-4cba-944e-09996874e5b4`

A system for doing peer assessments where the students rate the helpfulness of the assessments.

Redirect URIs:

- `http://api.scholar.test:8000/auth/callback/`
- `https://scholar.sklig.io/auth/callback/`

Registered 8 months ago Updated now

Figure 6.5: The OAuth2 Dashboard in Dataporten.

with the fact that GraphQL APIs expects the client to request specific data rather than the client to query a multitude of API endpoints, and then stitch the data together, was appealing.

6.4.3 Integrating with Third Party Services

To make using the system as low effort as possible, relying on integrations with other systems was important, for example, to reduce duplicating information. The two integrations this system used were Feide and Blackboard.

Dataporten


Section 6.3.1 describes Feide and Dataporten in detail; therefore, this section will focus on integrating an application with Dataporten.

Dataporten implements OAuth2 and OpenID Connect; standards for authorization and authentication, respectively. Since they are standards, it is straightforward to implement applications using them, because the standard defines how all the parties in the process should behave. Dataporten provided extensive documentation⁶ for their systems which made the implementation seamless. After having configured the OAuth2 client through the Dataporten Dashboard, it was ready for use. Figure 6.5 shows the Dataporten Dashboard.

When users visited the system, it needed to know who they were. The system redirected unauthenticated users to Dataporten to authorize themselves for the application through a standard Dataporten authorization (and if required, authentication) process, after which it redirected them back to the application. When they arrived back in the application, they had a token which the system could contact Dataporten with, to exchange for information about the user. That way, the user was identified and authenticated to use the system. Figure 6.6 shows the authorization progress, and Figure 6.7 shows the standard login prompt for Dataporten.

For more detail regarding the authorization process, refer to Section 6.2.4.

⁶Dataporten documentation: <https://docs.dataporten.no/docs/gettingstarted-tech/>




Jan ElevVGS Olsen

Feide test users

My services

Logout







Scholar - Peer Assessment System

provided by Håkon Larsen

Solbjørg

This service requests the following accesses on behalf of you:

	User ID
	Email address
	Name
	Profile photo

⌚ Access will only be granted for 8 hours after each login. You can [withdraw granted access at minside.dataporten.no](#).

Do you accept this?

✓ Yes


✗ No

Figure 6.6: The OAuth2 Authorization prompt in Dataporten.


Log in with Feide

Dataporten has requested you to log in with Feide.


UNINETT




Log in with your Feide account from NTNU. [Not your affiliation?](#)




Username





Password



[Forgot your username or password?](#)

Log in




Figure 6.7: The login form in Dataporten.

The implementation of an OpenID Connect client in the backend system was done by configuring the open source OpenID Connect client library ‘oauthlib’⁷. Such libraries remove the need for implementing something standardized over again, and they are usually tried and tested by many users.

Blackboard

The integration with Blackboard was developed using a demo version of Blackboard on a virtual machine, which made it possible to set up and test the REST API provided by Blackboard without requiring access directly to a production environment. It also meant that experimenting was possible with regards to setting up courses, students, and student groups.

However, some problems do not present themselves before a system is ready for production use, because of the difference in how the system works during development and in production. The following paragraph describes a problem which became apparent during a meeting with the Blackboard operations team at NTNU.

During a meeting with the operations team of Blackboard at NTNU, it became apparent that there was no simple way of finding a common denominator between the Dataporten user account and the Blackboard user account without having administrative access to Blackboard. Dataporten exposes unique usernames for all students, their student number as well as a Universally Unique Identifier (UUID) for each student through their API, but the only unique identifier provided by Blackboard was its internal user identifier in Blackboard, which Dataporten did not store. This problem originated in some privacy concerns in Blackboard because users have to actively opt into having their username displayed. The Blackboard team at NTNU said that they would be able to give this permission to the system so that the connection between Dataporten and Blackboard was possible.

Furthermore, to retrieve more information about the user to connect the user from Blackboard to the one from Dataporten, the system had to query the user-API one time for every student enrolled in the course to find the unique username which allows the system to connect the user accounts. If the system was relevant for multiple courses, the number of requests to fetch information about users could become quite significant, namely one API request per enrolled student per course. The Blackboard team at NTNU suggested that the students get mapped to a course when they first log in to the external system to reduce strain on Blackboard.

6.5 Prototype

This section showcases the developed prototype by detailing each step in the refined peer assessment process as described in Section 5.4. The implementation makes use of the technologies discussed previously in this section.

Title:

Evaluation of project 3 code

Description:

External resource:

Use this to link to e.g. a GitHub repository.

Publish date:

Date:

2018-01-01

Today

Time:

00:00:00

Now

Set a publish date, or it will be published immediately.

Due date:

Date:

2018-11-08

Today

Time:

23:59:59

Now

Review type:

individual

Schema:

IT1901 Schema

Allowed groups:

it2810_h17

it2810_h18

Hold down "Control", or "Command" on a Mac, to select more than one.

Number of peer assessments:

3

Select the number of peers assessing each deliverable.

Delete

Save and add another

Save and continue editing

SAVE

Figure 6.8: View of assessment creation

Change schema

Title:IT1901 Schema

HISTORY

QUESTIONS

TITLE

Is the project structured, with meaningful use

OBLIGATORY

☒

ORDER

1

DELETE?

X

TEXT QUESTIONS

TITLE

Suggest improvements, or leave some generic

DESCRIPTION

OBLIGATORY

☒

ORDER

1

DELETE?

X

+ Add another Text Question

RATING QUESTIONS

TITLE

Grade the project on a scale of 1-5

DESCRIPTION

OBLIGATORY

☒

ORDER

2

MINIMUM RATING

1

MAXIMUM RATING

5

DELETE?

X

+ Add another Rating Question

+ Add another Question

Delete

Save and add another

Save and continue editing

SAVE

Figure 6.9: View of rubrics creation

Assessments

Evaluation of project 3 code

- Test User 2's assessment of Test User 3
- Test User 2's assessment of Test User 4
- Test User 2's assessment of Test User 1

Figure 6.10: View of assessment tasks

Evaluation of project 3 code

Is the project structured, with meaningful use of directories, files, file names etc.?

Suggest improvements, or leave some general feedback

Yes, the project structure looks good. However, the files containing React components should be named the same as the component they contain. That's a pretty common thing to do.

Grade the project on a scale of 1-5

4/5

Save

Figure 6.11: View of a single assessment

6.5.1 Create Assignment

The course staff accesses the backend service of the system to set up an assessment and to create rubrics. Figure 6.8 shows the view for setting up an assessment and Figure 6.9 shows the view of creating rubrics.

When creating rubrics, they are part of a ‘schema’ which makes the rubrics reusable in other assessments.

Finally, when saving the assessment, it will automatically be distributed to the enrolled students. If some students register for the course in the future, they will be distributed too as well, as well as having assessments set up for their deliverable.

6.5.2 Hand in Deliverable

As previously mentioned, the developed prototype does not facilitate the process of handing in deliverables. However, the field for ‘external resource’ in Figure 6.8 allows hyper-linking to a resource, such as an assignment in an LMS.

Assessments of me

Evaluation of project 3 code

- Test User 3's assessment of Test User 2
- Test User 4's assessment of Test User 2
- Test User 1's assessment of Test User 2

Figure 6.12: View of received assessments

Is the project structured, with meaningful use of directories, files, file names etc.?

Suggest improvements, or leave some general feedback

The directory structure follows conventions described in Best Practices for JavaScript in 2015. File names fail to explain what the contents are. In React, it's common to name the file the same as the component it contains, which is not consistently done here.

Do you agree with this?

Any additional comments...

Grade the project on a scale of 1–5

4/5 (1–5)

On a scale of 1 to 10, how much do you agree with this review?

0/10

Figure 6.13: View of a single received assessment

6.5.3 Peer-Assessing Deliverable

When the peer assessment becomes available for the students, they will see a list of the assessments they are tasked to complete. Figure 6.10 shows a list view of the assessments the student is tasked to do, and Figure 6.11 shows the view where students assess their peers through the use of rubrics.

There is no difference between a draft or a final submission in this prototype, but students are not allowed to edit their submission after the due date has passed.

6.5.4 Rate Received Assessment

When the peer assessment becomes available for the students, they will see a list of the assessments they are tasked to complete. Figure 6.12 shows a list view of the received assessments, and Figure 6.13 shows the view of a single assessment which the recipient can rate.

Evaluation

Evaluation of project 3 code

- Test User 1
- Test User 4
- Test User 2
- Test User 3

Figure 6.14: View of the assessments ready to be evaluated

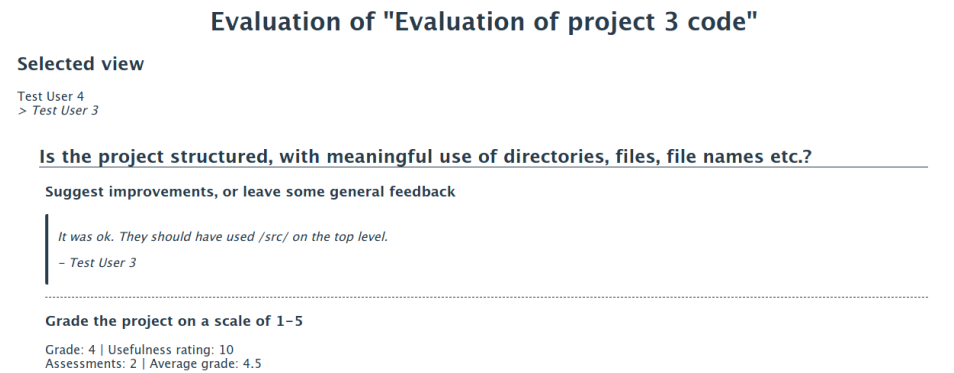


Figure 6.15: View of one assessments ready to be evaluated

6.5.5 Evaluation

The evaluation view for instructors shows a list containing all enrolled students and all the completed assessments for them. After having selected a student to view assessments for, the system will show a list of views to select from, namely the assessments done for this student. By selecting one of the assessments it is possible to see the details of that specific assessment, such as its grade and comments. Furthermore, it will show the rating the student has given that assessment as well as their comment, if they had one. The system also shows the number of assessments done for this student as well as the average grade given to the deliverable. Figure 6.14 shows the list view, and Figure 6.15 shows a view of a specific assignment done for one of the students.

⁷OAuthLib: <https://oauthlib.readthedocs.io/en/latest/>

Experiment

This chapter contains the experiments relevant to this thesis. The first experiment is a usability test, which tests the usability requirement for the application. Section 7.1 details the usability test. Following the usability test is a plan for an experiment to test the system, detailed in Section 7.2

7.1 Usability Test

This section contains the execution and results of a usability test conducted on some students, to test if the developed system satisfies the usability requirement. The test defined some scenarios for the user to accomplish.

Section 7.1.1 explains the process for conducting the usability test and Section 7.1.2 presents and discusses the results.

7.1.1 Usability Test Process

A usability test helped to make sure that the usability requirements were satisfied. To carry out the test, some students tried out the system with regards to one or more predefined scenarios based on one or more user stories. Appendix A.2.2 presents all the user stories, and Table A.6 shows the distribution of which test candidate tested which scenarios.

Before the test started, the candidate received a note with a username and password of a test user and was asked to pretend they were either a student or an instructor depending on which scenario to test. The test candidate used a computer which was set up with the system available, and the test started by the user opening a web page which contains the system, just as if they had clicked a link to get to the web page.

When the test started, the user carried out the scenario he or she was asked to complete. The candidate explained his or her thought process throughout the test and voiced the difficulty of the task after completion.

7.1.2 Usability Test Discussion

Appendix A.5 presents the granular findings from the usability test, and this section will briefly discuss the main findings. The main findings from the test is that the system has some room for improvement, mainly with regards to minor improvements such as naming things in a manner which makes it easy for the users to understand how the system is structured and providing more information in each of the views of the system. Additionally, the evaluation part of the system might require an overhaul with regards to its structure and design.

The consensus to be taken away from the usability test is that the system can be regarded usable in its current state, but a few small fixes can increase the usability drastically.

7.2 Suggested Experiment

To find out if the developed concept addresses the research questions for this thesis, this section details a plan for an experiment to test the system. Moreover, if testing the system in the same course as surveyed as part of this thesis, data is readily available for comparison.

7.2.1 Assignment Plan

For a full-scale test of the system, the assignment plans follow the one of the web development course studied, which had three assignments over the course of the semester which lasted for approximately four months. Because the concept introduces the rating of assessments, it might be required to revise the assignment plan to make sure the students have time to complete them all.

Another possibility is to conduct a test which only replaces one of the assignments, rather than to replace them all. This approach would require some more time for only one of the assignments rather than for all.

For each project to be completed, there will be three deliverables. The project itself, a peer assessment of one or more other groups' projects, and a rating of each received assessments. Combined, this could total up to a substantial amount of work, which is something to take into consideration when designing the assignment plan.

7.2.2 Assignment Content

The prototype supports the same kind of rubrics as the ones used in the web development course; namely numeric scales used to measure the degree of agreement or disagreement, like a Likert-scale. Additionally, it is possible to add open-ended text questions, primarily used for textual feedback. It is possible to extend the functionality of the system with other types of rubrics or questions if wanted.

The rubrics and questions used for the assessments must abide by the limitations defined above, namely a scale of agreement and open-ended text questions.

Is the project structured, with meaningful use of directories, files, file names etc.?

Suggest improvements, or leave some general feedback

Grade the project on a scale of 1-5

 0/5

Figure 7.1: Example of a peer assessment rubric

7.2.3 Assignment Delivery

The prototype does not facilitate the assignment delivery process. Therefore, students have to hand in their assignments in another system such as an LMS or as a hyperlinked archive. In the case of a hyperlinked resource, the system can show the hyperlink to the students. The only requirement is that the content the students are tasked to assess is available to them somehow.

7.2.4 Peer Assessment Process

The peer assessment process follows a flow in which students respond to the rubrics presented, much like a similar peer assessment system does. As mentioned in the previous section, the content to be assessed might require being downloaded beforehand from another system. Other than that, the process is quite straightforward.

As described in Section 6.4.3, the login-process consists of clicking on the ‘Log-in’ link, which goes through the authorization process described in Section 6.2.4. There is no need for students to create a new user account for the system.

7.2.5 Peer Assessment Rating Process

The peer assessment process allows the students to rate the assessment they received which ensures the quality of the assessments. The primary task for the students is to rate the helpfulness of the feedback in the assessment. Section 5.4.3 describes this process in more detail, and Figure 7.2 illustrates an example of a rating process.

The course staff can view the ratings as soon as the students submit them as well as after the deadline.

7.2.6 Evaluation Process

After students complete the assessment and ratings, it is time for the course staff to evaluate the assignments. The prototype can facilitate this process as described in detail in Section 5.4.4. Otherwise, the course staff can download the data for offline processing. Figure 7.3 shows one of the views in the evaluation part of the prototype. This view presents the assessments for a chosen student, combined with the number of ratings and the average ratings given to this assignment.

Is the project structured, with meaningful use of directories, files, file names etc?

Suggest improvements, or leave some general feedback

The directory structure follows conventions described in Best Practices for JavaScript in 2015. File names fail to explain what the contents are. In React, it's common to name the file the same as the component it contains, which is not consistently done here.

Do you agree with this?

Any additional comments...

Grade the project on a scale of 1-5

4/5 (1-5)

On a scale of 1 to 10, how much do you agree with this review?

0/10

Figure 7.2: Example of a peer assessment rating

Evaluation of "Evaluering av kode prosjekt 3"

Selected view

Test User 1
> Test User 2

Is the project structured, with meaningful use of directories, files, file names etc?

Suggest improvements, or leave some general feedback

The directory structure follows conventions described in Best Practices for JavaScript in 2015. File names fail to explain what the contents are. In React, it's common to name the file the same as the component it contains, which is not consistently done here.

- Test User 1

Comment regarding the assessment

Yes, it all makes sense. Thanks for the tip about naming the file the same as the component it contains.

Grade the project on a scale of 1-5

Grade: 4 | Rating: 10
Assessments: 2 | Average grade: 3.5

Figure 7.3: Example of a peer assessment evaluation view

Discussion

This chapter will discuss the findings of the research completed in this thesis. The background for this discussion comes from previously discussed topics; such as the literature review presented in Chapter 2, responses to the survey presented and discussed in Chapter 4 and the design of a concept described in Chapter 5.

For this chapter, Section 8.1 discusses how the concept targets and tries to solve each of the identified problems in state of the art peer assessment systems and Section 8.4 analyzes the study in itself. Section 8.5 brings up and discusses potential future work.

8.1 How the Concept Solves the Identified Problems

With an information system developed, based on some problems identified in state of the art research and survey data from students in a course where peer review was the primary evaluation strategy, it is now possible to discuss if the system, in theory, solves the problems the students faced.

The identified problems from Section 5.2 will be the primary points for discussion in this section, along with the points brought up by the survey responders in Section 4.1.4.

8.1.1 Improving Quality

One of the primary goals of the concept was to improve the quality of peer assessments, mainly with regards to the assessment part of the process. This focus originated by looking at previous assessments as well as in the literature study, and the survey responses confirmed that students were unsatisfied with the quality of the assessments.

As presented in Section 2.3.2, there are multiple ways to improve the quality of peer reviews. Some of them use authority figures such as course staff, and some methods make the students themselves ensure the quality of the work.

The prototype implemented as part of this research uses helpfulness ratings to rate the assessments, which allows the author to express how helpful the feedback was.

Other systems implement other methods, such as the calibration technique described in Section 2.3.2, but the choice for this prototype landed on helpfulness ratings, mainly due to the simplicity of implementing it combined with the low amount of extra work required for the course staff and students. Furthermore, it is possible to apply statistical functions to the ratings which make it possible to automate parts of the evaluation process without depending on comprehending open-ended texts.

Large communities of people often use reputation systems to award skilled contributors, such as the reputation system in StackOverflow¹, where users ask questions, and anyone can respond. The most useful responses gain votes from other community members and will be a permanent part of their track record on the site. Such a system could be helpful in a course as well, which could identify experts in the field, or maybe just some helpful peers. However, the process of gaining traction and reputation is long-winded, and university courses only last for a limited time, so such a system could end up not gaining users enough reputation to separate them from each other.

It is also possible to use meta-reviews where either peers or course staff assess the assessments. This process would be extra time-consuming in a course because the meta-reviewers have to study the underlying deliverable with regards to the assessment, which essentially doubles the time spent.

However, if the goal is to reduce work efforts from the course staff, methods which target a high number of students with a low effort from the course staff are preferred. Examples of such methods are reputation systems, meta-reviews and helpfulness ratings. Calibration reviews are also low effort, but it does require some initial effort, and depending on how granular identification is wanted, might require more calibration reviews.

There are multiple methods which can help improving peer assessment quality. Some of them require more efforts from course staff while some require more efforts from students. Some methods even require more effort from both parties. This prototype used helpfulness ratings as the method for gaining feedback about the assessments because it is a pretty simple step which does not add too much effort to the already time-consuming process of assessing open-ended student work.

8.1.2 Reducing Administrative Efforts

Coming up with an assignment is much work in itself, and then having to organize the delivery of such assignments, especially ones such as peer assessments which require considerable effort in itself, is time-consuming. Section 5.2.2 talks about utilizing information already registered in, for example, an LMS or a user directory. External systems can consume data such as students signed up for a course, course groups, assignments, due dates and so on from a central system. By doing this, work efforts which regard duplicating data are reduced, or even removed, and course staff, as well as students, can focus on the essential matters rather than the logistic ones.

Course staff spends a considerable amount of time on evaluating the assignments, especially with regards to the conducted peer assessments. By using statistical calculations, it is possible to discover which assessments agree and disagree with each other. Likewise, with the implementation of a rating system to quality control the assessments, it is possible

¹ StackOverflow: <https://stackoverflow.com/>

to identify which authors agree with their assessments. This statistical data help to identify outliers in the process; people who disagree with the norm. Reasons for disagreeing can be many, such as misunderstanding a question or people colluding to gain a higher grade. By identifying these anomalies, course staff can focus their work efforts on the students who vary from the norm.

As previous research suggests, defining useful rubrics can be time-consuming. However, re-use of the rubrics is possible, due to the openness of the assignments. Furthermore, designing rubrics is not time-constrained in the same way providing assessment is; the design of rubrics can happen before the course starts.

8.1.3 Reliability of Students as Evaluators

Section 2.3.4 presents previous work related to the reliability of students as evaluators. While there are problems with letting students evaluate their peers, such as due to lack of experience or the potential for collusion, allowing students to assess their peers by giving them specific criteria in the form of rubrics forces the students to evaluate based on a specific guideline. Since the students handed in the original assignment which expected them to follow these guidelines, they are expected to know those parts of the curriculum, and they should, therefore, be competent enough to judge others on it. Furthermore, previous work as presented in Section 2.3.4 confirms that it is possible to rely on students as assessors in peer assessments. The agreement rate between the grades given by students and teachers is between 60% and 70%. By defining expert groups, the reliability of their assessments can be as high as 94%. If students are unable to give a skilled assessment of their peers, it could reflect poorly on their grade. While this is an interesting discussion, how the grading criteria are set up and evaluated is out of the scope of this research.

Some students expected their reviewers to be experts in the field; which might be a fair expectation, but not a realistic one. Course staff often hire teaching assistants to help with grading work, and they are not necessarily experts in the field either. Peers can be inexperienced or experts, there is no way to tell beforehand. By using peer review calibration techniques or identifying expert groups, it is possible to gain information about the reliability of the students. By utilizing this information, it is possible to divide the work efforts even more, by accounting the reviews from experts higher than the ones from other students. While it does not entirely solve the problem at hand, it ensures to evaluate all students fairly. Section 8.5.1 further discusses the use of expert groups.

As presented previously, such as in Section 2.3.4, it is still crucial for course staff to overlook the process to ensure that the results are legitimate. Song et al. (2017) suggests that it is possible to identify collusion as long as collusion is not the norm. If the majority of reviewers partake in a collusion scheme, outliers will be the ones who fairly assess their peers, and collusion will be the new mainstream, making it hard to recognize.

8.1.4 Multiple Rounds of Feedback

Section 2.3.5 presents a study which researched the use of multiple rounds of peer assessment feedback. They worked with two rounds, one formative and one summative,

and experienced an improvement in the quality of the assessments. This topic could be interesting for further research as Section 8.5.2 presents.

The prototype developed as part of this thesis uses multiple rounds of feedback, but the author of the feedback changes from the assessor to the content author, so it does not happen in the same way as presented in Section 2.3.5.

8.1.5 Lack of Motivation from Students

Some of the survey responders, as well as some previous work, showed that students sometimes lack the motivation to complete peer assessments. The cause of this lack of motivation has not had much focus in this research. Even though the cause is unknown, based on survey results, it seemed that students would be more motivated to complete a peer assessment if they gained something from it; either the social effects of being rated on their work, or the even more if their assessments comprised part of their grade in the course.

Previous studies tried using gamification to increase motivation from students, and that is something that could warrant further study. Since the implemented system is entirely stand-alone, it is possible to extend it with such functionality.

Section 8.5.4 presents this topic as a field for further studies.

8.1.6 Large Courses

With the rise of online studies, such as MOOCs, it is crucial to be able to scale the assessment process without necessarily adding more course staff to do assessments.

Many courses with a high amount of students use multiple-choice quizzes as assignments. Depending on the course and the curriculum, this might be a good fit. Studies have shown that there is a correlation between the results on a multiple-choice examination and a similar open-ended exam (Wage et al., 2011).

However, the assignments in the surveyed course consisted of various projects where groups of students were to develop web systems using a predetermined technology. Explorative assignments such as these are hard to replace using multiple-choice questions.

Open-ended assignments require massive efforts to assess since the reader has to comprehend the content and identify how it answers the question posed in the assignment. As mentioned in Section 2.3.2, state of the art natural language processing has potential in application here, but until that is the case, human beings will have to do the work. Furthermore, exploiting the students to do the work both reduces course staff work while it also enhances the learning experience of the students.

8.2 Discussion of Research Questions

This subsection discusses the research questions defined in Section 3.1.

8.2.1 Research Question 1

The concept developed as part of this thesis tries to improve the usefulness of peer assessments by increasing the quality of assessments, making sure students are reliable assessors, and by implementing multiple rounds of feedback in the assessment process. These factors have shown promise in previous studies, and research has confirmed that students can be reliable assessors. Furthermore, by implementing a rating system, students are allowed to rate the helpfulness of the assessment, which has shown promise in a survey targeting students in a web development course at NTNU.

8.2.2 Research Question 2

The use of peer assessment in a course at NTNU required teachers to intervene for particular assignment submissions where students submitted rebuttals due to some assessments being of low quality. By implementing a rating system for the assessments, making the authors rate the feedback they received, the usefulness of the assessment shows the course staff which of the assessments or assessors might require intervention. Moreover, implementation of statistical measures allows course staff to filter out assessments where the author agrees with the assessor, to remove noise in the evaluation process, as well as to have an overview of the level of quality of related submissions readily available.

Previous research in the field of peer assessments has found peer assessors to be reliable graders with regards to grading students with an agreement rate of 60% to 70% compared to the grading done by the teacher. While a non-perfect agreement rate will not substitute a teacher, implementing algorithms which can identify the expertise and knowledge of individual students can help the system to define grades based on peer assessments. A study by Bejdová et al. (2014) found that expert groups can have agreement rates as high as 94%, so if such a group exists in a course, weighing their assessments higher when it comes to evaluation might alleviate more of the burden put on of course staff while still maintaining the legitimacy of the evaluation method.

While many processes can be automated; allowing computer systems to automate the evaluation might allow for collusion to occur. As mentioned in Section 2.3.4, algorithms exist to detect such schemes until the majority of students is part of such a scheme. Other imperfections may also exist, so the usage of a system requires further research and testing, particularly with regards to the concerns mentioned in future work of the research in Section 8.5.

8.3 Analysis of the Research Approach

Based on the findings in the literature review in Section 2.3 some research questions were defined. Following these came a survey targeting students in a web development course who had used peer assessments as an evaluation method in the course, to find out if the students identified with the issues present in state of the art research.

The research questions, together with the identified problems in the state of the art of peer assessment, were the groundwork for the conceptualization and design phases of this thesis. The concept tried to solve the issues present in state of the art peer assessment

which related to the research questions. Following the concept development, implementation of an information system ensued.

The plan was to test the information system in a course during the spring of 2018. Unfortunately, because the university recently changed LMS, the contacted course staff did not show interest in trying out yet another new system, so the experiment had to be canceled.

The implemented concept seems to solve the findings from the literature review and survey responses. While the research is missing an experiment which tests the system, the theory behind its implementation is grounded in sound research, and the problems it solves are present both in relevant theory and in practice.

8.4 Analysis of the Results and Discussion

This section targets concerns regarding the results and conclusion presented in this thesis. The primary concern regards how the lack of a field test impacts the conclusion of the thesis.

8.4.1 Missing Field Test

The results of the survey targeted at students enrolled in the web development course suggest that students find the features developed in the concept usable, namely with regards to the use of quality assurance to ensure that students spend time on their assessments. The results of the survey match the concerns identified in previous work in the field, and previous studies suggest ideas that the students agree would increase their motivation to complete proper assessments.

It was not possible to conduct a field test of the developed prototype during the spring semester of 2018, as initially planned, due to lack of interest from the multiple courses contacted.

The lack of such a field test of the concept and prototype makes it hard to conclude an answer to the research questions. However, the theory behind the concept has a firm grounding in related work as well as the conducted survey. Section 7.2 details a field test which can be carried out in the future.

8.5 Future Work

Studying relevant previous work found many theories which could improve peer assessment. Some of them require time, and some of them require efforts from course staff. To make the adoption of the system as inviting as possible, methods that reduced the time required as well as the efforts of course staff had the highest priority.

Therefore, some methods were dismissed in favor of others, even though they might have vast potential in other applications. This section will discuss future work relevant to this research and the prototype.

8.5.1 Expert Groups

Bejdová et al. (2014) found that experts in a field have potential to do an excellent job assessing their peers, so the use of experts or expert groups is of interest for further study. By identifying expert users, it is possible to reduce the work required by course staff even further. The way to identify such students, especially in big courses, could be tricky. One idea is to explore the use of calibration reviews to identify who did an excellent job with them, and see if that could mean that they are experts. Furthermore, meta-reviews, ratings or rankings could all be potential methods of finding experts.

8.5.2 Multiple Rounds of Feedback

The use of multiple rounds such as briefly discussed in Section 8.1.4, seems like an interesting approach for further studies. Song et al. (2016) studied this and found that the quality of assessments, both formative and summative, improved with more in-depth reviews while also lowering the empty response-rate.

Furthermore, instead of only two rounds of feedback, the process can consist of multiple rounds where the students iteratively improve upon the deliverable until a final delivery. Such a process allows for multiple assessors to assess the work at multiple stages, and a lot more people can set their eyes on the work, which increases the chances for spotting improvements. This process is similar to the one of agile software development, in which work undergoes continuous development until complete.

For university courses, long-winded assignments might be challenging to implement, due to the various criteria which need grading. Such a process could last multiple weeks, or even months, which requires the assignment to either be quite complicated or the course might not be able to visit all the topics at hand.

The number of times an assignment has to undergo assessment for it to be close to perfect has the potential for further studies, to determine the optimal number of rounds of feedback for an assignment.

Multiple rounds of feedback for peer assessment has the potential to improve the quality of assignments, as well as the quality of the assessments themselves, which ensures that the peer assessment process enhances the learning experience for both the author and reviewer.

8.5.3 Identifying Collusion

Song et al. (2017) presents an algorithm to detect collusion in a peer assessment setting. It could be relevant to implement this algorithm on the reviews to identify collusion schemes, which some students meant were apparent in the course surveyed in Chapter 4.

8.5.4 Increase Motivation

Some responses to the survey mentioned lack of motivation for completing the peer assessments. A study by Simionescu et al. (2017) reported the same thing and tried to solve it by implementing gamification in the peer assessment system. Further studies on why students lack motivation for peer assessment can be relevant, as well as how to make the

process more motivating to follow. A look at implementing gamification in the prototype can be one idea of increasing motivation.

Conclusion

This thesis explores the use of peer assessment as an evaluation method in an educational setting. The peer assessment idea comes from the use of peer reviews in scholarly work, which has a significant impact on how scientists produce scientific work, and it also has a proven beneficial effect on learners.

The course coordinator of a web development course at NTNU had identified some problems regarding the effort put into peer assessments in the course and wanted to find out how this could be solved. Based on the already identified issues, a literature review identified some further concerns regarding state of the art peer assessment which became the questions of research throughout this thesis.

Following the literature review came a survey targeted at the students enrolled in the course mentioned above. A majority of the survey respondents responded that they would have put more effort into their assessments if the recipient could rate them, and an even higher majority would do so if the rating could impact their grade in the course.

These findings formed the foundation for a concept which set out to solve the identified issues, and it came to life as an information system. The primary focus points of the implementation and conceptualization of a prototype regarded simplicity and automation while resolving the identified problems. By making use of well-defined standards within the software industry, anyone can use the application by connecting it to a generic user directory employing the same standards.

The concept shows promise with regards to previous research and the conducted survey. However, to verify that the prototype solves the identified problems, a real-life test was scheduled for a course during the spring of 2018. Unfortunately, due to lack of interest from any of the contacted course staff, it was not possible to carry through with the test. The lack of such a full-scale test makes concluding the practical impact of this thesis impossible at this time, and further research is required.

While it is not possible to conclude the practical impact of the work done without a full-scale test, the work in this thesis should be able to answer the research questions based on its theoretical background.

RQ1: How can the use of peer assessments be improved with regards to the usefulness and learning experience for students?

The use of helpfulness ratings has shown improvement in peer assessment quality in state of the art research as well as in a survey conducted for enrolled students in a web development course. The rating process requires low effort, and students can easily rate the helpfulness of assessments regarding their assignments. These findings show that the quality of peer assessments can improve, which will increase the usefulness of the assessments as well as maybe increasing the learning experience for the participating students.

RQ2: How can course staff spend less time evaluating peer assessments while still maintaining legitimacy?

The course staff has a lot to do with regards to managing peer assessments. Some of the tasks are; logistics of using external systems to conduct a peer assessment, distributing assessments to all the students, and manually connecting assessments with their respective students and authors. An information system can automate these tasks, which will allow the course staff to spend more time evaluating and less time organizing.

Furthermore, to reduce the time spent evaluating assessments and assignments, the students are asked to complete a helpfulness rating of their assessments, which indicate how much they found the assessment helping them, or how much they agreed with it. By aggregating the ratings collected through this process, it is possible to identify assessments which do not conform to the norm. With information like this available, course staff can prioritize and focus efforts where it is needed.

Bibliography

- Abramovich, S., Schunn, C., Higashi, R. M., Apr 2013. Are badges useful in education?: it depends upon the type of badge and expertise of learner. *Educational Technology Research and Development* 61 (2), 217–232.
URL <https://doi.org/10.1007/s11423-013-9289-2>
- Ames, C., 1990. Motivation: What teachers need to know. *Teachers college record* 91 (3), 409–421.
- Bass, L., Clements, P., Kazman, R., 2012. *Software Architecture in Practice*. Addison-Wesley.
- Bejdová, V., Kubincová, Z., Homola, M., July 2014. Are students reliable peer-reviewers? In: 2014 IEEE 14th International Conference on Advanced Learning Technologies. pp. 270–272.
- ConfTool GmbH, 2018. ConfTool: Conference management software. Accessed: 2018-04-11.
URL <https://www.conftool.net/index.html>
- Davies, P., 2009. Review and reward within the computerised peer-assessment of essays. *Assessment & evaluation in higher education* 34 (3), 321–333.
- Deterding, S., Dixon, D., Khaled, R., Nacke, L., 2011. From game design elements to gamefulness: Defining "gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. MindTrek '11. ACM, New York, NY, USA, pp. 9–15.
URL <http://doi.acm.org/10.1145/2181037.2181040>
- EasyChair Ltd, 2018. EasyChair smart cfp. Accessed: 2018-04-11.
URL <https://easychair.org/cfp/>
- Falchikov, N., Goldfinch, J., 2000. Student peer assessment in higher education: A meta-analysis comparing peer and teacher marks. *Review of educational research* 70 (3), 287–322.

-
- Fang, H., Wang, Y., Jin, Q., Ma, J., Dec 2017. Rankwithta: A robust and accurate peer grading mechanism for moocs. In: 2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE). pp. 497–502.
- Fielding, R. T., Gettys, J., Mogul, J. C., Nielsen, H. F., Masinter, L., Leach, P. J., Berners-Lee, T., June 1999. Hypertext transfer protocol – http/1.1. RFC 2616, RFC Editor, <http://www.rfc-editor.org/rfc/rfc2616.txt>.
URL <http://www.rfc-editor.org/rfc/rfc2616.txt>
- Gehringer, E. F., 2014. A survey of methods for improving review quality. In: Cao, Y., Våljataga, T., Tang, J. K., Leung, H., Laanpere, M. (Eds.), *New Horizons in Web Based Learning*. Springer International Publishing, Cham, pp. 92–97.
- Gerrit, 2018. Gerrit code review. Accessed: 2018-04-11.
URL <https://www.gerritcodereview.com/about.md>
- Hardt, D., October 2012. The oauth 2.0 authorization framework. RFC 6749, RFC Editor, <http://www.rfc-editor.org/rfc/rfc6749.txt>.
URL <http://www.rfc-editor.org/rfc/rfc6749.txt>
- Lunde, A., Sjøvoll, R. S., 2017. Peer Review Module Development in the Blackboard Learning Management System. NTNU.
- Moodle, 2018. Workshop activity. Accessed: 2018-05-18.
URL https://docs.moodle.org/35/en/Workshop_activity
- Mostert, M., Snowball, J. D., 2013. Where angels fear to tread: Online peer-assessment in a large first-year class. *Assessment & Evaluation in Higher Education* 38 (6), 674–686.
- Oates, B. J., 2006. *Researching Information Systems and Computing*. SAGE Publications.
- Parecki, A., 2014. OAuth2 simplified. Accessed: 2018-05-14.
URL <https://aaronparecki.com/oauth-2-simplified/>
- Ramachandran, L., 2013. Automated assessment of reviews. North Carolina State University.
- Sadler, P. M., Good, E., 2006. The impact of self-and peer-grading on student learning. *Educational Assessment* 11 (1), 1–31.
- Simionescu, S., Šuníková, D., Kubincová, Z., May 2017. Gamification of peer assessment in learning management system. In: 2017 18th International Carpathian Control Conference (ICCC). pp. 571–575.
- Song, Y., Hu, Z., Gehringer, E. F., Oct 2017. Collusion in educational peer assessment: How much do we need to worry about it? In: 2017 IEEE Frontiers in Education Conference (FIE). pp. 1–8.
- Song, Y., Hu, Z., Guo, Y., Gehringer, E. F., Oct 2016. An experiment with separate formative and summative rubrics in educational peer assessment. In: 2016 IEEE Frontiers in Education Conference (FIE). pp. 1–7.

Topping, K., 1998. Peer assessment between students in colleges and universities. *Review of educational Research* 68 (3), 249–276.

Turnitin LLC, 2018a. Screenshot of user interface in turnitin feedback studio. Accessed: 2018-05-11.

URL http://turnitin.com/assets/en_us/media/feedback-studio-demo/

Turnitin LLC, 2018b. Turnitin for higher education. Accessed: 2018-04-11.

URL http://turnitin.com/en_us/higher-education

Wage, K. E., Buck, J. R., Hjalmarson, M. A., Nelson, J. K., Jan 2011. Signals and systems assessment: Comparison of responses to multiple choice conceptual questions and open-ended final exam problems. In: 2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE). pp. 198–203.

Yanqing, W., Hang, L., Yanan, S., Yu, J., Jie, Y., Aug 2011. Learning outcomes of programming language courses based on peer code review model. In: 2011 6th International Conference on Computer Science Education (ICCSE). pp. 751–754.

Appendix

A.1 Questionnaire

The first part of the questionnaire was conducted by the professor of the course and is not included. None of the responses from that part are used in this research.

A.1.1 Question 6

Likert scale from 1 – 5 for each question.

Table A.1: ‘Identify with the following statements’

Question	Alternatives				
I had lots of earlier experience with regards to the technology in use in IT2810.	1	2	3	4	5
Survey responses	28%	29%	20%	15%	8%
I was satisfied with the thoroughness of the review I received for each project (assignment).	1	2	3	4	5
Survey responses	23%	40%	22%	14%	1%
Those who reviewed me were interested in me doing better next assignment.	1	2	3	4	5
Survey responses	43%	26%	23%	6%	2%
I was interested in making those who I reviewed better for next assignment.	1	2	3	4	5
Survey responses	14%	15%	29%	32%	10%

A.1.2 Question 7

Table A.2: ‘What fits best with regards to how you felt receiving feedback from your peers?’

Question	Survey response
The feedback helped me a little bit, but I am pretty experienced in the field already.	37%
The feedback gave me some help about things I did not know about, and other minor improvements.	62%
The feedback gave me quite a few tips about things I did not know about.	0%
I learned a lot from the feedback.	1%

A.1.3 Question 8

Table A.3: ‘What fits best with regards to how you felt giving feedback to your peers?’

Question	Survey response
I seldomly found something to comment upon, because I am pretty inexperienced in the field.	17%
I could comment on a thing here and there.	35%
I could comment on quite a bit for most of the reviews.	28%
I always found something to comment on.	20%

A.1.4 Question 9

Table A.4: ‘Thoughts about being able to rate the review’

Would you like to be able to ...	Yes	No
... rate the review you received if you got a bad score?	88%	12%
... rate the review you received if you got bad feedback?	85%	15%
... rate the review you received if you got a good score?	73%	27%
... rate the review you received if you got good feedback?	71%	29%

A.1.5 Question 10

Table A.5: ‘Thoughts about time spent revieweing’

Would you spend more time doing reviews if ...	Alternatives		
	Yes	No, I spent a lot of time revieweing	No, I would not change anything
... others could rate the review you did?			
Survey response	37%	54%	9%
... they impacted your final grade?	Yes	No, I spent a lot of time revieweing	No, I would not change anything
Survey response	57%	35%	8%

A.1.6 Question 11

Responses to the open-ended question where respondents were asked to generally comment the concept. Answers are kept original in their original language, but some of them might be translated and used in the thesis.

1 Synes det er helt feil å gi ansvar for vurdering til studenter, da faglærer eller undervisningsassistenter ikke står for noe som helst av dette. Det hele føles som en ansvarsfraskrivelse fra faglærer, og når vi i tillegg følte oss ‘truet’ med å få dårligere karakter dersom faglærer skulle vurdere prosjektet selv så føltet det hele rett og slett urettferdig (det var noe faglærer sa i en forelesning). Noen av de som vurderte meg trakk meg f.eks. for ting jeg hadde klart og tydelig skrevet i den lille dokumentasjonen hva som var grunnen, fordi vedkommende trolig ikke ville vurdere dette.

2 Opplevde at folk ikke gjorde det skikkelig, da det viste seg ved flere anledninger at de hadde oversett en del viktige ting, som gjorde at vi fikk dårligere vurdering.

3 Jeg synes kanskje det er bedre å ha folk som kan faget godt vurdere prosjektene som teller mot karakteren.

4 Problemet, slik jeg så det, var at det var mange som ga tilbakemeldinger basert på liten personlig erfaring og hadde liten forståelse for løsninger som var utført annerledes en hva de selv ville ha gjort. Det var enkelt å se hvilke tilbakemeldinger som kom fra personer som kunne dette fra før og de som ikke kunne det, da de som kunne det fra før var (konstruktivt) kritiske, mens de andre bare sa noe sånt som ‘bra’ og ‘fint’

5 Dette er ikke direkte relevant til selve systemet, men et problem med peer-review er når du enten blir evaluert av noen som ikke gjør en skikkelig jobb, eller av noen som åpenbart ikke har kompetanse til å vurdere deg. (Noe jeg opplevde i IT2810)

6 Det er litt rart å be studenter, som nettopp har lært en ny teknologi, og evaluere andre som har mye erfaring. Det vil da si at måten personen har lært teknologien på vil definere tilbakemeldingen som personen gir. Dette virket det som om skjedde ved flere tilfeller og skapte mye negativitet rundt faget.

7 ingen *Removed empty response*.

8 Jeg synes vurderingsmåten som ble brukt er en veldig dårlig løsning. Det var mye tid som ble brukt på å sette seg inn i andres kode uten at man nødvendigvis lærte så mye av det. Jeg ser poenget av å lære av andre og å være i stand til gå gjennom andres kode, men om det var det som var hovedpoenget burde det helles legges opp til en øving der man går gjennom et repo med høy kvalitet som faglærer har valgt på forhånd, eksempelvis <https://github.com/bertin/blog>.

9 Det er greit å lære rammeverker og bibiloteker alene, men da er det ikke noe poeng å ta faget på et universitet. Hadde vært bedre med litt mer stoff på forelesningene. Kanskje gjør noen enkelte prosjekter sammen i Angular/React osv. Kan vurderes å lære om Vue.js, da det er blitt ganske populær også. Kanskje bytte det ut mot React Native. Progressive Web Apps er framtiden.

10 Det var veldig stor grad av tilfeldighet når det gjaldt hva slags kompetanse gruppen som skulle vurdere vårt prosjekt hadde. Ofte fikk vi udugelige vurderinger og tilbakemeldinger fordi gruppene ikke ante hva de pratet om og bare skrev noe tull fordi det stod at man **måtte** skrive noe.

11 Nei *Removed empty response*.

12 problemet her er at noen studenter har mer og mindre erfaring en andre. Studenter men mye erfaring vurdere for stengt mens dårlige studenter ikke skjønner va de vurderer og derfor klarer de ikke å gi en rettfærdig vurdering, de kan trekke på ting som er bra fordi de ikke forstår teknologien. De fleste kommentarene vi fikk var dårlige, f.eks at editoren vi bruker la in space mens vi brukte tab så trakt de gjerne 5 av 10 poeng. Andre stender kunne jeg faktisk argumentere for at vi følge best praktis mens vi ble trukket av studenter som ikke skjønte teknologien. Vi fikk også kommentaren på siste øving at vi ikke hadde tester, vi hadde skrevet 30 - 40 stk. (der fikk vi 1 av 10 poeng).

Min erfaring er at ingen grupper vurdere samlet, alle sammen tar en gruppe hver og tar de fortest mulig.

Min erffaring fra NTNU er at hvis prosjekt blir stor og arbeidsmengden er den eneste som skiller karakterer vil alle jobbe mye, i dette faget måtte vi jobbe minst 20t i uken hver. Studentevalureingen ble et unødvendig tidleg som gjorde folk demotiverte og prøvde å

få gjennomført fortest mulig. Det er altså for stor arbeidsmengde uten at vi må evaluere andre.

Personlig synes jeg det virker som student evalueringen kun kommer pga latskap fra lærer staben. Men dette er også det gøyeste og en av de mest lærrike fagene jeg har hat. Jeg vil anbefale faget til andre studenter hvis det ikke var student evaluering. De fleste jeg har snakket med mener at studentevaluering er som å triller terninger.

13 1. Man må vurdere mindre enn 5 grupper om gangen.

Det var helt urealistisk at vi skulle bruke lang tid og gjøre en god jobb på å vurdere hele 5 grupper per prosjekt.

Det holder nok med 1 gruppe, eller maks 2. Hvis man skal vurdere flere grupper per prosjekt (og gjøre det skikkelig) så må resten av arbeidsmengden i faget justeres.

2. Man må kvalitetssikre vurderingene på en måte (slik dere spør litt om over her)

Vi opplevde for eksempel noen grupper ga oss lav score på noen av kriteriene uten å forklare hvorfor. Noen tilfeller der mesteparten av gruppene gir oss 9 eller 10 poeng av 10 mulige på et kriterie, mens noen grupper(mindretall) gir oss 3/4 av 10. Vi vet jo at vi har gjort en bedre jobb enn det, og likevel påvirker det scoren vår.

14 Når vi så på resultatene av et prosjekt var det alltid en gruppe som ikke tok det seriøst. Så vi fikk dårligst score på alle spørsmålene. Dette er ganske irriterende. Ellers fungerer det bra

15 Det største problemet med denne typen vurdering er de sprikende forståelsene av oppgavene og vurderingskriteriene. Det er greit nok at man kan gi tilbakemelding på om ting som skulle være tatt i bruk har blitt tatt i bruk, men å gi poeng ut fra hvor bra man synes noen har gjort det blir fort kaotisk. Kan kanskje være bedre å få en midlertidig vurdering, for så å få poeng ut fra hvor mye det har forbedret seg til siste vurdering. Dette krever da at samme personer vurderer de samme gruppene om igjen. Uansett bør det bli mye klarere hva som forventes at skal være med for de ulike punktene/kravene for oppgaven, med mulige eksempler på gode/beste løsning. Kan være greit å få med en vurdering fra en stud.ass/foreleser per gruppe også og/eller se bort fra veldig sprikende vurderinger (4 grupper gir 8-10 som poeng, mens en gruppe kun gir 1).

16 Leg lærte litt av å vurdere andres prosjekter og fikk litt bra tilbakemelding, men foreleseren mister pålitelighet av opplegget ettersom det kan virke som at han/hun bruker mye mindre tid på å rette prosjektene ettersom studentene allerede har 'rettet' hverandres prosjekter allerede.

17 Prosjektene burde vurderes basert på etterspurt funksjonalitet fra en som kan faget, ikke av andre studenter. Dette åpner også for at funksjonalitet som nesten er ferdigstilt vil gi uttelling. Studentene er i prosessen av å lære faget, og er ikke egnet for å gi tellende vurdering. Det er også problematisk at grupper kjenner hverandre og dermed får venner til å gi de en god vurdering.

18 Jeg synes fagfellelvurdering i seg selv kan være en god måte å evaluere på, i tillegg lærer man mye selv ved å se på andres kode. Ulempen var at det tok veldig mye tid og kunne være demotiverende. Faget opplevdes derfor som mye mer omfattende enn 7,5 poeng.

19 Studentvurdering fungerer *ikke* når det ikke har blitt klart fastsatt hvordan man burde vurdere et prosjekt. Dessuten, om man er helt kynisk her, så er den beste strategien for å yte best i faget (altså få høy karakter i et fag med lavt snitt) å gi laveste karakter til andre prosjekter uansett hva de har gjort. Dette er spesielt sant når man ikke innfører noen form for etterkontroll.

20 Det fungerer ikke så bra at studenter vurderer hverandre. Det blir for varierende vurderinger. Spesielt når vurderingskriteriene er dårlig eller tvetydig spesifisert. Det blir for mye misforståelser og feilvurderinger

21 Mitt problem med faget er at det er et 7.5 studiepoengs-fag som oppfører seg som om det er et 15 studiepoeng. Det krever for mye tid. Jeg kunne lett brukt flerfoldige timer på evalueringene, siden vi ble bedt om å evaluere så mange prosjekter ganske så grundig. Det var ikke nok timer i døgnet til å sette seg inn i prosjektene (hvordan man har løst oppgaven kan variere veldig, bruker fort en time på å sette seg inn i prosjektstruktur, hva ting er ment å gjøre, og så videre), og jeg prøvde til og med å spørre læringsassistentene hva jeg gjorde 'feil' siden jeg brukte å mye lenger tid enn det som var forventet. Fikk bare til svar at jeg ikke måtte gå så grundig til verks. Sånn som opplegget var så var det ikke tid til å gjøre noe annet enn en overflatevurdering, og hvis man synes tilbakemeldingene ikke var gode nok så er ikke løsningen å kreve mer tid av studentene, når faget allerede krevde arbeid langt over det som er rimelig. Jeg blir provosert av at 'løsningen' som foreslås er å vurdere vurderingene, og ikke en tilpasning av arbeidsmengden, som om studentene ikke brukte nok tid fra før.

22 Vurderingene tok ganske mye tid og det var lite betryggende å vite at noen brukte mer tid enn andre på vurderingen. I tillegg så var studentvurderingene med på å gi poengsummene i faget som var med å bestemt karakteren til studentene i faget. Det var ikke betryggende å vite at noen kunne brukt 2 minutt på å vurdere meg og at jeg dermed ikke får en representativ tilbakemelding. Det virket ikke som faglærer var med å vurderte studentene. I tillegg, hvis en gruppe brukte en teknologi jeg ikke kjente til så hadde jeg ikke verktøyene til å kunne vurdere dem i utgangspunktet.

23 Ikke noe spesielt. *Removed empty response.*

24 Studentvurderinger i IT2810 var fundamentalt svekket av at folk flest ikke kunne nok til å lage sitt eget prosjekt, og de kunne i hvert fall ikke nok til å vurdere andres prosjekter. Studentvurderingene kommer også til å være avhengig av kvaliteten til evalueringskriteriene. Når man har en stab av studasser kan man samle dem, klarifisere betydningene av kriteriene og bli kvitt mye subjektivitet. Når det er utallige studenter som håndterer det blir

det for mye rom til subjektiv tolkning. Et system for studentvurderinger (og vurderinger av studentvurderinger) må altså ha et robust system for å håndtere klagesaker, da det kommer til å være veldig mange av dem. Jeg vet egentlig ikke helt hvor mye nytte man hadde hatt av å vurdere andres vurderinger, da jeg vil si mange følte at det allerede ble brukt langt for mye tid på å vurdere hverandre i webdev. Å legge enda et lag på vurdering på det igjen vil trekke enda lengre vekk fra den faktiske læringen

A.2 Usability Test Stories and Scenarios

A.2.1 User Stories

This section contains some user stories which were used in the user test. A user story is a simple way of defining a requirement for how an end-user wants to use a system.

- As a student, I want to log in to the system.
- As a student, I want to view the assessments I have to do.
- As a student, I want to assess the assignments I have been asked to assess.
- As a student, I want to submit the assessment I have completed.
- As a student, I want to view the assessments others have done of my assignments.
- As a student, I want to rate the assessments others have done of my assignments.
- As a student, I want to view the rating others have given my assessments.
- As an instructor, I want to view the assessments of an assignment.
- As an instructor, I want to select which assessment (out of multiple) I want to view.
- As an instructor, I want to see the ‘outliers’ of an assessment process.
- As an instructor, I want to evaluate an assignment.

A.2.2 Test Scenarios

Logging in

You will be supplied with a username and a password for when you log in. When you are asked to select your “login provider”, select “Feide test users” on the right-hand side.

Scenario 1

Your role: Student

Setting: Nothing special.

Log-in credentials: asbjorn_elevg & 1qaz

Your goal: Log into the system and get an overview of what assessments you have coming up.

Scenario 2

Your role: Student

Setting: You have been asked to assess the project of one of your peers. Imagine that you have access to the source code for a web development project and use this as the basis for the assessment. For the questions you're asked, come up with some answers which fit the context.

Log-in credentials: asbjorn_elevg & 1qaz

Your goal: Log into the system and assess the assignment of a peer for the “Evaluering av kode prosjekt 3” assignment.

Scenario 3

Your role: Student

Setting: Imagine that you have completed a web development group project and that one of your peers have assessed your assignment.

Log-in credentials: bjorg_laererg & 2wsx

Your goal: Log into the system and rate the helpfulness and usefulness of the feedback you received for the “Evaluering av kode prosjekt 3” assignment.

Scenario 4

Your role: Student

Setting: You have been asked to assess the project of one of your peers. Imagine that you have access to the source code for a web development project and use this as the basis for the assessment. For the questions you're asked, come up with some answers which fit the context. However, you are a bit short on time; therefore, you fill out all the fields with some notes and keywords and plan to revisit the assessment at a later time.

Log-in credentials: cecilie_elevvgs & 3edc

Your goal: Log into the system and assess the assignment of a peer for the "Evaluering av kode prosjekt 3" assignment. After having filled out the notes, you log out and then back in, revisiting the system at a later time to finalize the assessment.

Scenario 5

Your role: Instructor

Setting: After having organized peer assessments for a course you are going to evaluate the assessments. You want to use the information from this overview to judge what the average scores and the levels of agreement across the assessments has been.

Log-in credentials: frank_foreleser & 6yhn

Your goal: You want to get an overview of the assessments done so far. Log into the system and get an overview of the submitted assessments.

Scenario 6

Your role: Instructor

Setting: After having organized peer assessments for a course you are going to evaluate the assessments. Some students reported one of the assessments as being inaccurate, and you want to evaluate the correctness of this assessment.

Log-in credentials: frank_foreleser & 6yhn

Your goal: Log into the system and evaluate the correctness of the assessment done by Cecilie for the "Evaluering av kode prosjekt 3" assignment.

A.3 Usability Test Consent Form

Samtykke om gjennomføring av brukertest

Navn på deltaker: _____

Deltaker har mottatt informasjon om studiet og brukertesten, og har hatt mulighet til å stille spørsmål ved eventuelle uklarheter. Deltaker er informert om at han eller hun når som helst kan avbryte testen ved et ønske om det, uten å oppgi noen grunn for avbrytelsen.

Data og informasjon innsamlet i denne brukertesten vil bli anonymisert slik at det ikke kan spores tilbake til testpersonen, og dataen vil kun bli brukt i sammenheng med den nevnte studien.

Jeg samtykker herved til ovenstående påstander og vilkår.

Trondheim, _____ (dato)

Underskrift deltaker: _____

Table A.6: Overview of scenarios tested

Test Candidate	Scenarios tested
Test Candidate 1	1, 2, 3
Test Candidate 2	4, 3, 1
Test Candidate 3	5, 6

A.4 Usability Test Transcript

A.4.1 Test 1

A student completed the first usability test with the role of a student for all the scenarios. Notes were taken during the test and transcribed for further studies. The following notes come from test number 1.

Scenario 1

- The front page of the system asks the user to log in, but clicking that text does not take you to a log-in page. Nothing happens. The user expected it to be possible to click the link, but rather clicked the log-in link in the navigation menu.
- The user was uncertain if he or she had found the correct page, and mentioned that the page showcasing assignments could show some more information, like for example a due date.

Scenario 2

- Saving gave no visual feedback until the user scrolled to the top, where some status messages had popped up.
- Unclear menu element names, not sure what to expect from each menu element.
- The rating slider was inconsistent between some questions where there were different rating scales. The user said it was mostly an inconvenience.
- The user suggested that the various elements on the assessment pages be grouped more together.

Scenario 3

- The user did not always rate all questions in the assessments, even though he or she chose to give feedback.
- The user spent some time understanding what grade he or she had been given.

-
- The same reaction regarding save as in Scenario 2, that there was no visual feedback of clicking the save button.

A.4.2 Test 2

A student completed the second usability test with the role of a student for all the scenarios. Notes were taken during the test and transcribed for further studies. The following notes come from test number 2.

Scenario 4

- The user clicked into the overview page. It turned out this was mostly to become familiar with the menu elements and how the system worked in an exploratory fashion.
- The user clicked the save button multiple times, expecting some visual feedback. When not receiving this, the user scrolled back to the top of the page and noticed the status messages.
- The user had forgotten to respond to one of the fields, so one of the status messages during the save operation was a message regarding a field which should not have been left blank. The message did not point to the offending field, but when the user read the message and scrolled back down through the rubrics, he noticed the empty field and corrected it.
- The user did not understand how to ‘finalize’ a delivery, since there was only a save button on the page.

Scenario 3

- The user did not always rate all questions in the assessments, even though he or she chose to give feedback.

Scenario 1

- The overview page was blank.
- The user found the correct page, but mentioned that it should show some more information about the assignments, like a due date.

A.4.3 Test 3

A student completed the second usability test with the role of an instructor for all the scenarios. Notes were taken during the test and transcribed for further studies. The following notes come from test number 3.

Scenario 5

- The user did not find the correct page at first. After some clicking around, the user received some input regarding which menu was the correct one, so that the test scenario could continue. The first page the user clicked was the overview-page, which was empty.
- The choice of which assignment to view was confusing, not really sure which assignment or user the system showed.
- The statistics regarding grades could have been separated further away from the assessment and feedback, it was quite close even though not having significant relevance.

Scenario 6

- The user clicked back to the assessment view, even though being an instructor and having the goal to evaluate.
- The menu elements were a bit confusing, not really sure where to click.
- The user did not fully understand how to select a specific assignment
- When presented with the various names, the user expected the system to work ‘the other way around’, that you watch the assignment that that person has done, not what that person has assessed to someone else
- The user wanted some sort of highlighting of relevant information so it was easy to spot, such as the grade given and if it was an outlier.
- The user suggested to show all grades instead of an average or mean if there were few grades, and in general requested a better user experience in the evaluation view.

A.5 Usability Test Findings

Scenario 1 findings

Two candidates tested Scenario 1, where the goal was to log in to the system and gain an overview of the upcoming assessments. The findings of this scenario should show how the students approach the system and navigate it to find information regarding their assignments and assessments.

The test candidates completed the scenario mostly with ease; however, two difficulties presented themselves: one regarding how to log in to the system and one regarding the overview itself.

One of the test candidates expected to be able to click the message asking the user to log in. The message read ‘Log in to the system to access your assignments’. When

clicking the message, nothing happened. The user noticed the menu on the left side of the system and clicked the log-in button in the menu instead and solved the task. Furthermore, one test candidate expected the 'Overview'-page to contain information about the assessments, but it seemed irrelevant for the time being. This page served no real purpose in the prototype, but it could be used to show upcoming assessments, received assessments, grades on assignments and assessments.

Both candidates mentioned that the overview should present more information, such as when the assignments are due. An LMS would most likely also present this information, but this test did not include an LMS. There would be no problem duplicating the information, and this is a simple fix for the presentation of the assignments.

Scenario 2 Findings

One candidate tested Scenario 2, where the goal was to log in to the system and submit a peer assessment. The findings of this scenario should show how the students approach the assessment components, such as the rubrics, as well as how they navigate to an assessment. Another candidate tested Scenario 4, which is similar to this scenario.

The test candidate navigated to the correct component after having clicked through multiple menu elements because he or she was unsure if it was the correct one. The candidate suggested finding more suitable names for the pages.

The candidate suggested to group the elements on the page together, so it is easier to see which components associated with which question.

Furthermore, the candidate found some inconsistencies in the summative rubric slider. The reason for this came from that one of the questions had its range set from 0–5, and the others had 1–5.

When the candidate saved the assessment, no visual feedback showed up. The user tried to save the assessment one more time before scrolling up through the rubrics. At the top of the page some status messages had appeared, indicating that the system had saved the assessment. These messages should appear on-screen at any point rather than at the top.

Scenario 3 Findings

Two candidates tested Scenario 3, where the goal was to log in to the system and submit a helpfulness rating of a received peer assessment. The findings of this scenario should show how the students approach the rating process with regards to rating the components available for rating the helpfulness.

Both candidates found the process similar to the previous one regarding assessing their peers. Even though they were instructed to rate the helpfulness, both candidates left out the rating for some of the feedback while still giving textual feedback on the assessment. If rating the assessment for every single question in the rubrics is wanted the system should set the rating as mandatory and disallow the user from not submitting the assessment if one rating is missing. However, if the fine-grained ratings are optional and only the overall

rating is obligatory, this can be left like it is. This decision should be up to the course to take, but the system can be more explicit regarding which questions are mandatory.

One of the candidates spent some time to understand what grade his or her peer had given the assessment. The display of the assessed grade showed quite close to the component for rating the helpfulness of the review, which might attribute to this finding.

Scenario 4 Findings

One candidate tested Scenario 4, where the goal was quite similar to the one in Scenario 2; namely, to submit a peer assessment. However, the goal of this scenario should also test the usability of the draft feature in the system, allowing students to save their current progress so they can come back to it later. The findings of this scenario should show how the students approach the feature of saving a draft of the assessment as well as submitting the assessment as a whole.

The setting in this scenario asked the user to fill out some notes first, before logging out of the system and come back to finalize the assessment at a later time.

The candidate testing this feature spent some time exploring the system before starting to assess their peer. After having taken notes of their peers' assignment, they saved the progress. The same problem as mentioned in the findings of Scenario 2 appeared here; namely, the lack of visual feedback when saving the assessment. Furthermore, the user had forgotten to fill out one of the fields, and the error message regarding this did not tell which field it was. The user scrolled through the questions and found the offending field and filled it out and saved their progress once more.

After realizing that the status messages were on the top of the page, the candidate logged out of the system, following the instructions. When logging back in to complete the assessment, the candidate was unsure how to 'finalize' the delivery, since there was no other way to save the assessment. The user ended up saving the assessment and explained that he or she figured the system would use the final saved copy when the deadline passed.

Scenario 5 Findings

One candidate tested Scenario 5, where the goal was to gain an overview of the submitted assessments. The findings of this scenario should show how an instructor approaches gaining an overview of the submitted assessments in a course they manage.

The candidate did not find the correct page at first. After having clicked around and expressed that they did not know what to do, the candidate received input regarding what page to visit so that the test could continue. The user explained that he or she expected the overview-page to show an overview of the assessments, but it was empty.

At first, the candidate found the display of assessments confusing, not being sure what was on the screen. After looking around a bit, the candidate understood what was going on. Not having designed the rubrics themselves could attribute to some of this confusion, but the user experience has the potential for improvement nonetheless.

The candidate spent some time looking through the various assessments to get the overview they wanted, but when they understood how the components connected and what

information was displayed, they were able to gain an overview of the conducted assessments.

Scenario 6 Findings

One candidate tested Scenario 6, where the goal was to evaluate one assignment. The findings of this scenario should show how an instructor approaches the evaluation process.

The candidate did not find the correct page at first and spent some time getting to the evaluation view. The candidate never selected an assessment but looked at the overview of the assignment as a whole. He or she asked for help regarding what to do after being visibly confused and received some tips regarding how to select an assessment.

With the knowledge of how to view individual assessments, the test candidate was able to gain enough information to evaluate the assignment. They were a bit confused at first at how selecting individual assignments worked, but by experimenting, they found out. The candidate suggested that the information relevant to the selected assessment could be separated a bit from the information regarding all the assessments for this assignment. Furthermore, he or she suggested that the system could display all grades instead of an average grade if there were a low number of assessments per assignment.