



**university of  
 groningen**

**faculty of science  
 and engineering**

**Managing capstone projects through  
 a web application while providing  
 a personalized learning experience**

Alex Gabor



**university of  
groningen**

**faculty of science  
and engineering**

**University of Groningen**

**Managing capstone projects through a web application  
while providing a personalized learning experience**

**Bachelor's Thesis**

To fulfill the requirements for the degree of  
Bachelor of Science in Computing Science  
at University of Groningen under the supervision of  
Prof. Fadi F. Mohsen, PhD  
and  
Prof. Dimka Karastoyanova

**Alex Gabor (s3787664)**

April 20, 2024

# Contents

	<b>Page</b>
<b>Abstract</b>	<b>5</b>
<b>1 Introduction</b>	<b>6</b>
<b>2 Background Information</b>	<b>7</b>
2.1 Capstone projects . . . . .	7
2.1.1 Product . . . . .	7
2.1.2 Process . . . . .	7
2.1.3 Progression . . . . .	8
2.1.4 Student-staff relationships . . . . .	8
2.2 Summative assessment . . . . .	8
2.3 Formative assessment . . . . .	9
<b>3 Challenges in teaching Capstone Projects</b>	<b>10</b>
3.1 Integration of knowledge . . . . .	10
3.2 Project management . . . . .	10
3.3 Assessment and evaluation . . . . .	11
3.4 Student motivation and engagement . . . . .	11
<b>4 Implementation</b>	<b>12</b>
4.1 Previous work . . . . .	12
4.1.1 Admin user . . . . .	12
4.1.2 Supervisor user . . . . .	12
4.1.3 Student user . . . . .	12
4.2 Project Planning and Design . . . . .	13
4.2.1 Student user . . . . .	13
4.2.2 Supervisor user . . . . .	14
4.2.3 Admin user . . . . .	15
4.3 Section: Feedbacks . . . . .	16
4.3.1 Mid evaluation . . . . .	17
4.3.2 Final evaluation . . . . .	17
4.4 Section: Sprint . . . . .	18
4.5 Section: Projects . . . . .	19
4.5.1 Grading a project . . . . .	20
4.6 User actions . . . . .	22
4.6.1 Admin user . . . . .	22
4.6.2 Supervisor user . . . . .	25
4.6.3 Student user . . . . .	28
4.7 Back-end . . . . .	29
4.7.1 Controllers . . . . .	29
4.7.2 Application Programming Interfaces (API's) . . . . .	29
4.7.3 Dependency Injection . . . . .	29
4.8 Front-end . . . . .	31

<b>5</b>	<b>Results</b>	<b>33</b>
5.1	Analyzing feedback statistics . . . . .	33
5.2	Analyzing project statistics . . . . .	34
<b>6</b>	<b>Limitations</b>	<b>36</b>
6.1	Measurement Instruments and Scope of the Study . . . . .	36
6.2	Data collection . . . . .	36
6.3	Subjectivity . . . . .	36
6.4	External Influences . . . . .	36
<b>7</b>	<b>Future work</b>	<b>37</b>
<b>8</b>	<b>Conclusion</b>	<b>38</b>
	<b>Bibliography</b>	<b>39</b>
	<b>Appendices</b>	<b>40</b>
A	Login . . . . .	40

## Abstract

This bachelor thesis presents an implementation of a web platform that sustains formative assessment on Bachelor projects, also referred to as "capstone projects". While wanting to create a good and efficient learning environment that could potentially be of future use, we have researched and found solutions that lead to creating multiple ways of giving feedback, at the user's convenience.

Doing so, we are planning to build an improved relationship between students and supervisors during the project implementation. While enhancing the learning process for students, we hope to improve the overall results and efficiency of capstone projects. The techniques that we use reflect an implementation of an agile environment. While the students benefit from receiving frequent feedback, the supervisors could also receive valuable feedback for the possibility of further improving their practices. Also, we are lead to believe that this approach will help students to easier integrate into a future workplace.

# 1 Introduction

Supervising capstone projects is presenting itself with many challenges [1]. In modern university learning process, one's progress is moderately evaluated at the end of a certain assignment. This raises some concerns not only from a student's, but also from a supervisor's point of view. Evaluating a project only based on the outcome can sometimes create deadlocks so that is why a good relationship between students and supervisors is very important [2]. The deadlocks can occur either on a psychological side, where students do not feel the need of finishing a project due to lack of motivation, or on an environmental side, where a learning system can be outdated. We believe that reflecting the learning progress into the process of determining the final evaluation would be of great use. Creating milestones of a learning progress and evaluating them would also give both students more motivation and supervisors a bigger picture of how well the student is adapting to each task.

At the University of Groningen, the main website (Brightspace) focuses on covering all the courses with the same (or slightly changed) database model. A feedback can be given through the platform only by receiving a grade, which in our situation, it does not help. Since each capstone project is different from one another, each student would have different milestones to receive feedback for, that do not necessarily include a grade. The grading section of each course on Brightspace contains the same assignments/grading parts for each student, they are not customizable for each project in particular. Our primary concept revolves around two personalized feedback sections, providing users with the freedom to select their preferred method of giving or receiving feedback.

In our approach, we want to make sure that students have a learning experience that closely resembles an agile environment that provides what they might encounter in a future workplace. We hope to provide them with the opportunity of achieving practical skills and knowledge that will be valuable in their careers. To achieve this, we emphasize the importance of regular and diverse feedback between students and supervisors: formative assessment. While it might be simpler to stick to occasional email exchanges, we believe that gathering data on the effectiveness of feedback is crucial. This allows us to better understand how well the feedback is being given and how students are performing. By offering multiple personalized feedback sections, we give students and supervisors the power to debate and choose the feedback methods that suit them best.

An agile environment keeps both sides close to a project, since when a student is updating a task frequently, then the supervisors know the progress and can interfere if needed. The duration of the capstone project is estimated to 420 hours and they need to be accountable: each task that is added to a project has hour estimations so, as a supervisor, you are always kept up to date with the progression of a project.

## 2 Background Information

There are different techniques of assessment that can be used when evaluating one's progress. In modern university learning process, there are two main types of assessment: summative and formative.

While there is usually a relationship between the two while teaching or studying, there is a tendency in the higher university to not fully understand the methods that are used for a proper formative assessment. Although there is no formula for creating an excellent teaching environment, there are some ways in which the learning and teaching process can be improved.

In this section, we will discuss the main information that a capstone project course experience involves.

### 2.1 Capstone projects

In a capstone project, the student is supposed to combine their accumulated knowledge and skills from the past university years. It is an opportunity to show maturity towards "team work, communication, client liaison, role playing, ethics, peer assessment and understanding the impact of the integration of their project on the wider community" [3]. The three P's of a capstone project: "Product, Process and Progression" [4] are identified to be the major grade factors when it comes to a capstone project:

#### 2.1.1 Product

The Product of a capstone project is identified as a combination between the expectations, the outcomes and the scope of a project. This is recommended to be evaluated objectively.

An objective evaluation can be sometimes not fair for a group evaluation. In a study that is mentioned in [3], 79% of students feel that there is a need for individual evaluation on group projects and 94% wanted to identify "free-riders", which are described as the students that fail to positively contribute within that group.

#### 2.1.2 Process

The Process of a capstone project is identified as a combination between the soft skills of a student which are usually presented as good communication, quality management, problem solving and more. Those skills become crucial in determining one's project execution and tend to reflect the student's readiness for real-world challenges.

It is highly recommended to adopt an objective assessment approach to ensure fairness and accuracy in evaluating soft skills. It is easier to identify areas that need improvement when keeping track of one's soft skills improvement while evaluating the progress of a capstone project.

### 2.1.3 Progression

The Progression of a capstone project is identified with the actions of taking initiative, professionalism and responsibility. This is recommended to be evaluated subjectively.

On another note, since we have stated above that a capstone project is a way in which a student prepares for a future workplace, we strongly believe the skills required for a good Progression of a project are also the skills that an employer looks for the most in employees. The key factor of a good Product is the Progression.

### 2.1.4 Student-staff relationships

The primary objective of the study [2] was to demonstrate the significance of student-staff relationships during the preparation of bachelor projects. To gather relevant data, a questionnaire was administered to a total of 14 students and 14 supervisors. The questionnaire aimed to obtain information regarding the importance of the relationships between students and supervisors at various stages of the bachelor project.

A subtheme of the research was represented in questions about the most important qualities of a supervisor. The qualities that are pointed out, in order, are shown below:

1. "knowledge of the relevant subject literature" (15 responses)
2. "supervisors should display a strong sense of realism (born out of supervisory experience)" (14 responses)
3. "responsiveness, receptivity and an ability to provide useful feedback" (13 responses)
4. "the importance of encouragement, nurturing and support, in combination with an ability to be 'firm when necessary'" (11 responses)

Those qualities were chosen by both students and supervisors in order to evaluate both impressions. A student/supervisor could choose more than one response.

## 2.2 Summative assessment

An evaluation tends to be summative when one's progress during a certain course or assignment is evaluated at the end of it. While there is a need to rely on this type of assessment, in order to improve it's final result it needs to be combined with other techniques, one of which being descriptive feedback "while there is still time to take action" [5].



## 2.3 Formative assessment

Formative assessment is a systematic process to continuously gather evidence about learning [6]. It can be defined and divided into four "core elements":

1. "identifying the 'gap'"
2. feedback
3. student involvement
4. learning progressions

Identifying the gap refers to understanding the leap that a student has to make in order to be successful in finishing a certain task. This leap can include both soft and hard skills that a student has to add to their compartment and/or implementation capabilities.

Student involvement refers to the actions that a student has to make on their own in order to actively be engaged in a project. The more a student wants to understand what skills they need to improve in order to finish a project, the more successful they are. Every student's action should begin with a perspective of "wanting", followed by the corresponding action.

Formative assessment must be linked to a learning progression in order to be of value. A learning progression is defined by describing and understanding the "subgoals" that a student has to achieve in order for a project to be successful.

Feedback is usually a tool from which the supervisors get a deeper view on the "level of understanding" a student has regarding a project and it leads to identifying what are the next steps in learning, in order to be able to finish a project in time. Moreover, following [5], those are a few examples of a good descriptive feedback:

- "You have interpreted the bars on this graph correctly, but you need to make sure the marks on the x and y axes are placed at equal intervals."
- "What you have written is a hypothesis because it is a proposed explanation. You can improve it by writing it as an "if ... then ..." statement. "
- "The good stories we have been reading have a beginning, a middle, and an end. I see that your story has a beginning and a middle, just like those good stories do. Can you draw and write an ending? "
- "You have described the similarities between ... and ... clearly in this paper, and you have identified key differences. Work on illustrating those differences with concrete examples from the text."

### 3 Challenges in teaching Capstone Projects

Teaching capstone projects presents numerous challenges [1]. In the present-day academic landscape, there are multiple approaches to concluding one's university studies. Certain approaches remain traditional, whereas others strive to enhance and innovate their teaching methodologies.

This section will provide information about the key factors that are required or expected to have learnt at the end or during a capstone project course. We will discuss areas like "Project management" where we will present a set of skills that a future successful student should have, and also "Assessment and evaluation", where we present a mid-evaluation study on capstone projects, that acquire valuable information about the state of a project and a student's progress.

Our study focuses also on finding areas of improvement in the way a Capstone Project is being taught. To provide more information about the subject, we included studies in which some applications of formative assessment evaluation were successful (e.g. "Assessment and evaluation" - presented above).

#### 3.1 Integration of knowledge

Capstone projects offer a unique opportunity for students: the integration of theoretical knowledge with practical implementation. Whether the approach of teaching capstone projects is well defined in terms of deliverables or not, it is clear that the process of it serves as a foundation for students preparing to enter the work environment [3].

#### 3.2 Project management

During a capstone project, a student demonstrates their project management skills and abilities. Those skills can and need to be also learned during the time that a student takes this course, even though the foundations of management skills, in most of the cases, are present. In a study that is presented in [7] we have found out that in order for a student to be prepared for the work environment, those are the top six important skills that lead to a good candidate:

1. "Honesty/integrity"
2. "Attitude"
3. "Willingness to learn new skills"
4. "Communication skills (oral and written)"
5. "Analytical skills"
6. "Professionalism"

The above skills are represented as soft skills [7], which are usually hard to interpret. Many definitions are "general and inconsistent". Even though it is hard to interpret them, the evaluation of them plays an important role in defining one's ability to make progress. The presented features are also included in the platform when giving feedback. (can be seen in section 4.4)

Those soft skills can provide information that a summative evaluation of the end product can not. The attitude of a student most of the times reflects their ability to make good progress towards a goal.

No one has got all the skills mastered, so we are lead to believe that pointing out the above presented skills while giving feedback, and also having a balance between negative and positive remarks, would improve the student's ability to grow educationally.

### 3.3 Assessment and evaluation

In a traditional way of teaching capstone projects, there are always ways of improvement when it comes to formative assessment. As it should, the main idea is always focused around feedback and response. In an article wrote by T. Clear, we have found out that a method of a mid-evaluation process, as a form of formative assessment, is helpful in many ways [8]. It is based on giving feedback on the three P's: "Product, Process and Progression" [4](as presented in the section 2.1) in order to find whether the project has potential or not. It also contributes when a project is made by a team of students, it helps in understanding how much each student is contributing to the project.

The main ideas and results of the mid-assessment are:

- "this simple technique enables a conscious and open discussion to take place about the students' learning"
- "Feedback can be given on their evidence of maturity, engagement, professionalism "
- As a conclusion: "the project may continue (with or without conditions); it may be cancelled; or the panel may require that the team and supervisor confer with the client prior to deciding the future of the project."

### 3.4 Student motivation and engagement

Following the previous section, "Assessment and evaluation", we have also found out in [8] that there are cases when a student, even though he manages to make good progress, he does not manage to show his progress to the full potential. Following a formative assessment evaluation, there are questions that have to be asked when showing a portfolio of work already done, in order to find out if a student just has a bad portfolio, even though he made good progress:

- "It exposes team members unable to evidence any noticeable contribution to the project. That may simply be a result of a poorly evidenced portfolio or may be a realistic indicator of limited contribution to the team's efforts."

## 4 Implementation

The concept behind the project is to initiate the implementation of a user-friendly experience environment where both students and staff actively engage in providing ongoing feedback to each other.

In order to tackle the challenges known in teaching capstone projects, the end product will try to improve the teaching methodologies by providing ways of formative assessment evaluation. By doing this, we not only try to improve the student-staff relationships, but the overall engagement into the capstone project courses.

We are lead to believe that while the focus is on improving the engagement of both student and supervisors, we will achieve better results, not only related to the final grade, but on the overall experience that both parties have. While the student is improving his skills based on the feedback, the supervisors could potentially improve their teaching ways also.

### 4.1 Previous work

We have started this bachelor project on top of an application that was made for the course "Short programming project". In this section, we will provide information for the application flow that was already implemented.

#### 4.1.1 Admin user

The main role of an admin user is to manage the users that are allowed to register through the platform. That means, at the server level of the application, there is a verification of the input email when registering as any user. When registering, if the provided e-mail was not added to the allowed user's to register database table, there is an error that is thrown.

#### 4.1.2 Supervisor user

In terms of the supervisor user flow, the main focus lies in initiating project instances. Once these instances are created, the next step involves students applying for these projects and after that, review the applications and make the decision of accepting or rejecting them.

#### 4.1.3 Student user

As a student user, you are interested in starting your bachelor thesis. After registering, the student users are able to browse through the projects that are listed by the supervisors and then apply for them by providing a compatibility description.

## 4.2 Project Planning and Design

Starting from the initial concept of developing a project management platform, we have shifted our focus towards researching and identifying solutions aimed at enhancing both student and staff performance.

Our primary focus lies in providing users with the freedom to choose their preferred approach when engaging in the feedback sections. After careful consideration, we have developed an idea that brings two separate sections into play, "Feedbacks" and "Sprint", each with its own specific purpose of facilitating the giving and receiving of feedback.

Our second focus lies upon giving an overall better and faster experience for all users while using the platform. In order to achieve it, we have come up with solutions that will be discussed in the next subsections: 4.2.1, 4.2.2 and 4.2.3 in which we will give a brief summary of actions that improve the experience of the users.

### 4.2.1 Student user

The action of filtering the projects while browsing them. (this includes opening and hiding the filters, resetting them, choosing the field of a project)



Figure 1: opened vs. closed filters

In (1)a we can see the list of projects, with the opened filters in which if the user writes a string in any or more of the filter choices, you can apply them and the projects will be filtered by that.

For easy use, we also added a button of clearing the filters in which if you apply it, it resets all the previous applied filters (and also the project field - which can be seen below the buttons of apply and clear filters in figure 1(a) or next to the button of open filters in figure 1(b)).

### 4.2.2 Supervisor user

While grading a project, you can either upload the according ".xlsx" file for grading a project and it automatically fills in the required fields in the platform, or you can use the platform for grading it and then after it is graded, you can download the according ".xlsx" type file that is already filled in.

Title of Bachelor's thesis	
First supervisor	
Second supervisor	
Final grade for thesis (weighted average grade from categories A-D)*	In
* The student can only pass the assessment if all subgrades of categories A-D are graded with at least 5.5 and the weighted average is at least 6.	
* If one or more of the subgrades (category A-D) are lower than 5.5, the student is offered a remediation trajectory (see Part 3) after which a second assessment takes place.	
* The weights of the subgrades of categories are A 40%, B 20%, C 20%, D 20%..	
Final grade for thesis after remediation	
Thesis available / present in repository	<input type="checkbox"/> k
	Unsat. Satisf. Very Good Outstanding N/A
	(<5.5) (6) (7) (8) (9) (10)
<b>1. Scientific quality of Research and technical contribution</b>	<b>40%</b>
Problem analysis	
Literature research	
Research strategy and methods/techniques used	
Quality, validity and relevance of results	
Analysis and interpretation of results	
Discussion of results	
Conclusions	

(a) Grading sheet

Date						
Title of Bachelor's thesis						
First supervisor						
Second supervisor						
	Unsatisfactory (< 5.5)	Satisfactory (6)	Very satisfactory (7)	Good (8)	Very Good (9)	Outstanding (10) N/A
Problem analysis						
Literature research						
Research strategy and methods/techniques used						
Quality, validity and relevance of results						
Analysis and interpretation of results						
Discussion of results						
Conclusions						

(b) Platform grading sheet

Figure 2: grading sheet vs. platform grading sheet

In the next picture, we can see the start of "grading a project" action in which if a user chooses to upload the excel, in the format from the figure 2(a), it automatically fills in the according fields in the platform.

If you want to fill in the sheet, please add the final assessment form (.xlsx file)  
 \*PLEASE NOTE: the grade will not be saved without submitting it\*

Choose File No file chosen

Upload file

Student number

Student name

Date

Title of Bachelor's thesis

First supervisor

Second supervisor  
pres@staff.com

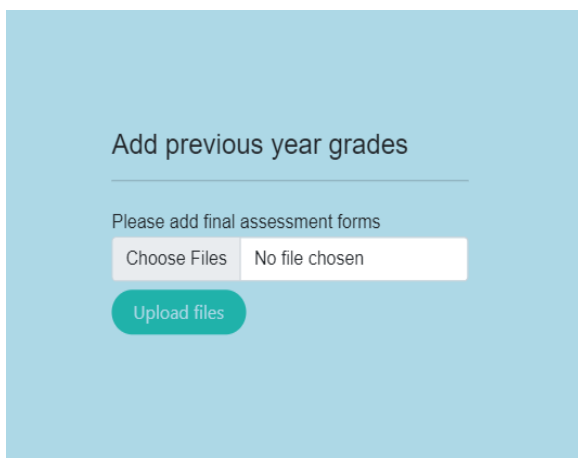
	Unsatisfactory (< 5.5)	Satisfactory (6)	Very satisfactory (7)	Good (8)	Very Good (9)	Outstanding (10)	N/A
Problem analysis							
Literature research							
Research strategy and methods/techniques used							
Quality, validity and relevance of results							
Analysis and							

### 4.2.3 Admin user

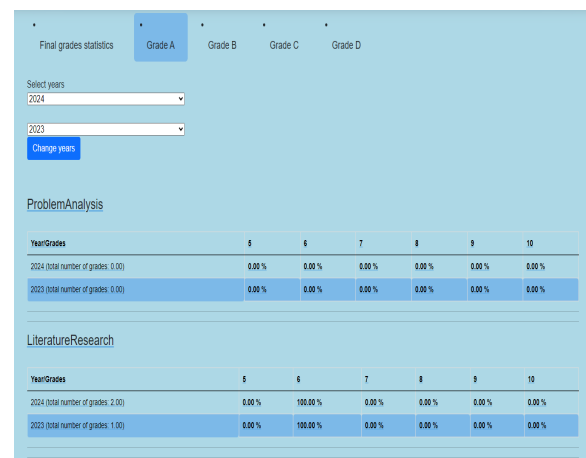
While browsing the projects, an admin user can download the associated grade sheet of a certain project (in excel format).



Also, an admin user can upload multiple already graded sheets (in excel format) to the platform to improve the statistics use in order to see if the grades improve over the years. The year and all associated grades are automatically added to the database.



(a) add final assessment forms



(b) choose years to compare

Figure 3: grade statistics

While an admin is choosing the years to compare (example: figure 3(b)) the page of grades is updated according to that year, comparing the results based on every subgrade of each category. (number of grades in each subgrade and percentages of grade ranges from insufficient to 10)

### 4.3 Section: Feedbacks

This section includes feedback that is associated with a project, a sender, a recipient, and a purpose. The image below shows how the form of creating a feedback looks like for a supervisor (the student page does not contain the right side information):

Give feedback

Title

Description

Project title  
<Choose a project>

Create feedback

When evaluating, please be sure to include values of:

- Honesty/integrity**
- Attitude**
- Communication skills (oral and written)**
- Analytical skills**
- Professionalism**

Examples of good descriptive feedbacks:

"You have interpreted the bars on this graph correctly, but you need to make sure the marks on the x and y axes are placed at equal intervals."

"What you have written is a hypothesis because it is a proposed explanation. You can improve it by writing it as an "if ... then ..." statement."

"The good stories we have been reading have a beginning, a middle, and an end. I see that your story has a beginning and a middle, just like those good stories do. Can you draw and write an ending?"

"You have described the similarities between ... and ... clearly in this paper, and you have identified key differences. Work on illustrating those differences with concrete examples from the text."

Back to feedbacks

Due to the various purposes of giving feedback, a user has the flexibility to provide it at their convenience, but it is also important to negotiate the terms with the supervisor/s: establish a certain frequency of the feedbacks, from both sides.

A feedback contains information like:

- Title
- Description
- Last time updated / created at
- Feedback given to
- Feedback created by
- Project title
- Grade (given by student/supervisor)
- Grade (given by admin user) - on the already given feedback, that belongs to a graded project.

To add more interaction between users, we have decided to also implement two feedback sections that act like a mid project evaluation and a final supervisor feedback. Those will be discussed in the next two subsections.



### 4.3.1 Mid evaluation

Based on the study [8], we have included the three P's into the mid project evaluation: "Product, Process and Progression". The mid evaluation section can only be created by a supervisor and be targeted to a student that is assigned to one of the supervisor's projects. The student can then view their results.

(a) Supervisor: give mid evaluation

(b) Student: see mid evaluation

Figure 4: create vs. view mid evaluation

### 4.3.2 Final evaluation

The final supervisor feedback is based on the values that are discussed in [2], also summarized at section 2.1.4. In the next figure, we will present how the action of creating a final evaluation and viewing it looks like:

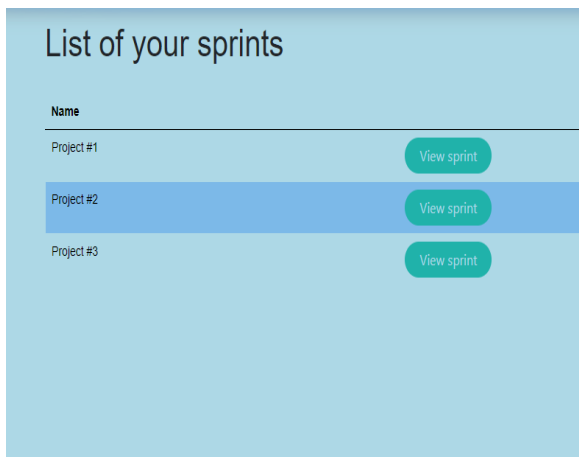
(a) Student: give final evaluation

(b) Supervisor: see final evaluation

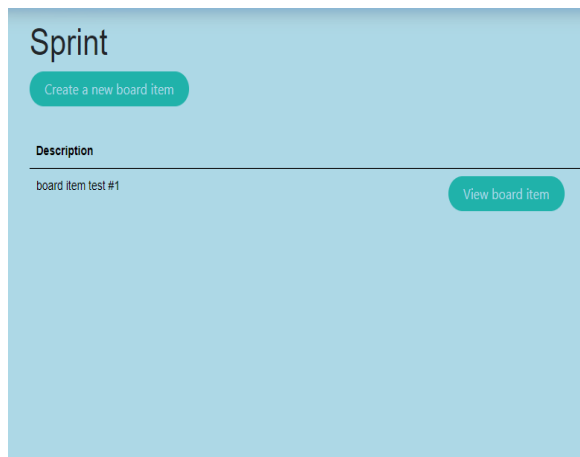
Figure 5: create vs. view final evaluation

## 4.4 Section: Sprint

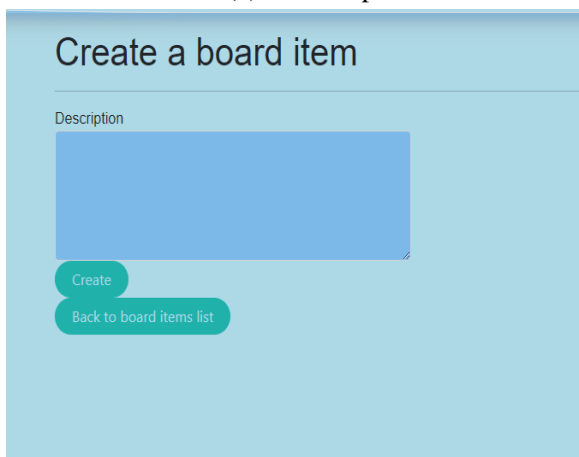
Since a capstone project prepares a student for the work environment, we have chosen to add a feature that may be used (although it is not necessarily) in order to keep track of the progress and give/receive feedback. It works as follows: when a project is created, the section of the sprint of that project is automatically available. Also, when a student is accepted to be part of that project, he gains access to this so-called "Sprint" section.



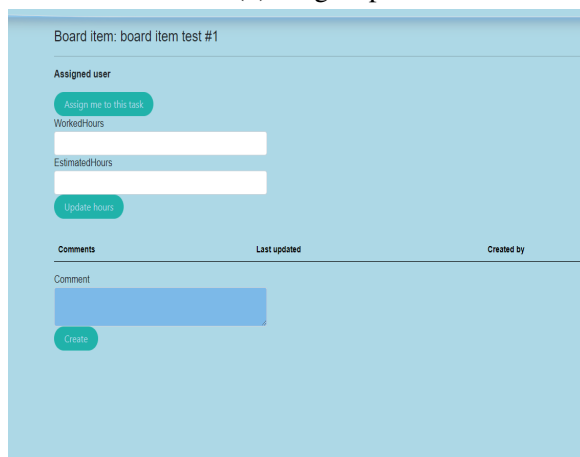
(a) List of sprints



(b) Single sprint



(c) Create a board item



(d) Board item view

Figure 6: Sprint pages

Each sprint is empty at first, but any user can create board items that will be linked to it. To perform this action, a user has to go to the sprint section, choose a project to see its associated sprint (figure 6(a) by pressing the button "View sprint") and create a board item with a description that can be seen in the list that is provided in the figure 6(b).

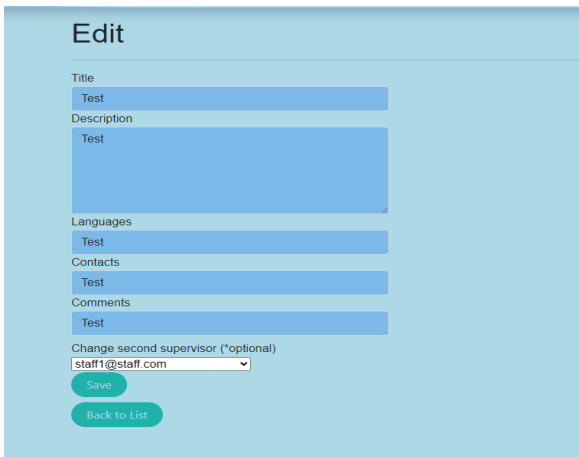
Each board acts like a milestone needed in order to finish a certain project. Any user can add new board items and also update them with information like:

- Description
- Assignees
- Hours already worked on it
- Estimated total hours
- Comments

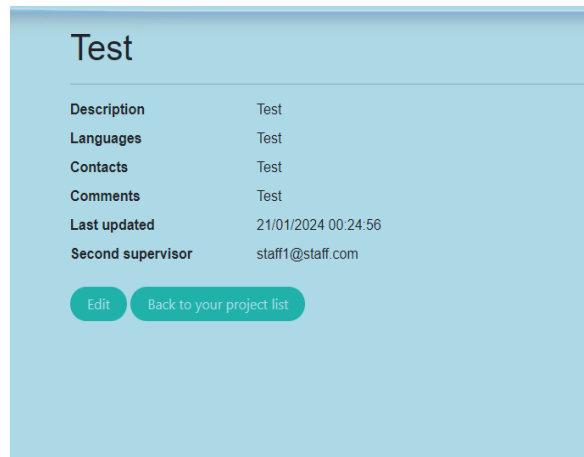
The comments section acts like a feedback section on what is already done for the milestone. It can also be a simple conversation between students and supervisors in order to help determine the state of the task. Each comment it's ordered by date, the first comment will always be the last one that has been sent, and provides information like: who created it and the actual comment, which is a description of what an user would like to say about this task until now.

### 4.5 Section: Projects

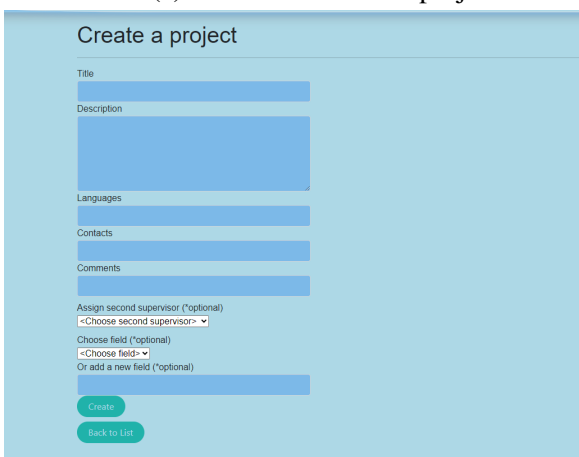
Firstly, we will provide a figure with all the actions that the project page includes:



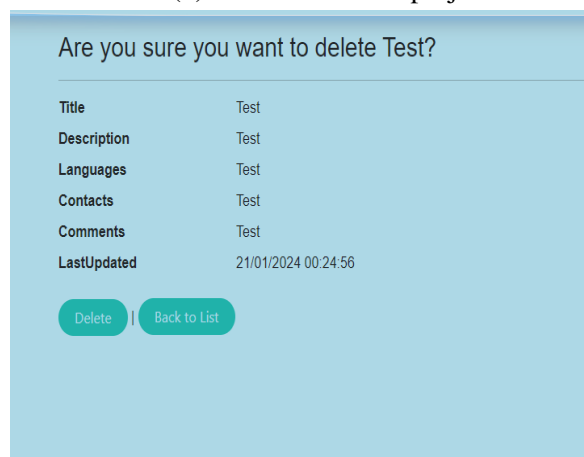
(a) edit the details of a project



(b) view details of a project



(c) create a project



(d) delete a project

Figure 7: Project pages

To provide a better experience while using this feature, we have implemented a possibility to filter the projects by: Contacts, Languages, Comments, Description, Title or Second Supervisor which are present in the base model of a project. Moreover, each project is linked to a field. If none is chosen while creating it, then the project will only be present to the field "All Projects". If a project is in another field, it will be present in this particular field and the "All projects" field.

While the supervisors can see only their projects (including the ones that are assigned to as second supervisor), the students can see all projects that are not graded (this view can be seen in figure 1 from section 4.2.1).

While editing a project, as a supervisor user, you can edit each mapped element of it (figure 7(a) from this section). The student can only see the details of a project (figure 7(b) also from this section), while the supervisor can also edit them. All the actions that include the section "Projects", and not only, will be discussed based on users in the next section (4.6).

In the next subsection of "Projects" (4.5.1), we will discuss the action of grading a project, the model of a grade and how the supervisor user can interact with the platform.

#### 4.5.1 Grading a project

In this section, we will present all the available grade types and functionality of grading a project.

First, let us present the model of a grade: (basic information)

- Student Number
- Student Name
- Date
- Title Of Bachelor
- First Supervisor
- Second Supervisor
- Year
- Compact Summary
- First Assessor
- First Assessor Date
- Second Assessor
- Second Assessor Date
- Final Grade
- Final Justification

Then, the first subgrade (Grade A) is formed by:

- Problem Analysis
- Literature Research
- Research Strategy
- Quality Validity
- Analisys Interpretation
- Discussion Of Results
- Conclusions
- Justification

Also, the second subgrade (Grade B) is formed by:

- Independence
- Initiative
- Motivation
- Focus
- Planning And Organization
- Collaboration
- Meeting Deadlines
- Communication Skills
- Adversities and Setbacks
- Justification

Then, the third subgrade (Grade C) is formed by:

- Contents
- Structure
- Clarity
- Presentation Skills
- Response To Questions
- Justification

Last, but not least, the fourth grade (Grade D) is formed by:

- Structure And Layout
- Problem Description
- Literature Study
- Materials Methods
- Results
- Discussion Of Results
- Conclusions
- Writing Skills
- Justification

This grading method is a format in which all Capstone Projects are graded at the University of Groningen.

Each subgrade (A,B,C or D) contains also a final grade for that subsection which is the sum of all graded elements divided by the number of graded elements (not every element has to be graded, it also has an option of Not Applicable). The final grade of all subgrades is the sum of them divided by 4.

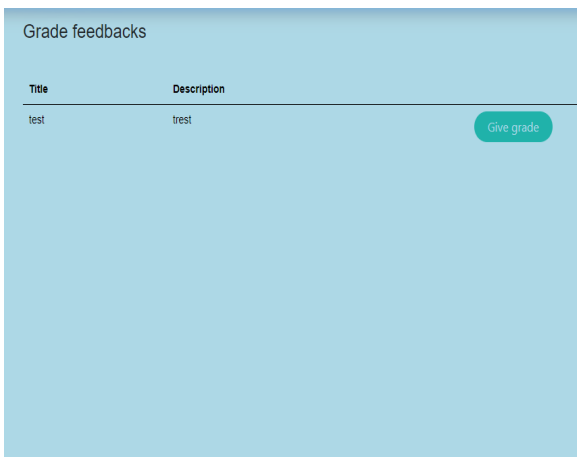
Also, each subgrade has a Justification which is a written description of why the grades were given like so in a certain section.

## 4.6 User actions

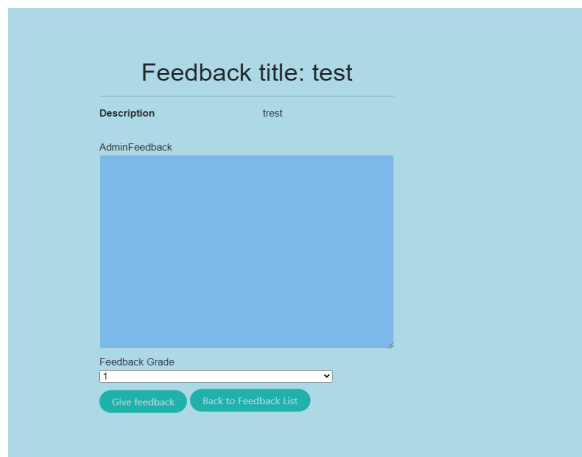
In this section, we will list all the actions that each user can perform based on his account type. On top of the previous discussed user flows, we have added some features that help improving the overall user experience and also create statistics that could potentially show capstone projects improvements over the years.

### 4.6.1 Admin user

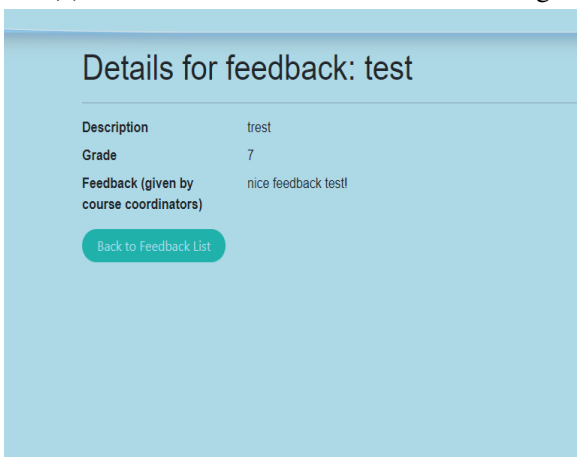
We have added, on top of previous discussed actions of adding allowed users to register, the actions of: grading a feedback of a closed project (when a project becomes graded by supervisor) and the ability to see and upload certain data compared between projects and academic years. (more to be discussed in section 5. "Results")



(a) Admin: List of available feedbacks to grade



(b) Admin: Give a feedback

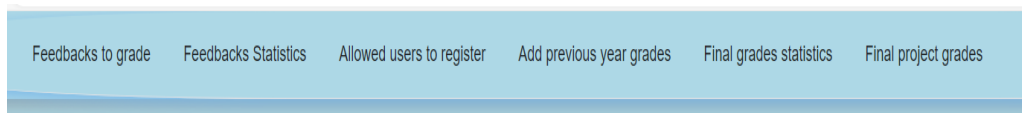


(c) Supervisor: view given feedback

Figure 8: Admin feedback pages and actions

1. Add, update and delete allowed users.
2. Add multiple allowed users by ".xlsx" type files.
3. View allowed student and supervisor users.
4. View statistics of feedbacks/graded projects.
5. Add multiple previous years grades by ".xlsx" type files. This can be done in order to create statistics based on project grades and years.
6. Grade feedbacks (of graded projects).
7. View statistics based on years on graded capstone projects.
8. Create customisable 2-year period of graded capstone projects which you want to compare in data.
9. Moreover, this customisable 2-year period can be related to 4 different type of grades (Final grade, Grade A, Grade B, Grade C and Grade D - which are the standard grading instructions that are used in our University).
10. View and download ".xlsx" grade type files of already graded projects.

Now, let us present all admin available pages:



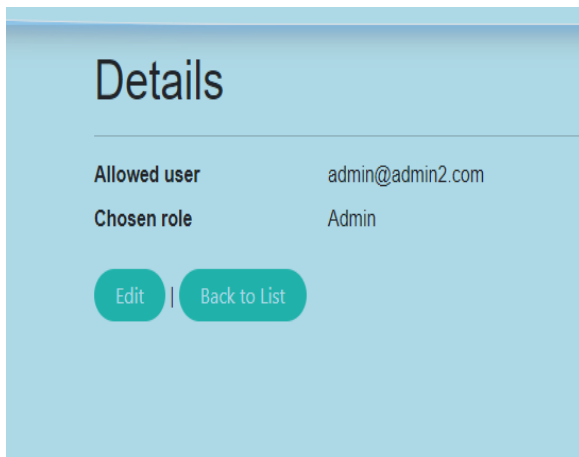
1. The "Feedbacks to grade" button opens a page in which you can see all the feedbacks that are available for grading (of already graded projects) - can be seen in figure 8(a) from this section.
2. The "Feedback statistics" button opens a page in which you can see the statistics of the given feedbacks of already graded projects. (More to be discussed in section 5. Results)
3. The "Allowed users to register" button opens a page in which you can see a list of the allowed users and also update it by simply adding a new user or an excel file that contains one or more users. Next, we will present two figures in which all the actions of this page can be seen.

Allowed user	Role given	Last updated	
admin@admin2.com	Admin		Edit Details Delete
supervisor@presentation.com	Staff	11/07/2023 12:00:24	Edit Details Delete
presentation@staff.com	Staff	11/07/2023 12:28:14	Edit Details Delete
pres@staff.com	Staff	12/07/2023 09:32:06	Edit Details Delete
staff@ing.nl	Staff	12/07/2023 10:32:53	Edit Details Delete
staff@pres-test	Staff	24/07/2023 21:32:46	Edit Details Delete
student@presentation.com	Student	11/07/2023 12:00:20	Edit Details Delete
presentation@student.com	Student	11/07/2023 12:28:00	Edit Details Delete
pres@student.com	Student	12/07/2023 09:32:21	Edit Details Delete
staff@pres-test5	Student	26/09/2023 22:11:33	Edit Details Delete

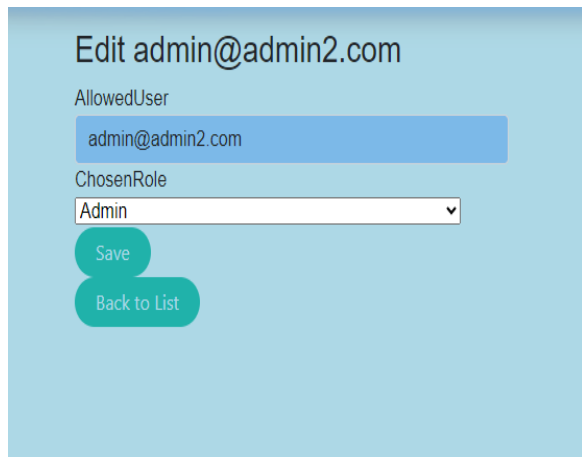
(a) List of allowed users to register

(b) Add single or multiple users

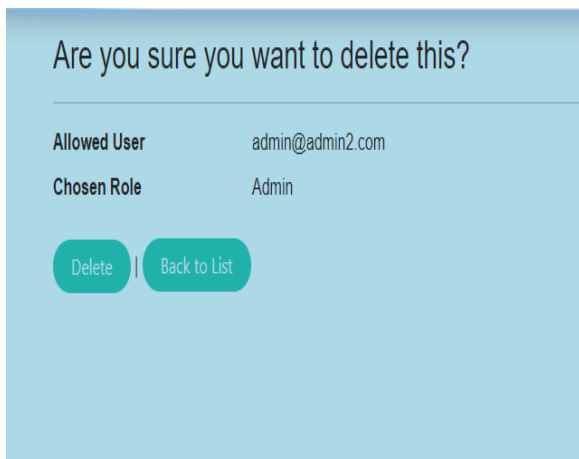
Figure 9: list of allowed users vs. add users action



(a) Details of a user



(b) Edit the details of a user



(c) Delete an user

Figure 10: Details, Edit or Delete allowed users to register

4. The "Add previous year grades" page was already discussed in section 4.2.3 (and can be seen in figure 3(a)). To summarize, it is a page in which an admin can add multiple already graded projects in order to improve the statistics of comparing grades/years.
5. The "Final grades statistics" page will also be discussed more in the section 5, "Results". It is used for comparing data (grades) between academic years, related to all types of subgrades discussed in the 4.5.1 Section, "Grading a project".
6. The "Final project grades" section is a page in which an admin can see all the graded projects and download the grades in excel format. It was also discussed at the start of section 4.5.1, "Grading a project".

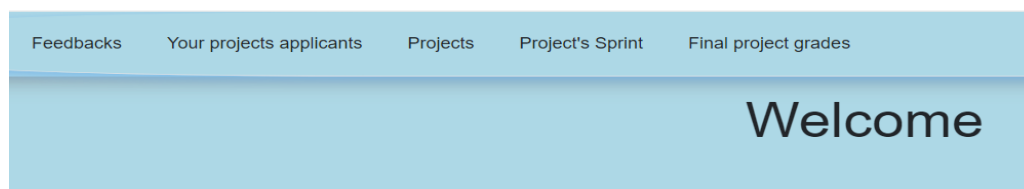


### 4.6.2 Supervisor user

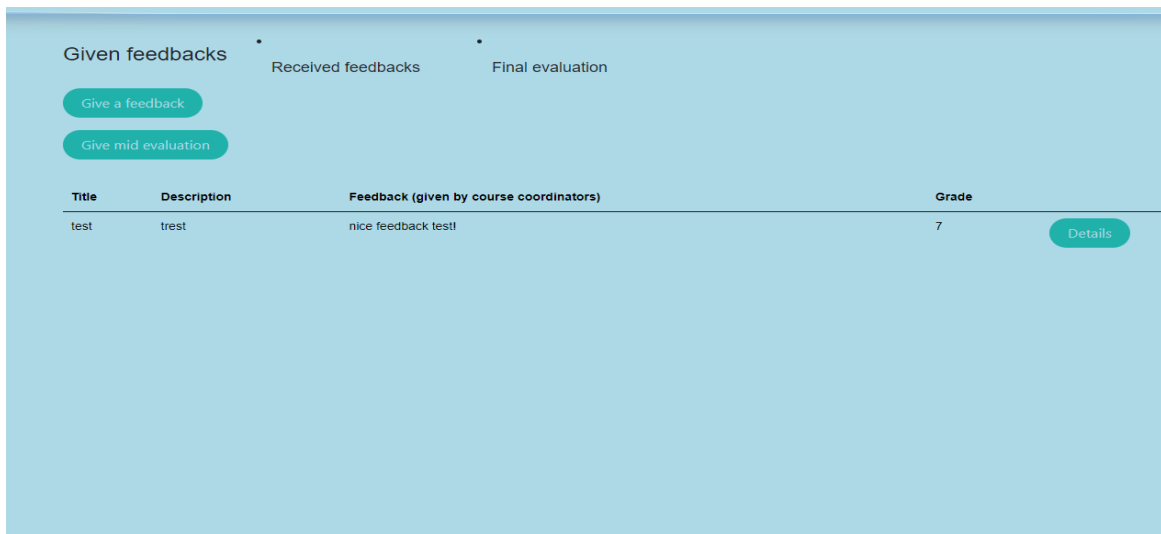
Regarding the supervisor user, we have added the actions starting at point 10 and changed the actions 8 and 9 according to the new feedback section that was explained at 4.3.

1. Create a new account.
2. Change password, close account.
3. Create a project.
4. Update or delete created projects.
5. While creating or updating a project, a supervisor can assign a second supervisor to it and he will have the same access to that project use cases.
6. While creating or updating a project, a supervisor can assign the project to a field (by choosing an already existing one or creating a new field (which can be later filterable in the projects page).
7. View created projects.
8. Filter your projects by Contacts, Languages, Comments, Description, Title or Second Supervisor.
9. View applicants for projects.
10. Accept or deny applicants.
11. Give feedback.
12. Update or delete given feedback.
13. View given feedback.
14. View received feedback, based on student and project.
15. View received final feedback.
16. Create mid-evaluation feedback.
17. Create, View, Update, Delete Sprint Board Item.
18. View your graded projects in a separate list in which you can download the according ".xlsx" file of the grade.

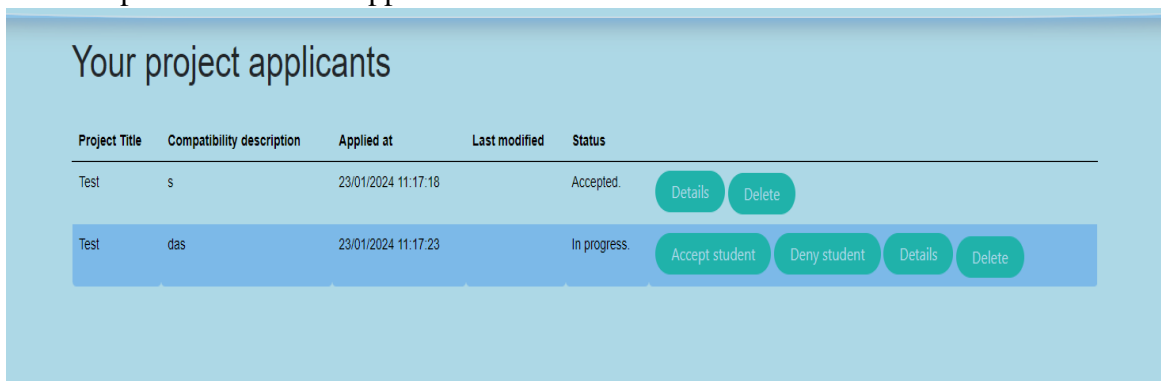
Now, let us present all supervisor available pages:



1. The "Feedbacks" button opens the feedbacks page in which the user can create a feedback, view the details of it, see received feedbacks, see his final evaluation and also give mid evaluation (those were presented in the section 4.3, "Feedbacks"). The page opens like so:

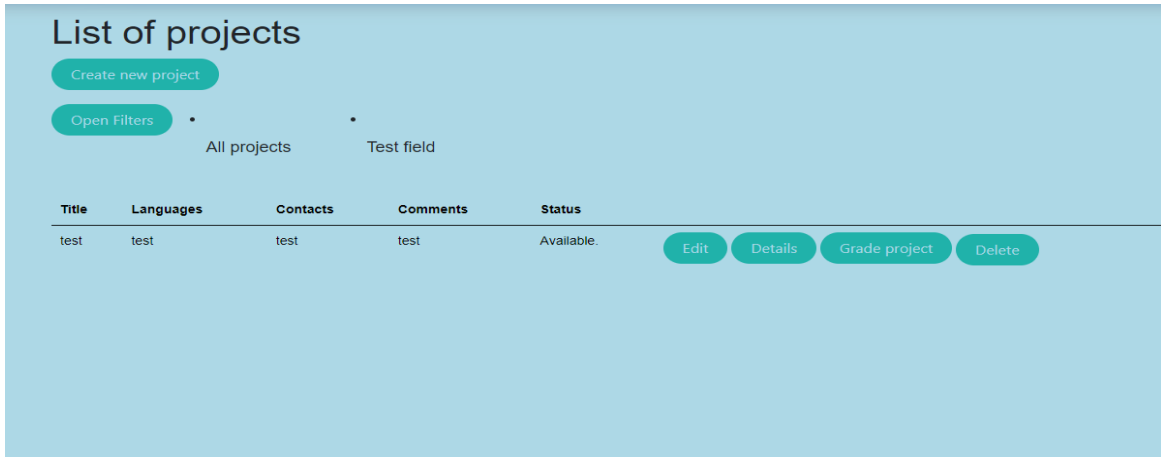


2. The next button, "Your projects applicants" opens a page in which the supervisor can see who applied to their projects and accept or deny students. In order for this flux to be complete, there are multiple statuses to the application:

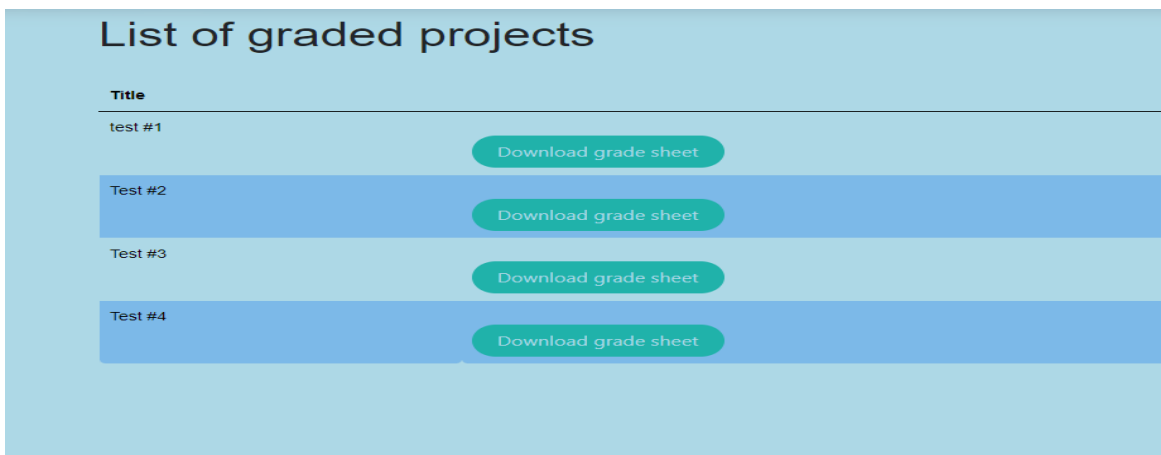


- (a) status 0: "Accept or deny student." - in which the supervisor has to take action in accepting or denying the student based on his Compatibility Description, which he writes when applying for a project.
- (b) status 1: "Waiting for student." - if the application was accepted when having the status 0, it moves to this status in which the supervisor has to wait for student approval. This status was created in order to be sure that the student that applied for this project finally chooses to work on it. - there are cases in which a student can apply to multiple projects and choose one of them to continue with.
- (c) status 2: "Application denied by you." - if the application is denied when it has status 0, then the status moves to this one and the student is notified that his application was denied.
- (d) status 3: "The student denied his application." - if the application was accepted when it has status 0 and then the student did not later accept, but denied the project application.
- (e) status 4: "Accepted." - if the student accepted the already accepted (by the supervisor user) application.
- (f) status 5: The application was deleted by the student. - it somehow moves to status 3 immediately, but will be shown only on the supervisor list (with this status) until he also deletes it.

3. The next button, "Projects", goes to the project section. The basic actions that can be done in this page are explained in the section 4.5 ("Projects"). A supervisor can only view their projects or the projects that he was assigned to.



4. "Project's Sprint" refers to the section 4.4, "Section: Sprint" and the actions that are linked to it are explained there.
5. The "Final project grades" button opens the page of already graded projects that are linked to the current users. There, the final grade assessments can be downloaded in excel format. It is similar to the first action that is discussed in section 4.2.3.

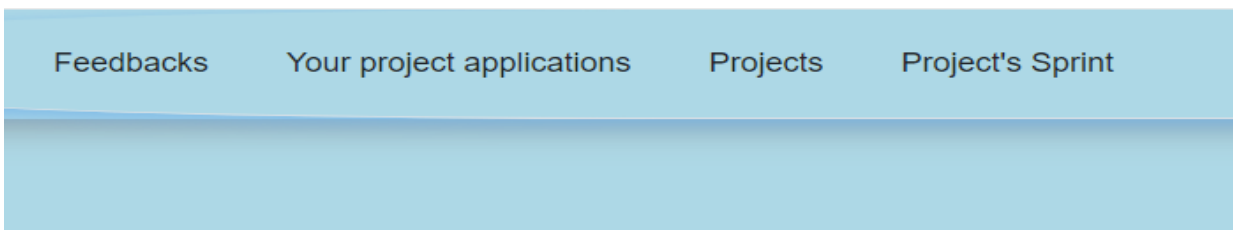


### 4.6.3 Student user

The actions that a student user can perform will be first described:

1. Create a new account.
2. Change password, close account.
3. Browse projects.
4. Filter projects by Contacts, Languages, Comments, Description, Title or Second Supervisor.
5. View applicants for projects.
6. Apply for a project.
7. View his applications for projects.
8. View if an application is accepted or not by the supervisor.
9. Accept/deny already accepted (by the supervisors) applications.
10. View assigned projects. (a project is assigned to a student when (7.) is accepted by the student)
11. Give feedback.
12. Update or delete given feedback.
13. Give final feedback.
14. View given feedback.
15. View received feedback, based on supervisor and project.
16. View mid-evaluation feedback.
17. Create, View, Update Sprint Board Item

Now, let us present the available pages for students:

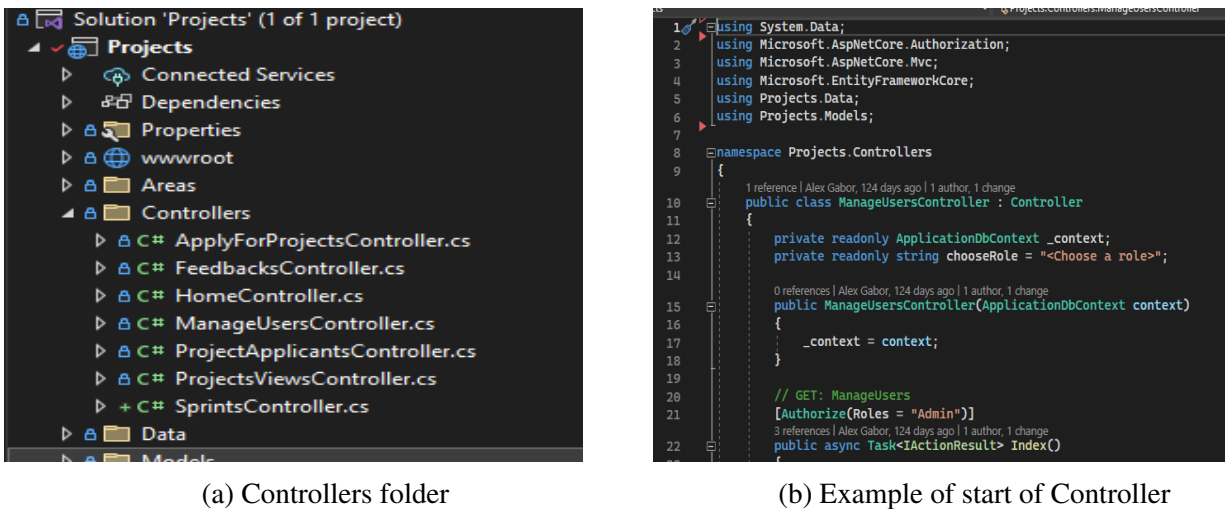


1. The "Feedbacks" button opens the feedbacks page which is similar to the supervisor's page, the only two differences are: the student can give a final evaluation for the supervisor and view his mid evaluation given by the supervisor.
2. The "Your project applications" button opens the page of each student's applications to projects, being similar to the one that the supervisor has. In last section, we have presented the statuses of each application and how it works (involving the student actions).
3. The "Projects" page opens the page with all the projects so that the student user can apply to them based on their preferences. How this page works, including the filtering of the projects, was explained previously in section 4.5 ("Section: Projects").
4. Last, but not least, the "Project's Sprint" page works exactly the same as the supervisor one, a student having the same actions as a supervisor.

## 4.7 Back-end

### 4.7.1 Controllers

Each section of the interface is linked to a Controller, which is used to determine the actions that a user can perform when is interacting with a part of the application. To give an example of the controllers that are used, and how it looks like, let us see the next image examples:



(a) Controllers folder

(b) Example of start of Controller

Figure 11: Controllers

### 4.7.2 Application Programming Interfaces (API's)

Each Controller contains (usually) more than one API, since a user can interact in multiple ways with a section from the application.

Each API has an authorize attribute that is based on his account type. That means that if a user does not have access to a part of the application, an error occurs. Those are out there only for security measures, because the front-end of the application also has (based on user role) some sections hidden from users that do not have access.

### 4.7.3 Dependency Injection

Dependency Injection is a technique that helps when it comes to design patterns. Instead of having all the logic in the API's, you create services that provide the database "selectors" in the backend application. It is also used as providing the channel that is formed through the connection of DTO's (Data Transfer Objects) that are linked to database tables.

In our application, in Controllers, we inject the previous explained uses of Dependency Injection. That helps when a method of selection in the database is used in more than one Controller or API, we do not need to write the selector again and instead we inject the service that contains the method of selection and make use of it.

To give some examples, we will discuss a method that is used in multiple API's, in order to not have duplicate code:

### 1. Create or update a project

```
//CREATE or UPDATE a project.
3 references | Alex Gabor, 5 days ago | 1 author, 1 change
public async Task CreateUpdateProject(ProjectView projectsView,
    string userId,
    string? projectId = null)
{
    var project = projectId == null ?
        new Projects.Models.Project(projectsView, userId, await GetProjectFieldId(projectsView)) :
        await _context.Project
            .Where(x => x.Id == Int32.Parse(Encryption.Decode(projectId))).FirstAsync();

    if (projectId == null)
    {
        _context.Project.Add(project);
    }
    else
    {
        Projects.Models.Project.UpdateProject(project, projectsView);
        _context.Project.Update(project);
    }

    await _context.SaveChangesAsync();
}
```

In the above image, we can see one of the simpler methods used in our solution which is used for creating or updating a project. Let us first provide two images with the Create and Update project API's, and then we will explain how it works:

```
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Staff")]
0 references | Alex Gabor, 62 days ago | 1 author, 2 changes
public async Task<IActionResult> Create(ProjectView ProjectsView)
{
    await _projectsService.CreateUpdateProject(ProjectsView,
        User.FindFirstValue(ClaimTypes.NameIdentifier));
    return RedirectToAction(nameof(Index));
}
```

(a) Create a project

```
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Staff")]
0 references | Alex Gabor, 62 days ago | 1 author, 2 changes
public async Task<IActionResult> Edit(int id,
    ProjectView ProjectsView, string encodedid)
{
    await _projectsService.CreateUpdateProject(ProjectsView,
        User.FindFirstValue(ClaimTypes.NameIdentifier),
        encodedid);
    return RedirectToAction(nameof(Index));
}
```

(b) Update a project

Figure 12: Example API's

In figures 12(a) and 12(b) we can see that the method `CreateUpdateProject` is called with different parameters. The difference in those calls are that when editing a project, we can catch the project id from the frontend of the application and add it as a parameter. The parameter has a default value of null (which is written as `"= null"` in the method parameters) which helps making the difference between creating or updating a certain project. This helps with making the controllers API's coding style very clean and also makes the code more readable since we do not have duplicity between database calls and actions.

## 4.8 Front-end

Since our platform is a back-end focused application, we have chosen to work with razor pages for the front-end. A razor page is like a html based page, but integrated with data which is provided by the API's. Each razor page is bond to a model or DTO and can iterate between lists or results. Mostly, the API's that we use in this application return a View Bag, in other words: a html component that has data provided in it, after making use of some selections and/or query's on the database.

Another important aspect of the front-end is that the data is encrypted, every id that belongs to a column in a table from the database comes encoded in a view of a page in order to improve the security of the website.

To give some examples, we will provide some razor page code and explain how it works:

1. Close the filters from the project's page

```
<a class="btn btn-primary a-button" asp-area="" asp-controller="Projects" asp-action="Index"
  asp-route-openFilters="false"
  asp-route-FilterContacts="@Model.FilterContacts"
  asp-route-FilterLanguages="@Model.FilterLanguages"
  asp-route-FilterComments="@Model.FilterComments"
  asp-route-FilterDescription="@Model.FilterDescription"
  asp-route-FilterTitle="@Model.FilterTitle"
  asp-route-FilterSecondSupervisor="@Model.FilterSecondSupervisor">Close Filters</a>
```

In this example, basically what we tell the code to do is the following:

- (a) first, the classes of "a" are written in a different file which is a ".css" one, which is used for styling how the button looks like on the website.
- (b) then, the asp-controller attribute tells that when pressing the button, we should go the the project's controller.
- (c) next, we have the action of Index, which is an API inside the project's controller that goes to the all projects page, based on user.
- (d) and last, but not least, we have the asp-route-"parameter" list that contains the information that is needed in order to close the filters and also keep their value that was already filtered on projects.

## 2. Create a list of projects

```
@foreach (var item in Model.Projects) {  
<tr>  
<td>  
    @Html.DisplayFor(modelItem => item.Title)  
</td>  
<td>  
    @Html.DisplayFor(modelItem => item.Languages)  
</td>  
<td>  
    @Html.DisplayFor(modelItem => item.Contacts)  
</td>  
<td>  
    @Html.DisplayFor(modelItem => item.Comments)  
</td>  
<td>  
    @if (item.StudentEncodedId != "")  
    {  
        <span>Taken.</span>  
    }else{  
        <span>Available.</span>  
    }  
</td>  
<td>  
    @if (User.IsInRole("Student"))  
    {  
        <a class="a-button" asp-area="" asp-controller="ProjectApplicants"  
asp-action="Create" asp-route-encodedid="@item.EncodedId">Apply for this project</a>  
        <a class="a-button" asp-action="Details" asp-route-EncodedId="@item.EncodedId">Details</a>  
    }  
    @if (User.IsInRole("Staff"))  
    {  
        <a class="a-button" asp-action="Edit" asp-route-EncodedId="@item.EncodedId">Edit</a>  
        <a class="a-button" asp-action="Details" asp-route-EncodedId="@item.EncodedId">Details</a>  
        <a class="a-button" asp-action="GradeProject" asp-route-EncodedId="@item.EncodedId">Grade project</a>  
        <a class=" btn btn-danger a-button" asp-action="Delete" asp-route-EncodedId="@item.EncodedId">Delete</a>  
    }  
</td>  
</tr>  
</tr>
```

In order to create the list of projects as a view, we send from the API a project list, which then we use as an iterable object for creating each row from the all projects view. We show the title, languages, contacts and comments of each one and then, based on user, we show some actions that the current user can perform, which are linked to different API's from the Project Controller.

Also, if the list is empty, we show a message that the API was not able to find any project with the applied filters.

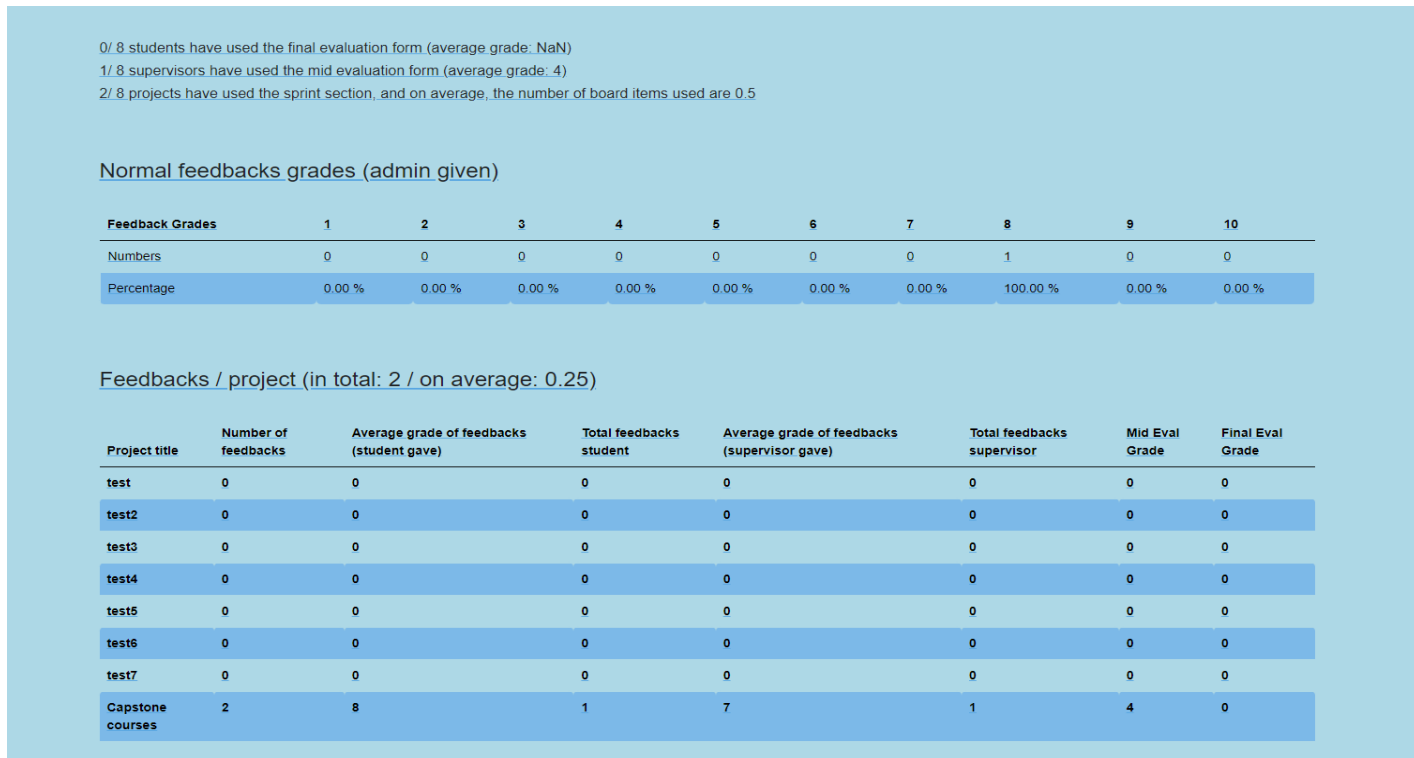
```
@if (Model.Projects.Count == 0)  
{  
    <h3> There are no projects that match the filters.</h3>  
}
```



## 5 Results

In this section, we will discuss how to analyze our platform's statistics, as an admin user. The statistics take into account only graded projects.

### 5.1 Analyzing feedback statistics

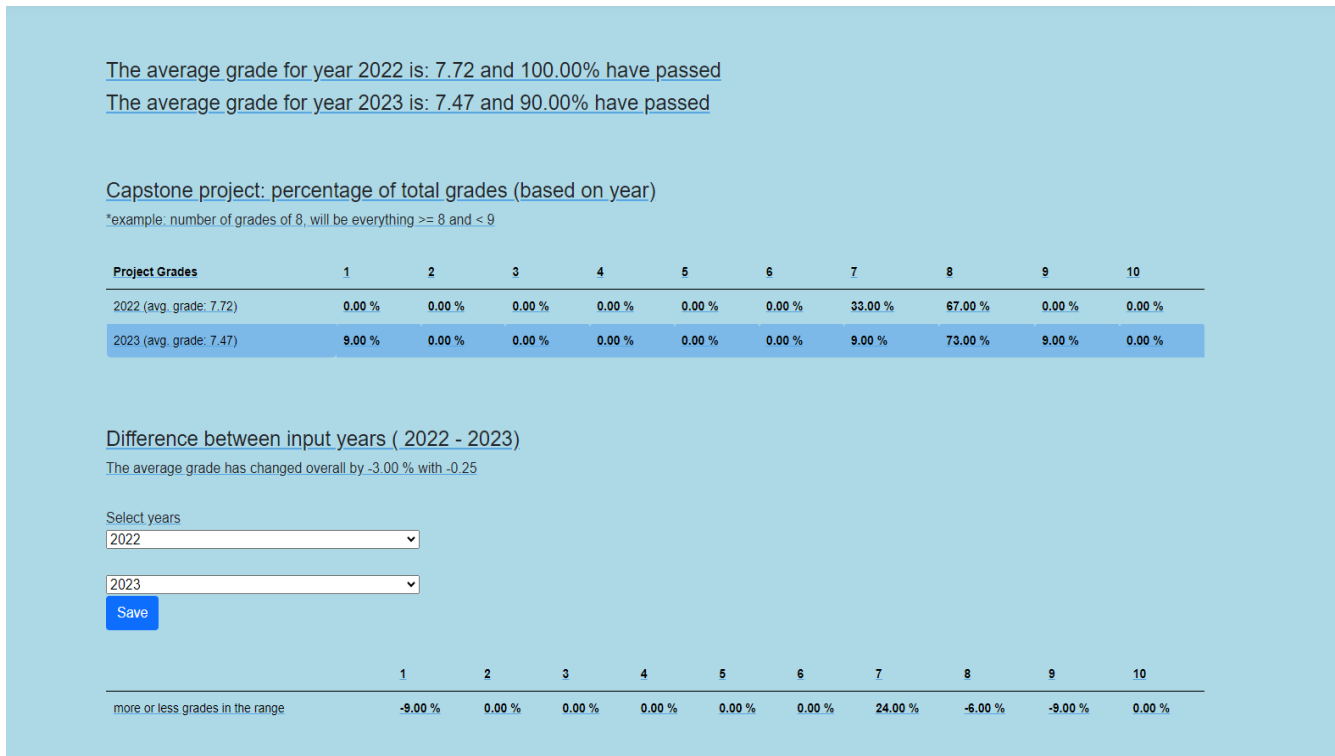


The "Normal feedbacks grades (admin given)" table presents what are the numbers and percentages of each grade that an admin gives to a feedback. This can be further evaluated, if discussing about future work, by dividing them into groups by year, to calculate the improvements that the staff and student users are making towards giving meaningful feedback.

The "Feedbacks / project" section presents, first in parantheses: the total number of feedbacks given and the average number of feedbacks/project that are given, and then a table that shows multiple counts and average grades that are given for a project (by both student and staff sides). This table can evaluate the whole progress of a student/supervisor user when giving feedback to one another.

## 5.2 Analyzing project statistics

### 1. Overall final grade statistics.



The section of analyzing project statistics contains, first, the average grades of every year projects that are added by admin users or graded by supervisors while using the platform.

Then, it contains a table of project grades percentages that are registered. To be specific, each column ( $x$ ) contains the grades ( $g$ ) that are in the range:  $x \leq g < x + 1$ .

The final table is filled up with the increase/decrease in percentages (like in the first table) between 2 input years. It also calculates the overall average grade increase (by percentage and value).

### 2. Grade statistics for each subgrade.

In this section, we will evaluate each subgrade (presented in the section 4.5.1 ("Grading a project")). There is a table created for each subgrade that takes into account the grade for that section from a certain year. Each table will represent a year, followed by percentages of how many grades were in a certain range. The years that are compared can be changed, so that the user gets a perspective of how well the students perform on different years.

To provide an example, let us present the statistics page:

Final grades statistics **Grade A** Grade B Grade C Grade D

Select years  
 2024  
 2023  
 Change years

ProblemAnalysis

Year/Grades	5	6	7	8	9	10
2024 (total number of grades: 0.00)	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %
2023 (total number of grades: 0.00)	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %	0.00 %

LiteratureResearch

Year/Grades	5	6	7	8	9	10
2024 (total number of grades: 3.00)	0.00 %	100.00 %	0.00 %	0.00 %	0.00 %	0.00 %
2023 (total number of grades: 1.00)	0.00 %	100.00 %	0.00 %	0.00 %	0.00 %	0.00 %

ResearchStrategy

Year/Grades	5	6	7	8	9	10
2024 (total number of grades: 3.00)	0.00 %	33.33 %	66.67 %	0.00 %	0.00 %	0.00 %

From the top of the page, we can see that there are multiple fields to choose from. The first field, "Final grade statistics" refers to the previous explained section. The next 4, refer to the grades that are presented in section 4.5.1.

Each subgrade has a total number of grades that were given in that section, in a certain year. For example, the first one, "Problem Analysis" has 0 grades in both years because, when grading, we have chosen to assign "N/A" to that section in the 4 graded projects.

Each page from Grades A to D have the same view, having all the subgrades in that section.

## 6 Limitations

In this section, we will provide some limitations that the current platform faces with.

### 6.1 Measurement Instruments and Scope of the Study

Currently, the platform is constructed only for the potential use of University of Groningen. If it were to be put to use, there would not be many entries of grades per year so that the platform's statistics could provide usable data.

It for sure can be compared to previous years, but the data should be tested through the period of multiple years of applying improved formative assessment techniques [6] in order to be relevant.

For the part of statistics for feedback, the grades that both the supervisor and student user receive from the admin one could not always be valid. In order for this use case to work, the admin should follow a grading system that covers all the values of formative assessment.

### 6.2 Data collection

There are no larger studies available in which formative assessment is frequently and in good practice used in order to improve the quality of learning environments. For example, in the study of [2] the number of respondents are between 3 and 15 and even though it is clear that the approach shown there is relevant, we can not say that we have an available study proving it. We can not provide the information regarding the improvements of a study environment over the course of years using ways of formative assessment.

### 6.3 Subjectivity

Since creating the platform's logic, use cases and implementation has been challenging, we might have missed some aspects of the study's practicality while applying it to capstone projects.

### 6.4 External Influences

The study is based on the current grading system of bachelor projects at University of Groningen. If it were to change, there should be constructed a way of relating the previous bachelor subgrades with the current ones.

Of course, if the platform were to be used, we can not say for certain that the users would interact with it to its fully potential. Currently, there are no filters available in the platform to only compare outcomes of projects in which the formative assessment tools are being used to extent.

## 7 Future work

- Following the subsections: "Measurement Instruments and Scope of the Study" (6.1) and "Data Collection" (6.2) from Limitations, we can safely admit that the platform's statistics would benefit greatly if the collection of data would be from multiple academic departments across different institutions. A solution to this would be:
  - Add a new table containing: an id (primary key), an institution name, a department name.
  - Then, create a new page in which admins can create such an entry in this table. After doing so, when an admin gives access to a certain user (so that he can create the account) it should also contain the information of institution + department they belong to.
  - Following this, when a user creates an account - they should state: institution + department in order to be able to create the account.
  - Then, the projects should also be linked to the table that was created and be always filtered by the current's user institution + department.
- To add to the previous point, in summary, it would mean that the platform needs to be updated with the possibility to include other methods of grading and thus, it could expand to grading not only capstone projects, but also other course types.
- Following the subsection: "Subjectivity" (6.3) we can state that in order for this platform to be continuously improved, there should be at least two sides working on it: someone that researches and finds also other ways of improving formative assessment and states them in use cases for the platform and someone that implements those use cases.
- Following the subsection: "External Influences" (6.4) we can safely say that the platform would benefit from a filtering of statistics data based on how often the feedback or sprint section are used. This filter part can be created visually the same as the one for projects, but the logic will contain either a count of how frequent the feedback section is used per project or either a minimum grade that the admin gave to certain feedbacks after project grade completion.
- Website deployment
- Emails: right now, the emails are send by a free subscription to SendGrid API's. We are restricted to sending 50 emails max per day, so coming up with a solution for this, when deploying the website, is necessary.
- Mobile Accessibility: optimizing the website to include mobile readability or create a mobile application.
- Database: currently, the database is created and stored locally. When progressing with the website deployment, our database should also be stored and maintained by a server.
- Testing database: a database that can be used for testing, while making code modifications. This can be mirrored from the live database from time to time, or it can contain dummy data.
- Continuous Integration and Continuous Deployment: implementing a pipeline to automate the process of deployment upon finishing tasks.
- Backup Recovery: setting up regular automated database information recoveries, in case of data loss.

## 8 Conclusion

The objective of this thesis was to research and find solutions that can be implemented in a platform in order to improve the quality of studying and teaching during capstone projects. We made the decision to integrate a continuous feedback functionality, allowing users to select their preferred approach to engage with it. The feedback sections, the sprint section or both combined, can increase the potential of capstone courses.

This platform creates an exciting space for students and staff to collaborate, exchange ideas, and offer their support. It encourages a culture of continuous feedback, where users can actively contribute to the improvement of their experience. Students and staff will have the opportunity to engage in meaningful discussions and provide two-way valuable feedback that mirrors a dynamic and supportive learning environment.

However, the platform is not ready for deployment. In the section on future work, we will explore possible ways to implement and identify necessary requirements in order for the platform to effectively function.

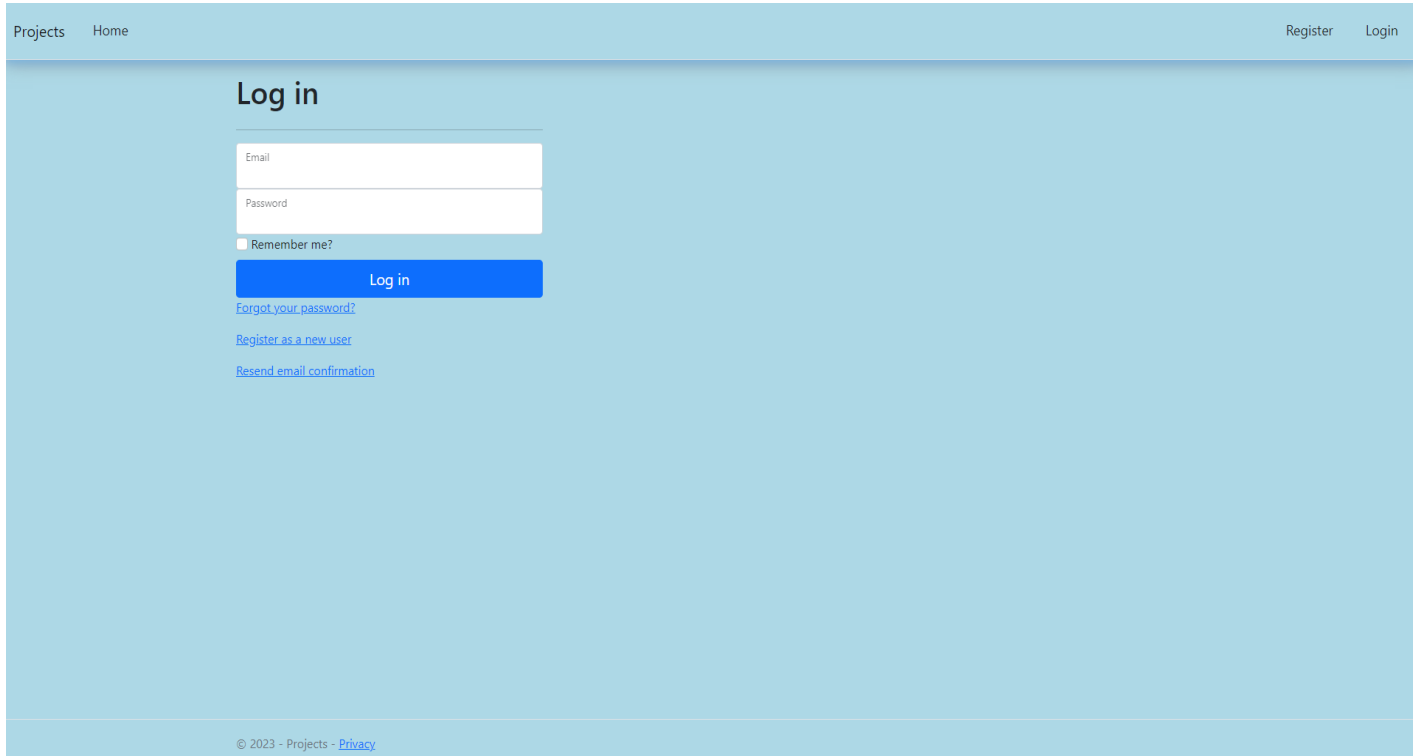
Overall, the implementation we've discussed is promising for a future platform that has the potential to enhance communication while improving teaching and learning skills during capstone projects. It aims to create an improved student-staff relationship, while using values of formative assessment.

## Bibliography

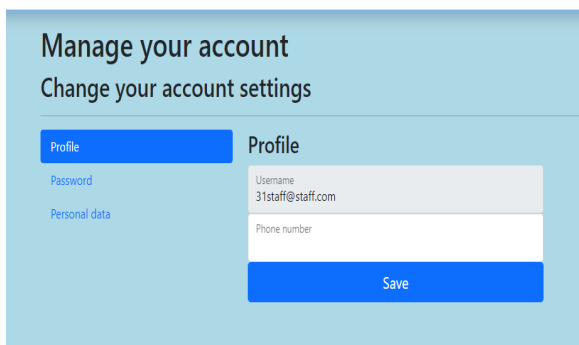
- [1] L. Adams, M. Daniels, A. Goold, O. Hazzan, K. Lynch, and I. Newman, “Challenges in teaching capstone courses,” *SIGCSE Bull.*, vol. 35, no. 3, pp. 219–220, September 2003.
- [2] J. Derounian, “Shall we dance? the importance of staff–student relationships to undergraduate dissertation preparation,” *Active Learning in Higher Education*, no. 12, pp. 91–100.
- [3] V. Farrell, G. Ravalli, G. Farrell, P. Kindler, and D. Hall, “Capstone project: Fair, just and accountable assessment,” in *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, (Haifa, Israel: ACM), 2012.
- [4] T. Clear, “Thinking issues: the three p’s of capstone project performance,” *SIGCSE Bull.*, vol. 41, no. 2, pp. 69–70, June 2009.
- [5] S. Chappuis and J. Chappuis, “The best value in formative assessment,” *Educational Leadership*, vol. 65(4), pp. 14–19, 2009.
- [6] M. Heritage, “Formative assessment: What do teachers need to know and do?,” *Phi Delta Kappan*, vol. 89, no. 2, pp. 140–145, Oct 2007.
- [7] C. Aasheim and J. Shropshire, “Knowledge and skill requirements for entry-level it workers: A longitudinal study,” *Journal of Information Systems Education*, vol. 23, no. 2, pp. 193–204, Jun 2012.
- [8] T. Clear, “Managing mid-project progress reviews: A model for formative group assessment in capstone projects,” *ACM Inroads*, pp. 14–15.

# Appendices

## A Login

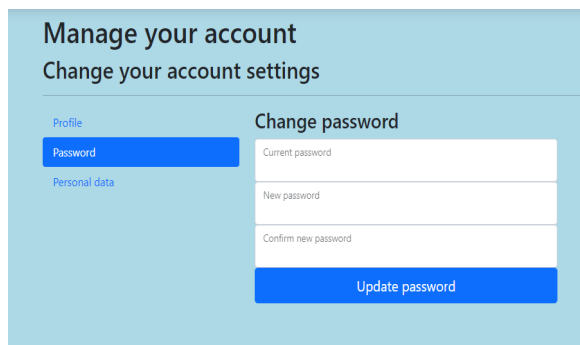


The screenshot shows a login page with a light blue background. At the top left, there are links for "Projects" and "Home". At the top right, there are links for "Register" and "Login". The main heading is "Log in". Below it, there is a form with two input fields: "Email" and "Password". There is a checkbox labeled "Remember me?". A blue button labeled "Log in" is positioned below the form. Below the button, there are three links: "Forgot your password?", "Register as a new user", and "Resend email confirmation". At the bottom of the page, there is a footer with the text "© 2023 - Projects - Privacy".



The screenshot shows a "Manage your account" page with the sub-heading "Change your account settings". On the left, there is a sidebar with three menu items: "Profile" (highlighted in blue), "Password", and "Personal data". The main content area is titled "Profile" and contains two input fields: "Username" with the value "31staff@staff.com" and "Phone number". A blue "Save" button is at the bottom.

(a) Change account settings



The screenshot shows a "Manage your account" page with the sub-heading "Change your account settings". On the left, there is a sidebar with three menu items: "Profile", "Password" (highlighted in blue), and "Personal data". The main content area is titled "Change password" and contains three input fields: "Current password", "New password", and "Confirm new password". A blue "Update password" button is at the bottom.

(b) Change Password