

Even Bryhn Bøe

# From Data to Insights: Efficiently Visualizing Peer-Review Data in Computer Science Courses

Master's thesis in Applied Computer Science

Supervisor: Rune Hjelsvold

June 2024



Even Bryhn Bøe

# **From Data to Insights: Efficiently Visualizing Peer-Review Data in Computer Science Courses**

Master's thesis in Applied Computer Science  
Supervisor: Rune Hjelsvold  
June 2024

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







# Abstract

This thesis presents the development and evaluation of a Learning Analytics Dashboard for visualizing peer-review data in computer science courses using the Computer Science Assignment Management System (CSAMS).

As courses grow in size, lecturers increasingly struggle to maintain personal relationships with students, losing a vital part of assessing student progress. The aim of the dashboard was to compensate for the missing personal assessment component by efficiently providing insights into peer-review data. Through an iterative approach, we designed, implemented, and evaluated the dashboard to enhance its usability and functionality.

The primary objective was to enable educators to efficiently gain an overview and insight into peer-review data, thereby helping the identification of challenges in the course and in quality assurance of the peer-reviews, ultimately to enhance educational outcomes. The implemented dashboard has functionality to identify challenging topics, struggling students, student trajectory as well as alerts for quality problems with the peer-reviews.

By moving application logic to the database, we could use database aggregations functions to calculate the average score as opposed to moving all the data to the application to calculate the score. This drastically increased the performance of the data aggregation by keeping the logic closer to the data thus reducing data transfer overhead.

The evaluation found that the dashboard was valuable in gaining insight into the peer-review data and helped in identifying problems.



# Sammendrag

Denne oppgaven presenterer utviklingen og evalueringen av et Learning Analytics Dashboard for visualisering av fagfellellevurderingsdata i datavitenskapelige fag gjennom Computer Science Assignment Management System (CSAMS).

Etterhvert som fag vokser i størrelse sliter undervisere mer med å opprettholde kjennskapen til enkeltstudenter og med det mister de en viktig del av evalueringsprosessen. Hensikten med dashbordet var å kompensere for denne manglende delen ved å tilby effektiv innsikt i fagfellellevurderingsdataen. Ved bruk av en iterativ metodikk designet, implementerte og evaluerte vi dashbordet for å forbedre brukervennligheten og funksjonaliteten dens.

Hovedmålet var å tilrettelegge for effektiv oversikt og innsikt i fagfellellevurderingsdataen og dermed hjelpe med undervisere med å identifisere utfordringer i faget og for å få innsikt i kvalitetsproblemer med fagfellellevurderingene med mål om å forbedre undervisningsresultatene. Det implementerte dashboardet har funksjonalitet for å identifisere utfordrende emner, studenter som sliter, studentens utvikling samt varsler om kvalitetsproblemer med fagfellellevurderingene.

Ved å flytte applikasjonslogikken til databasen, kunne vi bruke databaseaggregeringsfunksjoner for å beregne gjennomsnittsscore i stedet for å flytte alle dataene til applikasjonen for å beregne scoren der. Dette økte ytelsen til dataaggregeringen betydelig ved å holde logikken nærmere dataene og dermed redusere ytelse overhead ved overføring av data.

Evalueringen viste at dashbordet var verdifullt for å få innsikt i fagfellellevurderingsdataene og hjalp til med å identifisere problemer.



# Acknowledgements

I would like to thank my supervisor, Rune Hjelsvold, for his invaluable feedback and guidance. I would also like to thank the participants of the user tests for their interest and input, and to my family, friends, and classmates for their support.



# Contents

<b>Abstract</b> . . . . .	<b>iii</b>
<b>Sammendrag</b> . . . . .	<b>v</b>
<b>Acknowledgements</b> . . . . .	<b>vii</b>
<b>Contents</b> . . . . .	<b>ix</b>
<b>Figures</b> . . . . .	<b>xi</b>
<b>Tables</b> . . . . .	<b>xiii</b>
<b>Acronyms</b> . . . . .	<b>xv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Keywords . . . . .	1
1.2 Motivation . . . . .	1
1.3 Research Questions . . . . .	2
1.4 Outline . . . . .	2
<b>2 Background</b> . . . . .	<b>3</b>
2.1 Defining "Dashboard" . . . . .	3
2.2 Computer Science Assignment Management System (CSAMS) . . . . .	3
2.3 Learning Analytics (LA) . . . . .	4
<b>3 Related Work</b> . . . . .	<b>5</b>
3.1 Literature Search . . . . .	5
3.2 Literature Review . . . . .	6
3.3 Literature Review . . . . .	6
3.3.1 Dashboard Design Principles . . . . .	6
3.3.2 Dashboard Implementations . . . . .	9
3.3.3 Dashboard Requirements . . . . .	13
3.3.4 Dashboard Evaluation . . . . .	14
<b>4 Methodology</b> . . . . .	<b>17</b>
4.1 Design Science Research . . . . .	17
4.2 Design Science Research Methodology . . . . .	18
4.3 Validating the Artifact . . . . .	19
4.3.1 Qualitative Measurements . . . . .	19
4.3.2 Quantitative Measurements . . . . .	20
4.4 Deployment . . . . .	20
4.5 Dataset . . . . .	21
4.6 Query Benchmarking . . . . .	22
4.7 Success Criteria . . . . .	22

<b>5</b>	<b>Results</b>	<b>23</b>
5.1	First Iteration	23
5.1.1	System Design	23
5.1.2	UI Design	24
5.1.3	Development	28
5.1.4	Evaluation	34
5.1.5	Quantitative Evaluation	39
5.1.6	Conclusion - First Iteration	39
5.2	Second Iteration	40
5.2.1	UI Design	40
5.2.2	Development	45
5.2.3	Evaluation	52
5.2.4	Quantitative Evaluation	57
5.2.5	Conclusion - Second Iteration	57
<b>6</b>	<b>Discussion</b>	<b>59</b>
6.1	Database	59
6.1.1	Precalculated Score	59
6.1.2	Scalability	59
6.2	Dashboard Effectiveness	61
6.3	Dashboard Design and Implementation	61
6.4	Limitations	63
<b>7</b>	<b>Conclusion</b>	<b>65</b>
7.1	Summary	65
7.2	Research Questions	66
7.3	Future Work	68
7.3.1	More Iterations	68
7.3.2	Longitudinal Study	68
7.3.3	Student facing dashboard	68
7.3.4	Data-driven Dashboard	69
7.3.5	Better Utilization of Textual Data	69
	<b>Bibliography</b>	<b>71</b>
<b>A</b>	<b>Create Views Queries</b>	<b>75</b>
<b>B</b>	<b>Explain Analyze Output</b>	<b>77</b>
<b>C</b>	<b>Anonymization Query</b>	<b>81</b>
<b>D</b>	<b>Create Materialized View</b>	<b>83</b>



# Figures

4.1	Database diagram for CSAMS . . . . .	21
5.1	System Diagram for CSAMS with Analytics Service . . . . .	24
5.2	Assignment - Class Overview, Design . . . . .	25
5.3	Assignment - Individual Students, Design . . . . .	26
5.4	Course Overview, Design . . . . .	27
5.5	Pseudocode of Score Calculation Logic . . . . .	29
5.6	Sequence Diagram for Requesting Dashboard Page . . . . .	31
5.7	Assignment - Class Overview, Implementation . . . . .	32
5.8	Assignment - Individual Student, Implementation . . . . .	33
5.9	Course Overview, Implementation . . . . .	33
5.10	Assignment Overview, Design . . . . .	41
5.11	Alerts Inspiration . . . . .	42
5.12	Course Overview, Design . . . . .	43
5.13	Reviews Overview, Design . . . . .	44
5.14	Database View Dependency Diagram . . . . .	46
5.15	Assignment Overview, Implementation . . . . .	49
5.16	Course Overview, Implementation . . . . .	50
5.17	Reviews Overview, Implementation . . . . .	51



# Tables

5.1	Quantitative Measurements . . . . .	39
5.2	Quantitative Measurements . . . . .	57



# Acronyms

**CLS** Cumulative Layout Shift. 20, 39, 57

**CSAMS** Computer Science Assignment Management System. iii, v, ix, xi, 1–4, 13, 19, 20, 24, 28, 29, 31, 35, 36, 38, 39, 45, 52, 56, 61, 63, 65–68

**FCP** First Contentful Paint. 20, 39

**LA** Learning Analytics. ix, 4, 5

**LAD** Learning Analytics Dashboard. iii, v, 2, 4, 5, 8–11, 14, 15, 66, 68

**LCP** Largest Contentful Paint. 20, 39

**LMS** Learning Management System. 3, 10, 11, 13, 63

**NLP** Natural language processing. 23, 69

**NTNU** Norwegian University of Science and Technology. 6, 20

**SSE** Server-Sent Events. 48, 61

**SUS** System Usability Scale. 35



# Chapter 1

## Introduction

There are three kinds of lies:  
lies, damned lies, and statistics

---

*Mark Twain*

### 1.1 Keywords

- Development
- Learning Analytics
- Visualization techniques
- Dashboard design
- Data aggregation
- Computer Science Education

### 1.2 Motivation

Computer Science Assignment Management System (CSAMS) is an assignment management system originally developed as part of a bachelor thesis [1]. A central component of the system is peer-reviews where students will evaluate each others submissions (more details in section 2.2). Through this, a large amount of data is collected, which is currently being underutilized. This thesis aims to make this data more accessible and quickly understandable through visualizations. By transforming data into actionable insights, lecturers will be able to make more effective decisions to address educational gaps. These insights will aid in identifying both individual student challenges and collective learning hurdles within a course, enabling effective and accurate pedagogical interventions. This approach aims at enhancing educational outcomes by leveraging peer-review data to inform teaching strategies.

While the largest course in the system currently has a little over 100 students, any additions to the system should consider the possibility that the system could be adopted for larger courses and therefore consider the scalability of the additions.

## 1.3 Research Questions

- **RQ1:** What objectives do the lecturers using Computer Science Assignment Management System (CSAMS) wish to accomplish with a Learning Analytics Dashboard?
- **RQ2:** What technical and usability challenges need to be solved when implementing a Learning Analytics Dashboard for CSAMS?
- **RQ3:** How can data be efficiently aggregated in the CSAMS database?

RQ1 addresses what needs the lecturers have from a dashboard to inform the development. RQ2 evaluates the developed dashboard in relation to the discovered needs and addresses the challenges in implementing the solution. RQ3 addresses performance concerns and optimization of the database. In this context, "efficiently" means that data aggregation can be integrated into the front-end without causing performance issues, while maintaining the performance limits as described in subsection 4.3.2.

## 1.4 Outline

**Chapter 2 - Background** - This chapter provides theoretical background for some central concepts.

**Chapter 3 - Related Work** - This chapter contains the literature search and review.

**Chapter 4 - Methodology** - This chapter goes over the methodology followed.

**Chapter 5 - Results** - This chapter outlines the design, implementation and evaluation of the prototype.

**Chapter 6 - Discussion** - This chapter discusses overarching aspects of the results.

**Chapter 7 - Conclusion** - This chapter answers the research questions, summarizes the findings, and makes suggestions for future work.



## Chapter 2

# Background

### 2.1 Defining "Dashboard"

Few defines a dashboard as: "a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance" [2, p. 26].

However, this is an idealised version of dashboards where there is only a single screen. As we will go over in chapter 3, this is not necessarily possible or the best solution in all cases.

Therefore, we define a dashboard as: "a visual display of the most important information needed to achieve one or more objectives; arranged on a single screen or a system of multiple screens to allow information to be monitored at a glance".

### 2.2 Computer Science Assignment Management System (CSAMS)

CSAMS is an assignment management system that was originally implemented as part of a bachelor thesis[1] intended to solve some of the shortcomings in other Learning Management System (LMS) for programming assignments. The system allows a lecturer to create an assignment for a course with a submission form. Since the system is made for programming assignments, this is often just a text field to provide a link to a remote git repository. Then, once the assignment deadline has passed and if it is enabled for the assignment, the peer-review form will open. Here, the students are randomly assigned someone else's submission and asked to fill out a form to review the other persons work. The peer-review process is anonymous for both the reviewer and the reviewee. Although, since most submissions include a link to a git repository, the reviewer often knows who the reviewee is.

As courses grow in size, lecturers find it difficult to maintain a personal relationship with students and thus lose a vital part of assessing how the students are progressing. At this point, the lecturers become more reliant on the data gathered from reviews to compensate for this missing component of assessment in addition to getting a group sentiment. However, the current reporting and data aggregation of CSAMS is limited to a simple list of who has submitted and how many reviews they have done in addition to the ability to export data as an Excel spreadsheet that contains the average scores received for each student who has submitted

for each question of the review form. While this is better than nothing it still leaves something to be desired which has led to system administrators dumping the data and creating custom Excel spreadsheets to get the desired overview; this is not a viable solution at scale.

Since its creation, CSAMS has been subject to multiple theses. One thesis added group submissions[3] and another investigated using ChatGPT to evaluate peer-reviews[4].

## **2.3 Learning Analytics (LA)**

Learning Analytics (LA) involves collecting and analyzing large sets of student-related data to identify trends and challenges and improve educational outcomes. Initially, it focused on identifying at-risk learners to enhance retention. Modern tools now guide students towards better study behaviors. Visualization tools, such as learning dashboards, display data on student activity, interactions, and marks, helping educators monitor progress and provide timely feedback. These tools support education by making complex data accessible and actionable for educators. Learning Analytics (LA) combines data analysis techniques with effective visualization to provide insights through Learning Analytics Dashboard (LAD). Learning Analytics Dashboard (LAD) will later be referred to as just "dashboard".

## Chapter 3

# Related Work

In this chapter, we will explore the existing literature relevant to this thesis.

Our goal is to identify effective visualization techniques and dashboard design guidelines that has been used in educational settings and relates to LA. This exploration will help us understand what features are most beneficial for lecturers and how dashboards can be optimized to improve insight and learning outcomes.

Additionally, we are interested in reviewing LADs to get a clearer understanding of the types of questions these tools aim to answer and how data is visualized to answer them.

We are also interested in the evaluation of these LADs and their findings of real-world effectiveness and impact, what worked and what did not work. Additionally, any literature on the evaluation of LADs is also relevant.

### 3.1 Literature Search

While this is not a structured literature review it is based on it and is made up of parts to give this process credibility.

Evaluated for relevancy in order: title, keywords, abstract, intro, conclusion, full text. If still relevant after read full text, the article is mentioned later in the review (section 3.2). Only if it was clear the paper was not relevant was it dropped. When in doubt, the paper was included into the next step. In earlier steps there were more leniency i.e. if the keywords were not a complete match it was still included.

#### Inclusion Criteria

- Studies that describe the implementation of a dashboard
- Studies that describe visualization techniques
- Studies that describe dashboard design
- Studies that describe lecturer requirements of a dashboard

### Exclusion Criteria

- Full-text not available through NTNU network
- Paper is a presentation or keynote
- Duplicate paper
- Publication language is not English

### Queries

- "learning analytics" AND ("visualization techniques" OR "information visualization")
  - indented to capture visualization techniques used in learning analytics
- "learning analytics" AND "dashboard design"
  - indented to capture dashboard design guidelines used in learning analytics
- ("learning analytics" OR "educational data mining") AND "dashboard"
  - indented to capture learning analytics dashboards in other systems

### Databases

- Springer Link<sup>1</sup>
- IEEE Explore<sup>2</sup>
- ACM DL<sup>3</sup>
- Science Direct<sup>4</sup>

## 3.2 Literature Review

We found 18 papers that was deemed relevant and that has affected the work going forward. These were grouped into four categories. Here we present a summary of the papers, their findings, and what we learned from them.

## 3.3 Literature Review

### 3.3.1 Dashboard Design Principles

#### ‘Dashboard Design Patterns’ (Bach *et al.*) [5]

This paper outlines design patterns that can be applied in dashboard design based on a review of 144 dashboard. It identifies 42 design patterns split into Content and Composition categories. Content patterns focus on the types of data displayed, including detailed datasets and meta information, and their visual representations like tables, gauges, and progress bars to convey data at different levels of abstraction. Composition patterns address the dashboard’s layout

---

<sup>1</sup><https://link.springer.com>

<sup>2</sup><https://ieeexplore.ieee.org>

<sup>3</sup><https://dl.acm.org>

<sup>4</sup><https://www.sciencedirect.com>

and interaction, discussing page layouts, screen-space management, and interactive elements that allows for user engagement and data exploration.

Of particular interest are the patterns relating to drilling down to gradually more detailed information and the use of colors to quickly draw the attention of the user.

The paper also outlines the trade-offs between screen-space, number of pages, abstraction and interaction saying designers have to balance these variables to create a dashboard that fits a use-case. This is especially relevant to a discussion on single or multi-page dashboards where this paper does not present one as better than the other but rather that it is a trade-off between content and complexity in navigation while avoiding visual clutter and cognitive load.

### ***Information dashboard design: The effective visual communication of data (Few) [6]***

Found through snowballing from [7].

This book presents a comprehensive guide for designing effective visual displays. Central to this book is the mention of thirteen common mistakes in dashboard design along with other recommendations for colors, layout and other aspects of dashboard design.

One point Few makes is that in designing a dashboard you should always try to eliminate non-data pixels. If any space is used for non-data you should try to redesign the dashboard to eliminate this redundant space. Another point Few makes both in the definition of a dashboard and also in a dedicated section is that dashboards should fit on a single screen as to make sure the user has all the information in front of them without having to switch to somewhere else. This extends to scrolling; the user should never have to scroll for any information and as a designer you should not expect the user to see anything they have to scroll for.

Simplification and the removal of redundancies are also central to Few's design approach, enhancing the dashboard's visual appeal and reducing cognitive load. This makes the dashboard not only easier on the eyes but also more efficient for the user to interpret.

Context is another essential aspect covered in the book. Few points out that while it's beneficial to include various data points, their utility is limited if the user does not understand what they represent. Each piece of data should be clearly defined and its relevance explained, ensuring users grasp the full context of what they are viewing.

The appropriate use of colors is also discussed. Few recommends using colors to highlight significant information but also warns against excessive use. The goal is to draw attention to key data without overwhelming the user with too much visual stimulation.

Overall the book provides very good and approachable tips for designing a dashboard. However, Few's strong focus on simplicity can sometimes be a limitation. By prioritizing a clean and straightforward design, important details and context that could aid in decision-making might be left out. This minimalist style, while often making the dashboard neat and attractive, may overlook deeper insights that are crucial for a complete understanding. This is important to keep in mind when designing to balance making the dashboard easy to use and ensuring it provides enough detailed information.

**‘Single or Multi-page Learning Analytics Dashboards? Relationships Between Teachers’ Cognitive Load and Visualisation Literacy’ (Pozdniakov *et al.*) [8]**

This paper presents a study that examines the impact of single and multi-page LADs on the cognitive load of lecturers, considering their visualization literacy (VL). Through an experiment including 23 lecturers using eye-tracking technology, the research finds that lecturers with lower VL experience significantly higher cognitive loads with multi-page dashboards compared to single-page ones while participants with higher VL experience little impact with a very slight preference towards single-page dashboards. This suggests that dashboard design should consider the users VL and the paper advocates for single-page dashboards in scenarios with high monitoring loads, especially for users with lower VL, to allow for easier data processing and interpretation.

However, the study does not address the potential drawbacks of cramming information onto a single screen, which could also complicate data interpretation and negate the benefits of a simplified layout from having a single-screen design. There is no evaluation of whether a more crowded single-page dashboard could be as challenging as navigating multiple pages, an oversight that suggests the need for a balanced approach in dashboard design. The findings support single-page dashboards to aid users with lower VL under high monitoring loads, but the implications of dashboard density on usability is not explored.

**Summary**

In the design of dashboards, it is crucial that the most important information is immediately noticeable to users[6]. Ideally, dashboards should strive toward a single-page layout to minimize the need for navigation, although multiple pages may be necessary for different use cases, accommodating diverse and complex data[5, 8]. Drilldown features are particularly valuable for allowing detailed examination of data, such as enabling educators to focus on students who need more attention[5, 9]. Simplicity in visualization, using colors to rapidly convey insights, is crucial[6]. It is also important to provide hints and contextual help so users understand what they are looking at. Any content that requires scrolling should be considered optional and only supplementary, as it cannot be expected to be seen by all users[6]. The dashboard should limit the amount of information presented at once to make use of the visual working memory effectively, favoring graphical visualizations over numerical data where possible[6].

### 3.3.2 Dashboard Implementations

#### **‘Perceiving Learning at a Glance: A Systematic Literature Review of Learning Dashboard Research’ (Schwendimann *et al.*) [10]**

This paper provides a systematic literature review of LADs in educational settings with a focus on the representation and usefulness of data to various stakeholders. It identifies that LADs are commonly used in university contexts and primarily feature simple visualizations such as bar charts, line graphs, and tables. These choices reflect a preference for straightforward, easily interpretable formats. The review notes that most LADs rely on single data sources from systems like Moodle, though there’s a trend towards incorporating diverse data types. However, a significant gap is highlighted in the evaluation phase, with many dashboards lacking thorough, real-world testing to confirm their effectiveness. The paper concludes with a call for more comprehensive designs and robust evaluations to ensure that LADs are both practical and beneficial in genuine educational environments. By providing a simple overview of the most common visualization techniques this paper guides the choice of what visualisations to focus on.

#### **‘Educational Dashboards for Smart Learning: Review of Case Studies’ (Yoo *et al.*) [7]**

This paper is a review of 10 LADs, assessing their design, implementation, and effectiveness in enhancing learning through data visualization. To assess the dashboard, the paper presents a conceptual model for evaluating dashboards based on principles by Few and Kirkpatrick’s model. It reveals that bar plots, line charts, and tables are the most used visualizations for displaying real-time student performance and for tracking at-risk students. These visualization types are chosen for their straightforward interpretability which allows educators to quickly assess and react to student data. This paper also highlights a big gap in thorough real-world testing of the dashboards and expresses a need to integrate and pedagogically impactful visualizations. Overall, the paper highlights the importance of using intuitive visualizations that can be efficiently interpreted and lead to actionable insight.

#### **‘The student activity meter for awareness and self-reflection’ (Govaerts *et al.*) [9]**

This paper describes the development and evaluation of the Student Activity Meter (SAM), focusing on its iterative design process and the effectiveness of its visualizations. SAM gathers data using google Analytics and uses line charts to track time spent on learning resources, bar charts to display resource use, and parallel coordinates to compare student performance across multiple metrics. These visualizations helps identifying at-risk students and enable comparisons among students. It incorporates features like filtering and binning to drill down and diagnose specific challenges that the students face. The effectiveness of the visualizations was confirmed through evaluations over four iterations, which shows their utility in improving awareness and self-regulation. This iterative process and the choice of visualizations provide a valuable framework for developing and improving a LAD with specific importance on being adaptable to user needs.

**'A qualitative evaluation of evolution of a learning analytics tool' (Ali *et al.*) [11]**

This paper describes the iterative development and qualitative evaluation of LOCO-Analyst, which is a LAD to provide lecturers with visual feedback and insight on student activities and performance. The dashboard uses bar charts to effectively present the frequency and timing of interactions with learning content, making it easier for lecturers to track student engagement. It also uses color coding to help in quickly identifying different levels of activity. The evaluation revealed that some participants found the visualizations difficult to understand; we argue that the use of 3D bar plots might be a contributing factor in the confusion. As Few puts it "3D should always be avoided when the added dimension of depth doesn't represent actual data" [6, p. 106]. This is a great example of an iterative development and evaluation of a LAD, demonstrates how effective use of feedback can be used to improve a dashboard iteratively.

**'How did the e-learning session go? The Student Inspector.' (Scheuer and Zinn) [12]**

This paper outlines the development and evaluation of "The Student Inspector". The tool analyses log data gathered from e-learning activities, analyses the logs using AI, and visualizes the results to teachers to compensate for a lack of one-on-one tutoring caused by distance learning. The paper displays student performance through bar charts, misconceptions about topics as discussed in forums as a pie chart, and topic coverage through a combination of a bar chart and a table. The evaluation reveals that the visualizations used are intuitive and beneficial for identifying student needs and adapting teaching.

**'GLASS: a learning analytics visualization tool' (Leony *et al.*) [13]**

This paper presents GLASS, a web-based customizable visualization platform to support educators in analyzing data. The GLASS platform stands out for being based on a modular system where the lecturer can customize what data is used and how it is shown including filters. Most prominently is a frequency timeline to display the number of interactions. Additionally, there is a bar plot to indicate what category of activity was done. The paper mentions that they performed user tests and that the indications they received from the participants is that the visualizations and system as a whole needs to be intuitive to be usable. This system is a good demonstration of how customizability and filtering can be used in a LAD.

**'CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses' (Mazza and Dimitrova) [14]**

Found through snowballing from [11].

This paper discusses the design and evaluation of CourseVis, a tool to monitor online courses by extracting raw data from a LMS. CourseVis uses both 2D and 3D discussion plot to visualize discussion engagement. Additionally, they use a cognitive matrix to show student performance across various concepts. The evaluation showed that the participants preferred simpler, clearer visualizations and that they found those to be more effective. The conclusion highlights the importance of designing a dashboard that balances detail with usability.



**'Addressing learner issues with StepUp! an evaluation' (Santos *et al.*) [15]**

This paper describes the design and evaluation of "StepUp!" a LAD designed to enhance self-awareness and peer interaction among students by visualizing their educational activities. The dashboard integrates data from various sources like blogs, Twitter, and Toggl to display detailed activity records in a "Big Table" format and uses sparklines and bar charts for depicting weekly trends. This visualization aids students in reflecting on their time management and social media use, encouraging better engagement and study habits. The tool was evaluated through student surveys in actual courses, focusing on its impact on learning and self-regulation. In the evaluation, they received mixed feedback: "StepUp!" improved the awareness of the students, it was not as successful in motivating behavioral change leading to limited perceived utility. However, in the evaluation, they did not ask about the perceived effectiveness of the individual visualizations and thus we cannot know if that was the reason. It can be speculated that the "Big Table" approach might be overwhelming for users as it can be difficult to see trends when presented with a wall of numbers.

**'DDART, a Dynamic Dashboard for Collection, Analysis and Visualization of Activity and Reporting Traces' (Ji *et al.*) [16]**

The DDART paper presents a dynamic dashboard integrated into the Moodle LMS, aimed at enhancing self-regulation and metacognitive skills in learners through customizable data visualizations. The goal of this is to allow learners to monitor and adapt their learning behaviors through these visualizations. The dashboard uses line charts, pie charts, and bar plots via Google Charts<sup>5</sup> to visualize activity and reporting traces. This facilitates diverse analytical needs and learning styles of each student by making the data and the visualizations interactive and customizable. While the authors does not make claim on the effectiveness on the, the paper does not perform an evaluation of the tool and thus the we cannot trust the effectiveness of the tool and instead this is mentioned as future work. This tool highlights the use of selection and filtering as features that can be used to find the right information to best gain insight into the relevant context.

---

<sup>5</sup><https://developers.google.com/chart>

**‘AAT: a tool for accessing and analysing students’ behaviour data in learning systems’ (Graf *et al.*) [17]**

This paper presents the Academic Analytics Tool (AAT), which is a tool to analyse student behaviour data from learning systems. The tool offers a robust SQL-based query builder like user interface to gather data and presents the results in tables. While AAT enables great customizability in what data is included it falls short in visualizing the data. The complete lack of any graphical visualizations makes the data harder to comprehend and understand at a glance which stifles the usability of the otherwise great analysis tool. AAT exemplifies how great analysis tool can have the potential for insight limited by the lack of graphical visualizations. However, it also shows how flexibility and adjustability can be used to aid insight by answering the exact question the user asks.

In addition, the paper does not present any evaluation of the tool so the authors have no claim to the effectiveness of the tool.

**Summary**

Learning analytics dashboards typically focus on visualizing activity rather than direct results, with a significant majority using log data as a primary data source (75% [10]). Common visualization types include bar charts, line graphs, and tables, reflecting a trend towards easily interpretable and familiar designs. These elements are favored because they allow educators to rapidly assess student performance and engagement without the cognitive load of more complex visualizations[7, 9].

Significantly, the literature reveals that while many dashboards employ simple visual designs, there is a varied approach to how data is presented. For example, some dashboards focus on activity metrics, typically visualized through frequency timelines or histograms, to highlight patterns or trends in student engagement[13]. Others provide more detailed drill-down capabilities, allowing users to navigate from general overviews to specific details about student interactions, often using linked visualizations that maintain a cohesion across different data representations[18].

Furthermore, the incorporation of interactive elements like filtering and binning is noted as beneficial for allowing educators to customize views to better target students who require attention[9, 13]. This adaptability in dashboard design not only increases insight but also improves user engagement with the dashboard itself. However, despite these improvements, there is a recurring theme in the literature about the lack of comprehensive evaluation of these dashboards, suggesting a gap between dashboard capabilities and their proven effectiveness in educational settings[10].

### 3.3.3 Dashboard Requirements

#### **‘How Can Learning Analytics Improve a Course?’ (Hui and Farvolden) [19]**

This paper explores the use of learning analytics to enhance classroom and curriculum design in higher education by performing a case study of 69 participants of a CS1 course. The results showed that analyzing data like students’ interaction with the LMS, survey responses, and performance scores could reveal insights into student engagement, comprehension difficulties, and overall course effectiveness. For example, the data helped identify which course components (like assignments and midterms) significantly predicted students’ success on final exams and in the course. These insights can guide instructors on which areas of the course might require adjustments to improve educational outcomes and student engagement.

Of significant importance to this thesis is their outline of questions raised related to the needs of instructors [19, p. 3]. This formed a basis to the questions expected from the lecturers of CSAMS. The most important questions are:

- What are the current grades for each course component?
- Which concepts are generally more difficult?
- Do I need to spend more time on this material?
- Which students are doing well? Which are at risk of failing?
- What is the learning trajectory of the student?
- Are there any questions on the exam that is too easy or too hard?

#### **‘Getting to Know Your Student in Distance Learning Contexts’ (Zinn and Scheuer) [18]**

Found through snowballing from [12].

This is the precursor to [12] where the researchers identify the needs of teachers in remote learning environments.

The study shows that teachers prefer information related to student performance (e.g. success rate, mastery, or skill) and insight into common errors and misconceptions. While the data source differs from that is CSAMS this paper is relevant in forming initial assumptions about lecturer needs.

#### **Summary**

The reviewed papers highlight that learning analytics dashboards should provide detailed insights into student performance, engagement, and comprehension. Essential questions addressed by these dashboards include identifying current grades, difficult concepts, at-risk students, and overall learning trajectories. This information supports educators in making informed decisions to improve course effectiveness and student outcomes.

### 3.3.4 Dashboard Evaluation

#### ‘Learning Analytics Dashboard Applications’ (Verbert *et al.*) [20]

Found through snowballing from [7].

This article examines LADs that help learners and teachers by visualizing educational data. The review of 15 dashboards reveals limited documentation on their impact on learning outcomes, highlighting a need for further longitudinal studies to assess their effectiveness in educational settings.

As part of the review, the article proposes a model for evaluating the effectiveness and educational impact of LADs.

The levels, as described in the paper, are:

1. *Awareness*. This stage is concerned with just data, which can be visualized as activity streams, tabular overviews, or other visualizations
2. *Reflection*. Data in themselves are not very useful. The reflection stage focuses on users’ asking questions and assessing how useful and relevant these are.
3. *Sensemaking*. This stage is concerned with users’ answering the questions identified in the reflection process and the creation of new insights.
4. *Impact*. In the end, the goal is to induce new meaning or change behavior if the user deems it useful to do so.

#### ‘On evaluating information visualization techniques’ (Freitas *et al.*) [21]

The paper establishes criteria for evaluating information visualization, focusing on visual representation usability by assessing completeness, spatial organization, information coding, and the impact of user interactions on visual state transitions. It emphasizes the importance of a logical layout that enables easy location and understanding of data distribution, alongside the effective use of symbols to make the data easier to understand. In addition, the paper analyzes interaction mechanism usability through features that aid orientation, navigation, querying, and data set reduction, ensuring that users can efficiently manipulate and explore data. These criteria aim to improve both the expressiveness and the functional interaction with data visualizations.

While this is a short paper, it outlines many important factors that should be considered both when planning and also evaluating a LAD and is consistent with other guidelines [5, 6].

**'A checklist to guide the planning, designing, implementation, and evaluation of learning analytics dashboards' (Kaliisa *et al.*) [22]**

This paper presents a structured approach to developing a lecturer-facing LAD to ensure that it is effective. The proposed checklist consists of four key steps:

- **Planning**, which includes defining objectives, understanding user needs, and addressing ethical considerations
- **Designing**, focusing on user-centric interface design and appropriate data visualization choices
- **Implementation**, which involves integrating the dashboard into existing systems, providing user training, and conducting pilot tests
- **Evaluation**, which calls for ongoing assessment through both quantitative and qualitative methods to measure impact and user satisfaction.

This list is based on a combination of literature review and prevailing sentiment but the paper does not perform an evaluation. Instead, the paper ends with a call for other researchers to validate and extend the checklist based on empirical studies.

The steps given follow a standard, expected structure, yet this provides value not in the exact steps but in the LAD specific application of each step and gives specific guidance on what the steps mean in the context of LAD development.

**Summary**

Evaluations of dashboards reveal that while they are often successful in providing insights, they frequently fall short in inspiring behavioral change. This is not necessarily a reflection on the dashboards themselves, as influencing human behavior can be inherently challenging[13]. Furthermore, feedback from target users proves invaluable in informing subsequent iterations of the dashboard design[20]. This iterative process of design and feedback helps in refining the utility and user-friendliness of dashboards, thereby enhancing their effectiveness in real-world applications.



## Chapter 4

# Methodology

In this chapter we will present the iterative methodology used to conduct the research and development.

### 4.1 Design Science Research

The methodology for this thesis is grounded in the insights from "Design Science in Information Systems Research" [23]. This paper presents a design science approach in information systems, detailing a set of seven guidelines to for research efforts in this field. The following section outlines these guidelines, which serve as the foundation of our research methodology.

**Design as an Artifact** Design science in information systems research aims to create innovative and purposeful artifacts that address significant organizational problems. These artifacts can take the form of constructs (vocabulary and symbols), models (abstractions and representations), methods (algorithms and practices), and instantiations (implemented and prototype systems) that are rigorously designed and evaluated for their utility in solving identified problems.

**Problem Relevance** The relevance of design science research is anchored in its ability to address pressing and significant business problems through the development of technological solutions. By focusing on problems that matter to businesses and organizations, design science ensures that its contributions are valuable and impactful.

**Design Evaluation** Rigorous evaluation of the designed artifact is essential to demonstrate its utility, quality, and efficacy. Design science research employs a variety of evaluation methods, including observational, analytical, experimental, testing, and descriptive methods, to validate the effectiveness and applicability of the artifact in solving the problem it was designed to address.

**Research Contributions** Design science research contributes to the body of knowledge in three primary ways: through the creation of novel artifacts that solve unsolved problems, by developing new foundations (constructs, models, methods) that enhance the discipline's theoretical base, and through the formulation of new methodologies for evaluating and implementing artifacts.

**Research Rigor** Rigor in design science research involves the application of disciplined, systematic methods in both the creation and evaluation of artifacts. This includes leveraging existing knowledge bases and methodologies to ensure the artifact's design is well-founded and its evaluation is thorough and valid.

**Design as a Search Process** The design process in design science is characterized as a search for effective solutions, utilizing available means to achieve desired ends while satisfying the constraints and requirements of the problem environment. This process is iterative, often involving the exploration of alternative designs and the refinement of solutions based on evaluation feedback.

## 4.2 Design Science Research Methodology

This methodology, inspired by foundational principles from "Design Science in Information Systems Research" [23] (as described in section 4.1) outlines a systematic and iterative approach to develop a technological solution for enhancing learning outcomes through data visualization. It emphasizes continuous refinement based on feedback and evaluation.

### 1. Problem Identification and Motivation:

- 1.1. Define the problem and articulate its significance within an educational context.

### 2. Objectives Definition:

- 2.1. Outline the initial objectives for the solution, focusing on the desired impact on learning outcomes and teaching strategies.

### 3. Design and Development:

- 3.1. Develop the artifact, guided by the initially defined objectives and problem understanding.

### 4. Evaluation:

- 4.1. Employ both qualitative and quantitative methods to evaluate the artifact's effectiveness in achieving its objectives.



### 5. Iteration:

- 5.1. Reflect on the evaluation feedback to refine the artifact. This includes revisiting the **Objectives Definition** to ensure they accurately reflect the evolving understanding of the problem and solution spaces.
- 5.2. Based on refined objectives, proceed with subsequent iterations of Design and Development, and Evaluation, incorporating insights and feedback to progressively enhance the artifact.

### 6. Finalization:

- 6.1. Finalize the artifact design incorporating all feedback and insights gained through iterative development and evaluation.

### 7. Communication:

- 7.1. Document and communicate the research process, findings, and contributions to relevant stakeholders, including both academic and practical implications.

These steps aligns with design science research and allows for dynamic research which ensures that the development is rigorous and that the artifact is relevant and solved the problem.

## 4.3 Validating the Artifact

The validation of the developed artifact incorporates a balanced approach, using both qualitative and quantitative measures to assess its effectiveness, efficiency, and user satisfaction. This dual approach ensures a comprehensive evaluation of the artifact's design, functionality, and impact on educational outcomes.

### 4.3.1 Qualitative Measurements

The qualitative evaluation focuses on the artifact's usability, layout, and user interaction to offering insights into the user experience and the artifact's ability to solve the intended problem.

#### User Testing Methodology

A structured user testing session was organized, involving lecturers with prior experience using CSAMS. The session comprised of:

1. A brief introduction to the goal behind the iteration and what to expect in terms of features.
2. A series of tasks designed to interact with the new features, simulating real-world usage scenarios to assess intuitiveness, ease of use, and the effectiveness of the visualizations in conveying peer-review data insights. In addition, the tasks serve as an opportunity for the participants to get familiar with
3. Semi-structured interviews conducted post-task completion, gathering in-depth feedback on the user experience and suggestions for improvement. The interviews had a set of questions which had to be answered with unplanned follow-up questions based on the participants answers.

## Participant Selection

We selected participants who had prior experience with CSAMS as lecturers. There are four people who fit this requirement. In order to get different perspectives we set the minimum number of participants for an iteration to two. While this is a small number of participants, it represents a large percentage of the potential participants. The choice of two participants allowed us to compare and contrast their responses.

### 4.3.2 Quantitative Measurements

Quantitative metrics were employed to validate the technical performance and functionality of the artifact, focusing on aspects critical to the user experience and the system's efficiency.

We used Largest Contentful Paint (LCP) as the primary quantitative metric to ensure that the artifact performs well. LCP measures how long it takes for the largest content element visible in the viewport to fully render, reflecting the perceived load speed from the user's perspective. According to Walton and Pollard, the limit for what is considered a "good" LCP is 2.5 seconds [24]. Considering that dashboards often involve complex data interactions and visualizations, any value under 2.5 seconds is deemed acceptable.

In the process of gathering LCP we also get some secondary metrics that are useful in getting a more complete picture of the performance of the page load. These are First Contentful Paint (FCP)<sup>1</sup> which indicates the time between when load was initiated until the first content is drawn and Cumulative Layout Shift (CLS)<sup>2</sup> which indicates how much the layout moves as content is loaded and drawn giving a metric for the stability of the layout and thus perceived disorder in loading.

To measure the LCP of the different pages, we utilized Google Chrome's "Performance Insights" tool<sup>3</sup>. The tool was ran five times to ensure a representative sample. From these times, the highest and lowest samples were cut as to not include outliers which could have a drastically different time. Since LCP was considered the primary metric this was the discriminant for cutting out outliers. The remaining three samples were averaged to get the final result.

## 4.4 Deployment

For quantitative testing we needed a consistent and reproducible platform to test execution and page loading times. For this we deployed the system on the OpenStack instance of NTNU Gjøvik called SkyHiGh<sup>4</sup>. The single server instance was using the *gx3.4c8r* flavor with 4 CPU cores and 8GB or memory. The server was running *Ubuntu Server 22.04* and a local instance of MySQL version 8.0.

---

<sup>1</sup><https://web.dev/articles/fcp>

<sup>2</sup><https://web.dev/articles/cls>

<sup>3</sup><https://developer.chrome.com/docs/devtools/performance-insights>

<sup>4</sup><https://www.ntnu.no/wiki/display/skyhigh>

## 4.5 Dataset

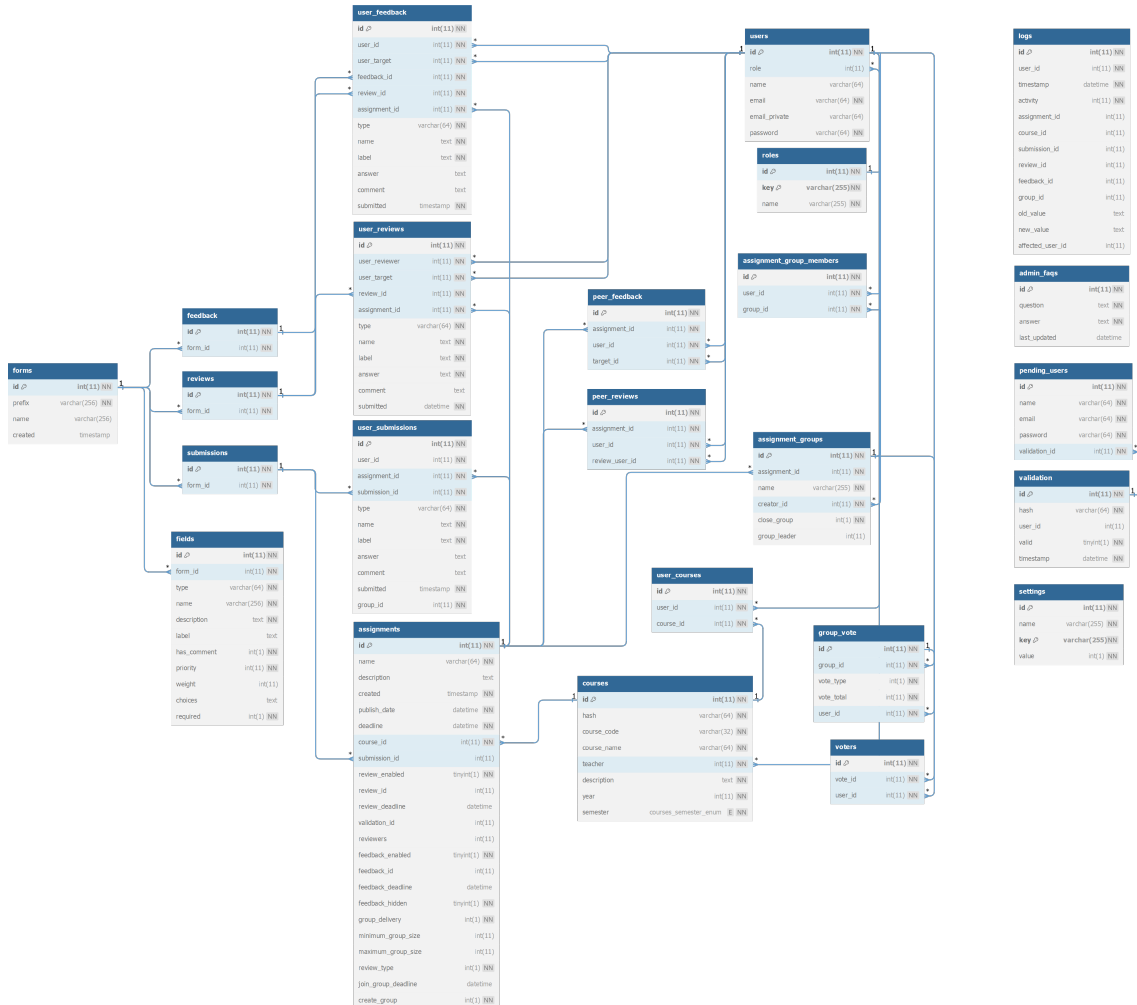


Figure 4.1: Database diagram for CSAMS

While most of the schema is as expected there are a few things to know about it. The first and most important is that a form is not submitted as a whole. As a result of this *user\_reviews*, *user\_submissions*, and *user\_feedback* are not tables to submitted forms but rather each submitted field of a form. There are actually no tables for a submission of form, instead this has to be inferred from the individual fields.

A second thing to know is that the *name* fields of tables *user\_reviews*, *user\_submissions*, and *user\_feedback* can be used to reference the *name* field of the *fields* table. As such the *fields* table includes the fields of all forms of all types (assignment submission, review, feedback). However, since the actual form can change after a submission has been made, there is no foreign key on these fields.

Thirdly, while most of the data relating to a submitted field is duplicated, for the radio field

type the submitted field has to be joined with the *fields* table to get the choices.

To work on development, we were supplied a data dump from last year. This data included two computer science courses for second year students with a focus on programming later referred to as Course 1 and Course 2. These courses had 105 and 40 users respectively with a total user count of 118 (some users were registered to both). The courses have 3 and 2 assignments respectively where the second assignment of Course 1 is group based with an average of 3 group members.

There are a total of 160 unique form fields of which 70 are radio buttons, 33 are checkboxes and 57 are text fields of different types (text, paragraph, URL). For these fields there was 2626 submitted fields for assignment submissions and 9468 submitted fields for reviews. A field is a question of the form. These terms are used interchangeably in this report.

Before we started working with the data, we made sure to anonymize it to protect privacy by removing names and emails. This was an important step to keep the data secure and confidential during our tests and development. The query used to anonymize the data is shown in Appendix C.

## 4.6 Query Benchmarking

All benchmarking was performed on the deployed database as described in section 4.4 with all parameters filled in. Whenever queries were benchmarked for the sake of comparison the parameters were the same as the execution times could differ based on what course or assignment was selected.

To get the execution time of queries we enabled profiling<sup>5</sup> with the query: `SET profiling = 1;`. The process of getting the execution times for queries we used a process similar to that for getting page load metrics; we repeat that here for clarity. The queries were then ran five times to ensure representative sample. From the times gathered from the five executions, the lowest and highest times were removed. The three times were then averaged to get the final execution time.

## 4.7 Success Criteria

‘Design Science in Information Systems Research’ states that "A design artifact is complete and effective when it satisfies the requirements and constraints of the problem it was meant to solve" [23, p. 85].

---

<sup>5</sup><https://dev.mysql.com/doc/refman/8.0/en/show-profile.html>

# Chapter 5

## Results

In this chapter we will present the design, implementation, and evaluation of the prototype dashboard. As described in chapter 4, this work followed an iterative approach where two iterations were performed.

### 5.1 First Iteration

This first iteration was crucial as it kicked off the development of the dashboard. In this iteration, we focus on constructing the prototype that allowed lecturers to (1) identify underperforming topics and (2) individual students who are struggling. Besides these primary goals, this initial iteration was also the start of the development and therefore this iteration also included system design and choices of tools and technologies.

#### 5.1.1 System Design

While the currently planned functionality is not any heavier than the existing functionality there could, in the future, be a need for heavier operations such as interactions with a NLP model which would take up more resources and time. It would therefore be a good addition to add a caching layer to reduce the need to reevaluate the same data when the operations performed are heavy.

To accommodate this potential future need, the analytics functionality was developed as a separate service. Since the services are logically separable with only a small interface between them this approach will ensure separation of concerns in addition to independent scaling of the two services. The system design is outlined in Figure 5.1.

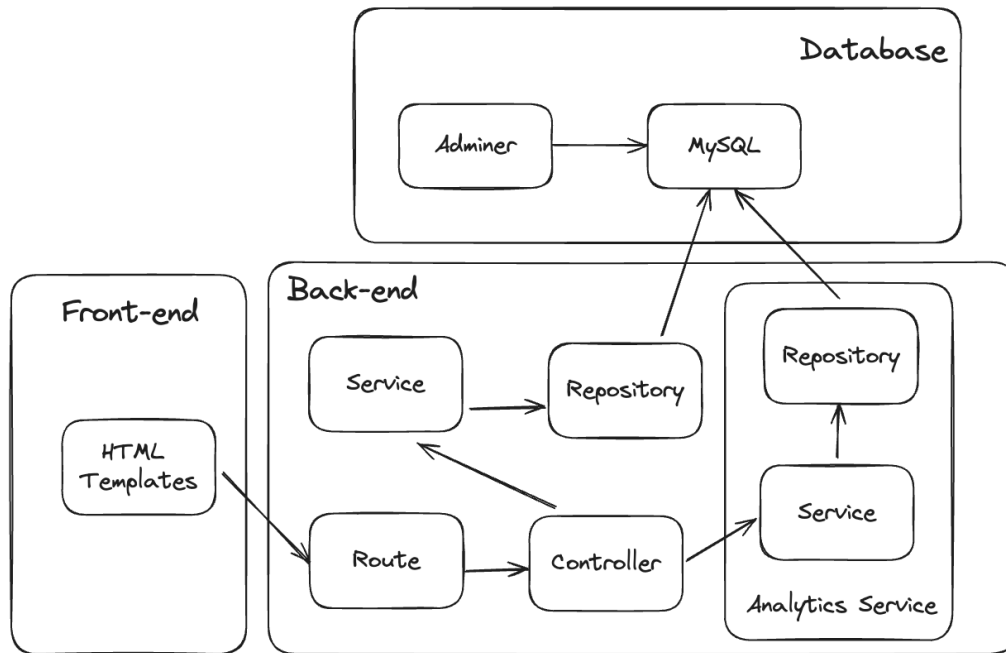


Figure 5.1: System Diagram for CSAMS with Analytics Service

Most of system design is inherited from CSAMS, however, the analytics service was added as a separate group. This also indicated how it is reliant on the existing controller to contact the service. Doing this alleviate the analytics service of having to handle users, instead opting to let the existing controller handle the users with only a simple token to authenticate requests between the controller and the analytics service. The data received from the analytics service is given as JSON which the controller will then render as HTML.

### 5.1.2 UI Design

Three screens were designed to accomplish the objectives of this iteration. Two of these were related to the an assignment and one related to a course. This follows a multi page layout pattern as outlined in [5].

#### Assignment - Group Overview

The first screen addresses the first objective of this iteration by providing an overview of how students reviewed each question/field of a peer-review form.

On the left half is an overview of the review form fields with a box plot showing the average and standard deviation along with a dashed line from the lowest to the highest values. In addition, there are colors to add an extra dimension to quickly identify the average with a traffic light color scheme. Below that is the same data presented in a table. This was done to supplement the bar plot with more accurate numerical data if that is desired. Still, careful consideration was made to making sure the most important element (i.e. the box plot) was the most prevalent on the screen.

On the right there are details of an individual field. There is a bar plot to show individual review scores for each student sorted by performance. This was mentioned in the preliminary investigation as a way to inspect the data for rapid changes / cutoffs to see if there are any performance boundaries.

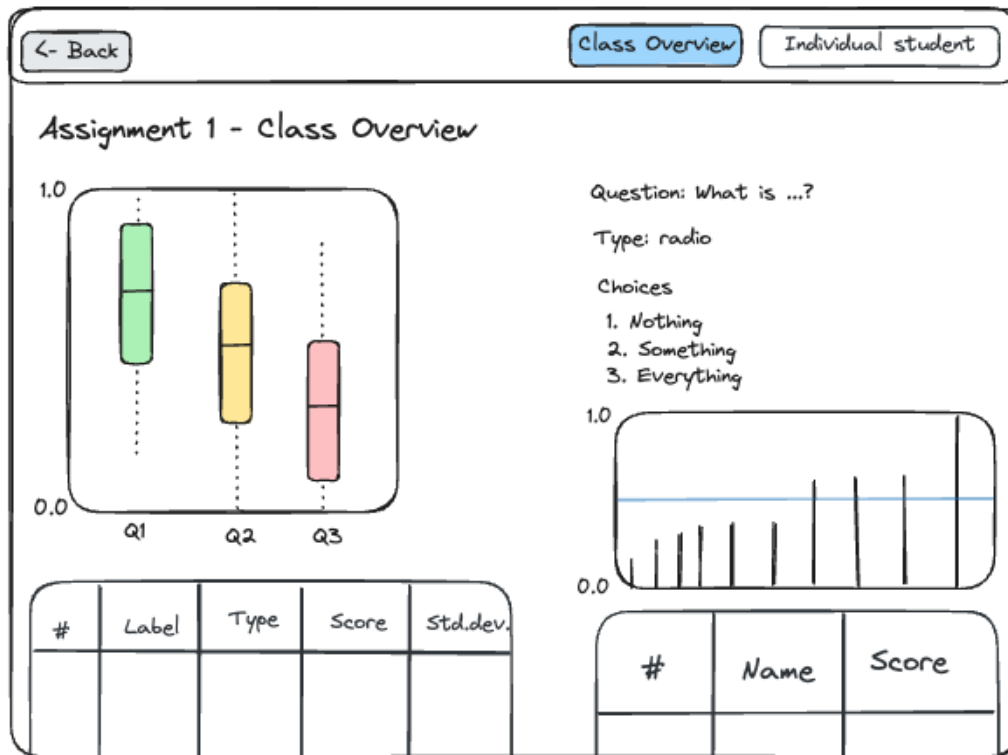


Figure 5.2: Assignment - Class Overview, Design

This split screen approach was done to maintain the context of the overall while simultaneously seeing a more detailed view and in that way being able to connect the granular details with the coarse details.

Before a single item has been selected to be shown on the right side of the screen that area is used to show instructions about the information shown and how to interact with the dashboard.

### Assignment - Individual Students

The second screen addresses the second objective of this iteration by providing an overview of the average of the received scores on the peer-reviews of each individual student.

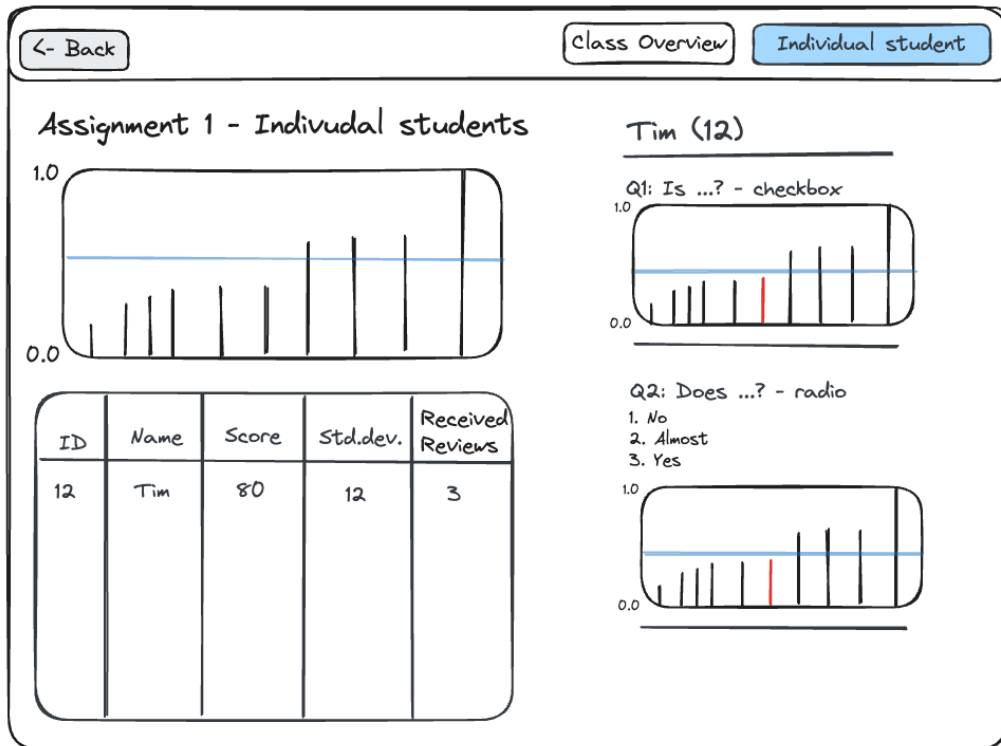


Figure 5.3: Assignment - Individual Students, Design

This screen again follows the split screen design with the left side featuring a bar plot of student performance. The bar plot is sorted by score for two reasons: (1) It has to be ordered on something and ordering on student ID gives little value, (2) gives insight into performance drop-offs and thresholds. In addition, the average value is added as a horizontal line to easily identify this threshold. The same information that can be seen in the bar plot can be seen in the table below which also functions as the way to interact with the dashboard and can be clicked to select the student to view on the right. The table also has the number of reviews received. This was added to the table for two reasons: (1) to know the basis for the data, and (2) identify students who have received too few reviews.

On the right is the view of the individual student with a list of each question, the type of question (paragraph, checkbox, radio), choices (if radio), and a bar plot of the student performance for that question with the selected student highlighted.

Like with the groups overview screen the right side is used for instructions before anything is selected.



## Course Overview

This screen is meant to show the student performance as the the course progresses to see performance issues cross-assignment. This screen is also related to goal of identifying individual students who are struggling but from a course perspective instead of assignment. On the left of the screen is a sankey diagram to show student performance binned into high, medium, and low performance in addition to showing those who are signed up for the class but has not submitted ("None"). On the right side of the screen is a table to show more granular information about each student the score for each of the assignments. In addition, there is an indicator for all but the first assignment to show the trend. This indicator is an arrow using both direction and color to show the trend with red and down arrow for negative trend, green up arrow for positive trend, and black dash for no change (<5%).

In addition to being a graphical representation, the sankey also functions as a filter for the table. By clicking either nodes or links in the sankey, the table will show only the relevant students along with an indicator above the table to give context. This will allow the user to drill down to more quickly find students that might be on a downward trajectory.

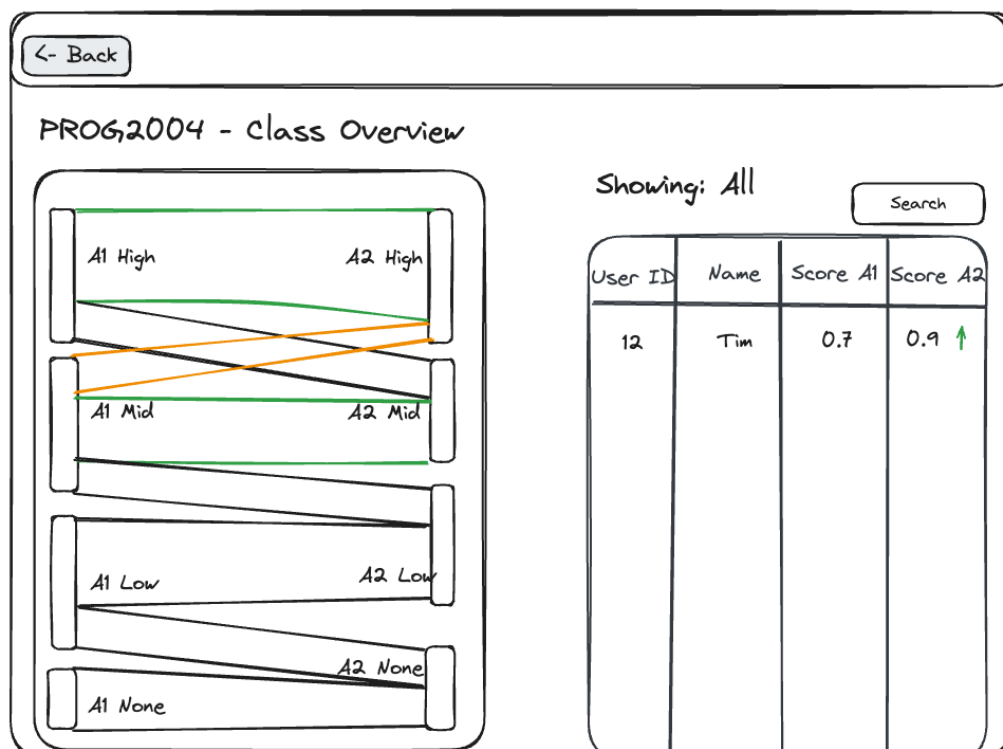


Figure 5.4: Course Overview, Design

### 5.1.3 Development

#### Analytics service

To develop the analytics service, we continued development in Golang (Go) to maintain consistency in the project.

The repository of the analytics service was developed using SQLC<sup>1</sup>. This is a code generation tool for create type-safe Go functions from a schema and a set of queries. SQLC generates type definitions (in Go: struct) based on the provided schema and functions to query the database and parse the results into a custom struct representing the unique set of columns returned in a query.

The web server for the analytics service was written using the Gin web framework. This choice is rather arbitrary, any web server would be sufficient since the routes are straightforward and was chosen because that is the web server we are most familiar with to ensure swift development. While the choice was one of preference, this is not to say that the choice is random since Gin is the Go router with the most stars on GitHub and has an extensive library of community written modules.

A client package was developed for the web service to interact with the analytics service. The package consisted primarily of a Client with a method for each endpoint that returns the parsed data and potentially an error. In addition to providing a clean interface, this approach consolidated the request generation to ensure that tokens were added to all requests and that any potential errors were handled and reported consistently. This package greatly benefited from the use of Go as the original struct declarations in the analytics service greatly simplifies the decoding of the returned JSON data.

#### Database Optimization

While getting familiar with CSAMS we observed that the existing submission overview page took multiple seconds to load. Because of the responsiveness requirements it was obvious that this had to be addressed for more complex queries and there was a need for a different approach to data aggregation.

The current approach places all the logic on the server by first fetching a single element and then fetching more elements based on that. An example of this is that to calculate the average score per user the application will get all the students of a course and then loop over the students to get the received reviews for that student. Additionally, if the assignment is group based, the application will fetch all the submitter in the group and then the result. This results in a huge multiplication of the latency between the server and the database along with having to wait for sequential executions. The solution to this is through the use of database procedures, better use of table joins, and the introduction of Common Table Expressions ("with" clause) for the queries to deal with the more complex logic without affecting the data model.

The procedure is a translation from the Go function currently used to calculate the score of quantitative fields. It is responsible for calculating a normalized score for a single peer-review field.

---

<sup>1</sup><https://sqlc.dev>

```
1 calculateScore(field):  
2     if field.type == 'checkbox':  
3         return field.value == 'on' ? 1:0  
4     else if field.type == 'radio':  
5         return field.value / field.choices.length  
6     return null
```

Figure 5.5: Pseudocode of Score Calculation Logic

These changes takes the response time for getting an assignment overview from 2.8 seconds to a 350 milliseconds (i.e. from not acceptable to acceptable) and was used in all queries that calculated averages and standard deviation along with the aggregation functions *AVG* and *STDDEV\_POP*. It is worth noting that it is not the calculation of the scores using the procedure that makes the query faster but rather that it enables us to use the aggregation functions which reduces the number of queries we have to execute and the amount of data transferred.

### Choosing a Visualization Tool

Given the overall system design, there was a need for a tool to make visualizations for out CSAMS. We briefly reviewed several JavaScript diagram libraries, with D3.js, Plotly.js, Chart.js and Google Charts being among the most popular. We will refer to these libraries without the .js suffix for clarity. To select the most fitting library, it was crucial to consider what requirements we have for flexibility and extendability. Although we do not anticipate needing many different visualizations and expect to use mostly simple diagrams, the ability to use more complex visualizations is important to support future needs. Additionally, to avoid a future redesign, the chosen library should support the all the currently planned diagrams.

Based on this initial review, D3 was chosen as the most appropriate tool for out needs. It offers the greatest flexibility and extendability ad allows for the creation of virtually any type of visualization imaginable from basic charts to complexity interactive diagrams.

While D3 has a steeper learning curve compared to other libraries like Plotly and Chart, its comprehensiveness justifies the learning and development time. This library not only meets our current requirements but also provides the robust framework necessary for any future expansion of visualization needs. In addition, D3 is a very popular library with over 100,000 start on GitHub and a large community that offers many extensions to the core library including a sankey diagram extension.

Other libraries such as Plotly, Chart, and Google Charts offer advantages in ease of use and quick setup for more standardized visualizations. However, they do not match the level of detailed control and customization that D3 offers. Plotly and Chart, while user-friendly, offer limited extendability in comparison, and Google Charts, though providing a range of ready-to-use charts, falls short in terms of customization and flexibility.

## Interactiveness

As per the design, the dashboard has an interactive component that switches element on screen without a full page reload. This required extending the existing rendering functionality ("view" package) with support for rendering individual components as opposed to the current functionality which only supports rendering full pages.

In addition there was a need for functionality to make requests and switch the contents of HTML elements. This could be accomplished with custom JavaScript but instead HTMX<sup>2</sup> was chosen for this purpose to boost productivity, increase extensibility and provide a concise way to define content switching. HTMX is a JavaScript library that extends HTML with extra attributes to annotate AJAX requests and element switching information. HTMX supports various triggering mechanisms to determine when a request should be sent. Among these is the "reveal" trigger which only sends the given request when the element is shown on screen. This trigger was used for the right side of individual student overview for a single assignment (Figure 5.3) to only send request for the data for the bar plot when the container element was in view. In addition, the "load" trigger was used to request the sankey diagram data when the page loaded to reduce the initial page load time to then show a loading indicator.

Here is a snippet that sends a request when revealed and replaces itself with the response of the request.

```
1 <div hx-get="/admin/fields/{{ .FieldName }}/students/{{ .UserID }}"
2     hx-trigger="revealed"
3     hx-target="this"
4     hx-swap="outerHTML"
5 ></div>
```

---

<sup>2</sup><https://htmx.org/>

## Implementation Results

The implementation of the screens was fairly straight forward with the dependencies already present in CSAMS which include JQuery and Bootstrap 4 including Bootstrap Table<sup>3</sup>. Bootstrap Table allows for a more concise implementation of search and sorting functionality in tables.

The interactions of the dashboard were carried out through a combination of HTMX when new data is needed (described in section 5.1.3) and JavaScript when the client already has the data and it just needs to be transformed or filtered.

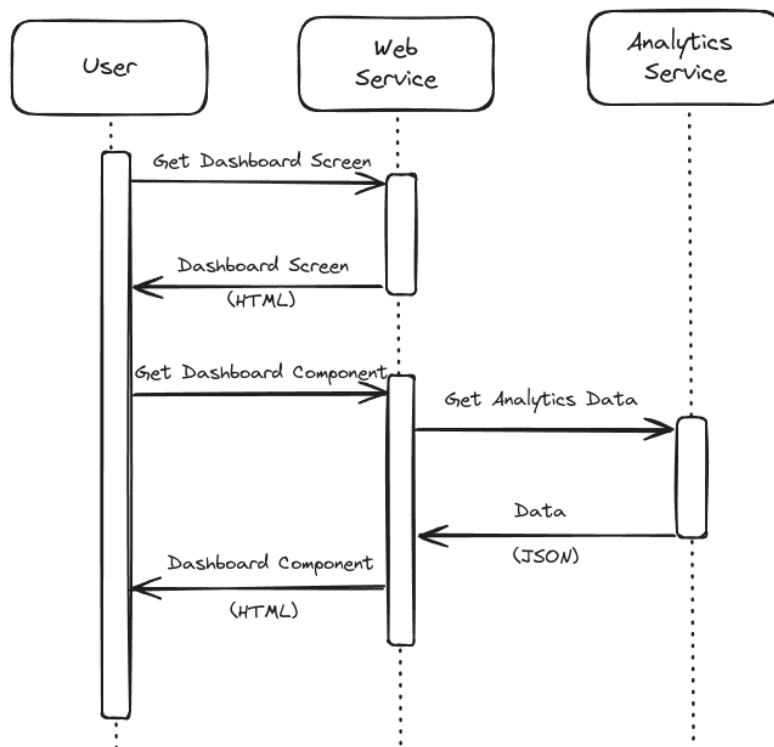


Figure 5.6: Sequence Diagram for Requesting Dashboard Page

As seen in the sequence diagram above, the web service will first send a page without any analytics to reduce the time to first load. This will then trigger the loading of a component by HTMX attributes and the web service will request the analytics data from the analytics service. The resulting data will be embedded into a Go HTML template along with the JavaScript required to generate the diagrams on the front-end.

<sup>3</sup><https://bootstrap-table.com>

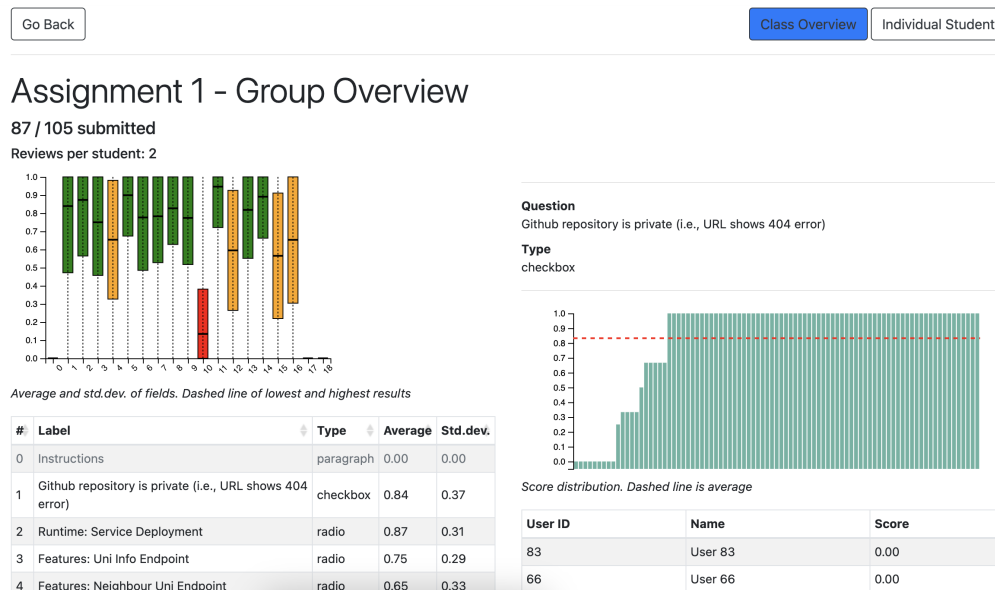


Figure 5.7: Assignment - Class Overview, Implementation

Changes from the design is that fields that cannot be evaluated (i.e. "paragraph" type) has a muted color to signify that they are not important. They can still be clicked and they work the same as other fields but they will always have an average and standard deviation of 0.

The colors of the boxes in the box plot uses a traffic light color scheme and the choice of the color is base on a simple threshold of the average value. Values under 0.25 are red, values over 0.75 are green and the rest (the middle) are yellow.

Interactiveness was implemented with HTMX to get the data for the right side and then swap the content with the selected question when clicking either a box in the box plot or a row in the table. The box plot was generated with D3 which generates HTML elements where HTMX attributes are added to make the diagram interactive in the same way as the table. Since HTMX parses the content as it receives it from a request and the elements of the box plot was generated with JavaScript we have to manually trigger HTMX to parse the elements in the box plot after the diagram has been generated.

The bar plot on the right includes a brush feature to filter the table underneath it. This can be used to single out groups of students and investigate thresholds that appear in the diagram.

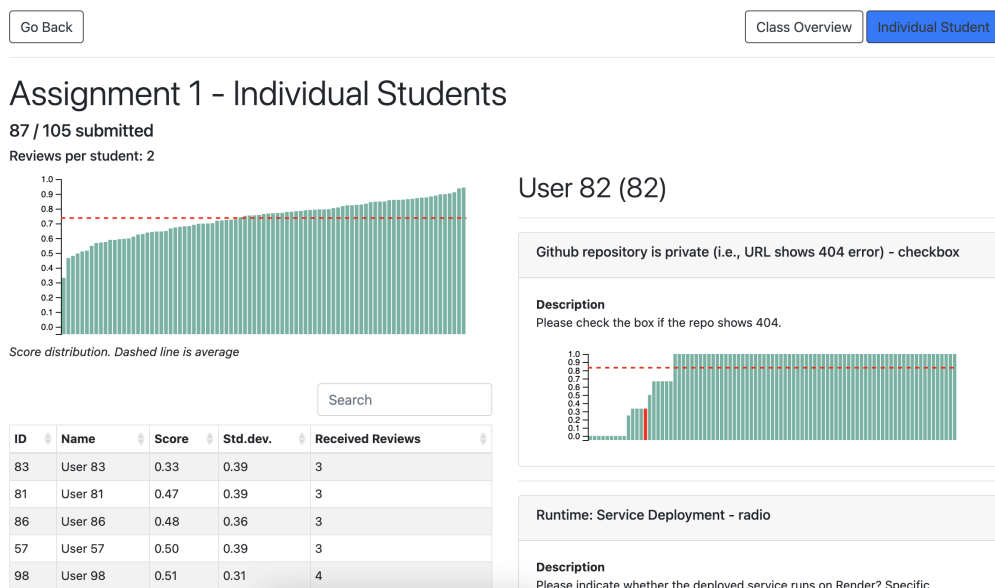


Figure 5.8: Assignment - Individual Student, Implementation

There has been added HTMX attributes to the table to swap out the right side to view a single students performance.

To be able to link to this screen from other screens, a URL parameter was added to be able to immediately include a student for the right side without having to click in the table; this was used in the sankey table.

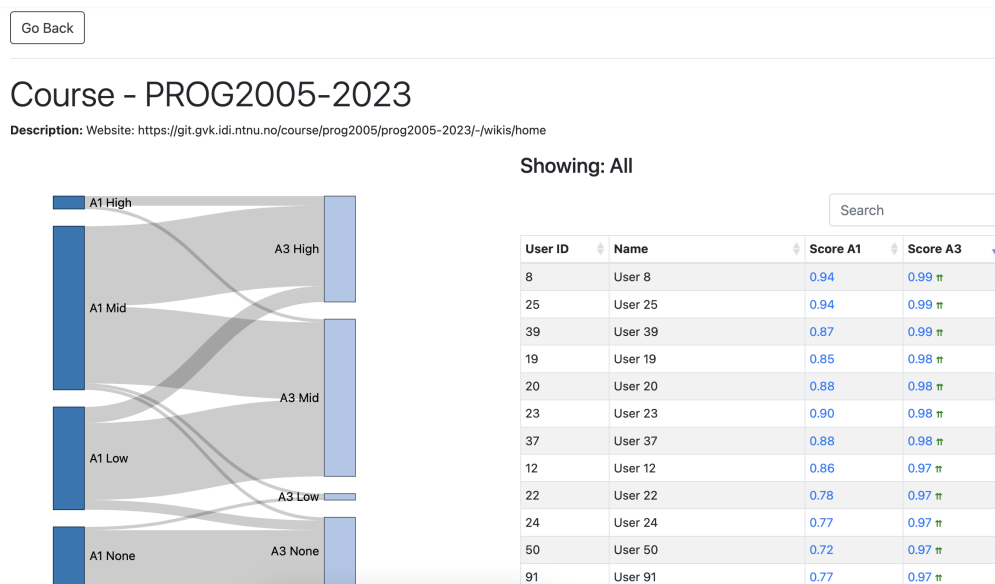


Figure 5.9: Course Overview, Implementation

Because of the limited usability of aggregation functions in MySQL and to increase the

usability of the data, the rows from the database are given in the format [user\_id, user\_name, assignment\_id, score]. These are then grouped by assignment\_id and binned using JavaScript to generate the nodes and links with weights in the format that D3 requires to generate the sankey. Additionally, they are grouped by the user\_id to generate the table. When filtering by clicking the sankey, the entire result set is filtered for appropriate results. When filtering for nodes only the given assignment and weight is considered but when a link is being filtered by both the source and target assignments of the node was considered. The functions are designed to work with any number of assignments however there were no courses with more than two assignments with peer-review in the dataset so this has not been tested.

It was observed after the implementation that there was a discrepancy between the compactness of the designed screens and the implemented screens which lead to a greater amount of white-space than intended. According to Few [6]: in designing a dashboard, the designer should strive for reducing the number of non-data pixels. Therefore, this consideration will be carried over into the next iteration.

#### 5.1.4 Evaluation

For the evaluation of the prototype in the first iteration we got three participants to take part in interviews of 45 minutes.

A tool is only useful if you can find it. Therefore, ensuring that the participants could find the screens was a the first step of every section of tasks the participants did. After navigating to a dashboard page the participants were asked to interpret the page without further input to get a first impression. Then, the participants were asked to explore the page and explain their actions and thoughts.

#### Tasks

1. Assignment - Group Overview
  - a. Navigate to the dashboard for "Assignment 1" of Course 1
  - b. Can you based on what you see on this screen identify any problems on a group level?
2. Assignment - Individual Student
  - a. Navigate to the dashboard to see individual students (still for "Assignment 1" of Course 1)
  - b. Can you based on what you see on this screen identify any individual students who received poor feedback?
3. Course Overview
  - a. Navigate to the dashboard for Course 1
  - b. Can you based on what you see on this screen identify any students or groups of students that raise any alarms for you?



## Questions

The questions asked in the post-task interview were inspired by the System Usability Scale (SUS)[25]. We opted to skew away from the SUS and the Likert scale to get more open feedback similar to what was done in [9]. This was done to allow for more open and relevant feedback. Since there were only three participants there was no need for data analysis to understand the results and therefore the questions were open ended to allow for greater flexibility.

A part of the evaluation was to inform the objectives of the next iteration. This was both to understand the problems with the current functionality and to understand what other objectives the lecturers wished to accomplish with the dashboard.

Here are the questions that were planned to evaluate the prototype and plan the next iteration.

1. What parts of the prototype did you find to be especially usable?
2. Was there any parts of the prototype that you experiences as confusing or difficult?
3. Were there anything that did not work as expected?
4. Was the layout of the pages intuitive and understandable? (inquiring about the course and two assignment pages individually)
5. Did you find the navigation to be intuitive?
6. Are there any additional objectives you want to accomplish with the dashboard?
7. Do you have any suggestions for improvements to the current functionality that would make the dashboard more valuable and usable to you?
8. Did you find the dashboard to be responsive?
9. If this dashboard was available in CSAMS, how likely are you to use it?
10. Any comments you would like to share about the dashboard?

For each question asked there were follow-up questions conversations to understand the full context of the participants opinion.

## Observations - Tasks

Overall the participants managed the tasks with minimal help. However, we made some observations during the tasks that we will present here.

### Navigating to the dashboard pages

An enduring problem with navigation in CSAMS is that there are three screens where the user can see a list of assignments but only one is up-to-date the others are multiple missing features; the same goes for courses. Therefore, we decided to continue the trend of only updating one of the lists. However, this was difficult for users and it was observed that each participant chose a different screen for navigating. While they all found a variant of the list of assignments where the correct assignment was present, they all landed on the wrong version. A slight correction was therefore needed for the participants to navigate to the correct version of the list of assignments. This was seen as a systematic problem with CSAMS and has very low priority for this project although it should be noted as a significant problem that impacts the overall usability of CSAMS.

### **Navigating within the assignment page**

The navigation between individual and class overview for assignments (Figure 5.2) was a new way of navigating within CSAMS and was not immediately obvious to some participants. The buttons are purposely out of view as to not be in the way but this also made it not immediately obvious to the participants and had to be pointed out to some. While this was commented by the participants as being a logical way of navigating, it demonstrates that this new way of navigating might require more work either in finding a new way of navigating to remove this system, making it stand out more, or add some instructions.

### **Observations - Questions**

#### **What parts of the prototype did you find to be especially usable?**

The sankey diagram stood out as particularly intuitive and praised for its effectiveness in identifying groups of students. Additionally, using the sankey as a filter was found to be an effective way to interact with the data making the sankey and accompanying table overall a very intuitive component.

The bar plots and box plots was also noted as being very straightforward and understandable visualizations for distributions, comparison, and, in the case of the bar plot, trends. Overall, the graphical visualizations were found to be more usable than the tables. While initially the tables were more intuitive, the graphical visualizations were later commented to be more usable.

#### **Was there any parts of the prototype that you experiences as confusing or difficult?**

Initial understanding of what the diagrams presented.

This was also observed in the tasks where participants initially looked at table to understand and only secondarily to the diagrams. After a while the participants understood the diagrams and at that point they found the diagram to be valuable. This indicated the need for further explanation of what was displayed. This should not take up much space and should be on-demand.

On participant found it to be unclear what could be clicked and not and thus being reliant on discovery i.e. clicking to see if anything happened. This was surprising since the dashboard already had hover effects to indicate with mouse cursor and color change to indicate something as clickable. Regardless, this goes along with the other feedback that indicates a lack of instruction.

#### **Were there anything that did not work as expected?**

After having used the dashboard and getting familiar with the diagrams and tables the participants found that the things that were implements were as expected with only a few bugs . The table on the right side of the group overview was missing sorting and searching. There is a lot more work making the table searchable and sortable and this table was missed in implementation. This demonstrates how important it is to have these mechanisms on a table. Since tables are less efficient visualizations than diagrams it is more important to have mechanisms to make navigating the table easier.

**Was the layout of the pages intuitive and understandable? (inquiring about the course and two assignment pages individually)**

As observed in the tasks, the first page had a steep learning curve but then the participants quickly understood the other pages.

The assignment group overview screen was found to be understandable. However, being the first screen shown, this was the one the participants struggled the most with. The participants understood the relation between the overview on the left and details on the right side.

After seeing the group overview, the assignment individual student screen was reported to be very easily understandable.

Being the most complex, the sankey did take some time to understand, especially here it's relation to the right. However, being the third time seeing the split screen design, the participants quickly understood the relation. One participant reported the expectation of a relationship before the relationship was understood.

Overall, the participants found the layout to be understandable. However, they again wished they had more instruction on the relation between the right and left sides.

**Did you find the navigation to be intuitive?**

The participants reported that the navigation to the dashboard was consistent with the existing navigation. However, as observed in the tasks, there are many ways to get to the lists of assignments and courses and this caused confusion in the tasks and this was also given as a response to this question.

**Are there any additional objectives you want to accomplish with the dashboard?**

Overall, the participants wanted more ability to drill in to issues and through various comments suggested the need for more levels and stronger linking. This includes both identifying the issued and then seeing what caused the issue. One of the points provided was: after seeing that a student has received poor reviews they want to see the reviews. Also, this means that values should link to a more detailed explanation or somewhere to get more info.

The participants also showed a strong interest in the standard deviation of reviews. In follow-up questions this was indicated as being used as a proxy for review quality and to determine how much agreement there was on a question. Following this, the participants expressed an interest in being able to see on what questions the reviewers disagreed since this would require manual intervention to edit reviews. One participant also suggested a tiered system to evaluate the quality of a radio button with comments enabled: (1) missing / no review, (2) review without comment, and (3) review with comment.

On the discussion of review quality and agreement, a participants mentioned that a review can be empty as in a student did not click any option of a radio button. Sometimes a student has given a comment but not clicked a button, this is clearly a mistake and requires attention from the lecturer. Based on this the participants expressed a wish to get an overview of what reviews were missing.

In addition, the participants requested insight into how many reviews the participants had done in addition to the current table column indicating how many reviews the participant received.

**Do you have any suggestions for improvements to the current functionality that would make the dashboard more valuable and usable to you?**

The participants all wanted more transparency in what the thresholds for values in box plot colors and sankey groups. In addition, they wanted to be able to change this. One participant also suggested the use of cluster analysis or percentiles to determine an initial value for the threshold and then having it customizable from there.

A participant wanted more information in the sankey table and suggested using a tiered table header to group columns by assignment and then giving score, standard deviation, number of received reviews, and number of given reviews as sub-columns.

As stated earlier, the participants also wanted more instruction and hints on what the diagrams were and how to interpret the data. This can be done either through static text or on-demand tool-tips on hover or click.

One participant also mentioned an interest in using color not directly to represent the received feedback but rather using it to indicate what required manual intervention from the lecturer. While this manifests itself in mostly the same way it indicates that the use of color for students that perform well might not provide valuable insight and instead be a distraction. Based on the feedback on the box plot this was not an issue there but it was used in the design choices for future use of color.

**Did you find the dashboard to be responsive?**

All participants agreed that the dashboard was as responsive as could be expected and no issues were reported.

**If this dashboard was available in CSAMS, how likely are you to use it?**

All participants said they would use the dashboard if it was available with the responses ranging from a simple yes to absolutely. Additionally, some participants noted that while they would be using the dashboard whenever appropriate that usually means once per assignment to check that everything is going as it should and that the ability to pick the system back up after a month or two of not using it is an important factor.

**Any comments you would like to share about the dashboard?**

None of the participants had anything to add.

### 5.1.5 Quantitative Evaluation

In evaluating the quantitative performance of the different dashboards screens we measured the LCP, FCP, and CLS for the different screens.

Screen	FCP (seconds)	LCP (seconds)	CLS
Course	0.35	0.72	0.008
Class	0.86	0.86	0.053
Individual	1.15	1.15	0.049

**Table 5.1:** Quantitative Measurements

These are very good numbers and are all well within the thresholds for good performance. This confirms the observations made by the participants that the dashboard is more than fast enough with no performance issues.

### 5.1.6 Conclusion - First Iteration

In the current iteration of the dashboard, the prototype functioned largely as anticipated with only minor bugs that did not hinder the overall usability. However, a critical need for additional guidance in the form of tool-tips and usage instruction was identified.

The feedback from the evaluation highlighted several areas of interest that should be prioritized in future iterations. Notably, participants expressed a great interest in the standard deviation of values and using that as a proxy to assess reviewer agreement and review quality. Furthermore, the importance of making tables sortable and searchable was an important factor in the usability of tables and this should be considered when designing tables in future iterations. Additionally, enhancing the dashboard with more links to related data, such as list of reviews given and received, was suggested as a way to make for an easier time drilling-in to information. This should be added even if that what is linked to has not been improved with visualization i.e. existing overviews.

The participants conveyed a need for functionalities that addresses the completeness and quality of the peer-reviews. Specifically, there is a need to identify and highlight missing reviews (in their entirety, given or received) as well as to flag review fields of low quality. Addressing "lazy" reviewers, or those who gave insufficient feedback, was also given as a concern. These additional objectives imply a desire to understand the quality and reliability of the given peer-reviews within CSAMS suggest the need to include tools for oversight and management of peer-review activities.

The observations made and the feedback given indicate that the dashboard was well received and allowed the completion of the outlined objectives. Despite its apparent effectiveness, some bugs were discovered during the interviews, and changes to existing features and additional features were suggested and so the work continues in the next iteration.

## 5.2 Second Iteration

After the first iteration of the analytics dashboard demonstrated its potential to meet user needs, the second iteration aimed to refine and enhance its features. This iteration focused on improving existing functionality that users found lacking and adding new features based on the feedback received.

Participants had indicated that the dashboard was difficult to navigate and utilize effectively due to a lack of instructions. To address this issue, we aim to add on-demand instructions across the dashboard to provide necessary clarifications.

Additionally, this iteration aims to introduce customizable settings for diagrams such as box plot and sankey diagram. Users will have the ability to set their own thresholds for categorizing data points as Low, Mid, and High, allowing for a more relevant view of the data that better suits the specific need in a given context.

As an additional goal to follow the guidelines in [6] and [5, p. 5] findings in [8] we made an attempt at compressing information of the assignment to fit in a single screen in an attempt to make it easier to make connections in the data and to lighten the cognitive load [8, p. 341]. Thought this we also aim to remove the non-standard navigation that some participants found a little confusing. In addition, this will remove the initial white-space on the right side of each of the assignment screens and better utilize the screen real-estate and reduce the non-data pixels [6, p. 103].

In the database and queries, we had a goal of standardizing some common logic that was at some points left out causing bugs and missing data.

A big goal of this iteration was to add a screen to gain insight into the reviews a student received on their submission of an assignment. There is already a screen to see the reviews but this requires a lot of scrolling and was reported to be very difficult to use to compare reviews. This screen should also allow the user to change the reviews if any inconsistencies or bad reviews were found.

Regarding quality problems, this iteration also aims to add functionality to report any quality issues with any reviews or review fields to make it easier for lecturers to find problems that need attention.

### 5.2.1 UI Design

In this iteration the two screens for assignment overview were compressed into a single screen while the course overview screen stays mostly the same. In addition, we added a screen to see the received reviews for a student.

Based on the objective of having more instructions and usage hints this new design includes boxes with question marks that can be used to show instruction usage. This follows the concept of information on-demand [5] as to not clutter the interface while simultaneously providing all the necessary information to use and interpret the dashboard.

## Assignment Overview

This screen combines the two assignment overview screens from the previous iteration. It is based on the old group overview but makes changes to the right side to have the same functionality as the old individual student overview. This was achieved by introducing options for filtering by field and highlighting a student in the bar plot.

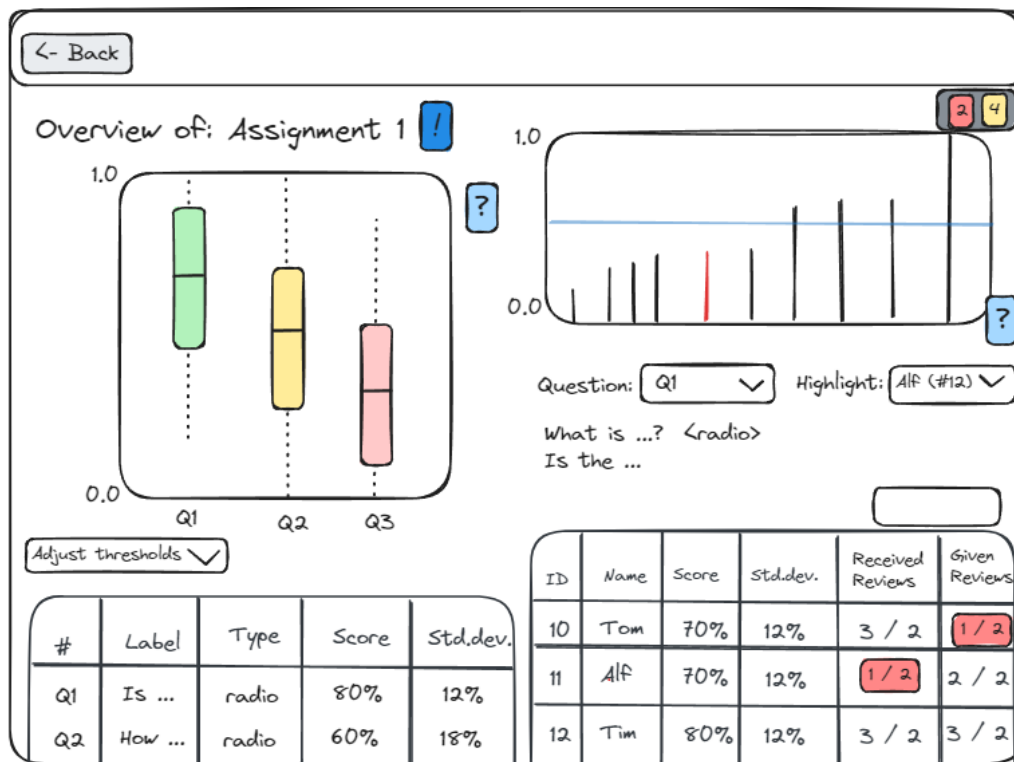
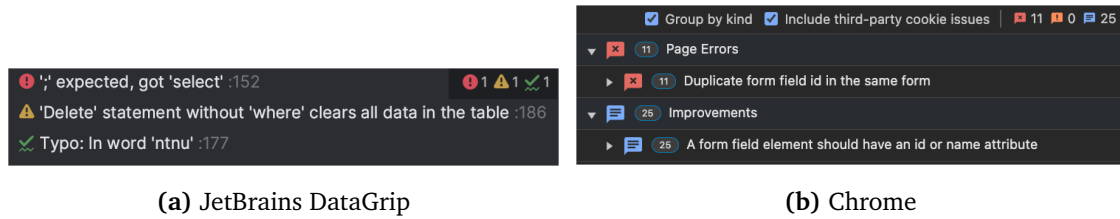


Figure 5.10: Assignment Overview, Design

The table on the right side uses color to indicate when the student has not performed enough reviews or when the student has not received enough reviews. This is to draw the attention of the lecturer to points of interest as per the feedback in the last iteration. The use of color is limited to only this point as to not distract the lecturer [6, p. 170] while the choice of using red it to draw attention to the point [6, p. 75].

With the addition of the reviews overview, this screen adds more linking in the individual students table. Additionally, the table includes the number of reviews performed and this also functions as a link to the existing and unchanged screen to see the reviews performed.

This iteration adds alerts to address the need for an overview of review quality that was reported in the feedback of the last iteration. The component for this can be seen in the top right of the screen. This is inspired by alerts in JetBrains IDEs and Chrome DevTools (Figure 5.11).



(a) JetBrains DataGrip

(b) Chrome

**Figure 5.11:** Alerts Inspiration

Alerts are categorized into errors (clearly a mistake), warnings (something wrong) and info (something that might be wrong). This component can be expanded to see the details of the alerts and each alert can be expanded to get a list of references to direct the lecturer to the location of the alert. Further details will be discussed in section 5.2.2.

We also included usage instructions on-demand as modals that can be accessed with the "?" buttons [6, p. 110]. The instructions include an explanation of the axes of the graphical visualizations, table columns, and how to interact with the dashboard. Additionally, a button was added in the title to get information about the assignment on-demand (details-on-demand) [5, p. 345].



## Course Overview

This screen is largely the same as in the previous iteration and only introduces some general improvements that was mentioned more generally in the feedback of the last iteration.

Most significantly is the introduction of adjustable thresholds. The thresholds are initially determined based on percentiles (25%, 50%, and 75%, non-adjustable) on a per-assignment basis in case varying performance in the different assignments. These thresholds can then be adjusted still per-assignment.

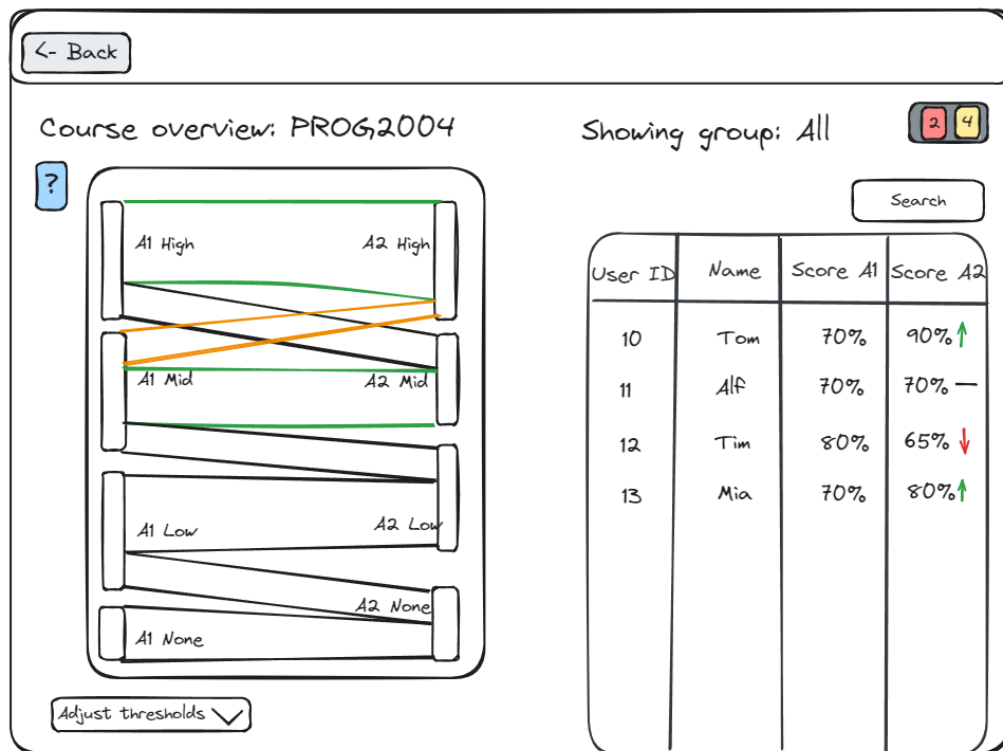


Figure 5.12: Course Overview, Design

As with the assignment overview screen there are also instructions on-demand on this screen. These instructions include the same as for the assignment overview but also includes more details about the thresholds since they are more complex for the sankey.

This screen also includes alerts. These are the same alerts that are shown for an assignment but without filtering on assignment.

## Reviews Overview

This screen gives an overview of the reviews that a student received on a submission. This addresses the need for insight into received reviews and disagreement among reviewers as reported in the feedback of the previous iteration.

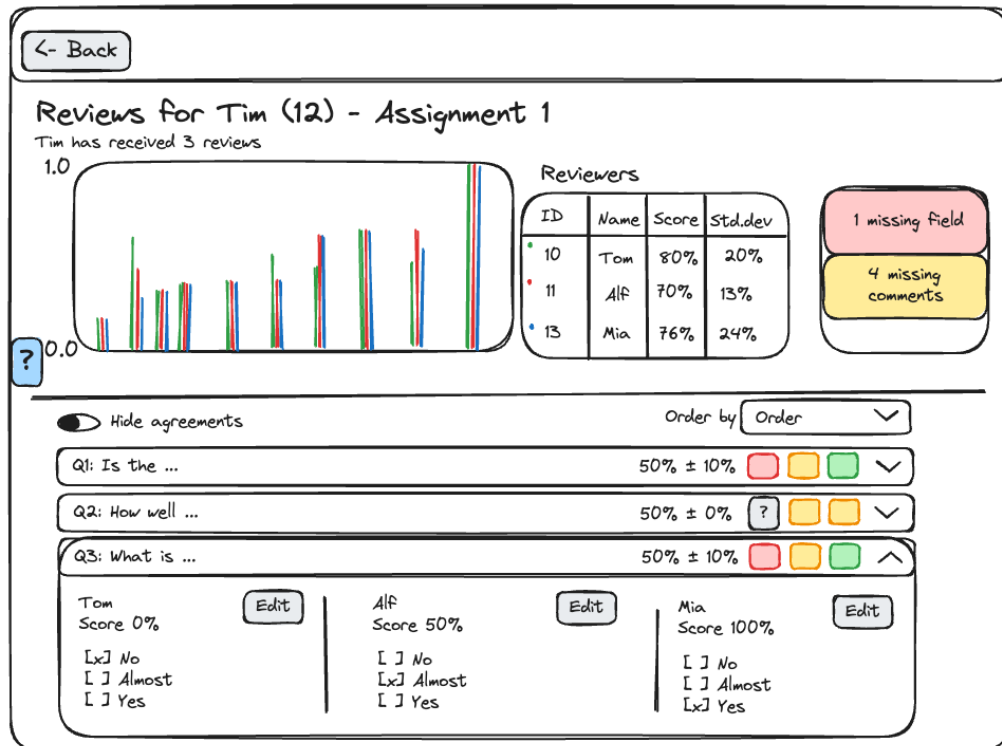


Figure 5.13: Reviews Overview, Design

The first element on this screen is the grouped bar plot. This is used to indicate the performance of each of the reviewers and to see the overall level for each question and also to see differences in the reviews.

Next to this is a table to show the reviewers with some information about their overall score and deviation for their review. The score can be used to compare to the other reviewers to see if they were in agreement overall while the deviation is intended to be used to identify lazy reviewer i.e. low deviation. This table also includes a colored dot in the top right of each row to indicate the corresponding color in the bar plot.

Similar to the other screen, this screen also has alerts. The scope of the alerts here is kept to the received reviews and can be used to jump to the correct row in the questions list. Different from the other screens is that this alerts container is always expanded. This was done simply because the overall layout of the screen allowed for it and this also makes it more visible with the intention that once the lecturer is on this screen they are more interested in finding individual fields with problems.

Below this first row is a list of questions showing an overview of each question. This over-

view consists of the question number, the question itself, score, deviation, and colored boxes to indicate the score. The colored boxes takes inspiration from [26, p. 58] to quickly visualize differences in the reviews. It uses a traffic light color scheme similar to the box plot to show the score and uses gray to indicate missing answer.

Above the questions list are controls to order and filter the list. The lecturer can order by question number, score, and disagreement. Additionally, with the toggle on the left side the lecturer can filter out questions where all the reviewers agree on the score (deviation of 0). Effectively this removes the noise and allows the lecturer to focus on the points with disagreements.

This screen also has usage instructions. On this screen there is an extra consideration in explaining the use of the questions list as well as the relation between the reviewers table and the bar plot. Since the scope of the aggregate scores is different in the questions list and in the reviewers table, this is also explained in the instructions.

It is also worth noting that, for consistency, the order of of the reviewers is kept the same in the bar plot, table, and the question list.

## 5.2.2 Development

### Expandable Content

Expandable content were implemented with Bootstrap since CSAMS was already using it, the decision to continue with this framework was straightforward. There were three types of expandable content implemented: instructions, hints, and hidden elements.

**Instructions** were implemented with the Bootstrap Modal component<sup>4</sup>. This places the instructions in a box in the center while darkening all other content. The instructions included information on the diagram axes, table columns, and the interaction between them as well as some instruction on the alerts.

**Hints** were implemented with the tooltip component<sup>5</sup> to show text when the user hovers over an element. This was implemented for all graphs to give information about the element hovered over as well as instructions for buttons. Since the instructions were thought to be a better and less cluttered way to provide instructions the use of this component was limited to additional information and also used to indicate what action could be done such as "Click to edit". However, the tooltip component was found to be very effective and easy to work with for displaying non-crucial additional information.

**Hidden elements** such as adjustable thresholds and alerts, were implemented with the collapse component<sup>6</sup>. Since the trigger for a collapse component requires a unique *ID* HTML attribute to reference its target, some elements used a slug<sup>7</sup> to differentiate elements that inherently had no obvious identifier such as alerts.

---

<sup>4</sup><https://getbootstrap.com/docs/4.3/components/modal>

<sup>5</sup><https://getbootstrap.com/docs/4.3/components/tooltip>

<sup>6</sup><https://getbootstrap.com/docs/4.3/components/collapse>

<sup>7</sup><https://itnext.io/whats-a-slug-f7e74b6c23e0>

## Data Attributes

To create the ordering and filtering functionality of the reviews overview we used HTML data attributes along with the JavaScript method `getAttribute` to get the attribute. Each element of the questions list had data attributes for question order, score, and disagreement. By adding a `onchange` listener to the "Order By" select we could get a list of the elements, order them by the correct attribute and set the content of the container element.

To hide agreement we used a CSS selector to hide elements that had a disagreement data attribute value of 0. The "Hide Agreements" toggle was used to toggle a class on the container element. This way the whole filtering can be defined by a single CSS selector.

```
.hide-agreements .question[data-disagreement="0"] {
  display: none;
}
```

## Database Optimization

While the queries in the last iteration showed great promise in reducing execution time it had some flaws that we wanted to address in this iteration. Firstly, each query was very long; about 30 lines each. This led to a lot of inertia when creating new queries or making changes to existing ones. Secondly, writing each query from scratch meant that they each had to include common logic such as accounting for group submissions and selecting distinct submissions. This introduces bugs where this common logic was not included notably from the prototype in the last iteration where the course overview accounted for group submissions but the individual overview did not so in following link from course overview to individual overview it was very likely that the link would not work.

This constituted the need for a different approach. The approach chosen was to introduce views to abstract away this common logic. We introduced 5 views that build upon each other to combine data. The views are shown in Figure 5.14.

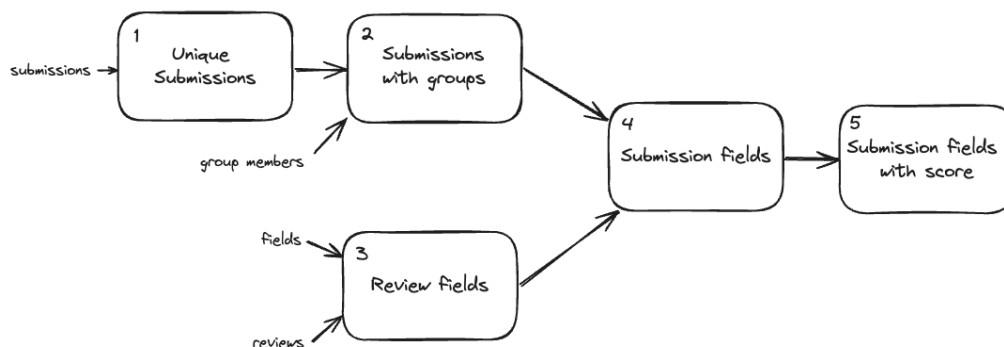


Figure 5.14: Database View Dependency Diagram

There is no table for just submission but rather there is just a single table that contains each submitted field so view 1 selects distinct submissions for a user and assignment to account for this.

Groups are implemented such that a single member of the group submits an assignment and the other group members are implicitly associated with the submission by having signed up to the same group. However, not all assignments use group submissions in which case the group ID will be 0. View 2 associates each submission with all group members by doing a left join of the assignment with the group members. If the group ID is 0 then there will only be the one row representing the person who submitted.

The review table is similar to the submissions table in that it does not store the overall submission but rather each field. However, there are a few important attributes missing from the review table so this has to be joined with the fields table to get all the attributes necessary to calculate the score. This is view 3.

Then, view 4 associates each review with each submission by each group member. At this point we have all the information about all the reviews received by each student.

View 5 adds a score column to view 4. This is set as a separate view to be used when the query does not need the score and is only interested in the review information.

View 5 (referred to later as just "the view") can be grouped by any of its attributes to create a different result. Grouping by *field\_name* gives average score for a question while grouping by *user\_id* gives the average score for a user. This means that all common logic is abstracted away in the views making the individual queries much simpler which makes adding new queries easier.

View creation queries can be found in Appendix A.

Since the view accounts for all common logic that was previously omitted in some queries in the last iteration, some queries were observed to be slower after implementing the view. Selecting every row in the view itself only took 79 milliseconds so there were no issues with the view itself but when we added grouping and aggregations the queries were significantly slower with indicators that it was these additional operations that impacted execution time the most. Running *explain analyze* (output in Appendix B) there were two things that took an excessive amount of time. The first was a very costly left join; there is only one left join in the views namely that for joining view 3 into view 2 to get view 4. To mitigate this issue, we added an index for the *name* column of the *fields* table. This took the execution time from 830 milliseconds to 350 milliseconds and according to the *explain analyze* output the left join went from 550 milliseconds to 70 milliseconds i.e. accounting for all the difference and a huge improvement. The second issue uncovered was with the aggregation functions. The test query used was the same as the query to generate box plot. This query therefore included four aggregation functions: *AVG*, *STDDEV\_POP*, *MIN*, and *MAX*. These four aggregation functions accounted for 320 milliseconds. Mitigating this requires larger changes to be made to the model and since the inclusion of the index has brought the execution time within expectations this is not required with the current system requirements. A discussion on the future possibilities can be found in subsection 6.1.2.

## Alerts Component

As described in the design, this iteration adds alerts to report any issues with review quality. This is done through the use of Server-Sent Events (SSE) which is a technology that allows the server to push real-time updates to the client over a single long-lived HTTP connection.

To implement this feature we used the SSE extension of HTMX<sup>8</sup>, which uses the *EventSource* JavaScript interface<sup>9</sup> to manage the connection with the server. This interface allows us to send a payload consisting of an optional event name and associated data.

The core of our implementation consists of two components: the *Event Dispatcher* (front-end) and the *Dispatcher* (back-end). The *Event Dispatcher* handles the routing of live data to specific elements of the web interface, using HTMX attributes to determine where and how data should be displayed. The *Dispatcher* struct manages the execution of multiple background tasks simultaneously. It orchestrates the parallel processing of tasks and ensures that each completed task's data is sent to the client as soon as it is available, without waiting for other tasks to finish. This simultaneous processing and immediate data streaming improves the responsiveness of the system, providing lecturers with instant updates on identified issues. Since the tasks are independent of each other this approach supports very heavy queries that takes a long time to execute. After all the tasks have been completed, a "close" event is sent which removes the SSE container and thus closes the connection. Without this the client would reconnect to the server thus triggering a new round of tasks.

Our implementation currently includes a variety of alerts that were determined to be essential as per the feedback from the last iteration. These implemented alerts are:

- (*error*) Radio buttons that have comments but no item selected
- (*warning*) Radio buttons that are not selected and do not have comments
- (*warning*) Students who have completed too few reviews
- (*warning*) Potential lazy reviewers, identified by a low standard deviation in their given review scores
- (*info*) Short or missing text answers

These alerts were chosen to evaluate the usability of the alert system and provide a solid foundation for future extensions. Additional alerts could easily be added as needed.

Beyond specific alerts, the Event Dispatcher's capability can be applied more broadly to dynamically populate a front-end dashboard. This is a change from the current system of populating the dashboard where each element on a front-end requests the exact data intended for the container with no consideration to its importance. Instead, a server-driven approach could evaluate and decide what important information is needed to be displayed and sends it directly to the relevant parts of the front-end. This would not only make the system more adaptive but also ensure that the most critical information is presented to lecturers, improving decision-making and overview.

---

<sup>8</sup><https://htmx.org/extensions/server-sent-events>

<sup>9</sup><https://developer.mozilla.org/en-US/docs/Web/API/EventSource>

## Implementation Results

The implementation of this iteration used the same technologies and techniques as the last iteration beyond the changes already described with database queries and alerts.

Since this iteration includes a greater ability to drill-down into problems, this prototype opens screens in a new browser tab when drilling-down e.g. clicking received reviews to get to the reviews overview screen.

## Assignment Overview

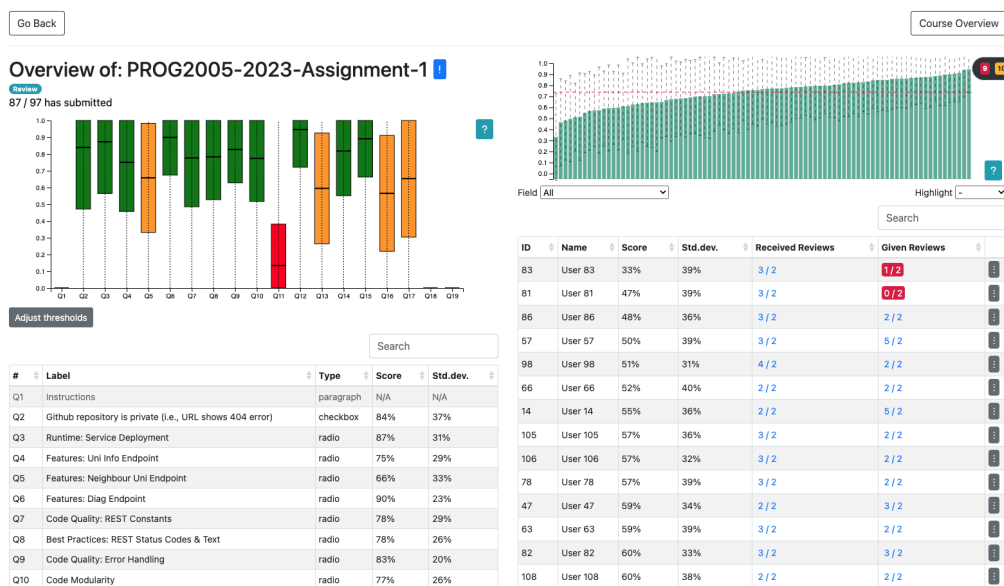


Figure 5.15: Assignment Overview, Implementation

The left side is mostly the same as in the previous iteration. However, it now had adjustable thresholds which uses *onchange* JavaScript listeners to redraw the diagram. The diagram is also redrawn when the screen resized with *debounce* to reduce potential redundant redraws.

The bar plot has functionality to handle clicking actions. Left click adds a hash to the URL, triggering a scroll down to the associated row in the table. Right click selects the clicked bar as the highlighted bar.

## Course Overview

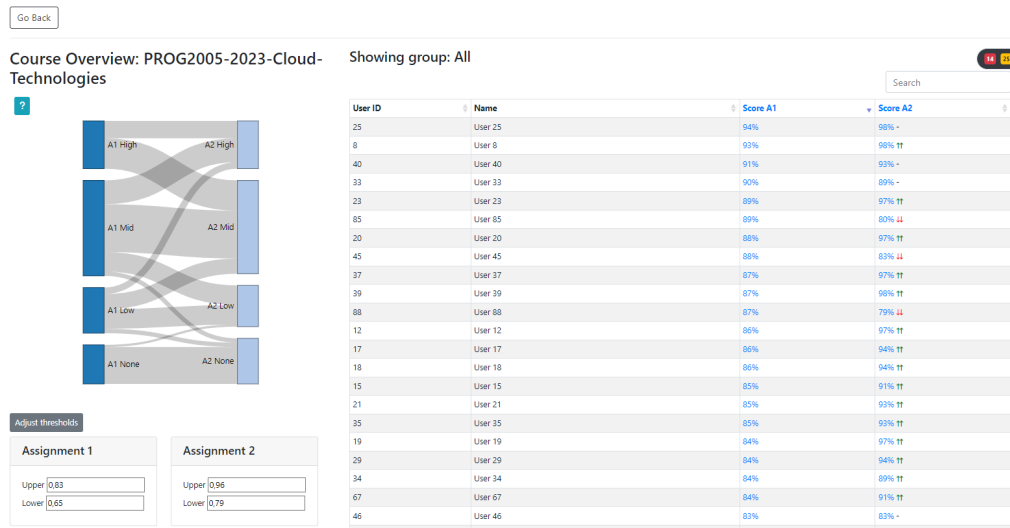


Figure 5.16: Course Overview, Implementation

We added adjustable thresholds and redrawing of the diagram on this screen the same way as for the assignment overview.

Due to time constraints for the implementation of this screen, we elected to not implement separate tasks for the alerts, instead opting to reuse the assignment and double up . This ensures that all the alerts are present but it also means that some alerts are shown twice such as two alerts for missing fields when really they are just for two different assignments. We considered it more important that all the alerts were present both for demonstration purposes but also for thoroughness of the solution rather than leaving it out completely as future work.



## Reviews Overview

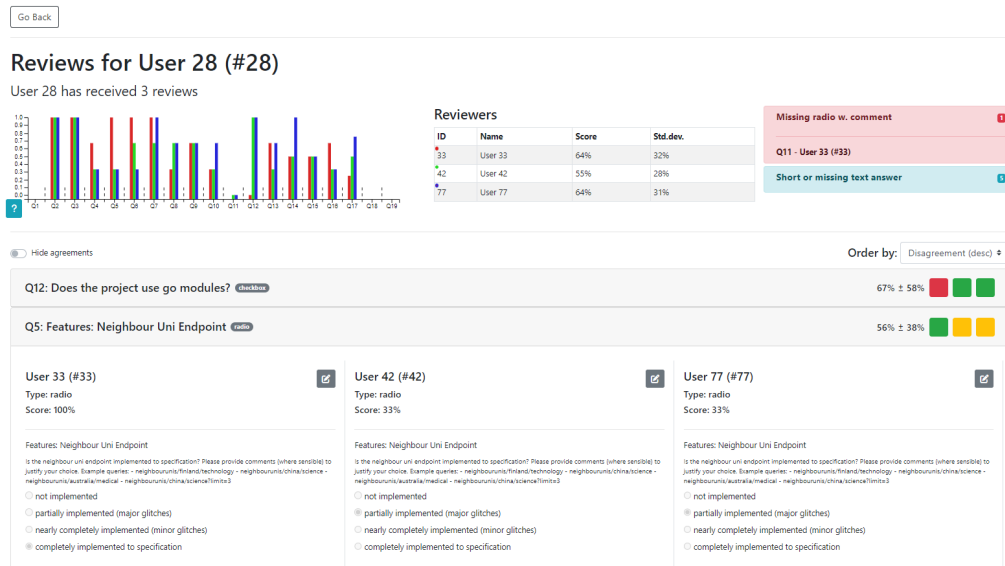


Figure 5.17: Reviews Overview, Implementation

Depending on how many reviewers there are, the boundary between the groups of the bar plot can be more difficult to see than it looked like in the design. Therefore, we added a small line between the groups to make this boundary more visible even with many reviewers.

The colors in the bar plot and reviewers table were based on HSL color wheel with a simple rotation of the hue with increments based on the number of reviewers. For three reviewers this gives the hues of 0, 120, and 240 degrees i.e. red, green, and blue.

### 5.2.3 Evaluation

The evaluation of this iteration stayed the same as for the last iteration. This was chosen both for the simplicity but also, since there were no problems with the evaluation, this allowed for easier comparison.

#### Tasks

The tasks for this iteration was largely the same as for the last iteration for the course and assignment overview. The assignment overview adds a task to draw the user to either alerts or the colors in the table if they had not already noticed that. With the newly introduced reviews overview there are tasks there were also a some tasks related to this screen. While the other screens have tasks for navigating to them, the reviews screen does not have that as the navigation to this screen should be done as part of the last task of the assignment overview.

1. Assignment Overview
  - a. Navigate to the dashboard for "Assignment 1" of Course 1
  - b. Can you identify any struggling students or topics based on the dashboard?
  - c. Can you identify any errors with any reviews?
    - Attempt to draw the attention to the alerts
2. Reviews Overview
  - a. Can you identify the largest disagreements among the received reviews?
  - b. Can you fix any of the issues with the received reviews?
3. Class Overview
  - a. Navigate to the dashboard for Course 1
  - b. Can you identify any students of interest based on the dashboard?

#### Questions

The questions used in this iteration was the same as for the first iteration. The participants were encouraged to draw comparisons to the previous iteration if possible.

1. What parts of the prototype did you find to be especially usable?
2. Was there any parts of the prototype that you experiences as confusing or difficult?
3. Were there anything that did not work as expected?
4. Was the layout of the pages intuitive and understandable? (inquiring about the screens individually)
5. Did you find the navigation to be intuitive?
6. Are there any additional objectives you want to accomplish with the dashboard?
7. Do you have any suggestions for improvements to the current functionality that would make the dashboard more valuable and usable to you?
8. Did you find the dashboard to be responsive?
9. If this dashboard was available in CSAMS, how likely are you to use it?
10. Any comments you would like to share about the dashboard?

## **Observations - Tasks**

### **Assignment Overview**

Being the first screen shown, much of the common functionality was explored here such as alerts and instructions.

The left side was mostly ignored. This was not surprising as this is at first glance the same as in the last iteration.

Participants did not initially find the instructions buttons and therefore the screen was much more confusing than it otherwise would. This also meant that much of the functionality of the bar plot was not obvious.

Most of the participants did not notice the alerts and the one who did, did not understand what it was and this was also not included in the instructions. As a result of this, we had to explain what the alerts was and how to use it. This was information that should have been in the instructions.

### **Course Overview**

This screen was largely unchanged from the previous iteration and most of the new features was explored in the assignment overview such as alerts and instructions. However, all the participants started playing around with the thresholds. Here, we notices a few different ways the participants selected the thresholds: one participant tried to find where the optimal thresholds were to separate the groups while another set the thresholds to that of grades to see grade distribution.

### **Reviews Overview**

The participant immediately notices the bar plot. This is expected as this is the first component of the screen and the participants were primed to look at diagram from the other screens. On the other hand, they did not seem to pay much attention to the reviewers table, some even noting that they did not understand what it was.

While all the participants notices the alerts, only one of them clicked one of the alerts to expand it. This participants however, did not click any further indicating that they did not notice that it was a link.

Eventually the participant looked at the questions list although they did not use the controls, instead all the participants initially just scrolled through the list while looking at the colored boxes on the right side. The participant eventually clicked a row to expand it and while they understood the side-by-side of the reviews they did not recognize the edit button.

After pointing out the list controls the participants tried to order the list, still, most of the participant were hesitant in using the option to hide agreements.

## Observations - Questions

### What parts of the prototype did you find to be especially usable?

The participants found that the linking within the dashboard pages to be very intuitive and a huge improvement over the previous iteration in how this allowed for faster and easier navigation to points of interest.

In the same tone one participant noted that the prototype of this iteration was way more in-your-face with the information much more accessible with less of a need to "hunt" for the right information. This was said about the right side of assignment overview both in relation to the filtering and also the linking.

In addition to the improved linking it was remarked that there was now a way to use the dashboard to navigate all the way to the "Edit Review" screen and fix any errors found through the dashboard as opposed to the previous iteration where the navigating to the edit screen was not available.

The participants found that the ability to adjust the thresholds particularly in the sankey diagram was a much needed addition overall made the sankey much more valuable.

In the reviews overview, the participants found the colored boxes of the questions list to be very intuitive and effective in quickly seeing the given scores and any difference. The side-by-side review was also perceived as extremely helpful and referred to by one participant as "brilliant".

A general feedback for this question was that this prototype was more intuitive than in the last iteration.

### Was there any parts of the prototype that you experienced as confusing or difficult?

While the participants did find the sankey threshold adjustment to be very usable, they did all find the wording and what the values represented to be a little difficult to understand at first.

Because standard deviation is used in many different places and for with different scopes some participants noted that it was unclear what standard deviation was referring to in the different tables. This was explained in the instructions but the participants also found this difficult to spot. However, this does illustrate the importance of the instructions in explaining what the values represent.

The participants had a hard time finding the expandable content e.g. questions list items and threshold adjustment. From the design, these should have ticks on the right side as an indicator that they are expandable however in implementation this was overlooked.

One participant mentioned that they found it difficult to relate the reviewers table to the grouped bar plot based on the colors. They said that they initially thought these were to indicate performance using the traffic light colors. This is a coincidence as when there are three reviewers the HSL hue rotate generates red, green, and blue which can easily be confused as traffic light colors at a glance. The participant recommended that colors used to indicate scores should not be used as labels. This observation was also extended to the red highlighting of a student in the assignment overview.

**Were there anything that did not work as expected?**

All participants said that when highlighting a student in the assignment overview that the corresponding row in the table should also be highlighted.

Since buttons and clickable elements have a background colors some participants noted that they expected that the various badges such as the one in the questions list to be clickable. One participant recommended only using background colors for clickable elements and only using border color for badges.

One participant noted that the browser tab names should be more descriptive. After using the dashboard for a little while there are many tabs opened and differentiating between these is difficult and so more descriptive names would help.

**Was the layout of the pages intuitive and understandable? (inquiring about the screens individually)**

Participants found that the layout of the pages in the prototype was largely intuitive and easy to understand. The assignment and course pages, in particular, were very clear, likely due to the split-screen design and the connection between these sections that was introduced in the previous iteration.

However, the participant noted that the placement of the instructions button was a little confusing and a little all over the place. One participant noted that they expected to see a help button in the bottom right.

**Did you find the navigation to be intuitive?**

The participants found that navigation within the dashboard to be very intuitive and that the linking was much better than last time, while the navigation to the dashboard was still confusing.

**Are there any additional objectives you want to accomplish with the dashboard?**

Two participants said that at this point the dashboard accomplished all the goals they could think of. However, implicitly as part of the other answers we found that one participants wanted to know more about how much students agreed with other students i.e not deviation within question (as in reviews overview) or deviation in reviews given or received (as in assignment overview) but rather overall how much a reviewer agreed with other reviewers on all scores given for each field.

Another participant expressed a desire to gain more insight into group dynamics but without any specific solutions in mind.

Yet another participant mentioned the desire to see how the disagreement changes over time with the expectation that as the students become better at reviewing they will start to converge in their reviews and disagreement will go down. Additionally, this participant mentioned a general interest in trends, while acknowledging that there might not be enough assignments in an any course to get any insight from this.

A participant mentioned wanting some visualizations of the text answers with possible solutions being word cloud, sentiment analysis (potentially with traffic lights), or keyword search.

**Do you have any suggestions for improvements to the current functionality that would make the dashboard more valuable and usable to you?**

The only feedback on functional changes was the ability to order on delta between assignments in the class overview table. This already has the trend so the context is there but this participant wanted the addition of way to order on that.

In addition to that functional change, a participant also wanted a way to save the set thresholds to they did not have to reset them when reloading page. The participant mentioned *localStorage*<sup>10</sup> as a potential solution.

**Did you find the dashboard to be responsive?**

All participants noted that the loading speed is more than sufficient and that there were no issues with the responsiveness of other actions either. One participant noted that the speed could be lower as long as the data correctness was kept intact.

**If this dashboard was available in CSAMS, how likely are you to use it?**

All participants said they would use it especially for the purpose of fixing review quality in addition to getting an overview of student performance. Participants also mentioned that at this point the dashboard has enough information to replace the need for the other overview screens.

**Any comments you would like to share about the dashboard?**

Participants noted that there was a huge improvement in the overall usability of the dashboard with the addition of the reviews overview that links all the way to editing reviews and usage instructions. No further comments beyond that.

---

<sup>10</sup><https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage>

### 5.2.4 Quantitative Evaluation

This section details the quantitative performance metrics of the latest prototype across various screens. While the participants of the user tests were very positive towards the performance these measurements gives us a objective view of the performance of the various screens. In addition, this gives us a point of comparison with the prototype from the previous iteration.

Screen	FCP (seconds)	LCP (seconds)	CLS
Course	0.81	1.28	0.007
Assignment	0.74	0.74	0.001
Reviews	0.53	0.53	0.008

**Table 5.2:** Quantitative Measurements

These numbers are consistent with the user feedback and proves their observations that the dashboard is performing well in loading.

The results are slightly slower than in the last iteration. This is likely due to the queries including more logic by using the views. At the same time the index lowers the time down to about the same time overall.

While the CLS was very good last time, this time it is even better. This is due to us gaining a better understanding of bootstrap and its flexbox styling options resulting in a more rigid and stable layout.

### 5.2.5 Conclusion - Second Iteration

This iteration aimed to address the shortcomings of the first iteration as well as adding features to gain insight into received reviews. In this regard, the second iteration of the prototype showed significant improvement over the first prototype, both in terms of functionality and usability.

The merging of the two assignment screens into a single, unified interface was well-received. Participants found it easier to understand and navigate, indicating a reduced cognitive load. This integration helped them make quicker and more effective use of the dashboard without having to switch between different screens.

The course overview screen remained mostly unchanged but included adjustable thresholds, which were seen as a valuable addition. These thresholds allowed users to customize data views to better fit their specific needs, making the tool more flexible and useful.

The new reviews Overview screen was another standout feature. It was found to allow for quick comparisons where the of colored boxes and side-by-side views of reviews were found to be particularly effective. This setup made it easy to spot discrepancies and understand review data at a glance, enhancing the dashboard's ability to provide detailed insights into the reviews.

Significant technical improvements were also made in this iteration, particularly concerning the database's structure and efficiency. By implementing views within the database, the system saw a reduction in bugs and a simplification in the process of adding new queries and features. This improvement streamlined the queries, making them more understandable and

less complex. These database views abstracted complex queries into simpler, reusable components, reducing the potential for errors and increasing the efficiency of data retrieval. With the additional logic added to the queries, the performance of some of them went down. To account for this we added an index on the name column of the fields table which took the execution back down to about the same levels as before maintaining the load time of the interface.

The use of alerts was another major addition in this iteration. Alerts was found to be a valuable feature for identifying and acting on issues within the dashboard. They provided immediate notifications about inconsistencies and errors in the review data, allowing lecturers to focus on critical areas in need of attention. This feature not only enhanced the dashboard's functionality but also its educational effectiveness, as lecturers could more efficiently manage and rectify issues in peer reviews.

For future quality-of-life improvements, a participant suggested adding a mechanism to save and reload set thresholds. Future iterations could focus on adding more information in the sankey table going for a big-table approach as in [15] by adding more columns and potentially header-levels thereby enabling ordering on delta, as suggested in the feedback.

Overall, participants were pleased with the prototype and the insights it provided. They had fewer suggestions for improvements this time, possibly indicating that the initial needs were well met. The dashboard successfully achieved its intended goals.



# Chapter 6

## Discussion

The discussion points in this chapter comes in addition to the discussions in chapter 5.

### 6.1 Database

#### 6.1.1 Precalculated Score

Instead of calculating scores on-the-fly for each query, a more efficient approach could involve adding a column to the *user\_reviews* table to store the precalculated scores which could be maintained using database triggers to keep the scores up-to-date. However, it was found that the performance impact of real-time score calculation is minimal. The comparison between selecting from views with and without score calculations (views 4 and 5 in Figure 5.14) showed only a 5 millisecond difference (74ms vs. 79ms). The primary performance challenges are related to grouping and filtering operations rather than score calculations. Implementing this approach would also necessitate significant changes to the data model, adding complexity without substantial performance benefits at the current scale.

#### 6.1.2 Scalability

As originally outlined in the motivation (section 1.2), we should consider the scalability of the additions in the system. Here we discuss some aspects that should be considered when scaling the system.

#### Indices

We have already added an index to the system which ensures that all the features added are within expected performance limits. While no other fields were indicated as potential candidates for indices this does not mean that there is no need to any new indices as requirements change. It is expected that, if the system has more assignments, an index on the *assignment\_id* field of the *user\_reviews* table will be required to rapidly separate the reviews of an assignment. The same considerations needs to be made for other tables since for now the assignment count is so low that most of the rows of a table will be included in any join or filtering and thus an

index can not be used effectively to separate the rows, however, as the row count increases an index scan might be more efficient than a sequential scan.

### Materialized View

A materialized view can be described as a precomputed view that stores the results of a query physically. While MySQL (as of version 8.0) does not natively support materialized views as PostgreSQL does, it is possible to simulate this functionality by creating a table from a selection and periodically updating it by truncating and refilling it with the latest data.

Implementing materialized views could potentially make each query more efficient and faster, especially for complex queries that involve large amounts of data aggregation. By pre-computing and storing results, the system could reduce computational load at query time, making data retrieval quicker and more responsive. This approach could be particularly beneficial for queries that are accessed frequently, as it would turn complex calculations into simple select statements with predicates for filtering on assignments or courses. This would require creating a separate materialized view for each grouping e.g. one for grouping on *field\_name* and one for grouping on *user\_id* to get scores from different perspectives.

However, the use of materialized views comes with several trade-offs. One significant drawback is that materialized views are not updated in real-time; they would always be somewhat out of sync with the latest data, typically for use-cases like here by a few hours. This delay might be acceptable in some contexts, but it can be problematic when up-to-date information is critical. Additionally, since the dashboard is used relatively sparsely, many updates to the materialized views would likely be unnecessary, resulting in inefficient use of resources. Balancing the update interval to keep data fresh without causing undue overhead is a challenging aspect of this approach. Ideally the update interval is decided by the access frequency. However, there comes a point where making the interval any longer would result in too stale data; that is the dilemma we find ourselves in here. While the access frequency indicates that we only need to update maybe as little as once per day, that could be too long as more reviews could have come in during that period which would make the data inaccurate.

Additionally, introducing materialized views would introduce additional complexity into the data model. This added complexity includes managing the update schedules and ensuring data consistency, which could complicate maintenance and development efforts along with complicating debugging and problems with data correctness. Given the current scale of the system and the need to keep the model straightforward, the decision was made to not use materialized views. The current approach maintains data freshness and minimizes complexity while still delivering adequate performance. However, as the system scales, it could become necessary to revisit this decision. A proposal for creating and updating a simple materialized view with an event scheduler for timed updates can be found in Appendix D, offering a potential solution if future needs justify it.

Furthermore, from an environmental perspective, choosing materialized views that update needlessly can lead to unnecessary computational resource consumption, which is undesirable in terms of sustainability.

Therefore, the current strategy aims to balance performance needs with environmental considerations by avoiding unnecessary updates and maintaining an efficient, streamlined database structure.

## 6.2 Dashboard Effectiveness

Following the evaluation model outlined in [20] ( section 3.3.4 ), the dashboard accomplished the first level (awareness) with the various visualizations, the second and third levels (reflection and sensemaking) by prompting the user to evaluate the data and its relevancy while being able to filter for values to get a more granular overview. However, we cannot evaluate the fourth level (impact) because, as the paper points out, this requires a longitudinal study on the impact of the dashboard over time which is left as future work. Although we can make some arguments as to how it would reach this fourth level. Firstly, an impact outlined by the paper is that the dashboard would make behavioural changes. In the CSAMS dashboard, this change could be measured by: frequency or depth of data analysis conducted, changes in the way the lecturer addresses student performance issues, modification in the peer-review process and quality control, and overall engagement with the dashboard. Secondly, the lecturers overview of student performance could help the lecturer better guide the students in their learning and help point out struggling users and in that way make an impact.

## 6.3 Dashboard Design and Implementation

Here we will discuss and justify some of the choices made on the final version of the dashboard design and implementation.

### Alerts

The decision to implement a complex alert system using Server-Sent Events (SSE) rather than a set of predefined *GET* requests was based on the need for flexibility and maintainability. While the SSE-based implementation added complexity to the back-end, it reduced redundancy and centralized the alert logic, thus simplifying the overall code base. This design means that there is only one SSE endpoint instead of multiple endpoints for each alert type, making the system more streamlined. Moreover, this approach supports easier maintenance and extension, as adding new alerts involves modifying a single part of the code rather than updating numerous HTML templates.

The SSE-based alert system serves as a prototype demonstrating a flexible mechanism for dynamically populating the dashboard with critical information. This modular approach allows for the system to be easily extended to other areas of the dashboard, enhancing its adaptability and responsiveness to new requirements. Few mentions the importance of reducing non-data pixels in a dashboard [6, p. 103] however we can go a step further and say that we should reduce non-insight pixels i.e. all pixels that does not provide insight should be removed. A data-driven population of the dashboard could help in removing non-insight pixels by filling the dashboard with only the most important aspects of the data potentially providing more insight. Exploring and implementing this feature is left as future work.

### **Change to Percentage**

Even though participants didn't explicitly ask for a switch from decimal numbers to percentages, their confusion about the scale and normalization of the numbers suggested that percentages would be clearer. Percentages, ranging from 0 to 100, are universally understood and remove any ambiguity, making the data easier to immediately understand. While we cannot say for certain if it is because the confusion was cleared up in the first iteration, the issue was not mentioned in the second iteration. This also fits with Few's mentions of understanding the context of the data [6, p. 53].

### **Split Screen Design**

In the assignment overview, the split-screen design combines diagrams and tables effectively, which is a well-accepted pattern in interface design[5]. Tables often need the full height of the screen, so adding a second column for diagrams or extra information made sense. While a three-column layout would be too crowded, a two-column layout is a good balance. This design lets lecturers see both group-level and individual-level data at the same time, giving them better insights into how students are performing relative to their peers. Participants found the design a bit confusing at first but quickly found it intuitive and useful. This indicates that, despite a brief learning curve, the split-screen design improves the dashboard's overall effectiveness. Despite a general advice to stay away from scrollable content as Few points out that anything that has to be scrolled for is assumed by the user to be less important and will likely never see it [6, p. 52]. In this case the tables go beyond the height of the screen, requiring scrolling to see all the content. However, in this is accepted as the content outside the screen is just more of what is already on the screen. Therefore, there is not any additional points being made that the user has to scroll for; just more of the same.

### **Different Alert Designs**

The design choice to use different presentations for alerts—boxes (as in the reviews overview) versus small pills (as in assignment and course overviews) was driven by layout constraints. It was challenging to fit the full alert component from the reviews overview into the existing layouts for assignment and course without a larger redesign. Leaving the layout as it was allowed for evaluation of user preferences regarding alert style and usability. Future iterations could refine this based on user feedback to optimize alert display without compromising the interface's cleanliness.

### **Grouped Bar Plot**

The decision to use a grouped bar plot instead of a box plot in the reviews overview was based on the typical number of reviewers. With usually only 2-3 reviewers, a box plot would not provide a meaningful representation of the data. Grouped bar plots offer clearer insights in these cases. However, a dynamic plot type that switches to a box plot when there are more than four reviewers could be a future enhancement, preventing clutter and enhancing readability. Potentially, this could also be manually set with a toggle.

## 6.4 Limitations

### Participant Limitations

The participant pool for this study was extremely limited, involving only three lecturers. This number closely represents the total of four possible lecturers who met the criteria for familiarity with the CSAMS. The small sample size restricts how many different perspectives we were able to capture and thus the results may not capture the full range of potential user experiences and feedback with the dashboard.

There is the consideration that we could have involved lecturers that did not have experience with CSAMS and given them instructions on the use beforehand. However, we opted to not do this for two reasons: (1) we got three participants early on and as such a greater participation than we initially required, and (2) CSAMS differs from other LMS in that it is less focused on the raw results, instead focusing more on the quality and consistency which is a mindset shift that could be difficult to properly instill in participants not already familiar with CSAMS and as such the goals those participants would wish to achieve would not be the best fit for CSAMS.

### Accessibility Considerations

Modern professional interfaces are typically expected to include some accessibility considerations. Since the dashboard was designed primarily as a prototype, and considering its experimental nature aimed at evaluating core functionalities rather than broad usability, accessibility was not considered in this thesis. This decision allowed us to focus resources and attention on performance and essential features for the initial prototype.

### Dataset Constraints

Since we only had a single dataset available we could not definitively generalize the findings for larger datasets. Although educated guesses and common optimization strategies were suggested, these are based on theoretical extensions rather than empirical evidence from diverse or larger datasets.



## Chapter 7

# Conclusion

This chapter summarizes the project, provides answers to the research questions, and outlines future work.

### 7.1 Summary

The CSAMS analytics dashboard project aimed to enhance the visibility and usability of student performance data derived from peer-reviews. Throughout the development process, two iterations were conducted to refine the dashboard's functionality and user interface based on feedback from lecturers.

The dashboard successfully integrated features that addressed the lecturers' needs, such as identifying which specific questions or concepts students were struggling with and tracking the learning trajectory of each student. By incorporating drill-down capabilities, standard deviation metrics for peer reviews, and customizable thresholds, the dashboard provided a comprehensive tool for lecturers to gain insights into student performance and review quality.

Technical challenges such as ensuring efficient data aggregation and optimizing query performance were addressed through the use of stored procedures and careful indexing. The decision not to implement materialized views was made to maintain data freshness and minimize complexity. Instead, real-time query optimizations were utilized to balance performance and accuracy.

Overall, the CSAMS analytics dashboard has proven to be a valuable tool for visualizing student performance data, providing lecturers with the necessary insights to support and improve student learning effectively.

## 7.2 Research Questions

### **RQ1: What objectives does the lecturers using CSAMS wish to accomplish with a Learning Analytics Dashboard?**

Lecturers using the CSAMS analytics dashboard identified several key objectives that focus on understanding and improving student performance. One primary objective is to identify which specific questions or concepts students are struggling with. This allows lecturers to pinpoint areas of the curriculum that may require additional instructional time or alternative teaching methods. By identifying these challenging areas, lecturers can adjust their teaching strategies to better support student in their learning.

Another significant objective identified was to track the learning trajectory of each student over time. Understanding how students progress allows lecturers to identify those who are improving and those who are falling behind, enabling lecturers to make interventions to correct students. This extends into identifying students at risk of failing the course. By analyzing patterns in submission and review activities, as well as overall performance levels, lecturers can early detect struggling students. This proactive approach allows for timely interventions that can help at-risk students get back on track.

Lecturers also expressed a strong interest in evaluating the quality of peer-reviews. Using metrics such as standard deviation, lecturers aim to gauge the consistency of reviews and identify instances where reviewers significantly disagree. This information is important for assessing the reliability of peer-reviews and determining when further training or manual review is necessary to maintain the integrity of the evaluation.

### **RQ2: What technical and usability challenges need to be solved when implementing a Learning Analytics Dashboard for CSAMS?**

Implementing the CSAMS analytics dashboard involved the need to address several central technical and usability challenges. One of the primary challenges was ensuring intuitive navigation within the dashboard. Participants reported difficulties in finding specific screens and understanding the layout without sufficient instructions. To address this, better navigation aids and on-demand instructions were necessary to make the dashboard more user-friendly.

Efficiently aggregating data from the CSAMS database without causing performance issues was another significant challenge. This included ensuring that data updates were timely without introducing excessive computational overhead. Using stored procedures enabled the use of aggregation functions in queries along with grouping which greatly reduced the efficiency of data aggregation as compared to the existing application logic. Combining this with views to abstract away the complexity and logic of the data model made the database much easier to work with and less error-prone.

Incorporating user feedback effectively to improve functionality was also an important factor in the success of the dashboard. Features such as drill-down capabilities were implemented to allow lecturers to access detailed data, such as specific student reviews, from summary views. This was done through logical linking between different pages in addition to automatically scrolling to a specific location. This was important for understanding the context behind performance data. Standard deviation was highlighted as an important metric and was integrated into the dashboard to assess the consistency of peer reviews and highlight areas of



significant disagreement among reviewers. This functionality helped lecturers evaluate the reliability of peer-reviews.

Ensuring visual elements were clear and consistent was another big challenge. Participants noted issues with understanding what elements could be clicked and the meanings of certain visual indicators. To address this, we standardized the use of colors and interactive elements, and provided clear explanations for visual data representations. Additionally, the dashboard incorporated tool-tips and on-demand instructions to aid in data interpretation and usage.

The dashboard was also improved with features to track review completion and quality. This included identifying missing or incomplete reviews, which ensures the quality of the peer-reviews. Adjustable thresholds were implemented to make sure the diagrams better fit their specific analysis needs. In addition, all tables included sorting and searching to make them more navigable.

**RQ3: How can data be efficiently aggregated in the CSAMS database?** Efficient data aggregation in the CSAMS database can be achieved through several strategies that were implemented or considered during the project. One effective approach used was the implementation of stored procedures to enable complex aggregation functions within the database. This reduces the load on the application layer and ensures faster data retrieval, as the use of stored procedures means we have the scores needed on the database without having to move the data to the application first.

Queries were optimized to handle large datasets efficiently. This involved indexing critical columns and using appropriate query strategies to minimize execution time and resource usage. Complex queries were abstracted into simpler, reusable views to reduce errors and complexity of these individual queries.

Materialized views were considered as a potential solution for pre-aggregating data to improve query performance. However, this approach was ultimately not used due to concerns about data freshness and the complexity of managing materialized views within the existing system. The decision was made to rely on real-time query optimizations instead.

Maintaining data freshness without excessive computational overhead required carefully balancing the update intervals for aggregated data. This involved analyzing access frequency to determine the optimal update schedule. For example, the dashboard was designed to update certain metrics in real-time, while less frequently accessed data was updated at regular intervals.

Efficient data aggregation also considered environmental sustainability by minimizing unnecessary computational resource consumption. This involved avoiding redundant updates and maintaining a streamlined database structure. Efforts were made to ensure that the aggregation processes were as efficient as possible, reducing the overall computational load on the server.

Combined, these strategies results in a performance within the accepted thresholds for the CSAMS analytics dashboard, as well as considerations for how the system could further be improved for larger scales.

## **7.3 Future Work**

### **7.3.1 More Iterations**

Future work on the CSAMS analytics dashboard should include further iterations to refine and enhance its functionality and usability. One critical area for improvement is the consistency of navigation within CSAMS. As observed during the user evaluations, participants faced challenges in finding specific screens due to inconsistent navigation paths within CSAMS. While this is not an issue with the analytics features themselves, finding them is an important factor in their usability and should be improved.

Another area for future iterations is the expansion of alert jobs. The current implementation of alert jobs successfully identifies some issues such as missing reviews and poor feedback. However, adding more types of alerts can further aid lecturers in identifying and addressing a wider range of issues. These might include alerts for significant drops in student performance, patterns of late submissions, or more anomalies in review consistency.

Efforts should also be made to condense the dashboard layout to eliminate the need for separate screens for the course overview. A more integrated approach could involve displaying course-level data alongside individual assignment data within a unified interface. This would streamline the user experience and make it easier for lecturers to obtain a holistic view of student performance without navigating between multiple screens.

### **7.3.2 Longitudinal Study**

To rigorously evaluate the long-term effectiveness of the CSAMS analytics dashboard, longitudinal studies should be conducted. These studies would track the impact of the dashboard on student performance and teaching strategies over extended periods. By collecting and analyzing data across multiple semesters, researchers can assess whether the insights provided by the dashboard lead to sustained improvements in student outcomes and instructional practices. This long-term perspective is essential for understanding the true value of the dashboard and identifying any areas where further adjustments might be necessary.

### **7.3.3 Student facing dashboard**

The concept of a student-facing dashboard emerged as a suggestion during the first iteration of the project. Implementing a LAD for students could allow the students themselves to take a more active role in their learning. By providing students with insights into their own performance, strengths, weaknesses, they can self-direct their learning more effectively. This addition would not only enhance the utility of CSAMS but also allow for more engaged and proactive students.

#### **7.3.4 Data-driven Dashboard**

Future iterations should also focus on making the dashboard more data-driven, particularly by leveraging the alerts component. The alerts system, which currently highlights issues such as missing reviews and inconsistencies, can be expanded to incorporate predictive analytics. For example, alerts could notify lecturers of students who are predicted to struggle based on current performance trends or highlight emerging patterns that might require pedagogical adjustments. By integrating more sophisticated data analysis techniques, the dashboard can provide even more actionable insights, helping lecturers to make informed decisions and intervene proactively. Additionally, the dashboard could be populated based on what results were found to be interesting and not just a static set of elements. This would require much more resources but could be an interesting topic to ensure the dashboard value and relevancy in addition to potentially de-cluttering by not showing irrelevant information.

#### **7.3.5 Better Utilization of Textual Data**

For now, the textual data is almost entirely untouched. Future projects could focus on visualizations of this data. Participants mentioned simple examples like word cloud, keyword search, frequency analysis. However, more advanced text processing techniques like NLP models or sentiment analysis could be used to get a better view of the data and to be able to generate scores from text data that could be aggregated in the same way as



# Bibliography

- [1] J. Aanesen, B. F. Klausen and S. A. Danielsen, 'Administrasjonssystem for datavitenskapsoppgaver,' Bachelor Thesis, NTNU, 2019. [Online]. Available: <http://hdl.handle.net/11250/2617889>.
- [2] S. Few and P. Edge, 'Dashboard confusion revisited,' *Perceptual Edge*, pp. 1–6, 2007.
- [3] H. K. Hoel, 'A study of how peer review used in group and individual assignments in computer science impacts students' learning outcome,' Master's Thesis, NTNU, 2023. [Online]. Available: <https://hdl.handle.net/11250/3079164>.
- [4] J. E. Hægdahl, 'Exploration of automatic quality assurance of code reviews,' Master's Thesis, NTNU, 2023. [Online]. Available: <https://hdl.handle.net/11250/3079158>.
- [5] B. Bach, E. Freeman, A. Abdul-Rahman, C. Turkay, S. Khan, Y. Fan and M. Chen, 'Dashboard design patterns,' *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 1, pp. 342–352, 2023. DOI: 10.1109/TVCG.2022.3209448.
- [6] S. Few, *Information dashboard design: The effective visual communication of data*. O'Reilly Media, Inc., 2006.
- [7] Y. Yoo, H. Lee, I.-H. Jo and Y. Park, 'Educational dashboards for smart learning: Review of case studies,' in *Emerging Issues in Smart Learning*, G. Chen, V. Kumar, Kinshuk, R. Huang and S. C. Kong, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 145–155, ISBN: 978-3-662-44188-6.
- [8] S. Pozdniakov, R. Martinez-Maldonado, Y.-S. Tsai, N. Srivastava, Y. Liu and D. Gasevic, 'Single or multi-page learning analytics dashboards? relationships between teachers' cognitive load and visualisation literacy,' in *Responsive and Sustainable Educational Futures*, O. Viberg, I. Jivet, P. Muñoz-Merino, M. Perifanou and T. Papatoma, Eds., Cham: Springer Nature Switzerland, 2023, pp. 339–355, ISBN: 978-3-031-42682-7.
- [9] S. Govaerts, K. Verbert, E. Duval and A. Pardo, 'The student activity meter for awareness and self-reflection,' in *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '12, Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 869–884, ISBN: 9781450310161. DOI: 10.1145/2212776.2212860. [Online]. Available: <https://doi.org/10.1145/2212776.2212860>.
- [10] B. A. Schwendimann, M. J. Rodríguez-Triana, A. Vozniuk, L. P. Prieto, M. S. Boroujeni, A. Holzer, D. Gillet and P. Dillenbourg, 'Perceiving learning at a glance: A systematic literature review of learning dashboard research,' *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 30–41, 2017. DOI: 10.1109/TLT.2016.2599522.

- [11] L. Ali, M. Hatala, D. Gašević and J. Jovanović, 'A qualitative evaluation of evolution of a learning analytics tool,' *Computers & Education*, vol. 58, no. 1, pp. 470–489, 2012, ISSN: 0360-1315. DOI: <https://doi.org/10.1016/j.compedu.2011.08.030>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360131511002089>.
- [12] O. Scheuer and C. Zinn, 'How did the e-learning session go? the student inspector,' Jan. 2007, pp. 487–494.
- [13] D. Leony, A. Pardo, L. de la Fuente Valentín, D. S. de Castro and C. D. Kloos, 'Glass: A learning analytics visualization tool,' in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, ser. LAK'12, Vancouver, British Columbia, Canada: Association for Computing Machinery, 2012, pp. 162–163, ISBN: 9781450311113. DOI: 10.1145/2330601.2330642. [Online]. Available: <https://doi.org/10.1145/2330601.2330642>.
- [14] R. Mazza and V. Dimitrova, 'Coursevis: A graphical student monitoring tool for supporting instructors in web-based distance courses,' *International Journal of Human-Computer Studies*, vol. 65, no. 2, pp. 125–139, 2007, ISSN: 1071-5819. DOI: <https://doi.org/10.1016/j.ijhcs.2006.08.008>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1071581906001303>.
- [15] J. L. Santos, K. Verbert, S. Govaerts and E. Duval, 'Addressing learner issues with stepup! an evaluation,' in *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, ser. LAK '13, Leuven, Belgium: Association for Computing Machinery, 2013, pp. 14–22, ISBN: 9781450317856. DOI: 10.1145/2460296.2460301. [Online]. Available: <https://doi.org/10.1145/2460296.2460301>.
- [16] M. Ji, C. Michel, É. Lavoué and S. George, 'Ddart, a dynamic dashboard for collection, analysis and visualization of activity and reporting traces,' in *Open Learning and Teaching in Educational Communities*, C. Rensing, S. de Freitas, T. Ley and P. J. Muñoz-Merino, Eds., Cham: Springer International Publishing, 2014, pp. 440–445, ISBN: 978-3-319-11200-8.
- [17] S. Graf, C. Ives, N. Rahman and A. Ferri, 'Aat: A tool for accessing and analysing students' behaviour data in learning systems,' in *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, ser. LAK '11, Banff, Alberta, Canada: Association for Computing Machinery, 2011, pp. 174–179, ISBN: 9781450309448. DOI: 10.1145/2090116.2090145. [Online]. Available: <https://doi.org/10.1145/2090116.2090145>.
- [18] C. Zinn and O. Scheuer, 'Getting to know your student in distance learning contexts,' in *Innovative Approaches for Learning and Knowledge Sharing*, W. Nejdl and K. Tochtermann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 437–451, ISBN: 978-3-540-46234-7.
- [19] B. Hui and S. Farvolden, 'How can learning analytics improve a course?' In *Proceedings of the 22nd Western Canadian Conference on Computing Education*, ser. WCCCE '17, Abbotsford, BC, Canada: Association for Computing Machinery, 2017, ISBN: 9781450350662. DOI: 10.1145/3085585.3085586. [Online]. Available: <https://doi.org/10.1145/3085585.3085586>.

- [20] K. Verbert, E. Duval, J. Klerkx, S. Govaerts and J. L. Santos, 'Learning analytics dashboard applications,' *American Behavioral Scientist*, vol. 57, no. 10, pp. 1500–1509, 2013. DOI: 10.1177/0002764213479363. eprint: <https://doi.org/10.1177/0002764213479363>. [Online]. Available: <https://doi.org/10.1177/0002764213479363>.
- [21] C. M. D. S. Freitas, P. R. G. Luzzardi, R. A. Cava, M. Winckler, M. S. Pimenta and L. P. Nedel, 'On evaluating information visualization techniques,' in *Proceedings of the Working Conference on Advanced Visual Interfaces*, ser. AVI '02, Trento, Italy: Association for Computing Machinery, 2002, pp. 373–374, ISBN: 1581135378. DOI: 10.1145/1556262.1556326. [Online]. Available: <https://doi.org/10.1145/1556262.1556326>.
- [22] R. Kaliisa, I. Jivet and P. Prinsloo, 'A checklist to guide the planning, designing, implementation, and evaluation of learning analytics dashboards,' *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, p. 28, 2023.
- [23] A. R. Hevner, S. T. March, J. Park and S. Ram, 'Design science in information systems research,' *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, ISSN: 02767783. [Online]. Available: <http://www.jstor.org/stable/25148625> (visited on 28/02/2024).
- [24] P. Walton and B. Pollard. 'Largest contentful paint (lcp).' (19th Feb. 2024), [Online]. Available: <https://web.dev/articles/lcp> (visited on 10/04/2024).
- [25] J. Brooke, 'Sus – a quick and dirty usability scale,' in Jan. 1996, pp. 189–194.
- [26] K. Upton and J. Kay, 'Narcissus: Group and individual models to support small group work,' in *User Modeling, Adaptation, and Personalization*, G.-J. Houben, G. McCalla, F. Pianesi and M. Zancanaro, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 54–65, ISBN: 978-3-642-02247-0.





## Appendix A

# Create Views Queries

```
1  create or replace view unique_submissions as
2  select distinct user_id, assignment_id, group_id
3  from user_submissions;
4
5
6  create or replace view submissions_with_groups as
7  -- group submissions
8  select ag.assignment_id, agm.user_id, s.user_id as submitter_user_id,
9         ↪ s.group_id
10 from unique_submissions s
11 inner join assignment_groups ag
12         on ag.id = s.group_id
13 inner join assignment_group_members agm
14         on ag.id = agm.group_id
15 where s.group_id <> 0
16
17 union
18 -- individual submissions
19 select assignment_id, user_id, user_id as submitter_user_id, group_id
20 from unique_submissions s
21 where group_id = 0;
22
23
24 create or replace view review_fields as
25 select ur.id as review_id, ur.user_reviewer, ur.user_target,
26        ur.answer, ur.comment, ur.assignment_id as review_assignment_id,
27        ↪ ur.type, ur.name as field_name,
28        f.weight, f.choices, f.label, f.priority, f.has_comment,
29        ↪ f.description, f.required
```

```
28  from user_reviews ur
29  inner join fields f
30      on ur.name = f.name;
31
32
33  create or replace view submission_review_fields as
34  select sg.user_id, sg.submitter_user_id, sg.group_id, sg.assignment_id,
35         ↪ rf.*
36  from submissions_with_groups sg
37  left join review_fields rf
38      on rf.user_target = sg.submitter_user_id
39         and rf.review_assignment_id = sg.assignment_id;
40
41  create or replace view review_field_scores as
42  select srf.*,
43         CalculateReviewWeight(type, choices, answer, weight) as score
44  from submission_review_fields srf;
```

## Appendix B

# Explain Analyze Output

Query

```
1  select field_name, priority, type, label,  
2         avg(score) as score,  
3         stddev_pop(score) as stddev,  
4         min(score) as min_score,  
5         max(score) as max_score  
6  from review_field_scores rc  
7  where assignment_id = 1  
8  group by field_name, priority, type, label;
```

Explain analyze output

```
1  -> Table scan on <temporary> (actual time=509.390..509.392 rows=19  
   ↳ loops=1)  
2  -> Aggregate using temporary table (actual time=509.388..509.388  
   ↳ rows=19 loops=1)  
3  -> Nested loop left join (cost=10628966.20 rows=106228476)  
   ↳ (actual time=1.453..335.763 rows=4952 loops=1)  
4  -> Table scan on sg (cost=1544.79..1636.75 rows=7159)  
   ↳ (actual time=1.299..1.330 rows=87 loops=1)  
5  -> Union materialize with deduplication  
   ↳ (cost=1544.78..1544.78 rows=7159) (actual  
   ↳ time=1.299..1.299 rows=87 loops=1)  
6  -> Nested loop inner join (cost=723.64 rows=7046)  
   ↳ (actual time=0.093..0.093 rows=0 loops=1)  
7  -> Nested loop inner join (cost=1.41 rows=3)  
   ↳ (actual time=0.092..0.092 rows=0 loops=1)
```

```

8         -> Covering index lookup on ag using
          ↳ assignment_id (assignment_id=1)
          ↳ (cost=0.35 rows=1) (actual
9         ↳ time=0.092..0.092 rows=0 loops=1)
          -> Index lookup on agm using group_id
          ↳ (group_id=ag.id) (cost=1.06 rows=3)
          ↳ (never executed)
10        -> Index lookup on s using <auto_key0>
          ↳ (group_id=ag.id) (cost=769.43..775.63
          ↳ rows=23) (never executed)
11        -> Materialize (cost=769.14..769.14
          ↳ rows=2294) (never executed)
12        -> Table scan on <temporary>
          ↳ (cost=508.57..539.73 rows=2294)
          ↳ (never executed)
13        -> Temporary table with
          ↳ deduplication
          ↳ (cost=508.56..508.56 rows=2294)
          ↳ (never executed)
14        -> Filter:
          ↳ (user_submissions.group_id
          ↳ <> 0) (cost=279.15
          ↳ rows=2294) (never executed)
15        -> Table scan on
          ↳ user_submissions
          ↳ (cost=279.15 rows=2549)
          ↳ (never executed)
16        -> Table scan on s (cost=101.41..105.28 rows=113)
          ↳ (actual time=1.162..1.169 rows=87 loops=1)
17        -> Materialize (cost=101.38..101.38 rows=113)
          ↳ (actual time=1.162..1.162 rows=87 loops=1)
18        -> Table scan on <temporary>
          ↳ (cost=86.26..90.13 rows=113) (actual
          ↳ time=1.145..1.151 rows=87 loops=1)
19        -> Temporary table with deduplication
          ↳ (cost=86.23..86.23 rows=113)
          ↳ (actual time=1.143..1.143 rows=87
          ↳ loops=1)
20        -> Filter:
          ↳ (user_submissions.group_id = 0)
          ↳ (cost=74.98 rows=113) (actual
          ↳ time=0.231..0.985 rows=1125
          ↳ loops=1)

```

```
21         -> Index lookup on
           ↳ user_submissions using
           ↳ assignment_id
           ↳ (assignment_id=1)
           ↳ (cost=74.98 rows=1125)
           ↳ (actual time=0.226..0.918
           ↳ rows=1125 loops=1)
22     -> Nested loop inner join (cost=1533.74 rows=14839)
           ↳ (actual time=0.017..3.841 rows=57 loops=87)
23     -> Filter: (ur.assignment_id = sg.assignment_id)
           ↳ (cost=24.68 rows=93) (actual time=0.010..0.137
           ↳ rows=57 loops=87)
24     -> Index lookup on ur using user_target
           ↳ (user_target=sg.submitter_user_id) (cost=24.68
           ↳ rows=93) (actual time=0.010..0.129 rows=103
           ↳ loops=87)
25     -> Filter: (ur.`name` = f.`name`) (cost=0.00 rows=160)
           ↳ (actual time=0.010..0.065 rows=1 loops=4952)
26     -> Table scan on f (cost=0.00 rows=160) (actual
           ↳ time=0.001..0.055 rows=160 loops=4952)
```



## Appendix C

# Anonymization Query

```
1  update users
2  set
3      name = concat('User ', id),
4      email = concat('user', id, '@ntnu.no'),
5      password = '...'
6  where id != 1; -- admin user is 1
```

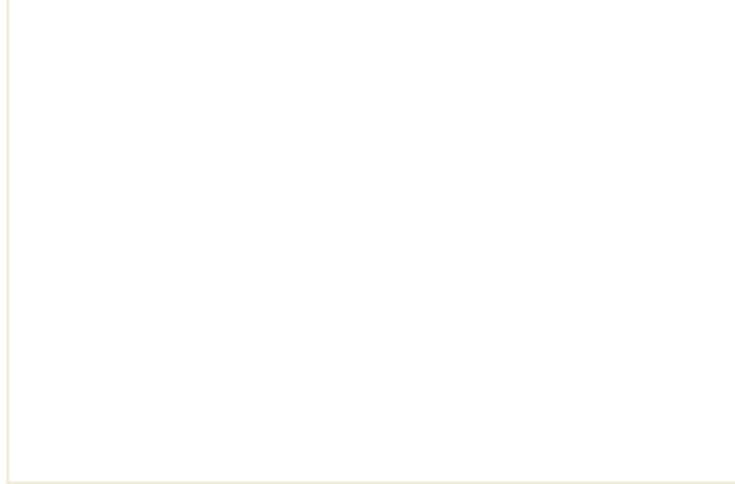
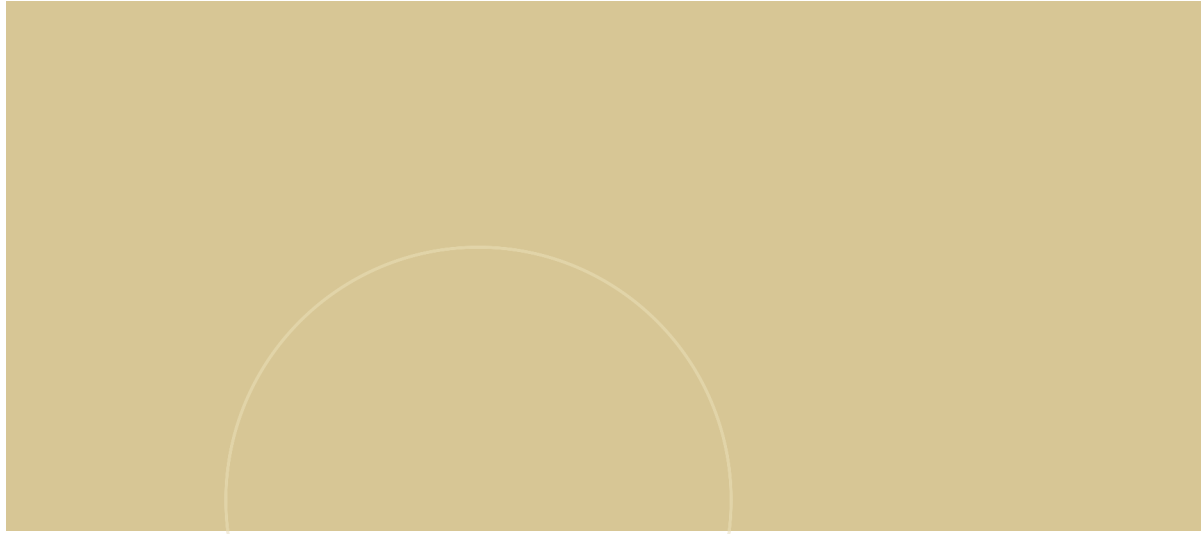




## Appendix D

# Create Materialized View

```
1  create table review_field_scores_mat as
2  select * from review_field_scores;
3
4  SET GLOBAL event_scheduler = ON;
5
6  DELIMITER |
7  CREATE EVENT update_review_field_scores_mat
8      ON SCHEDULE EVERY 12 hour
9      DO
10     BEGIN
11         start transaction;
12         truncate table review_field_scores_mat;
13         insert into review_field_scores_mat
14             select * from review_field_scores;
15         commit;
16     END |
17  DELIMITER ;
```



 **NTNU**

Norwegian University of  
Science and Technology