

# ECS Final Year Project Marking System (FYPMS)

Moo Plah Soe Kusay

**Abstract**—The Final Year Project Marking System (FYPMS) addresses the need for an efficient assessment process for Engineering and Computer Science (ECS) final year projects at Victoria University of Wellington. These projects carry significant weight within the ECS curriculum, characterised by unique individual projects and multiple examiners involved in grading. Moreover they are worth double the points of other ECS courses. The current reliance on the ECS Assessment System for marking presents challenges, including manual cross-referencing and reduced productivity for both examiners and course coordinators. To address these challenges, the development of a dedicated web-based marking system is proposed and implemented. This system is designed to optimise the assessment process with distinct user interfaces for course coordinators and examiners. Course coordinators can monitor progress and ensure fairness in grading, while examiners can focus on efficiently marking their assigned submissions. The web-based system is intended to elevate the grading and feedback process, ultimately benefiting both students and faculty. The chosen technology stack includes TypeScript and Next.js for full-stack web development. Microsoft's Fluent UI Library is used for visually consistent components, with Tailwind CSS introduced for enhanced design flexibility. Meanwhile, a MySQL database and schema serves as the foundation for the system's relational data storage. An evaluation of the system's effectiveness and usability stems from a questionnaire-based survey conducted on faculty staff. Finally, future work is presented, which is now possible due to the completion of the marking system.

## I. INTRODUCTION

The Final Year Project, facilitated by the courses ENGR489 and COMP489, plays a vital role in the Engineering and Computer Science (ECS) curriculum at Victoria University of Wellington. It offers students the opportunity to utilise the knowledge and skills accumulated throughout their studies. The Final Year Project is unique in that students are assigned their own unique project and are supervised and examined by different lecturers. Spanning 2 trimesters, it is also worth twice the points that other courses are worth. Considering this high weightage, marking each project adequately and fairly is of high importance. The current approach to marking is using the ECS Assessment System, which is utilised throughout all ECS courses and is mainly used by tutors. This is sufficient for general ECS courses but falls short in addressing the unique requirements of the Final Year Project courses.

To mitigate the potential bias from supervisors marking their own students' work, the current approach involves additional examiners reviewing each submitted project. One drawback of the current system is that all users see the same view, including all student submissions. This makes it difficult for examiners to view which submissions they are assigned to.

Currently, examiners have to cross-reference a separate list of students, such as in an Excel Spreadsheet, to see who

they have to mark, before looking up the students in the assessment system. This has been a tedious and inefficient task. Considering the amount of marking needed to be done is significantly increased due to having multiple markers per submission, this creates an issue of productivity loss for both examiners and course coordinators. From a course coordinator point of view, manual tasks such as keeping track of examiner marking progress and determining when a submission has a high variance in grades between multiple examiners, also results in productivity loss. The considerable weightage of the Final Year Project and its unique requirements necessitate a dedicated system which would improve organisational productivity for both examiners and the course coordinators. The current system for marking final year projects leaves room for organisational productivity loss. This project aimed to design, develop, and evaluate a new system for examining final year projects, within the span of a school term.

The central issue to address was the development of a new system capable of meeting the needs of the Final Year Project, enhancing the efficiency of the marking process, and improving the user experience for both examiners and course coordinators. The proposed solution involved the creation of a web application for marking final year project deliverables, such as reports and artefacts. The application is named the ECS Final Year Project Marking System (FYPMS). FYPMS aims to streamline the marking process by allowing examiners to view their assigned submissions without cross referencing a separate list. The solution also aims to facilitate the provision of grades by multiple examiners and enable course coordinators to monitor marking progress and quickly identify significant variance in grades. The system's efficacy has been evaluated through a questionnaire survey involving 11 ECS lecturers.

Considering the web-based nature of this project as well as the small scope of the project, environmental and sustainability impacts were relatively minimal. Nevertheless, the project strove to be as sustainable as possible in order to comply with the United Nations 17 goals for sustainable development [1], especially the 12th goal of responsible consumption and production. Given that the project had no budget, the solution avoided unnecessary consumption of resources. While energy use is inevitable in software development, efficiency and speed in the development process can minimise this impact. To limit resource usage, the system was not deployed to any remote server until user testing commenced.

## II. RELATED WORK

Numerous online marking and grading systems have been developed to streamline the assessment process in educational institutions. In recent years, researchers have explored various

aspects of these systems, shedding light on their functionalities, usability, and effectiveness.

#### A. An online marking system conducive to learning

In their 2016 paper, “An online marking system conducive to learning” [2], Z. Jianxiong et al. introduced an online marking and storage system designed to streamline marking processes. Within this system, teachers are assigned specific components to assess, this was proposed to enhance marking efficiency and teachers’ valuable time. The system employs a Browser/Server structure as opposed to a Client/Server structure, allowing teachers to access and operate the system from any location with internet access. The business logic of the application is implemented on the application server-side, while the user’s response behaviour is handled at the web server. The system’s database is built on Microsoft SQL Server, which was chosen for its performance, scalability, reliability, and compatibility with other server software integrations. The researchers highlighted the software’s scalability in distributed organisations. The database is composed of 3 tables, Teacher\_Info, Exam\_Info, and Student\_Info, Exam\_Info having relation to Teacher Info and Student Info, whilst Student Info has a relation on Teacher Info. To ensure fair grading, the system assigns each subjective question to at least 2 teachers for marking. If there is a significant difference between the scores given by the teachers, a team leader is responsible for determining the final score. This approach helps reduce subjectivity in grading. The system allows real-time score entries and maintains records of examiners and paper reviewers’ information, facilitating the calculation of averages and identification of errors. The practical results of testing the system on an exam showed that it generally improved marking efficiency. However, the research has some limitations. User feedback was not gathered during the development or evaluation of the system, so it is unclear how users perceive its usability or user interface.

The paper provided insight into the requirements of the online marking system, the database design including an entity relationship diagram (ERD), and its approach to handling fair grading. It also presents an architectural design of the system, including the design of the database and the chosen technologies. Additionally, the paper offers an example of how high variance in grading is handled. However, there is still a research gap regarding what would be necessary to fully understand the user experience and the system’s usability

#### B. Web-based Usability Measurement for Student Grading Information System

In their 2018 Study, “Web-based Usability Measurement for Student Grading Information System” [3], Hendra et al. evaluated and examined the implementation of a student grading information system at Atisa Dipamkara High School in Indonesia. Prior to the system, traditional methods, such as Microsoft Excel, were used to manually process student grade reports. However, these processes resulted in difficulties for the homeroom teachers and headmaster in processing and monitoring grades. To address these challenges, the school

developed an online data processing information system for student grade reports. The main goal of implementing this information technology was to enhance efficiency and productivity in education management. Despite the system’s operation for several months, a formal method for measuring its effectiveness and efficiency, and for gauging teachers’ satisfaction, was not in place. In response to this, the study proposed the use of usability measurement, using the USE (Usability, Satisfaction, and Ease of use) Questionnaire. This questionnaire was distributed to 25 teachers at the high school using the new grading system to evaluate its usefulness, satisfaction, and ease of use. The implementation of the system involved three user roles - headmaster, teacher, and student, each with specific functionalities. The study found that perceived usefulness, ease of use, and ease of learning had a significant impact on user satisfaction. However, on an individual level, only usefulness and ease of use significantly influenced satisfaction, while ease of learning did not. The study showed that ensuring a system’s perceived usefulness and ease of use is vital in digital grading systems for improved usability and user satisfaction.

The paper provided a technique for evaluating the system using the USE questionnaire. It also took an in-depth look at the system itself, discussing how it manages various user roles, with distinct user interfaces. Use cases for these different roles are depicted in a Use Case diagram, and the paper includes screenshots of the system to provide a visual aid.

#### C. Usability measurement for SIPROK-MRK

In their 2021 paper “Usability measurement for Sistem Proyek Akhir Manajemen Rekayasa Konstruksi (SIPROK-MRK) measured with the system usability scale” [4], W. Wahiddin et al. discuss an online thesis examination system developed during the COVID-19 pandemic for the Construction Engineering Management study program. The system is named SIPROK-MRK, and its usability was evaluated using the System Usability Scale (SUS). The study involved the use of the questionnaire which featured 16 questions, rated on a 7-point scale from Strongly Disagree to Strongly Agree. It assessed factors such as ease of use, error correction, clarity of information, interface quality, and overall satisfaction with the system. The paper also examines the implementation of the system. SIPROK-MRK utilised Laravel, a PHP-based full-stack web framework, as well as a MySQL database. Its features are tailored for lecturers, students, and final project examination committee members. Since the development time for the software was only 1 month, the system was built using the prototype method. The system includes functionalities such as scheduling, grading, and revisions management. Results from the user questionnaire showed that the system had a usability score of 0.82, which is the total score of respondents over the maximum score. W. Wahiddin et al. emphasise that this usability measurement is valuable for future iterations and improvements to the system.

The paper provided techniques on rapid development, as well as providing the technology used such as a MySQL database and a full-stack web framework. The paper also

provided a way to evaluate the system using a questionnaire based survey.

#### D. Summary

These previous studies offer valuable insights into the design, development, and usability of online marking and grading systems in educational settings such as: the importance of scalability, database structure, usability, and user satisfaction. While existing literature provides a foundation for understanding online marking systems, FYPMS's novelty lies in its dedicated focus on addressing the specific needs of the Final Year Project at ECS. By designing and developing a web application tailored to these unique requirements, this project aimed to enhance efficiency, streamline the marking process, and improve the overall user experience for examiners and course coordinators. The literature has offered various technology choices and techniques for system evaluation, which have inspired FYPMS. These inspirations include the use of a full-stack web framework, the use of a relational database, and the incorporation of user testing for evaluation. The literature also provided approaches to designing and creating diagrams for the system.

### III. DESIGN

The development process began with the identification of core requirements. The system is designed to cater to two distinct end-user roles: Course Coordinators and Examiners, each having their own dedicated views. However, Course Coordinators should have the flexibility to switch between Coordinator and Examiner views. The list of system requirements encompasses functionalities for both user types, including data entry, data management, sorting and filtering capabilities, and real-time features.

The Course Coordinator / Admin should be able to:

- **Req1** View list of assessment items
- **Req2** View each item's submissions
- **Req3** View list of examiners
- **Req4** View examiners' assigned submissions and marking statuses
- **Req5** View list of students
- **Req6** View students' submissions
- **Req7** Sort and filter each list
- **Req8** See if a submission has high variance in given grades
- **Req9** Add discussion notes for a submission
- **Req10** Remind examiners of high variances or late marks
- **Req11** Share / export grades for all or selected item submissions

The Examiner should be able to:

- **Req12** View their own assigned submissions
- **Req13** Search, sort, and filter assigned submissions
- **Req14** Enter and edit marks, including comments, for submissions if marking is open for that submission
- **Req14** Add discussion notes to a submission if they are the primary supervisor of the student
- **Req15** See if a submission has high variance in marks if they are the primary supervisor of the student

#### A. Use cases

A use-case diagram was designed to show the use-cases of the system. These use-cases are intended to facilitate the functional requirements identified throughout the requirements gathering phase. *Figure 1* Shows the functionality of the system via a use-case diagram.

#### B. User Interface

A number of wire-frames and designs were created for each page. These designs were created in Figma [5]. The first round of designs consisted of low fidelity wire frames of each page in order to provide a sense of layout. The second round of designs consisted of higher fidelity prototypal wire-frames that painted a better picture of what the user interface should look like. The latter designs included designs for individual components, colours, and font styles. 2 wire-frames of the examiner view are shown in *Figure 2* and in *Figure 3*, while 2 wire-frames of the Course Coordinator view are shown in *Figure 4* and *Figure 5*. The complete wire-frames are provided in Section A of the report appendix. The Course Coordinator view consists of 7 views/pages: an items page, item page, submission page, students page, student page, examiners page, and an examiner page. The Examiner view consists of 2 pages: an assigned submissions page and a marking page. The design of the components were based off of Fluent UI, a collection of UX frameworks from Microsoft [6]. Fluent UI was chosen because it adheres to Microsoft's design language. It should be familiar to users, in this case ECS examiners, as Microsoft software is frequently used by staff. This had potential to increase the overall usability of the system.

The design of the user interface was important as it provided a clear understanding of which components and pages to implement as well as how to structure and style the interface, thus saving time having to make decisions during the development phase.

Title	Username	Date Submitted	Marking Due
Final Report	smithjan	14/10/2023 18:58:00	23/10/2023
Final Report	smithjan	14/10/2023 19:58:00	23/10/2023
Final Report	smithjan	14/10/2023 20:58:00	23/10/2023
Final Report	smithjan	14/10/2023 21:58:00	23/10/2023
Final Report	smithjan	14/10/2023 22:58:00	23/10/2023
Final Report	smithjan	15/10/2023 07:30:00	23/10/2023
Final Report	smithjan	15/10/2023 07:45:00	23/10/2023

Fig. 2. Examiner view of assigned submissions



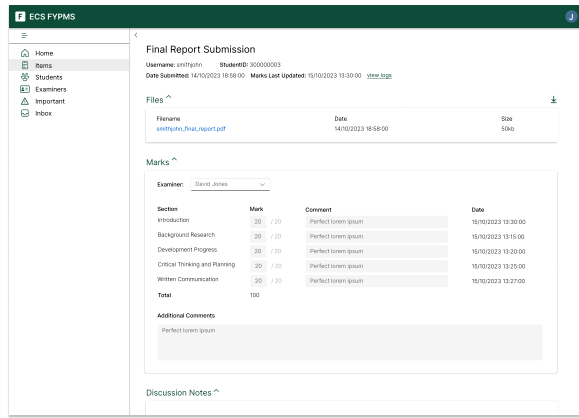


Fig. 5. Course coordinator view of submission page

### C. Entity Relationship

A database was necessary for the project to ensure that data is up to date and consistent for multiple users. In this case, a relational model was chosen over other models, such as NoSQL, because of the robustness offered. Literature has also shown the efficacy of a relational model in similar systems [2][4]. The database is designed with the system's functional requirements in mind. In total there are 13 tables. The Examiner, AssessmentItem, and Student tables were designed first, with subsequent tables being dependent on them.

Figure 6 shows an ERD of the system's database. Here is a breakdown of each table:

- **AssignedMarking** tracks which examiners are assigned to which submissions. It also stores additional comments left by examiners as this is currently a one-to-one relationship.
- **DiscussionNote** stores discussion notes related to a submission and which examiner wrote them.
- **Examiner** stores information about examiners, including their names, email addresses, and whether they are administrators.
- **MarkingEntry** represents individual marking entries.
- **MarkingLogsEntry** logs changes to completed submission marks given by examiners.
- **MarkingSchemeEntry** contains details about marking scheme sections. Multiple marking scheme entries make up one assessment item's marking scheme.
- **Student** holds information about students, including their names and usernames.
- **Submission** stores information about student submissions, including item, student, final grade, submission date, and more. There is one unique AssessmentItem and one unique Student to each submission.
- **Supervising** manages the supervisor relationship between examiners and students, specifying whether an examiner is a primary supervisor.
- **UploadedFile** keeps track of files uploaded for a specific submission.
- **AssessmentItem** contains details about assessment items, including titles, descriptions, due dates, marking dates, marking guides, and variance thresholds.

- **MarkingVariance** tracks variances in marking scores for specific marking scheme sections and submissions.
- **Notification** stores notifications, including dates, titles, messages, and a read status.

## IV. IMPLEMENTATION

### A. Next.js

The application was developed using Next.js, a full-stack React-based framework known for its capabilities in server-side rendering and static site generation [7][8]. Next.js's ability to perform server-side rendering was advantageous in enhancing page load performance and improving the overall user experience. Server-side rendering ensured that content was readily available to users without additional client-side processing. Statically generated pages were useful in cases where data remained static, such as student details, examiner details, and assessment item details. Next.js also allowed FYPMS to function as a single page app (SPA) [9], offering users the ability to navigate between routes without the need for a full web-page reload. This feature contributes to a smoother and more responsive user experience. Next.js also provides an integrated routing system that simplifies navigation within the application, eliminating the necessity to build the routing system from scratch. Furthermore, TypeScript was the chosen programming language for FYPMS because of its type safe system [10]. Next.js seamlessly integrates with TypeScript, allowing for better error checking, improved tooling support, and enhanced development speed and reliability.

Other React frameworks can achieve similar results, such as Remix [11] or Gatsby [12], however Next.js tends to be more suitable for scalability [13].

The App Router [14] in Next.js was chosen over the alternative Pages Router [15] for the sake of a better developer experience. However, this choice did present some challenges, as the app directory, being relatively new, had less documentation and lacked the battle-testing that the Pages Router had. Nevertheless, choosing the App Router was a proactive measure to future-proof the system, as the Pages Router will eventually be phased out [16].

### B. User Interface

The user interface (UI) of the application was implemented using a combination of Tailwind CSS [17] and Microsoft's Fluent UI Component Library [18]. The Fluent UI Library was chosen for its wide range of pre-built and customisable components that align with Microsoft's design, as discovered during the design phase. Using these pre-built and customisable components allowed for quick development, avoiding the need to build the components from scratch.

A challenge faced during the development process was the need to upgrade to a newer version of the component library, Fluent UI 9 [19]. This involved migrating all the components to a different library, which took up some valuable time. Another problem introduced by this migration was that some components that were available in the old version were not available in the newer version, such as a search bar component

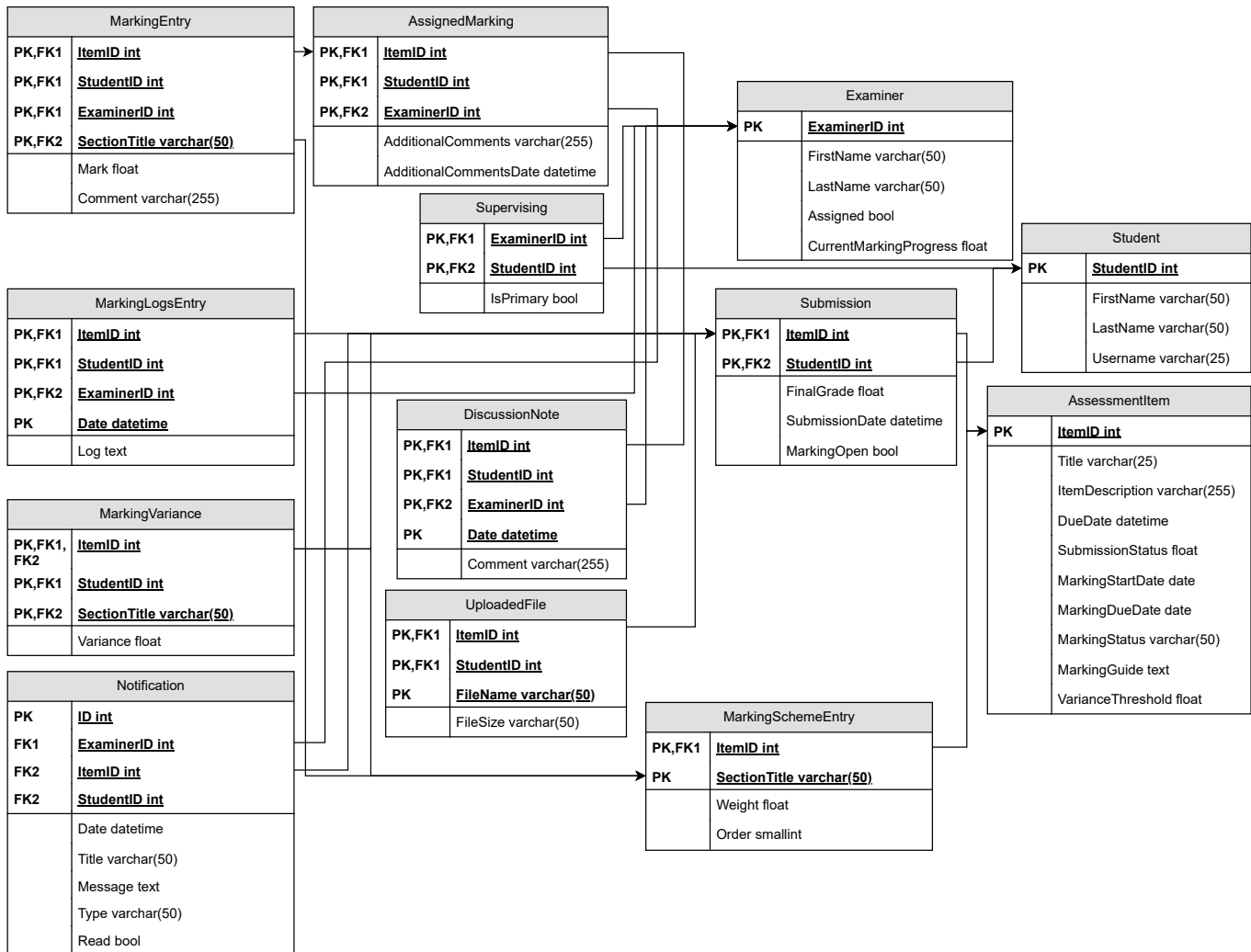


Fig. 6. Entity Relationship Diagram

and a drawer component. These components had to be custom-built with the help of Tailwind CSS. However, the upgrade was justified by the newer version's introduction of lightweight components. Lightweight components had the potential to improve the application's loading times, which would lead to a better user experience. This move was also a proactive measure to future-proof the application, avoiding having to rely on deprecated components in the future.

Tailwind CSS played an important role in enhancing the developer experience by enabling quick styling and layout of components through its utility classes. This approach significantly reduced the time and effort required compared to traditional CSS. The use of Tailwind also facilitated quick changes at the component level without concerns about shared styles or CSS classes.

### C. Authentication

User authentication was a fundamental aspect of the application, given the importance of system security. Non-examiners or course coordinators should not have access to the system,

and similarly, non-course coordinators should not be able to access course coordinator or admin views. Authentication served to safeguard data, including student submissions and grades, while preventing unauthorised tampering with this information. NextAuth.js [20], an authentication library for Next.js, was integrated to provide secure and seamless user authentication to meet these requirements. NextAuth was chosen because of its high compatibility with Next.js including the new App Router. NextAuth.js offered various authentication options, including support for different providers. For FYPMS, the credentials provider was chosen, with a basic password and email setup. This configuration was straight forward and met the project's requirements.

### D. MySQL Database with Prisma ORM

The back-end of the system required a robust and reliable database solution to ensure that data is kept up to date. MySQL, a widely-used relational database, was selected for its suitability in managing structured data efficiently and securely. The system required a stable database to store various types

of data, including student submissions, examiner information, and assessment items to mitigate any inconsistencies. Such inconsistencies could potentially result in bugs, resource-intensive queries, infinite loops, and user data visibility issues, all of which would impact the user experience negatively. MySQL's established robustness and reliability made it the ideal choice for this purpose.

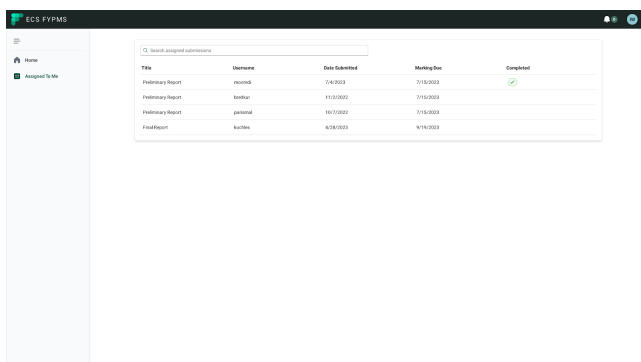
To interface with the MySQL database, Prisma ORM (Object-Relational Mapping) was used [21]. This choice was driven by MySQL's compatibility with Prisma, simplifying database operations like querying and data manipulation. Additionally, Prisma integrates well with Next.js, which allowed for a good developer experience.

### E. Requirements Fulfilment

Some features listed as requirements were not implemented. This was due to the short development time and the need to prioritise certain features over others. In the end, 4 of the 15 requirements were not properly met, these were **Req9**, **Req10**, **Req11**, **Req14**, and **Req15**. Discussion Notes were de-prioritised in the implementation due to their real-time data requirements which would have taken up more development time than the project allowed for, and because this functionality could be simulated externally, for instance, through email discussions. Similarly, the requirement of administrators sending notifications to examiners was also de-prioritised. However, notable features that were implemented included high variance detection and notifications to course coordinators.

A deviation in the implementation and design was removing the two distinct Course Coordinator and Examiner views. In the end, the Course Coordinator was given protected routes for admin activities, accessible via a separate tab in the navigation bar. This decision was driven by the limited number of final pages and the convenience it offered the Course Coordinator to easily switch between the different pages.

2 screenshots of the Examiner view can be found in *Figure 7* and *Figure 8*, while 2 screenshots of the Course Coordinator view can be found in *Figure 9* and *Figure 10*. Screenshots of all the implemented pages can be found in Section B of the report appendix.



File	Username	Date Submitted	Marking Due	Completed
Preliminary Report	jeffrey	1/15/2023	1/15/2023	✓
Preliminary Report	jeffrey	1/15/2023	1/15/2023	
Preliminary Report	jeffrey	1/15/2023	1/15/2023	
Final Report	jeffrey	4/18/2023	4/18/2023	

Fig. 7. Examiner Assigned Items Page

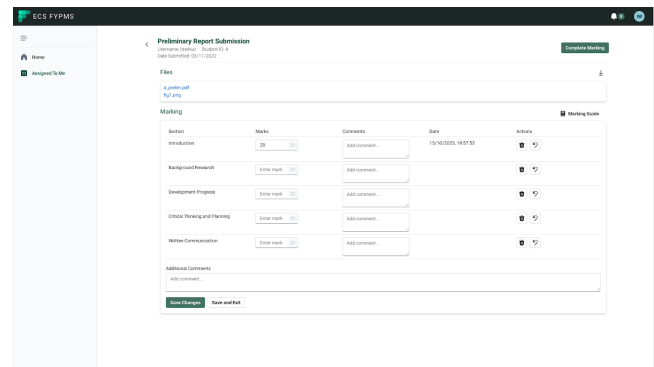
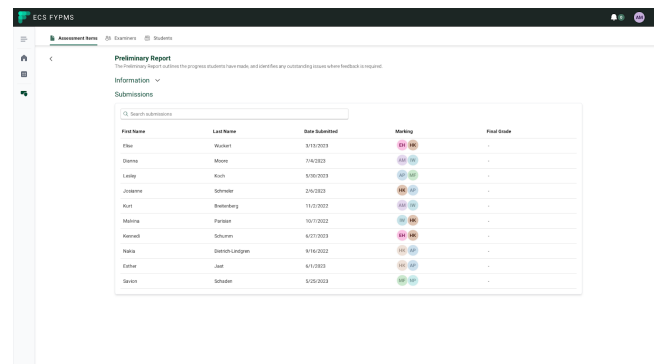
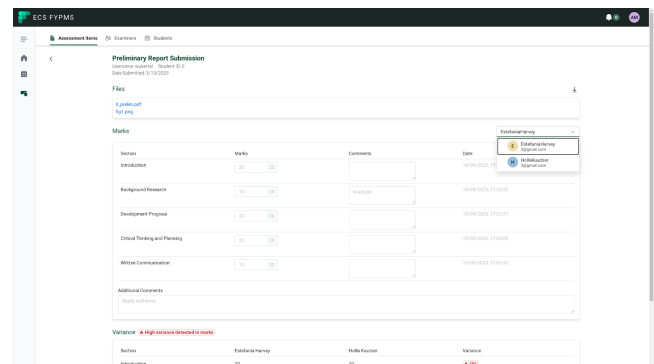


Fig. 8. Examiner Marking Page



First Name	Last Name	Date Submitted	Marking	Final Mark
File	Wicket	5/15/2023	🟡 🟢 🟠	-
Dennis	Moss	1/4/2023	🟡 🟢 🟠	-
Lenny	Koch	5/10/2023	🟡 🟢 🟠	-
Jessica	Spencer	2/6/2023	🟡 🟢 🟠	-
Art	Benderling	11/5/2022	🟡 🟢 🟠	-
Melissa	Parsons	10/1/2022	🟡 🟢 🟠	-
Kenned	Schuren	6/21/2022	🟡 🟢 🟠	-
Nadia	Deborah-Lutzman	6/16/2022	🟡 🟢 🟠	-
Esther	Jost	6/11/2022	🟡 🟢 🟠	-
Sarah	Schuler	5/25/2022	🟡 🟢 🟠	-

Fig. 9. Course Coordinator Assessment Item Page



Variance: ▲ High variance detected in marks!

Section	Final Mark	Marking Due	Variance
Introduction	20	1/15/2023	🟡 🟢 🟠

Fig. 10. Course Coordinator Submission Page

### F. Architecture

The architecture of the final implementation can be found in *Figure 11*. There are slight deviations from the initial design, which can be found in Section C of the report appendix. One difference is the integration of the authentication system within the application, as opposed to having it be an external service, with direct communication to the database for user credential validation. Another change includes the addition of Pusher.js [22], a real time data service, which currently facilitates the notifications feature. One more change involves the internal division of the app into client components and server-rendered components/pages. Server-rendered pages are hosted on the server, enabling them to directly manage data, while client

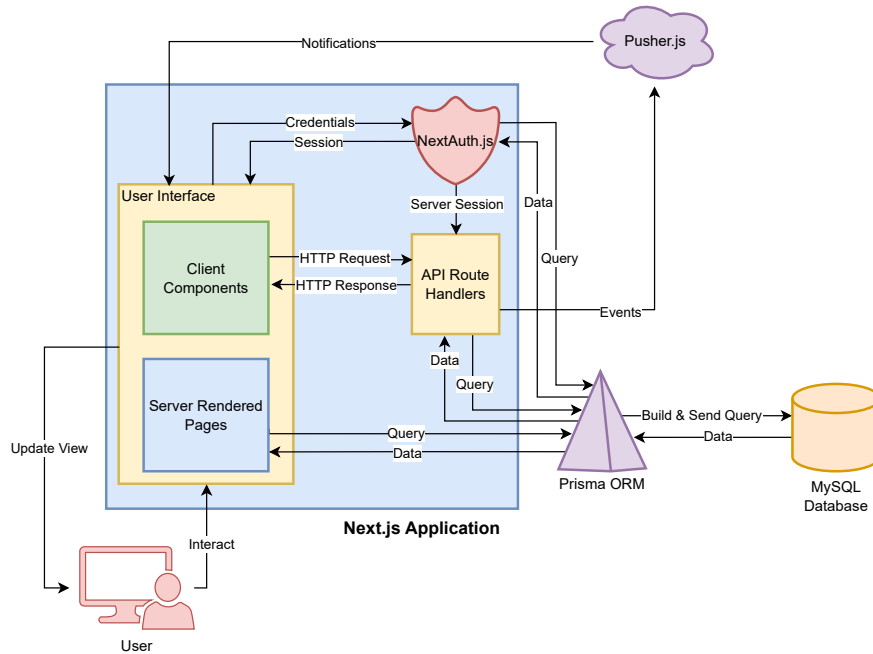


Fig. 11. System Architecture Diagram

components rely on a Restful API to request and receive data. Apart from these distinctions, the data flow remains straightforward. Users interact with the user interface, which fetches and receives data, the Prisma ORM intermediates between the application and the MySQL database to construct queries and retrieve data. Meanwhile, the authentication system receives credentials from the front-end and provides session protection, ensuring authorised access to both the pages and API endpoints. Lastly, changes in the data are reflected in the user interface.

## V. EVALUATION

The evaluation of the system was done with a user test, to evaluate the system’s usefulness, satisfaction, and ease of use. In total, 11 staff at ECS were recruited. The recruitment of ECS staff was done for 2 main purposes. 1: Convenience sampling, as staff are readily available where the project was being developed, and 2: ECS staff are intended to be the end user of the system, so their feedback is valuable. For convenience, the data collection was performed at the offices of the participants. Participants used their own computers, with access to the internet, to access the system. This was done in person to allow the participant to request assistance and to allow easy communication between the participant and the conductor. Because the user testing would be conducted on ECS staff, who are not covered by ECS’s approval for Usability Testing [23], a formal application needed to be submitted to Victoria University of Wellington’s Ethics Committee [24]. Relevant user testing documents and ethics application documents are found in Section D of the report appendix.

### A. Methodology

Each participant received an identical set of tasks, aiming to replicate real-world system usage, encompassing activities like data entry, navigation, and information gathering. These tasks were categorised into two segments: “Examiner Activity” and “Course Coordinator Activity.” The first set of tasks simulated the responsibilities of an examiner, involving the completion of all marking assignments assigned to the examiner. The tasks involved reviewing submitted files, accessing the marking guide, inputting marks and comments, as well as saving and finalising each marking.

The second set of tasks simulated the role of overseeing the marking progress and included accessing the notifications menu to observe a high variance alert. Additionally, participants were asked to compare the marks provided by two examiners involved in the high variance submission. Furthermore, they were tasked with finding more information about these two examiners and their other completed marking assignments.

Upon the completion of these tasks, each participant was instructed to complete a questionnaire, specifically the USE (Usefulness, Satisfaction, and Ease of Use) Questionnaire [25]. Participants were required to rate each question on a scale ranging from 1 to 5, where 1 signified “Strongly Disagree” and 5 represented “Strongly Agree”, participants could also answer a question as “N/A” or not applicable. An example of this situation was the question “It is fun to use” as many participants did not find the marking process enjoyable. When a question received 2 or more “N/A” responses, it was excluded from the final calculation of mean scores. This was an easy way to determine irrelevant questions. This choice was made to ensure data integrity, as using responses to these questions for drawing conclusions could be unreliable due to the limited



TABLE I  
USE QUESTIONNAIRE RESULTS

Question	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	Mean	SD
"It helps me be more effective."	4	4	3	4	3	4	3	4	3	4	5	3.73	0.65
"It helps me be more productive."	4	4	3	4	3	4	4	5	4	4	5	4	0.63
"It is useful."	4	4	4	4	4	4	4	5	4	5	5	4.27	0.47
"It makes the things I want to accomplish easier to get done."	3	4	4	4	4	4	4	5	4	4	3	3.9	0.54
"It saves me time when I use it."	3	5	3	4	4	4	4	5	4	4	5	4.09	0.7
"It meets my needs."	4	3	4	4	3	4	n/a	5	3	5	4	3.9	0.74
"It does everything I would expect it to do."	4	4	4	4	3	4	4	3	3	5	4	3.82	0.6
"It is easy to use."	5	4	4	5	5	4	4	4	4	5	4	4.36	0.5
"It is simple to use."	4	4	4	5	5	5	4	4	4	4	4	4.27	0.47
"It is user friendly."	3	3	3	5	4	4	4	4	4	4	4	3.82	0.6
"It requires the fewest steps possible to accomplish what I want to do with it."	4	4	3	5	4	4	5	3	3	2	4	3.73	0.9
"Using it is effortless."	4	4	4	5	4	3	3	3	3	3	4	3.64	0.67
"I can use it without written instructions."	4	2	5	5	4	4	4	4	4	5	3	4	0.90
"I don't notice any inconsistencies as I use it."	5	5	5	5	5	4	4	3	4	5	4	4.45	0.69
"Both occasional and regular users would like it."	4	4	3	5	5	4	2	4	4	5	n/a	4	0.94
"I can recover from mistakes quickly and easily."	4	5	4	5	5	3	2	3	3	2	4	3.64	1.12
"I learned to use it quickly."	5	4	4	5	5	4	4	4	4	4	5	4.36	0.5
"I easily remember how to use it."	4	4	5	5	5	5	4	4	4	4	5	4.455	0.52
"It is easy to learn to use it."	5	3	4	5	5	4	4	4	4	4	5	4.27	0.65
"I quickly became skillful with it."	4	5	4	5	5	4	2	4	3	4	4	4	0.89
"I am satisfied with it."	5	4	4	4	4	4	4	4	3	4	5	4.09	0.54
"I would recommend it to a friend."	4	4	4	4	n/a	4	4	4	4	4	5	4.1	0.32
"It is wonderful."	4	3	5	4	4	4	n/a	3	4	4	4	3.9	0.57
"I feel I need to have it."	5	4	4	4	n/a	4	4	4	3	3	4	3.9	0.57
"It is pleasant to use."	4	3	4	4	n/a	4	4	3	4	4	4	3.8	0.42

number of responses. In addition to the USE questionnaire, participants were also asked to provide feedback regarding both the most negative and most positive aspects of the system. This feedback is intended for guiding future improvements rather than assessing the current implementation.

For each question, a mean score was calculated, and subsequently, a total mean score for each section was calculated. This resulted in the generation of a Usefulness score, a Satisfaction score, and an Ease of Use score. This method of processing USE scores is consistent with other work found in literature such as [26].

### B. Findings

The findings from the user testing revealed positive perceptions of the system's usability and user satisfaction, with scores being above average and low standard deviations (SD) indicating low variability between answers. The mean score for the system's overall usefulness was 3.98, this indicate that participants found the system to be somewhat useful. Additionally, the system was generally regarded as easy to use, as evidenced by a mean Ease of Use score of 4.09, which also included the ease of learning. This suggests that participants found the system user-friendly and approachable. Furthermore, the participants reported a good level of satisfaction with the system, as indicated by a mean satisfaction score of 3.97. This suggests that the system met or exceeded the expectations and needs of the users. Table I shows the results of each respondent along with the mean ratings and SD of the ratings. Table II shows a breakdown of the average means and SDs for each section.

TABLE II  
USE QUESTIONNAIRE SECTION RESULTS

Criteria	Mean Score	SD
Usefulness	3.98	0.62
Ease of Use	4.09	0.69
Satisfaction	3.97	0.55

### C. Limitations

While the evaluation provided valuable insights into the usability, satisfaction, and potential of the ECS final year project, there are some limitations that may impact the generalisability and scope of the findings. The sample size of 11 participants represents only a fraction of potential users. This limits the extent to which the findings can be generalised to a broader and more diverse user base. Additionally, 10/11 of the participants were male. Expanding the participant pool to include a more diverse set of users could provide a broader perspective on the system's usability and effectiveness. The evaluation was conducted in a controlled testing environment, which may not fully mirror the complexities and dynamics of real-world usage scenarios. User interactions and experiences can differ significantly in their day-to-day work compared to a controlled testing setting. While the tasks were designed to simulate real-world scenarios, they may not have encompassed the full spectrum of tasks and responsibilities undertaken by ECS staff. The evaluation provided insights into users' initial experiences with the system. However, long-term usability, user satisfaction, and the evolution of challenges over extended usage have not been fully explored. Furthermore, the evaluation focused primarily on usability and user satisfaction, without delving deeply into context-related factors and specific

challenges that ECS staff members might encounter in their daily tasks. More qualitative research methods, such as in-depth interviews, could provide additional insights into these aspects.

## VI. CONCLUSIONS AND FUTURE WORK

The results of the user testing for the ECS final year project reveal promising outcomes due to an overall positive reception to the system.

Participants found the system to be useful, easy to use, and satisfying, which indicates a strong foundation for its potential. The positive user feedback, along with participants' prior experience with similar tools, underscores the system's viability as an effective tool for ECS staff members. This project showcases the effectiveness of the system in simulating real-world tasks performed by ECS staff, including data entry, marking, and information gathering. The overall user satisfaction and favourable usability scores provide evidence that the system aligns well with the needs and expectations of its intended users.

Building on the results from this evaluation, there are several avenues for future work. Future development should focus on integrating additional features to enhance the system's capabilities, such as adding discussion notes. Valuable user feedback collected during the testing phase should be also be incorporated into updates to the system. Regular user feedback sessions would also help in refining the system's user experience and identifying any evolving user needs. This will ensure that the system continues to evolve to meet user needs and preferences. To improve data management and streamline administrative tasks, integrating the system's back end with the school's existing back-end infrastructure is essential. This integration would facilitate seamless data exchange and enhance the overall efficiency of the system. It is also important to design the system with scalability in mind. Future work should focus on ensuring that the system can accommodate a growing user base and increasing data volumes. Additionally, making the system accessible to a wide range of users, including those with disabilities, should be a core consideration in future development.

The ECS Final Year Project Marking System has shown promise in enhancing the efficiency of ECS staff's tasks. With continued development, feature integration, and responsive updates based on user feedback, the system has the potential to become a tool for the ECS department and contribute to a more streamlined and productive environment.

## REFERENCES

- [1] United Nations, "The 17 sustainable development goals," United Nations, 2015. <https://sdgs.un.org/goals>
- [2] J. Zhou, J. Yang, H. Song, S. H. Ahmed, A. Mehmood, and H. Lv, "An online marking system conducive to learning," *Journal of Intelligent & Fuzzy Systems*, vol. 31, no. 5, pp. 2463–2471, Oct. 2016, doi: <https://doi.org/10.3233/jifs-169088>.
- [3] Hendra, S.Kom., M.T. and Y. Arifin, S.Kom., M.M., "Web-based Usability Measurement for Student Grading Information System," *Procedia Computer Science*, vol. 135, pp. 238–247, 2018, doi: <https://doi.org/10.1016/j.procs.2018.08.171>.

- [4] W. Wahiddin, A. R. Syulistyo, and M. N. Shoumi, "Usability measurement for Sistem Proyek Akhir Manajemen Rekayasa Konstruksi (SIPROK-MRK) measured with the system usability scale," *IOP Conference Series: Materials Science and Engineering*, vol. 1073, no. 1, p. 012015, Feb. 2021, doi: <https://doi.org/10.1088/1757-899x/1073/1/012015>.
- [5] Figma, "Figma: the collaborative interface design tool.," Figma, 2022. <https://www.figma.com/>
- [6] "Home - Fluent UI," [developer.microsoft.com](https://developer.microsoft.com/en-us/fluentui/#/).
- [7] Vercel, "Next.js by Vercel - The React Framework," [nextjs.org](https://nextjs.org/).
- [8] Meta Open Source, "React," [react.dev](https://react.dev/), 2023. <https://react.dev/>
- [9] "SPA (Single-page application) - MDN Web Docs Glossary: Definitions of Web-related terms — MDN," [developer.mozilla.org](https://developer.mozilla.org/en-US/docs/Glossary/SPA).
- [10] Microsoft, "TypeScript - JavaScript that scales.," [Typescriptlang.org](https://www.typescriptlang.org/), 2015. <https://www.typescriptlang.org/>
- [11] "Remix - Build Better Websites," [remix.run](https://remix.run/).
- [12] "The Fastest Frontend for the Headless Web," [Gatsby](https://www.gatsbyjs.com/).
- [13] "Next.js vs. Gatsby: The Difference and Which Framework to Choose," [ninetailed.io](https://ninetailed.io/blog/next-js-vs-gatsby/). (accessed Oct. 15, 2023).
- [14] "App Router — Next.js," [nextjs.org](https://nextjs.org/docs/app). (accessed Oct. 15, 2023).
- [15] "Pages Router — Next.js," [nextjs.org](https://nextjs.org/docs/pages). (accessed Oct. 15, 2023).
- [16] "Upgrading: From Pages to App — Next.js," [nextjs.org](https://nextjs.org/docs/pages/building-your-application/upgrading/app-router-migration#nextjs-version). (accessed Oct. 15, 2023).
- [17] tailwindcss, "Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.," [tailwindcss.com](https://tailwindcss.com/), 2023.
- [18] "Home - Fluent UI," [developer.microsoft.com](https://developer.microsoft.com/en-us/fluentui#/controls/web). (accessed Oct. 15, 2023).
- [19] "Fluent UI React," [react.fluentui.dev](https://react.fluentui.dev/?path=/docs/concepts-introduction-page). (accessed Oct. 15, 2023).
- [20] "NextAuth.js," [next-auth.js.org](https://next-auth.js.org/).
- [21] Prisma - Database tools for modern application development, "Prisma," [Prisma.io](https://www.prisma.io/), 2019.
- [22] "Pusher — Leader In Realtime Technologies," [Pusher.com](https://pusher.com/), 2020.
- [23] "Human Ethics Application Application ID : 0000029386 Application Title : Software usability testing for ENGR301." Accessed: Oct. 15, 2023. [Online].
- [24] V. U. of Wellington, "Human Ethics Committee — Research," [Victoria University of Wellington](https://www.wgtn.ac.nz/research/support/ethics/human-ethics-approval/human-ethics-committee), Mar. 21, 2022.
- [25] A. Lund, "(PDF) Measuring Usability with the USE Questionnaire," [ResearchGate](https://www.researchgate.net/publication/230786746), Jan. 2001.
- [26] T. V. Machado Faria, M. Pavanelli, and J. L. Bernardes, "Evaluating the Usability Using USE Questionnaire: Mindboard System Use Case," *Lecture Notes in Computer Science*, pp. 518–527, 2016, doi: [https://doi.org/10.1007/978-3-319-39483-1\\_47](https://doi.org/10.1007/978-3-319-39483-1_47).