CEEP: An Administration Platform to Assist Student Volunteers in Hosting Evaluations to Improve Modules

Authors

Andrea Onofrei
An Pan
Arsalaan Khan
Diana-Alexandra Bozea
Kristiyan Velikov
Sanghyeop Park

Supervisor

Wallace Corbo Ugulino

Abstract

The **Committee for Education Evaluation Panels (CEEP)** is an initiative responsible for organizing and overseeing evaluation panel sessions led by students for the bachelor study programs Technical Computer Science (TCS) and Business and IT (BIT). While currently CEEP encourages participation and constructive feedback, operational inefficiencies hinder its efficacy. To address these challenges, a comprehensive design project has been initiated to create a more efficient system. The proposed platform serves a dual purpose. Primarily, it seeks to provide CEEP members with advanced tools for the seamless monitoring of progress and the refinement of evaluation planning. This also means automation of a lot of tasks previously conducted manually and repeatedly. Additionally, the platform seeks to streamline the process by which students and instructors enrol in evaluations and access past evaluation records, creating a one-stop place for all the CEEP information. This report will present the web platform for the CEEP evaluation panel, explaining in detail the process of its creation and testing and the motivation behind the design choices that were made.

CONTENTS

1 INTRODUCTION

In this chapter, we will explain the background of the project by discussing the current system that is used by the CEEP. Furthermore, we will analyze our system's domains. Domain analysis is needed in order to establish a clear understanding of the development process from the start of the project.

1.1 Background

The Committee for Education Evaluation Panels lacks a dedicated application to meet its requirements. Presently, they rely on a university website page where individuals can learn about CEEP and register for module panel discussions. To facilitate registration, CEEP uses an office form that stores the data on their drive. For communication purposes, such as notifying students about evaluations, inviting coordinators and teachers, arranging for food, and booking evaluation rooms, they manually compose and send emails on each occasion. To access past evaluation records, students and teachers are required to send an email to the CEEP. After the evaluation took place the CEEP members have to first send emails with minutes to all the participating students and if they have no comments, send it to the module coordinator for approval. As evident from the observation, the existing system consumes a significant amount of time for students, teachers, and CEEP members. In section 1.3, a more detailed exploration of the current system's procedures is presented.

1.2 Domain Analysis

1.2.1 Client and Users of the system

The client of the application is the Committee for Education Evaluation Panels. The users of the system are students and teachers of the Technical Computer Science (TCS) and Business and IT (BIT) programs. The domain of our project covers implementing a user-friendly application where students and teachers can both register for upcoming evaluations and see the records for the past ones. Moreover, the system should automate the tasks of the CEEP members e.g. email composition for inviting coordinators and teachers, arranging for food, and booking evaluation rooms and allow for easier collaboration and task division between the CEEP members. For module coordinators, students and CEEP members it simplifies the process of approving and making changes to the report of the evaluation panel. Appendix C, Figure C-1 provides a comprehensive overview of all the project stakeholders.

1.2.2 Software Environment

As the client has not expressed a specific software environment preference for the project, we have made the deliberate choice to employ a programming language that our team is proficient in, combined with a framework known for its ease of use and scalability. Specifically, we have utilized Python in conjunction with the Django framework. Section 4.2 of this document will

provide a more in-depth examination of the technology stack we have selected, along with the reason behind our choice of this system environment.

1.2.3 Similarities to other UT Software

In designing the system, our approach has been driven by the unique user base, consisting of University of Twente teachers and students. Our choice of authentication method aligns with the university's commitment to security and user convenience. We've integrated the OAUTH 2.0 flow, a robust and widely recognized authentication standard used across the university, providing a secure and user-friendly login process. To further enhance the sense of continuity and belonging within the university community, we've adopted the same colour scheme and design elements found on official university websites. This design consistency aims to create a visual connection, making the system feel like a natural extension of the UT digital landscape. By mirroring the design and authentication principles used by the University of Twente, we aim to ensure that our system seamlessly integrates into the existing technological ecosystem, providing a coherent and user-friendly experience for both teachers and students.

1.2.4 Procedures of the Current System

In this paragraph, a more in depth exploration of each of the steps of the CEEP workflow is described.

Contacting the Module Coordinator

The first step for organizing a panel evaluation involves contacting by email the module or course coordinator, informing them of the goal of the CEEP and asking them to participate in the evaluation.

Date and Location selection

The CEEP members have to agree on a date when they want to organize the panel, update the evaluation spreadsheet with the selected date and put the evaluation in the CEEP calendar.After that inform the coordinator and based on the expected number of attendees, arrange a location by writing an email to the Booking Office. If the module coordinator agrees to include the evaluation in the module schedule, then CEEP has to contact the Timetable Office.

Promotion

For promotion, the CEEP members have to send an email to the module coordinator with a promotional text that can be posted on Canvas to gather participants. The promotion text contains a link to the CEEP university website where students can find an office form to register for the panel evaluations.

Arranging for Lunch

At least two days before the evaluation, CEEP members have to write an email to Appèl to arrange lunch.

Evaluation

During the evaluation panel the committee chair leads and facilitates the meeting, encouraging

participants to voice their opinions on the module. The secretary writes down the minutes and includes comments that provide detailed explanations of the participants' opinions.

Aftermath of the Evaluation

After the evaluation session, the secretary finishes the minutes and sends them by email to the participating students. If the students don't have any comments, the secretary sends the report by email to the module coordinator. In case the module coordinator approves it, the secretary has to archive it on the CEEP drive. If necessary the secretary will send the evaluation report to the appropriate program committee.

2 METHODOLOGY AND COLLABORATION APPROACH

In this section, we will discuss the frameworks we used to manage the project and to collaborate with the client

2.1 Frameworks

We have used Agile practices in order to manage this project. We thought Agile would be a good methodology to manage the project as it uses iterative development and allows for easier feedback integration from the client. The term Agile refers to frequent reassessment and adaptation of plans and techniques and dividing tasks into shorter tasks for efficiency. One of the advantages of agile development is that it ensures customer satisfaction since this methodology requires customers to be actively involved in the development process [1]. Since we were also working for a client, satisfying the client's demands was one of our main requirements which led us to choose the agile development method. Particularly we have chosen the SCRUM framework [2], over other Agile frameworks as Kanban or Lean as we wanted to have more structure and well-defined priorities when implementing the project. It is based on an incremental software development process. In the SCRUM method, the entire development cycle is divided into a series of iterations where each iteration is called as a sprint. SCRUM is ideal for rapidly changing, accumulating requirements and the work estimates are much easier. SCRUM also provides customer satisfaction by optimizing turnaround time and responsiveness to requests.

2.2 Requirement Elicitation

From the start of the module, we organized every week meetings with the client, particularly the current CEEP chairman, to ask questions about how the committee envisions the system and update them on our progress. During the first week of the project, the client offered us a project plan created during the BIT module Business Innovation Through IT Projects. The document contained the problem statement and context of the project as well as the requirements. After studying the document we addressed the remaining questions to the client during the next meetings, as we started developing the application. As we mentioned, we have developed the project using the SCRUM framework, and at each sprint review, we were defining new requirements for the project based on the feedback offered by the client.

3 REQUIREMENTS ANALYSIS

3.1 Requirements Proposal

In order to create an efficient and practical web platform, it is imperative to thoroughly identify the specific requirements. This section has organized the requirements into 2 different types: functional and non-functional. To prioritize these requirements using the MoSCoW method, we have categorized them into Must-haves (M), Should-haves (S), Could-haves (C), and Won't-haves (W).

According to a study by Hatton (2008), using a technique such as the MoSCoW method to prioritise gives the development team an excellent starting point to determine value. MoSCoW has been shown to take little time to do, and is easy to understand thus showing that it would be a quick source of information about priorities. It also has the characteristic, that since it assigns an independent classification to each requirement, it would easily deal with the addition of new requirements without voiding the previous prioritisation. [3]

3.1.1 Functional requirements

Functional requirements are features that a developer must implement to allow users to complete particular tasks in the application.

Must-have requirements

1. *The CEEP system must support the CEEP chair in assigning evaluations to CEEP members, facilitating efficient task distribution within the evaluation process.*

This requirement is crucial to ensure that evaluation tasks are allocated to the appropriate CEEP members efficiently. It streamlines the evaluation process by reducing the risk of mismanagement and ensuring that each evaluation is assigned to the right team member.

2. *The CEEP system must enable users to view the evaluation reports, ensuring easy access to essential evaluation data for informed decision-making.*

Providing easy access to evaluation reports is essential for transparency and informed decision-making. This requirement ensures that stakeholders can quickly access and review evaluation data, which is critical for making improvements and informed decisions.

3. *The CEEP system must allow CEEP members to upload evaluation reports, facilitating the seamless sharing of evaluation information within the system.*

This requirement allows CEEP members to share their evaluation reports within the system, ensuring that all relevant data is available for review and analysis. It promotes a centralized and organized approach to evaluation data management.

4. *The CEEP system must enable CEEP members to view the participants associated with specific evaluations, ensuring transparency and accountability within the evaluation process.*

Transparency and accountability are vital in the evaluation process. This requirement allows CEEP members to see who is participating in each evaluation, ensuring that the process is fair and accountable.

5. *The CEEP system must facilitate CEEP members in viewing their tasks, promoting effective time management and coordination within the evaluation process.*

Effective time management and coordination are crucial for the successful execution of evaluations. This requirement ensures that CEEP members can easily monitor tasks and deadlines, reducing the risk of delays or oversights.

6. *The CEEP system must enable students who have signed up for the evaluation to provide comments on the evaluation report, encouraging active participation and feedback from the student body.*

Encouraging student participation and feedback is vital for continuous improvement. This requirement allows students to provide comments on the evaluation report, promoting their active involvement in the process.

Should-have requirements

7. *The CEEP system should enable module coordinators to officially approve the evaluation reports from their respective modules, ensuring the validation and verification of evaluation data by authorized personnel.*

This requirement adds an additional layer of validation to the evaluation process. Module coordinators can officially approve reports, ensuring that the data is verified by authorized personnel.

Could-have requirements

8. *The CEEP system could support CEEP members in creating custom templates, allowing for flexibility in reporting formats as required by the users.*

This requirement offers flexibility to CEEP members by allowing them to create custom report templates. It ensures that reporting formats can be adapted to specific needs and preferences as required by the users.

Won't-have requirements

9. *The system won't integrate with social media platforms for sharing evaluation reports or other system activities.*

While social media integration can enhance user engagement, it won't be a priority during the initial system deployment. This is also because evaluation reports are not something to be shared on social media. The system will primarily focus on core functionalities.

3.1.2 Non-functional requirements

Non-functional requirements are requirements that are concerned with the quality and performance of the software which do not affect the functionality of the application.

Must-have requirements

1. *The system's user interface must be user-friendly, ensuring that all stakeholders, regardless of technical expertise, can easily navigate and use the system.*

A clear and intuitive interface is essential for user satisfaction and adoption.

2. *The system must load user dashboards within 10 seconds and generate evaluation reports in under 20 seconds.*

Quick response times are vital to maintain user productivity and prevent user frustration.

3. *The system must implement robust security measures to protect user data, evaluation reports, and sensitive information by securely storing user passwords in a protected way, using salted and hashed algorithms.*

This requirement is categorized as a "Must-have" because the security of user data and sensitive information is of paramount importance. In the context of an educational institution, protecting user privacy, ensuring data integrity, and preventing unauthorized access are non-negotiable.

Should-have requirements

4. *The system architecture should support horizontal scaling to handle a 50 percent increase in users and data volume over the next 3 years*

Scalability is important for accommodating future growth but may not be an immediate requirement.

5. *The system should be compatible with various web browsers and devices, including the latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari on both Windows and macOS.*

6. *The system could be designed for maintainability, with code documentation and ease of updates.*

While important, maintainability may not have an immediate impact on the initial system deployment and can be addressed in later phases.

Could-have requirements

7. *The system could implement data integrity measures, including data validation rules at the database level, to prevent the storage of invalid or corrupt data.*

While data integrity and auditing are important for maintaining data quality and ensuring data consistency, these requirements are categorized as "Could-have" as they may not be as critical as the "Must-have" security measures.

Won't-have requirements

8. *The system won't focus on immediate interoperability for integrating with other systems.*

While interoperability for potential future integration is a valuable feature, it is not a priority for the initial system deployment.

9. *The system won't focus on multi-language localization features. It will primarily support content in the primary language of the institution.*

While multi-language support can be valuable for international users, the system won't prioritize this feature in the initial deployment. It will primarily support content in the institution's primary language which is English.

10. *The system won't provide a dark mode or an alternative color scheme for user interface customization.*

While dark mode can enhance user experience in certain applications, it won't be a feature in this application. The focus will be on core functionality and user interactions, and customization of the user interface color scheme won't be prioritized.

3.2 User Stories

Identifying the user stories is crucial step to make efficient comprehensive web platform for all users. Within this section, we pinpoint these user stories to develop a comprehensive system that satisfy to all users based on the SMART framework.

Student
1. As a student, I want a seamless sign-up process for evaluation sessions to contribute to the improvement of the study programs, ensuring an efficient and user-friendly experience for student participation
2. As a student, I want to access past evaluation minutes.
3. As a student, I want to see details of the evaluation session.
4. As a student, I want to specify allergies for panel meeting.
5. As a student, I want to upload my feedback for evaluation.

CEEP Member
1. As a member of CEEP, I want to see the dashboard to monitor the progress of evaluations.
2. As a member of CEEP, I want to see details of the participants who join the panel meetings.
3. As a member of CEEP, I want to upload new evaluation meeting plan on web platform.
4. As a member of CEEP, I want to upload/delete the agenda on web platform.

Teacher
1. As a teacher, I want see an intuitive platform for scheduling evaluation sessions
2. As a teacher, I want to see an intuitive platform for enabling me to collect insightful feedback to enhance the quality of the modules.

Module coordinators
1. As a module coordinator, I want to see the dashboard to monitor the past evaluation report.
2. As a module coordinator, I want to approve the evaluation report.
3. As a module coordinator, I want to leave the feedback on evaluation report
4. As a module coordinator, I want to request the changes to the evaluation report
5. As a module coordinator, I want to review my feedback from the evaluation.

4 GLOBAL DESIGN

In this chapter, we have provided an in-depth discussion of the overall design and architectural decisions. We conducted a high-level analysis of the system's structure, and we also offered an overview of the system, outlining its various pages and comprehensive functionalities.

4.1 Global Design Choices

The primary objective behind creating the centralized evaluation platform for CEEP is to simplify the organization of evaluation sessions for the TCS and BIT bachelor programs while also re-configuring the associated workflows and procedures. In doing so, the aim is to automate repetitive routine tasks, thereby establishing a smoother, more efficient, and streamlined workflow. This marks a significant difference from the previous reliance on predominantly manual inputs and data manipulation. Additionally, the system's key design components were identified through an analysis of the client's requirements and preferences, ensuring that the system is made to meet the user's specific needs.

4.1.1 Key Design Elements

The system's design serves to improve work processes and procedures while making the system as user-friendly as feasible. To that end, the system is primarily intended to improve usability and convenience of use. Involving clients in the development process is essential as it promotes collaboration and a user-focused approach. By engaging clients from the very beginning, we gain valuable insights into their specific needs, preferences, and objectives. This not only ensures that the final product aligns closely with their expectations but also mitigates the risk of misunderstandings or misalignments during the project's development. A close collaboration with the client encourages transparency and open communication, enabling real-time feedback and adjustments, which can be crucial in avoiding costly revisions and fixes.

Furthermore, we have developed the application with future enhancements in mind,so as to create a foundation that allows for seamless adaptability and scalability. This proactive approach ensures that as the client's needs evolve, the application can readily accommodate new features and improvements, saving time and resources in the long run. Ultimately, this collaborative partnership promotes a sense of ownership and trust, leading to the creation of a solution that genuinely resonates with the client and fulfills their unique requirements while being well-prepared for future growth.

4.1.2 Redesigned Workflow

The system's main goal is to significantly improve the efficiency and efficacy of the entire process of hosting an evaluation session for the TCS and BIT modules. To achieve this goal, we redesigned the entire workflow. This re-imagined workflow, as shown in Figure 4.1, specifies a series of well-defined tasks and processes meant to assure the smooth and successful execution of a module assessment session. Our mission was to create a solution resulting in a

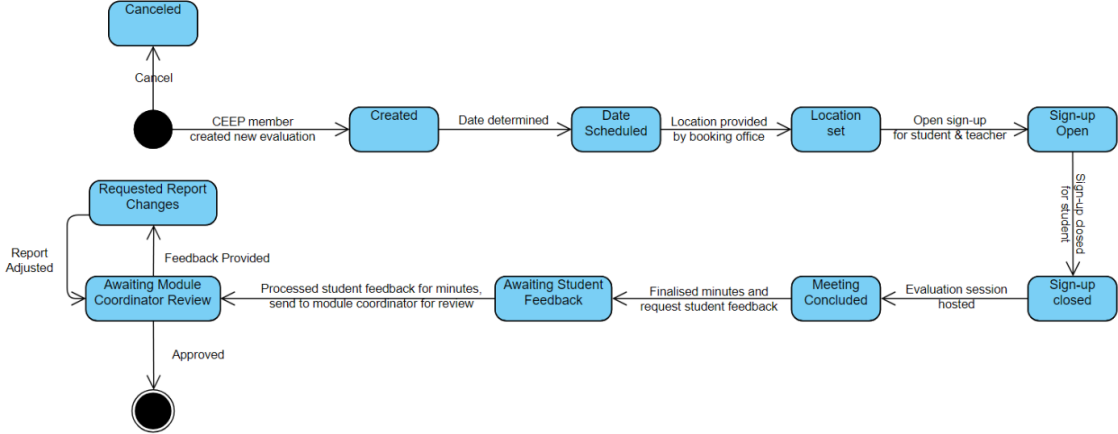more user-friendly, organised, and efficient experience for both the organising committee and the attendees.

Figure 4.1: State Machine Diagram outlining the Evaluation Process

4.2 Preliminary Design Choices

In this section, the preliminary design choices are discussed, namely the programming languages, frameworks and libraries used, as well as the main architecture of the system.

4.2.1 Programming Language

In this project, Python has been selected as the programming language of choice. Python is renowned for its accessibility and is commonly employed for back-end development in web applications. Notable web platforms like Dropbox, Pinterest, and Instagram are powered by Python. A significant factor in favor of Python is the existing expertise within the team, eliminating the need to acquire proficiency in a new language. While an alternative like Java was considered, it was disregarded due to its comparatively verbose nature when contrasted with Python's concise syntax. Python's brevity facilitates accelerated development, a crucial advantage given our project's time constraints. Moreover, Python's comprehensive documentation further reinforces its suitability for ensuring system maintenance post-project completion.[4]
Furthermore, we have chosen to develop the front-end using JavaScript. It was the logical selection for our front-end development, primarily due to its widespread utility and adaptability. It is universally supported by modern web browsers, ensuring a wide-reaching user base. Its seamless integration with HTML and CSS simplifies the development process and facilitates user interface interaction. JavaScript's rich selection of libraries and frameworks, including React, Vue, and Angular, eases the creation of dynamic and responsive interfaces. JavaScript's versatility, coupled with its extensive toolset and robust developer community, positions it as a dependable and practical solution for our requirements.

4.2.2 Frameworks

Django was chosen as a high-level web framework. Django is well-documented, has a large support community, and is incredibly scalable. This aspect was crucial because the system might be distributed over the entire TCS and BIT bachelor programs of the university. Django encourages speedy development and straightforward, pragmatic design due to the many features that are already available. Django is also known for its simplicity. To do so, Django uses

14

principles of rapid development which means developers can do multiple iterations at a time without starting the whole schedule from scratch [5]. This web framework was already familiar to several team members, which was preferable to a framework that no one on the team was familiar with. Django also has good security mechanisms in place, by default blocking common attack vectors like SQL injection, CSRF, and XSS [6].

The adoption of React as our front-end development framework is rooted in its versatile and practical attributes. React's cross-platform compatibility enables us to create a unified user experience, streamlining development efforts and resource allocation. Leveraging JavaScript, a well-known language among our development team, ensures an efficient development process, reducing the learning curve and expediting project advancement. React promotes code re-usability and efficient updates, ensuring consistency and maintainability throughout the application. Additionally, its extensive library of pre-built components and robust community support contributes to a faster development process, a crucial factor given our project's time constraints. For the authentication system, as also depicted in Figure A.6, we have chosen to use the OAUTH 2FA system that also the university uses. This means that we are using the Microsoft framework for authenticating the users of our system, using the university accounts. We have chosen this framework so as to not only give a sense of continuity with the rest of the university's system, but also for an added level of security.

Furthermore, we have chosen to host the mock information about the evaluations and the CEEP members on the university's server of PostreSQL. We have done so due to the fact that we want to simulate a database on the cloud, as well as having an extra layer of protection against common attacks such as the ones mentioned previously.

4.3 System Overview

In this segment, we have analyzed every page of our web application, highlighting the specific functions they serve. This compartmentalization is intended to enhance the user's interaction with the system. Moreover, we have recognized the necessity for three distinct views tailored to various potential users, namely CEEP members, students, and module coordinators. Consequently, the pages and their respective functions have been expounded upon, aligning with the specific user groups they are designed for. The relationship between each entity is described in Figure A.7.

4.3.1   CEEP Member View

In this subsection we shall be discussing the pages included in the access level of the CEEP members.

Admin Page

In this page, CEEP members have the possibility to view all registered users, their roles and are able to edit any user information, as well as remove a user (for example, a CEEP member graduates and he is no longer required to have access to the system.). As system administrators, they are able to view all possible modules that can be evaluated for TCS and BIT, and edit the relevant information about them. They can also remove a module that is no longer taught or add a new module. The page also has a tab for logs. CEEP members have access to a log table as well where they can view detailed descriptions of all create, edit, and update actions performed on the database.

Evaluations Page

CEEP members can access a comprehensive overview of all evaluations through a dedicated table view. Clicking on the "info" or "edit" button for a specific evaluation will display and enable actions related to that evaluation.

Within each evaluation, CEEP members can view detailed statistics on participating students, including their dietary restrictions and primary discussion points. They can also review registered students' dietary preferences in aggregate. Additionally, the platform allows CEEP members to upload the agenda for upcoming meetings and the evaluation report upon its completion. They can also preview and incorporate feedback from students and module coordinators if avaliable to refine the evaluation report.

CEEP members can also utilize provided templates to generate text for various purposes, including email communications. These templates enable CEEP members to send emails to specific recipients, groups (such as registered students, instructors, or the evaluation module coordinator), or the module coordinator of a particular evaluation.

Tasks Page

In this page, the CEEP members are able to see all the upcoming tasks open for all evaluations. There are 2 main sub-pages: *All tasks* and *My tasks*. As the names also suggest, the first one displays all available tasks in correlation with their respective evaluation, as well as the CEEP members assigned to the evaluation who must also complete the tasks. In the latter tab, the user can view the tasks assigned to them and mark them as completed. A new task is created automatically based on the state of the evaluation, in case it has not yet reached the finished state.

Templates Page

CEEP members have access to a collection of templates for various purposes, such as composing booking emails to Timetable Officers for room reservations during evaluations or crafting announcements on Canvas to publicize upcoming evaluation sessions. These templates can be tailored to specific needs, and new ones can be created or existing ones can be deleted. Templates are equipped with dynamic placeholders, such as "module", "scheduled date time", "allergy statistics", "user count", which automatically get populated with relevant information from the evaluation data when the template is used in the evaluation page.

4.3.2 Student View

In this subsection we shall be discussing the pages included in the access level of the students.

Evaluations Page

On this page, the students are able to see all evaluations. It has 3 sub-pages.

All evaluations
In this tab, students are able to see all the evaluations that are open for sign up. (These are filtered based on the study that the student is enrolled in. This ensures that a student may only sign up for evaluations of the courses that they have taken.) For each module evaluation there is a button that opens a pop up through which a student can sign up for the evaluation lunch session. The students may list the points that they would like to discuss during the meeting, as well as any allergies or intolerance they may have. Students are also able to view more details about the module, its description and the module coordinator assigned.

Own evaluations

This tab provides the logged in student with a list of all the evaluations that they have signed up to. It includes both open evaluations, and finished ones. For the open evaluations, until the day of the meeting, the student has the chance to sign out of an evaluation, thus removing themselves, their talking points and dietary restrictions from the database of enrolled students for that particular evaluation lunch. Students are able to see all available information for the evaluations, including the agenda of the points to be discussed in the following meeting, as well as the finalized report, after the evaluation session has finished.

Past evaluations

In this tab, students can see all the evaluations that have been concluded for their study program. These include the evaluations that they have signed up in the past. Students are able to see all the details regarding a module, including the agenda of the talking points discussed and the final report that has been approved by the module coordinator.

4.3.3   Module Coordinator View

In this section, we discuss the pages included in the access level of the module coordinator, as well as the functionalities included within these pages.

Evaluations Page

This page, while resembling the other versions of it, serves a different purpose. The module coordinator is automatically assigned and is expected to participate in the evaluation session for the module that they are coordinating. Furthermore, they have the responsibility of giving feedback to a published report after a meeting has concluded. After feedback has been given and there is nothing more to modify, the module coordinator must sign and approve the finalized report. After this step, an evaluation is marked as finished.

5 DETAILED DESIGN ANALYSIS

In this chapter, we have presented a more in depth look of the technicalities and the implementation of the elements that fulfill the functional requirements presented in Section 3.1.

5.1 System Overview

In the following class diagram from Figure 5.1, we have the models and all the relations between the focal elements in our system. All the tables follow a naming convention, thus all the tables that model elements for our system start with the prefix "api", to separate them from the other elements in the database, such as authentication elements or Django elements.
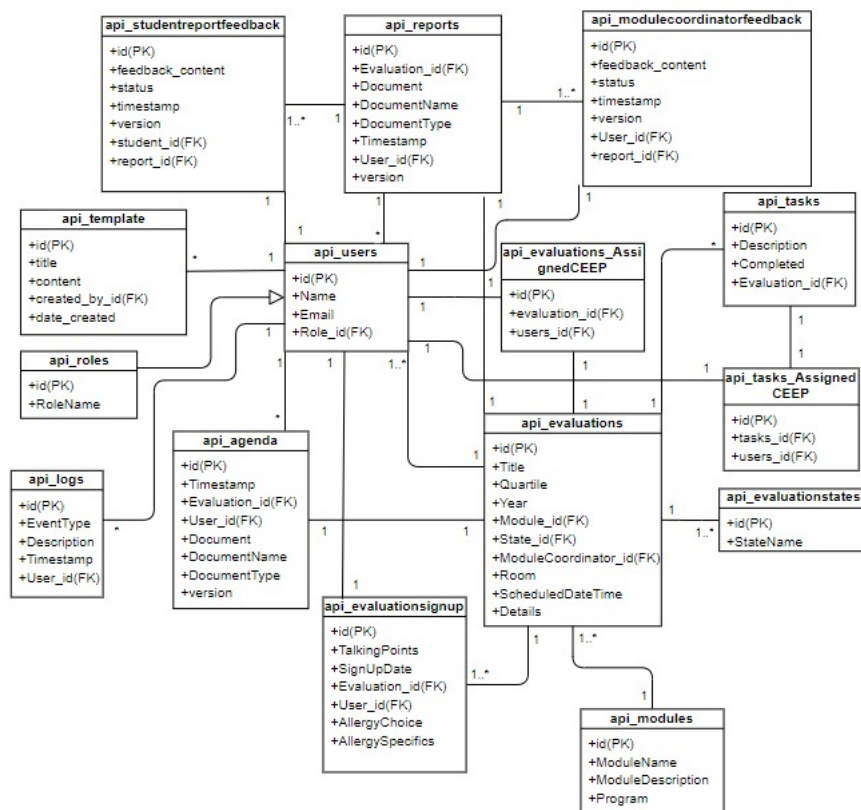

Figure 5.1: Class diagram of the models from the database

As such, we define the following tables and the relations between them:

5.1.1 Users

To streamline the user experience within our system, there's no requirement for users to set up a separate platform account. Instead, we've integrated the Microsoft OAuth 2.0 authentication

18

API, enabling users to sign in with their University of Twente student/employee credentials. Upon the initial login using their university accounts, the Microsoft OAuth API will return to our system's backend with user information, including usernames and emails, which will then be stored in the *api_user* table.

5.1.2 Roles

As previously elaborated in section 4.3, we have delineated three distinct access tiers within the application, which are categorized into four roles: **student**, **module coordinator**, **CEEP member**, and **teacher**. Notably, the teacher role is endowed with the same access privileges as students. These roles are subject to administrative oversight and are stored in the *api_roles* table. Upon user registration, each individual in the *api_user* table is initially designated as a 'student.' However, the system administrator holds the authority to adjust these roles to other designations through the system's admin page.

5.1.3 Modules

As our platform is specifically designed to assess modules offered within the Bachelor of Technical Computer Science and Business and IT majors, it is imperative to maintain a comprehensive record of all modules available to these programs. We achieve this by utilizing the *api_modules* table, which stores information about the modules, including their names and descriptions.

5.1.4 Evaluation

A key function of our platform is to present and manage a comprehensive record of all evaluations, both past and scheduled, for different modules. We gather and retain detailed information regarding these evaluations, encompassing the date, time, location, module coordinator, year, and the academic term (quartile) in which the module evaluation occurs in the **api_evaluations** table. This data is easily accessible to all users through the evaluation overview page. Additionally, we ensure that the status of each evaluation is kept current, enabling us to track the progress of each evaluation. These statuses are referenced from the **api_evaluationstates** table.

5.1.5 Evaluation States

Throughout the lifecycle of every evaluation, from its inception to its conclusion, it progresses through a sequence of distinctive states. These states, including "created," "date scheduled," "location set," "sign-up open," "sign-up closed," "meeting concluded," "awaiting student feedback," "awaiting module coordinator review," "requested report changes," "finished," and "canceled," were defined based on input from the client (CEEP member) and are stored in the **api_evaluationstates** table. These states are utilized by the **api_evaluations** table to track the current status of each evaluation.

In addition to the various states, each evaluation involves specific tasks that must be completed as it advances. For instance, during the "sign up close" state, the responsible CEEP member is tasked with creating the agenda for the evaluation and ordering lunch based on allergy statistics generated from the sign-ups. To facilitate the monitoring and management of these tasks, we have incorporated a column named "task description," which stores the tasks associated with each state. CEEP members responsible for each evaluation can access their required tasks within the evaluation portal's "tasks" tab. They can mark a task as complete once it's done, and this action will trigger the transition to the next state with its associated to-do tasks.

5.1.6 Evaluation Sign Up

When the status of an evaluation is set to "sign-up open," students have the chance to enrol in the evaluation of their choice. As part of this process, they must fill out a form that includes details about the topics they'd like to discuss concerning the module and any allergies they may have to specific ingredients. This sign-up information is then stored in the **api_evaluationsignup** table, where each sign-up is linked to the corresponding module through a foreign key. Users with CEEP member privileges can access and review all the sign-up details on the evaluation detail page.

5.1.7 Evaluations Assigned CEEP

In each evaluation, a minimum of two CEEP members are assigned responsibility for the entire process. This includes tasks such as scheduling a date with the module coordinator, arranging the location and catering, as well as hosting the evaluation and taking minutes. To facilitate the tracking of which members are responsible for a particular evaluation, CEEP members can select the individuals involved in these roles during the evaluation creation process. All the assignments related to each evaluation are stored in the **api_evaluations_AssignedCEEP** table.

5.1.8 Student Report Feedback

Following the conclusion of the evaluation, CEEP members will upload the minutes of the evaluation, which also serves as the final report. This report is initially reviewed by students and teachers to identify any unclear or necessary adjustments in the minutes. All the feedback given by students and teachers is stored in the **api_studentreportfeedback** table. This table includes not only the content of the feedback but also the timestamp when the feedback was provided. Additionally, it tracks the version number of the feedback, as there can be multiple versions of the report for the same evaluation, particularly if there are multiple rounds of feedback provided by students and teachers. All of this feedback is accessible to users with CEEP member privileges on the report details page. They can view and review the feedback provided by students and module coordinators.

5.1.9 Module Coordinator Report Feedback

Once there is no further feedback from students, a new version of the report is uploaded based on the feedback provided by students and teachers. Subsequently, the CEEP member sends a request to the module coordinator for their review and approval. If the module coordinator provides additional feedback, necessary changes are made, and a new version is uploaded for final approval. All the feedback is stored in the **api_modulecoordinatorfeedback** table. This table follows a similar format as the **api_studentreportfeedback** table, capturing the content of the feedback, the timestamp, and version numbers.

5.1.10 Reports

All these various versions of the report for the same module are stored in the **api_report** table. This table includes not only the document but also the date of upload, the CEEP member who uploaded it, the associated module, and the version number of the report.

5.1.11 Agenda

After closing the sign-up for each evaluation, CEEP members are tasked with reviewing the talking points submitted by students for discussion during the evaluation session. They select the most important ones and use them to create a meeting agenda. Once the agenda is finalized, it is uploaded to the designated module evaluation page. This agenda is accessible for preview and download by all users who have signed up for the evaluation. The agendas are stored in the **api_agenda** table, which includes information such as the time, the specific module to which the agenda pertains, the person who uploaded the agenda, and a version number.

5.1.12 Template

During various phases of an evaluation, different emails and announcements need to be sent to various recipients. For instance, when the evaluation state is set to "sign-up open," a CEEP member may request the module coordinator to send an announcement on the Module Canvas page, encouraging students to sign up for the evaluation. Another example is when the CEEP member needs to book a room to host the evaluation; they also need to send an email to the Rooster officer of the EEMCS faculty. What we've recognized is that these emails generally follow a similar structure across different evaluations, with variations primarily in course-specific information. To streamline this process, we've implemented a template functionality. CEEP members can create different email templates with predefined variables (e.g., module name, evaluation location, evaluation date, time, etc.). This allows them to generate an email or announcement with a single click. All these templates are stored in the **api_template** table, and CEEP members can add and modify templates in the "template" tab of the evaluation portal.

5.1.13 Logs

In every system, it is crucial to preserve logs for the purpose of monitoring the system and tracking user activities. These logs are of great value when it comes to diagnosing issues and maintaining the system's functionality. In our system, we store these logs in the **api_log** table, and users with CEEP member privileges can access and review these logs through the admin page. This access proves highly beneficial when diagnostics or tracking back specific activities are required.

5.2 Design Choices

In this section we shall be discussing the choices that we have made throughout the development process of the system.

5.2.1 Aesthetics

The main focus for this aspect is to maintain a clean and simple interface to facilitate the user in interacting with the system. We have also tried to use the colours and the overall feel of the official University websites, so as to give a sense of continuity, and also for a potential integration with the rest of the system.

5.2.2 Uploading the Agenda

We have decided that the CEEP Members that are in charge of a specific evaluation should be the ones creating an agenda document based on the talking points of the students that have signed up to the evaluation. Afterwards, they are able to upload the agenda. The document has to be a PDF, and it is visible to all users of the system, no matter what role they have.

5.2.3 Uploading and Updating the Report

After a meeting has concluded, a report of the evaluation must be uploaded as an official document confirming the proceedings that have taken place during the evaluation session. As such, this report must be in accordance to the opinions of the students, the ideas discussed in the session and the potential response of the teachers present at the session. As such, a preliminary version of the report must be written by the CEEP member and uploaded on the evaluation page so as to receive feedback. The teacher would now be able to see which evaluation is awaiting report approval. The flow of this functionality is described in the Appendix section, in Figure A.4.

5.2.4 Sign up form

As a main requirement of the system, students must be able to sign up to an evaluation session. They must be able to state the points that they would like to discuss regarding a specific module. Furthermore, due to the fact that the evaluation sessions often happen during the time allocated to the lunch break, the participating students are offered lunch. Due to the University's policy, the lunch options offered are all vegetarian, but students may still have a certain intolerance. As such, as part of the sign up form, students are required to select an option from a drop-down list of the most common allergies such as lactose or gluten intolerance, or select other, in which case they have to specify themselves. The flow of this functionality is described in the Appendix section, in Figure A.1.

5.2.5 Canvas Announcements

Any upcoming evaluation for a module is first announced and promoted through a Canvas announcement. This is done through the admin account of the CEEP members, and posted on the Canvas page of each respective module. Due to the fact that we do not have access to the credentials of this account, we have discussed within our team and with the client that it would be best to simply generate the text needed for the announcement with he data regarding the evaluation from the database. In this case, on the details page of each evaluation, the CEEP member is able to generate the text for the canvas announcement using the click of a single button, from a drop-down menu of the available templates. If they choose they can also generate a new template from the template page using the dynamic placeholders to their liking.

5.2.6 Catering and Booking Email

All upcoming meetings for an evaluation need a room and catering for the lunch. As such, we have created a template for both types of emails: requesting catering for the lunch offered, as well as a room in which the session could be held. For these two functionalities, we have created a new email address from which the timetable office as well as the catering company is contacted. Both of these had been previously done using a CEEP member's personal student email, so creating a custom email address seemed like a natural choice. Both of the templates are filled in with the information gathered from the sign up forms of the students. The booking email contains the amount of students signed up for the evaluation, and asks the Timetable Officers to book a room that is in proximity to the room where the students had their previous class. The catering email requests lunch for the amount of students that have registered for the evaluation, and specifies how many of the options must be custom: based on the allergies mentioned in the sign up form. The flow of this functionality is described in the Appendix section, in Figures A.2 and A.3.

6 TEST STRATEGY

6.1 Test Plan

6.1.1 Purpose

The test plan serves as a comprehensive road-map that outlines the strategy, objectives, and methodologies for assessing the quality and performance of our project. In this context, our project refers to the development of a centralized evaluation platform tailored for the exclusive use of the CEEP committee at the University of Twente. This platform is designed to streamline and optimize the evaluation processes within the university, providing a centralized and efficient system for the committee's operations. The primary purpose of this test plan is to ensure that our project not only meets but exceeds the expectations and requirements of the CEEP committee and the broader University community. By clearly defining our testing strategy, objectives, and procedures, we aim to systematically evaluate the platform's functionality, performance, security, usability, and compatibility to guarantee a robust and dependable solution for our stakeholders. This test plan will play a pivotal role in guiding our testing efforts, facilitating the identification of issues, and ultimately delivering a reliable, user-friendly, and secure centralized evaluation platform.

6.1.2 Scope

This test plan covers a comprehensive evaluation of the centralized evaluation platform, encompassing functional, performance, security, usability, and compatibility testing. Functional testing ensures all features operate according to specifications, while performance testing evaluates system responsiveness and scalability. Security testing safeguards sensitive data, usability testing focuses on user experience, and compatibility testing ensures harmonious operation across supported browsers and devices. These assessments collectively assure the platform's quality, reliability, and effectiveness.

6.1.3 Objectives

Testing of a centralized evaluation platform is a critical step in its development and deployment. First and foremost, it is essential to ensure that the platform functions correctly, guaranteeing that all its features and functionalities work as intended. Additionally, verifying the platform's performance under various loads is crucial to ensure it can handle user traffic without slowing down or crashing. Security is a top priority, and therefore, validating the platform's security measures is paramount to protect sensitive data and maintain the platform's integrity. To provide a seamless experience for end-users, assessing the platform's usability is essential, focusing on user-friendliness and accessibility. Lastly, confirming compatibility with supported browsers and devices is necessary to ensure that the platform is accessible to a wide range of users, enhancing its overall effectiveness and reach.

6.2 Test Objectives

6.2.1 Functional Testing

In this phase, we are committed to conducting thorough manual tests to evaluate the platform's features from the perspectives of its three main user categories: CEEP members and administrators, module coordinators and teachers, and students. To accurately mimic real-world usage, we have utilized our individual University of Twente accounts to assess the specific functionalities and access privileges associated with each user role. Furthermore, we have also hosted testing sessions with some CEEP members, as well as fellow students from TCS and BIT programs. It's important to note that to use the platform, individuals must possess a University of Twente domain account, which is a fundamental requirement mirroring actual usage conditions. This testing methodology ensures a comprehensive assessment of the platform, closely aligning with the expected user experience and meticulously assessing all features and access permissions to cater to the diverse needs and expectations of these user groups.

Security Testing

Our platform's security is well-established thanks to its utilization of the University's OAUTH 2FA system, a trusted authentication method. Furthermore, the mock data is securely stored in a database on the University's server, adding an extra layer of protection. User communication is intentionally limited, reducing the need for encryption. Additionally, to enhance database security, we've avoided direct SQL syntax, thus minimizing the risk of SQL injection attacks. These collective security measures emphasize our dedication to safeguarding the platform and the data it manages.

Unit & Integration testing Testing

In order to make sure that our system is acting as desired and there are little to no bugs present in the system, we have created automated unit and integration tests. Table 6.1 presents an overview of the tested functionalities as well as their priority level, ranging from H (High) to L (Low). We have made sure to create tests for all endpoint methods that communicate between the front-end and back-end. For each test mock data is created and then an endpoint method is called and tested to see if it perform the desired actions. For each endpoint we test the main functionality and some edge cases, such as trying to request data as Student from endpoint which is restricted for Admin/CEEP member, or trying to retrieve data for objects that do not exist, etc.

Functionality	Priority level
Edit user information	L
Delete an already existing user	L
Edit module information	M
Add a new module with details	H
Delete an already existing module	L
Display evaluations properly	H
Sign up to an evaluation	H
Edit an evaluation	H
Upload documents for evaluation	H
Send email	M
Generate text for emails/announcements	H
Update evaluation states through Task page	H

Table 6.1: Tested functionalities and their priority level

Usability Testing

Our main focus for the testing plan was testing the easy of use and client to system interaction of the project. A such, we have organized usability tests with BIT and TCS bachelor students (1, respectively 4 students) and members of the CEEP committee. Usability testing is a method by which users of a product will be asked to perform certain tasks to measure how easy it is to use the product, and the user's perception of the experience. There are various advantages of including user testing in the development process of an application. It discovers the real demands and tasks of the user early in the design process , provides well defined evidence for design recommendations and minimizes costs by anticipating and eliminating potential user roadblocks [7]. The students had to complete simple tasks e.g creating a new evaluation, signing up for it and de-enrolling from such an evaluation. The feedback received was positive, with our test subjects having no difficulties completing the tasks.

Test Data

We define 3 roles: CEEP member, Student and Module Coordinator:

- For the CEEP member, we have used some of our student accounts, designated in the database with the role of administrator. As such, we have tested if the CEEP members, as potential users, have the correct level of access to the data and functionalities. They must be able to create new evaluations for any module, view all the students that have signed up for a certain evaluation, their dietary requirements for the evaluation lunch, send promotional emails for the evaluation, upload the agenda and the final report, as well as request catering.

- Students must be able to sign up for evaluations, fill in a form through which they can send their main ideas that they would like to discuss regarding the module, as well as their allergies and any intolerance they may have. They should also be able to see the details of the modules of their study program only, and for the past and upcoming evaluations of those modules. Furthermore, they should also be able to de-register from an evaluation, in case they can no longer attend.

- Module coordinators are technically required to be present at the evaluations, so they are automatically added as a participant in the evaluation, however, in the case that they cannot attend, they should be able to de-register. Furthermore, the main responsibility of the module coordinator is giving feedback and approving the report created by the

CEEP member after finalizing the evaluation meeting. As such, they have access to the documents of an evaluation as well as the module details.

Detailed scenarios & steps for user testing

As such, we have created a list of steps that could describe a potential flow of work for our application for each one of the predefined roles. Every scenario has taken into consideration the responsibilities that each role must fulfill and is in accordance with the flow of the already existing solution, described in Section 1.2.4. Thus, we have defined the following 3 scenarios based on the role fulfilled by the user. The steps are written in second person, as instructions given to a potential user blind testing the application.

CEEP members

For the role of the CEEP member, we have defined the following tasks to be completed:

1. Log in to the system.

2. Create a new evaluation for a module of choice.

3. Generate the text for the Canvas announcement promoting the evaluation.

4. For the aforementioned evaluation, assign yourself as the CEEP member responsible for.

5. Check the tasks. For the evaluation created, follow the description of the task and mark it as completed.

6. Close the sign ups for one evaluation.

7. For the previous evaluation, upload the mock agenda.

8. Send the email requesting catering.

9. Assuming that the evaluation meeting took place, mark the evaluation as such.

10. For the same evaluation, upload the (initial) mock report.

11. Send an email to the module coordinator requesting approval or feedback of the report.

12. For the (mock) feedback received from the module coordinator, upload a new version of the report.

13. Log out of the system.

Students

For the role of the student, we have defined the following tasks to be completed:

1. Log in to the system.

2. Check the available evaluations for sign up. Sign up to an evaluation of choice.

3. For a past evaluation, view the agenda and the report.

4. For an evaluation that you have signed up for and is in the state *Awaiting student feedback*, send feedback for the report.

5. De-register yourself from an evaluation that you have signed up for.

6. Log out of the system.

Module Coordinators

For the role of the module coordinator, we have defined the following tasks to be completed:

1. Log in to the system.

2. For a past evaluation, view the agenda and the report.

3. For an evaluation that you are assigned to and is in the state *Awaiting module coordinator review*, send feedback for the report.

4. For an evaluation in the same state as mentioned previously, approve the report.

5. Log out of the system.

Compatibility Testing

Due to the fact that this is a web based platform, we have tested its compatibility with different browsers such as Chrome, Safari and Mozilla Firefox. We have not encountered any issues, and as such we deem the application to be compatible with the most popular search engines. However, we did not test the compatibility with any mobile devices, due to the fact that we mainly intend the platform to be used on a laptop or a PC. Mobile integration is something to be discussed for future development.

6.2.2 Testing the System & Results

Testing Approach

We use mainly manual testing for the functionalities. We focused on the main requirements, such as creating and signing up for evaluations, and vaping a clean, organized platform in which the details of each evaluation and the progress is visible. We are trying to take into account each edge case and test all possible actions that a user can take on any page of the platform.

Test Environment

Due to the fact that we use Django Python and React for the back-end and, respectively, the front-end, we are using PyCharm/Visual Studio code to run the back-end and analyze the log of the Django server, and a browser of choice to analyze the actual user platform, functionalities and potential user interaction.

Results

The user testing sessions have resulted in positive feedback. This serves as an objective confirmation of its functionality and effectiveness. It highlights the platform's alignment with the Committee for Education Evaluation Panels' (CEEP) unique needs, effectively addressing earlier operational issues. The good response validates the platform's ability to automate repetitive processes and the successful implementation of our design choices. Furthermore, it verifies the platform's dual purpose, which includes enhanced tools for CEEP members to monitor progress and organise evaluations, as well as simplified enrollment and access for students and teachers. This good feedback strongly suggests that the platform is ready to improve the CEEP initiative's overall efficiency and effectiveness.

Furthermore, we have made sure to eradicate the most severe bugs with the unit and integration tests, so as to have the best version of the system as possible. We have made sure that in case a functionality with a lower priority has partially failed the tests but does not interfere with the usage of the system, to work on improving it in the following sprint. In case the functionality has not passed the tests at all or if it interferes with the rest of the system, then we reconsidered if it is truly needed for the final product of our project. Thankfully, that was not the case for any of the functionalities that we have tested.

7 RESULTS AND DISCUSSION

7.1 Outcome of Project

The overall project outcome has effectively met the client's expectations by satisfying all the mandatory and conditional requirements outlined in section 3.1. In this section, we will elucidate how each requirement has been addressed and implemented within our system to ensure alignment.

Assignment of CEEP members to evaluations

One of the requirements identified in the initial phase was the ability of the CEEP member, particularly the chairman, to assign the CEEP member responsible for each created or upcoming evaluation. We have implemented this by incorporating a field in the evaluation creation form (shown in figure B.2), enabling CEEP members to assign multiple CEEP members to the evaluation being created.

Evaluation agenda & report Preview

To ensure that all users have access to comprehensive information for each evaluation, we've established an evaluation details page (shown in figure B.3). This page allows all system users to view and download the agenda (available after sign up closed) and the report (available after the evaluation session has taken place). This aligns with the requirement of the CEEP system to enable users to access and view evaluation reports, facilitating easy access to crucial evaluation data for informed decision-making.

Upload evaluation agenda & report

In line with the functionality that allows all users to preview and download evaluation agendas and reports, we were also required to provide CEEP members with the ability to upload these reports and agendas. This functionality has been implemented within the "edit evaluation" page, where a dedicated section allows CEEP members to upload reports and agendas for the corresponding evaluation (shown in figure B.4).

Evaluation sign up overview

In compliance with the requirement that CEEP members should have the ability to view all participants associated with a specific evaluation, we have implemented two fields within the CEEP member's evaluation details view (shown in figure B.5). These fields allow CEEP members to access information about all the students and teachers who have signed up for the evaluation, including details about the topics they would like to discuss and any dietary allergies they may have. Additionally, we have created a separate view to provide statistics regarding the dietary preferences of all the participants.

Task overview

To fulfil the requirement of allowing CEEP members to have effective time management and coordination within the evaluation process, we have introduced a "Task" tab. Within this tab, CEEP members can access a comprehensive list of tasks that need to be completed for various evaluations (shown in figure B.6. Additionally, all CEEP members can see who is assigned to each task. In addition to the "All Tasks" tab, CEEP members have a "My Tasks" tab where they can view tasks specifically assigned to them (shown in figure B.7. They can access task details and have the option to mark a task as complete once they have finished it. This functionality aids in task management and coordination for CEEP members.

Evaluation Report Review from Student

To fulfil the requirement that allows students to provide feedback on the report after the evaluation, we have introduced a new tab. This tab becomes accessible to students when the evaluation's state changes to "awaiting student feedback." Students can access it by clicking the "Provide Feedback" button on the "My Evaluation" page, as illustrated in Figure B.8. Within this tab, both students and teachers can submit their feedback. The feedback provided will be displayed in the comments section of the report in the CEEP member view as shown in Figure B.12, allowing for a transparent and collaborative feedback process.

Evaluation Report Review from Module Coordinator

The "should have" requirement, which allows stakeholders to request approval or provide feedback on the report, has been successfully implemented in the system. The process is similar to the one for students. Module coordinators gain access to the report feedback and approval page when CEEP members change the evaluation state to "awaiting module coordinator review." This can be achieved by clicking the "Approve / Request Changes" button on the "My Evaluation" page, as shown in Figure B.10. On the review page, module coordinators can either provide feedback or approve the report, as shown in Figure B.11, facilitating a seamless and collaborative review process.

Customizable Templates

The "could have" requirement for customizable templates has been successfully integrated into the system. A dedicated tab allows CEEP members to create various templates for emails and announcements, utilizing the variables provided in the grey box, as illustrated in Figure B.13. Once these templates are created, CEEP members can employ them on the evaluation detail page, which features a designated "Generate Template" section. In this section, CEEP members can generate evaluation-specific announcements and emails, with the added convenience of sending out these announcements via email with a single click on the same page, as shown in Figure B.14. This feature enhances the efficiency and flexibility of communication within the system.

7.2 Future Development

While our system has already incorporated numerous features, we recognize that there is still substantial potential for further enhancement to improve the experience for both students and CEEP members. In this section, we will explore various functionalities that could be implemented in future development to continue improving the system.

Integration with Canvas API

The current system faces limitations in identifying the courses or majors that students are studying, and it allows all students to sign up for evaluations without considering their course enrollment. Integration with the Canvas API could offer valuable improvements:

Customized Evaluation Overview: With Canvas integration, we could access additional student information, such as the modules they are enrolled in. This information could be used to tailor the evaluation overview page, showing students only the modules relevant to their studies. This customization reduces distractions and enhances the user experience.

Automated Announcement Sending: Currently, sending out announcements to promote evaluations is a manual process, relying on module coordinators to post announcements in Canvas. By integrating the Canvas API, we could automate this process, allowing CEEP members to send announcements with a single click from within our system. This streamlines communication and increases efficiency.

These potential functionalities can significantly enhance the user experience and administrative processes in our system, providing a more tailored and efficient environment for students and CEEP members.

Email Notification

In future development, implementing an email notification system can be highly beneficial for all primary stakeholders in our system, enhancing their experience and ensuring important tasks and events are not missed:

CEEP Members: Email notifications would serve as a valuable tool for CEEP members. They would receive immediate email notifications when they are assigned to a specific evaluation. This ensures they are aware of their responsibilities in a timely manner.

Module Coordinators: As module coordinators may not frequently use the system, email notifications can play a crucial role in keeping them informed. They would receive email notifications when their attendance is required for an evaluation or when they need to review a report. This ensures they are well-informed even without regular system access.

Students: Email notifications would be helpful for signed-up participants. They could receive reminder emails one day before or on the day of the evaluation, ensuring they do not miss the event. Additionally, students could be notified when they are eligible to provide feedback on the report. These notifications enhance the student experience by keeping them informed about key events and deadlines.

By introducing email notifications, our system can elevate the user experience to a new level by providing timely reminders and updates to all stakeholders, improving communication, and ensuring essential tasks and events are not overlooked.

Import Evaluation

Considering the CEEP committee has already conducted a lot of of evaluations in the past, a valuable feature for future development could be the ability to import past evaluations into the system. This functionality would enable the committee to easily transfer older evaluation data

into the system in a specific format. The ability to import past evaluations can be a practical and time-saving feature for the CEEP committee, supporting the effective management and preservation of older data within the system.

Export Sign Ups

In future development, a valuable feature could be enabling CEEP members to export a list of sign-ups for specific evaluations. This feature offers multiple advantages for the CEEP committee. It provides an efficient way to keep track of the attendance at evaluation sessions and serves as valuable documentation for record-keeping and auditing purposes.

Upvoting System

Our project coordinator suggested a valuable feature that can be added for future development - an upvoting system. The students will be able to upvote a discussion point or an report review comment. In this case CEEP would know which discussion point really concerned the students or what else should the secretary add in the report for the module coordinator.

Record Physical Sign Ups

In response to feedback from the CEEP committee, it is anticipated that in the future, paper sign-up forms may be introduced as a means for students to register for evaluations. As part of future development, a valuable feature could be the ability for CEEP members to input these paper sign-up forms into the system manually. This feature would enable CEEP members to fill in the details from paper sign-ups, providing a means to keep track of students who have registered for evaluations using physical forms.

Mobile Integration

Our system is currently designed for computer use, but for future development, mobile integration could be considered. This enhancement would allow CEEP members to conveniently access and utilize the system on the go using their mobile devices. This mobile integration aims to provide greater flexibility and accessibility for CEEP members, ensuring they can efficiently manage evaluations and tasks, even when not at a computer.

7.3 Timeline of Project Development

Throughout the entire project, we followed the Agile system and used Trello board, which involved weekly sprints and scrum meetings to assess progress from the previous week and outline tasks for the current week. Additionally, we conducted client meetings every Friday to showcase the project's progress, gather feedback, and address any questions or concerns. In this section, we will provide a weekly overview of the project's development, highlighting key milestones and achievements.

Week 1 - Week 2

During Week 1 and Week 2, our initial contact with the client led to the receipt of concise documentation explaining CEEP and their expectations. After a thorough review of this document, we arranged a meeting with the client to delve into a more detailed understanding of the entire evaluation workflow. With a clearer perspective in mind, we commenced the process of listing requirements in collaboration with the client. Priority for these requirements was determined

using the MoSCoW method. With all these foundational steps in place, we drafted our project proposal.

Week 3

In Week 3, we initiated the process by selecting the technologies we intended to employ for our system. We determined that we would use React for the frontend and Django for the backend, with PostgreSQL serving as the database. Once this technology stack was established, we proceeded to outline the functionalities to be implemented in the sprint. This included setting up the frontend using React and configuring the backend with Django. Additionally, we implemented OAuth 2.0 authentication and authorization system with Microsoft and established a basic admin page for viewing registered users.

Week 4

During this sprint, we conducted a thorough evaluation of our progress and confirmed that all the components we aimed to implement in the previous sprint were successfully completed. We also began the crucial task of planning and designing the database schema that would serve as the backbone for the entire system. This database schema is essential for the implementation of various functionalities that depend on structured data.

Additionally, we implemented the student evaluation view, a feature that enables students to access information about various evaluations. Notably, during this phase, we did not include the ability to add or edit evaluations, which meant that each evaluation had to be manually entered into the database. This served as an initial step in building the foundation for further functionalities in the system.

Week 5

During this sprint, we made significant advancements in the evaluation page of the system. We introduced two distinct pages for the student view, differentiating between upcoming evaluations, signed-up evaluations, and past evaluations. Among the functionalities developed was the ability for students to sign up for upcoming evaluations. They could specify their talking points and indicate any dietary allergies during the sign-up process.

In addition to progress in the student view, we implemented the CEEP members' view for the evaluation page. Users with CEEP member privileges gained the capability to access a comprehensive set of features. This included viewing all evaluations, editing existing evaluations, adding new evaluations, and even deleting evaluations when necessary. Furthermore, CEEP members could access detailed information for each evaluation, which encompassed student sign-ups and dietary allergy information. These developments significantly enhanced the functionality and usability of the system for both students and CEEP members.

Week 6

In this sprint, we implemented three critical functionalities to enhance the system.

First, we enabled CEEP members to upload reports and agendas for specific evaluations. Moreover, all system users gained the ability to access and download these uploaded agendas and reports. This feature enhanced the documentation and sharing of essential information within the system.

33

Second, we implemented a task page that allows CEEP members to track their tasks based on the various states of evaluations. This feature simplifies task management and coordination, ensuring that responsibilities are efficiently managed throughout the evaluation process.

The third significant addition was the template page, where CEEP members can create various templates for emails and announcements. These templates can include specific variables, making it easy to generate various communication types, such as catering requests, promotional announcements, and room bookings. This feature streamlines communication processes and ensures consistency in messaging within the system.

Week 7

During this sprint, we successfully implemented three of the final essential functionalities for the system.

Firstly, we implemented the capability for students to provide feedback on the evaluation report after the evaluation session has concluded. This feature empowers students to contribute their insights and opinions, enhancing the feedback process.

Additionally, we implemented the functionality for the module coordinator to provide feedback on the report and approve it after the evaluation session. This step is crucial in ensuring the accuracy and quality of the evaluation reports.

Furthermore, we added extra functions to the admin page. On this page, CEEP members can modify and delete user accounts, as well as add, edit, and update module information. This feature provides comprehensive control over the user and module management aspects of the system, streamlining administrative tasks.

Week 8

In this sprint, we introduced an additional feature that allows CEEP members to send emails through the system to specific user groups, such as all users who have signed up for an evaluation, or to individual recipients, like catering suppliers. This functionality is built on the template functionality, which generates email content using templates. CEEP members can utilize the "send email" functionality to distribute these generated templates to the designated user groups or individuals.

Furthermore, we conducted user testing sessions to assess the user interface and usability of the system. These sessions involved real users providing valuable feedback. Based on the testing results and user feedback, we made further refinements to enhance the system's performance, user-friendliness, and overall quality. This iterative process ensures that the system aligns with user expectations and requirements.

Week 9

During this sprint, we focused on incorporating integration tests and addressing any identified bugs within the system. Integration tests are essential for ensuring that various components of the system work harmoniously together. Bug fixes involve resolving issues, errors, or inconsistencies in the system to enhance its overall functionality and reliability. This sprint aimed to improve the robustness and stability of the system.

Week 10

In this sprint, we are in the final stages of completing our product and report, and we are also preparing for the final presentation. This phase involves wrapping up any remaining tasks, and ensuring that the product meets all the requirements and is thoroughly tested. Additionally, we are working on the final documentation and presentation to present our work and achievements to stakeholders.

7.4   Reflection on the process

As we near the completion of our project, it's essential to reflect on the overall process and the journey we've taken to develop this system. In this section we are going to discuss some key reflections:

Team Collaboration

The entire project was primarily conducted remotely due to various personal circumstances affecting multiple team members. While initial communication challenges were encountered, the team adapted and ultimately managed to establish effective communication, allowing for efficient collaboration throughout the project.

Agile System

Throughout the entire project, we adhered to the Agile framework, conducting weekly meetings and setting specific goals for each sprint. This approach was instrumental in helping us manage our progress effectively. It allowed us to establish deadlines and act as a self-checking mechanism to ensure that we remained on track and met our project milestones.

Challenges with Technologies

In the initial phase of our project, we did not thoroughly consider the technologies we would use for our system. Consequently, we ended up choosing technologies that most of the team members were not familiar with. This resulted in a steep learning curve at the beginning, which proved to be challenging and even demotivating for some team members. However, Kristiyan, with his motivation and experience, stepped in to assist and support the team in adapting to the new technologies we were using. His guidance and expertise played a crucial role in helping the team overcome these technological challenges.

User-Centric Approach

Throughout the entire project, we managed to keep the client actively involved in every stage of our work. We scheduled weekly meetings with the client, during which we presented our progress and addressed any issues or questions related to the system and workflow. This client-centric approach proved invaluable in ensuring that the system we delivered was closely aligned with their requirements and expectations.

Communication Challenges

Remote working posed several challenges for our team throughout the project. One recur-

ring issue was the lack of real-time communication (aside from scum meetings), which led to instances where not all team members were up to date with the project's progress. In light of these challenges, we believe that incorporating physical, face-to-face sessions in the future could be more effective for maintaining effective communication and ensuring that everyone is on the same page.

Another communication issue arose from the online nature of our collaboration. Some team members faced delays in receiving and responding to messages, which occasionally hindered the flow of information and teamwork.

7.5 Personal Contribution

In this section, we will provide an overview of each team member's individual contributions throughout the entire project. We'll highlight the unique responsibilities and roles that each team member undertook to accomplish our project's objectives.

An:

Full Stack Developer: Responsible for the implementation of the Admin Page
Scrum Master: Responsible for setting up scrum meetings, checking development progress, providing suggestions and overview of the system design and logic.
Report Editor: Responsible in writing section 5.1 and 7 of the report
Diagram Maker: Responsible for drawing sequence diagram, use case diagram and state diagram
Presenter: Responsible for presenting during the peer review sessions

Andrea:

Client and Project Supervisor Liaison: Responsible for being a point of contact with the client and project coordinator. Informing them about the progress of the project, and asking questions that came from the team
Report Editor: Responsible for writing the Abstract, Introduction and parts of the Proposal sections of the report
Proposal Editor: Responsible for writing the Planning, Coordination with the client, and Test plan section and parts of the risk analysis
Stakeholder Onion Creator: Responsible for creating the Stakeholder Onion diagram
Junior Full stack developer : Responsible for the initial Templates page and Sending booking e-mail functionality.

Arsalaan:

Junior Full stack developer: Responsible for the initial Templates page and Sending booking e-mail functionality.
Report Writing : Requirements specification, Poster design, Junior editor.
Proposal Writing : Risk analysis and requirements specification

Diana:

> **Full stack developer:** Responsible for the implementation of the Task page for the CEEP members
> **Head of Documentation Maintaining and Writing:** Created all documents required for documentation , Wrote Sections 4, 5 and 6, Editor in Chief,
> **Usability Testing Coordinator:** Created the test scenarios and conducted the user tests with the clients
> **Designer:** Design of the Poster and Presentation Slides

Kristiyan:

> **Tech Lead & Full Stack Developer:** Responsible for the setup and development of the Django back-end server and the React front-end, establishing the core infrastructure for the project's development and operation.
> **UI/UX Designer:** Crafted the Design for the user interface and experience.
> **Evaluations Page Architect:** Implemented the Evaluations page, tailored to support various functionalities for Students/Teachers, Admin/Ceep, and Coordinators.
> **Database Design Lead:** Leading role in designing the database architecture, leveraging Django ORM for secure and efficient data handling.
> **Integration Testing Coordinator:** Led Integration testing in collaboration with team member Park, certifying the reliability of API endpoints.
> **Security and Authentication Lead:** Utilized Django ORM's built-in protections as part of the data handling strategy, deployed two-factor authentication with Microsoft OAuth 2.0, and implemented role-based access controls at API endpoints.
> **System Enhancement Developer:** Developed a detailed logging mechanism and enhanced the email sending functionality, including improvements to the template system.

Sanghyeop:

> **Testing developer:** Make system testing to increase reliability of our system
> **Report Editor:** Contributing to write down for section 1, 2, 3, and abstract for the report
> **Diagram designer:** Making use case diagram and class diagrams for the database and web system

We strongly recommend granting the green card to our team member, Kristiyan. He played a pivotal role by assisting the entire team in navigating new technologies and consistently putting forth significant effort in implementing the system.

7.6   Conclusion

Throughout this journey, we faced lots of challenges, but we are happy with our end result. We managed to fulfil the requirements and produced a system that met clients' expectations, while maintaining our independence from the supervisor.
We have listed several future development ideas, this includes mobile integration, email notifications, and the ability to import past evaluations. These future developments have the potential to streamline further and elevate the CEEP system. The CEEP committee could consider having these requirements done by the next Design Project Group.

8 MANUAL

8.1 For CEEP (Admin User)

The CEEP Web Platform is your command center for managing evaluations, tasks, templates, and administrative settings in the web application. This guide focuses on the nuanced functionalities within the Evaluation section.

Navigating the Dashboard

After logging in, you will find a navigation bar with the following sections:

- Evaluation

- Tasks

- Templates

- Admin

8.1.1 Evaluation Page

Admin View – All Evaluations for CEEP Members

The Evaluation page begins with a comprehensive table listing all evaluations. This includes functionalities for:

- Search and Filter

- Column Customization

- Data Export

- Adding New Evaluations

Initial Evaluation View

When you first access the Evaluation page, you can view and manage evaluations in a summarized format. Each row in the table represents an evaluation and includes:

- Basic Information: ID, Module, Quartile, Year, Scheduled Date Time, Room.

- State and Management Tools: User Sign-up Icon, State, Edit Button, Module Coordinator, Assigned CEEP Members.

Detailed Evaluation View (After Clicking 'Edit')

When you click the 'Edit' button on an evaluation, a new tab opens within the Evaluation page, providing a detailed view of that specific evaluation.

- Evaluation Information: View all details about the evaluation.

- Edit and Delete Options: Top right buttons allow you to edit the evaluation's information or delete it.

- Editing: Brings up a form to change details, including the evaluation state.

- Deleting: A confirmation pop-up ensures intentional deletion.

- Files Management: Upload, view, replace, or delete files like agendas or reports. Track document versions and manage feedback.

- Template Integration: Select and use templates for communication related to the evaluation.

- Signed-up Users Table: Detailed view of students/teachers registered for the evaluation.

- Allergy Statistics: Comprehensive data on dietary restrictions of participants.

Implications of Changing Evaluation States

- Created: The initial state for new evaluations.

- Sign up Open: Makes the evaluation visible to Students and Teachers for registration.

- Awaiting Students Feedback: Allows registered students to provide feedback.

- Awaiting Module Coordinator Review: Enables the coordinator to review and provide feedback or approve the evaluation.

8.1.2   Tasks Page

Task Management

This section is essential for tracking and managing tasks related to the evaluations.

- Overview of Tasks: Provides a snapshot of all current tasks associated with different evaluations.

- All Tasks Tab: Lists all tasks across evaluations. Key details include:
    - Evaluation Name: Identifies the associated evaluation.
    - State of Evaluation: Shows the current state of each evaluation.
    - TODO: Describes the task to be completed.
    - Assigned CEEP Members: Displays who is responsible for each task.

- My Tasks Tab: Focuses on tasks specifically assigned to you.

- Task Details: Each task includes information about the related evaluation, its current state, and the specific action required.

- Completion Feature: Mark tasks as complete, which updates the status of the associated evaluation and triggers the next steps in the evaluation process.

Navigational Features

- Filter and Sort Options: Tailor the view to focus on specific evaluations, states, or assigned members.

- Task Progress Tracking: Easily monitor the progress of tasks and their impact on evaluations.

8.1.3 Templates Page

Template Creation and Management

The Templates page is a powerful tool for standardizing and streamlining communications related to evaluations.

- Template Library: Access a collection of pre-designed templates for various communication needs.

- Creating New Templates: Use a user-friendly form to create new templates. This includes setting up dynamic placeholders like "module name", "date", or "participant count", which automatically populate with relevant data from evaluations.

- Applying Templates: Seamlessly integrate templates into evaluation communications, ensuring consistency and saving time. This could be done in the evaluation page when you have selected an evaluation.

8.1.4 Admin Page

Comprehensive Administration

The Admin page is the hub for managing the overall functionality and users of the web application.

- Logs Tab:

  - Activity Monitoring: View a detailed log of all activities that affect the database.
  - Event Tracking: Includes information like event type, user involved, and a description of the action.
  - Time-stamped Entries: Each log entry is time-stamped for accurate record-keeping.

- Users Tab:

  - User Management: View and edit user profiles, including their roles and personal details.
  - Role Assignments: Assign or change user roles, which determines their access levels and functionalities within the app.

- Modules Tab:

  - Module Overview: Manage all module-related information, crucial for setting up evaluations.
  - Editing Capabilities: Add new modules, edit existing module details, or remove modules that are no longer relevant.
  - Comprehensive Details: Includes module name, description, and associated actions.

8.2 For Student/Teachers

As a student or teacher, you have access to the Evaluations page, which is tailored to provide you with a streamlined experience for participating in module evaluations. This guide will help you navigate and utilize the features available to you.

8.2.1 Accessing the Evaluations Page

After logging in with your university Microsoft account, you will find the Evaluations tab on the navigation bar. This is your gateway to all evaluation-related activities.

8.2.2 Navigating the Evaluations Page

The Evaluations page is divided into three primary tabs:

- Open Evaluations: View and sign up for ongoing evaluations.

- My Evaluations: Manage evaluations you've signed up for.

- Past Evaluations: Access evaluations that have concluded.

Open Evaluations Tab

Overview

This tab lists all evaluations currently open for sign-up, tailored to your course enrollments.

- Table Layout: The table displays key information including Module, Quartile, Year, Scheduled Date Time, Room, Module Coordinator, and Sign-up options.

- Info Icon: Clicking this reveals more details about each evaluation, such as the agenda, reports, and specific details about the module.

Signing Up for Evaluations

- Sign-up Process: Use the 'Sign Up' button to register for an evaluation. A pop-up form will appear where you can input your discussion points and dietary requirements if any.

My Evaluations Tab

Managing Your Evaluations

- Current and Past Sign-ups: This tab shows evaluations you have signed up for, both ongoing and completed.

- Feedback Option: When an evaluation is in the "Awaiting Student Feedback" state, you can provide your feedback through a pop-up form.

- Withdraw from Evaluation: You have the option to withdraw from an evaluation before it occurs. This will remove your details and contributions from the evaluation database.

Past Evaluations Tab

Accessing Completed Evaluations

- Evaluation History: View all completed evaluations, including those you participated in and others in your study program.

- Detailed Information: Access full details of each evaluation, including discussion agendas, final reports, and module coordinator notes.

Additional Features

- Search and Filter: Utilize search and filter functionalities to easily find specific evaluations or information.

8.3 For Module Coordinator

As a Module Coordinator, you have a specialized dashboard that allows you to manage and participate in evaluations for the modules you oversee. This guide will walk you through the functionalities and features available to you for effective coordination.

8.3.1 Accessing the Evaluations Page

After logging in with your university Microsoft account, you will find the 'Evaluations' tab on the navigation bar. This tab is your primary interface for managing module evaluations.

8.3.2 Navigating the Evaluations Page

The Evaluations page for Module Coordinators is divided into two main tabs:

- Coordinator View: Manage evaluations that you are coordinating.

- Past Evaluations: Access and review completed evaluations.

8.3.3 Coordinator View Tab

Overview

In this tab, you will find a list of evaluations you are assigned to, offering a comprehensive view of each module's evaluation process.

- Table Layout: The table includes Module, Quartile, Year, Scheduled Date Time, Room, Assigned CEEP Members, Info, and Actions.

- Info Icon: Clicking this icon reveals additional details about each evaluation, including the agenda and report.

Managing Evaluations

- Evaluation Actions: Depending on the state of the evaluation, you can perform various actions such as approving reports, providing feedback, or viewing existing feedback.

- Approval and Feedback: When an evaluation is in the "Awaiting Module Coordinator Review" state, you can either approve the report or provide feedback. Approving the report marks the evaluation as finished.

8.3.4 Past Evaluations Tab

Accessing Completed Evaluations

- Reviewing Past Evaluations: This tab allows you to view all evaluations that have concluded, including those you coordinated.

- Detailed Records: Access the full details of each evaluation, including discussion points, reports, and feedback from participants.

8.3.5 Additional Responsibilities

As a Module Coordinator, you play a crucial role in the evaluation process. Your feedback and approval are essential in finalizing the evaluations and ensuring that they accurately reflect the discussions and outcomes.

REFERENCES

[1] G. Waja, J. Shah, and P. Nanavati. Agile software development. *International Journal of Engineering Applied Science and Technology, 5(12).*, 2021.

[2] M. Mahalakshmi and M. Sundararajan. Traditional sdlc vs scrum methodology - a comparative study. *International Journal of Emerging Technology and Advanced Engineering, 3(6).*, 2013.

[3] S Hatton. Choosing the right prioritisation method. *Australian Software Engineering Conference*, 2008.

[4] Vineesh Cutting and Nehemiah Stephen. A review on using python as a preferred programming language for beginners. *International Research Journal of Engineering and Technology (IRJET), 8(8)*, 2021.

[5] D. A. Chopade, S. P. Deshmukh, M. S. Khune, and C. M. Manwar. A survey: Django framework. *Special Issue on Research and Innovations in Science and Engineering, 5(12s).*, 2020.

[6] `https://docs.djangoproject.com/en/4.2/topics/security/`. [Online; accessed 02-November-2023].

[7] Niranjanamurthy M, Archikam Nagaraj, Himaja Gattu, and Puneeth K Shetty. Research study on importance of usabilitytesting/ user experience (ux) testing. *International Journal of Computer Science and Mobile Computing*, 2014.

A APPENDIX

A.1 Sequence diagrams for functionalities

A.1.1 Evaluation session sign up
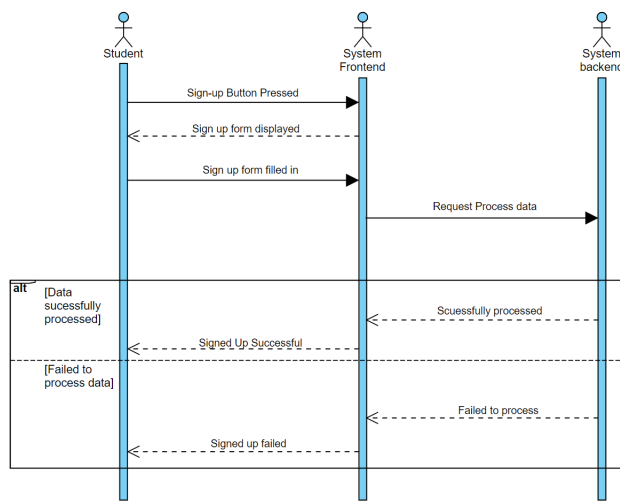
Figure A.1: Sequence diagram for signing up for an evaluation

A.1.2 Sending catering email
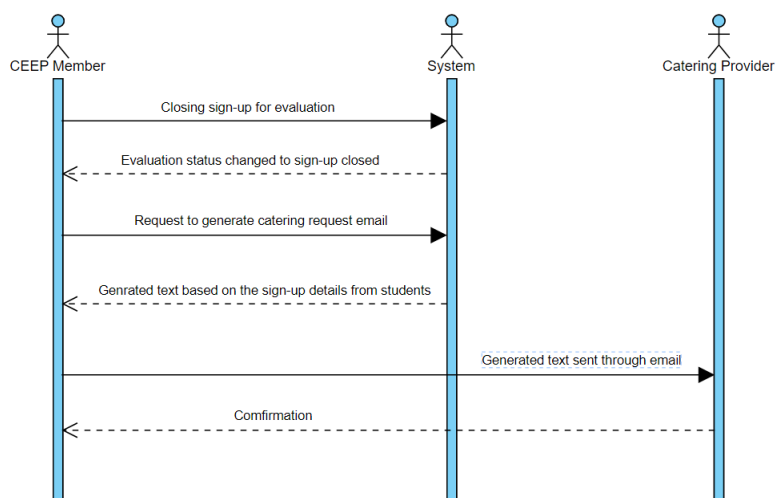
Figure A.2: Sequence diagram for sending the email for requesting catering
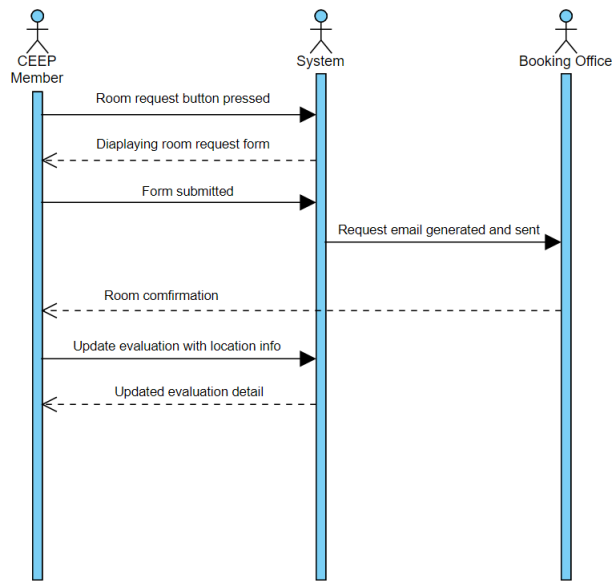
A.1.3 Sending booking email

Figure A.3: Sequence diagram for sending the email requesting booking a room
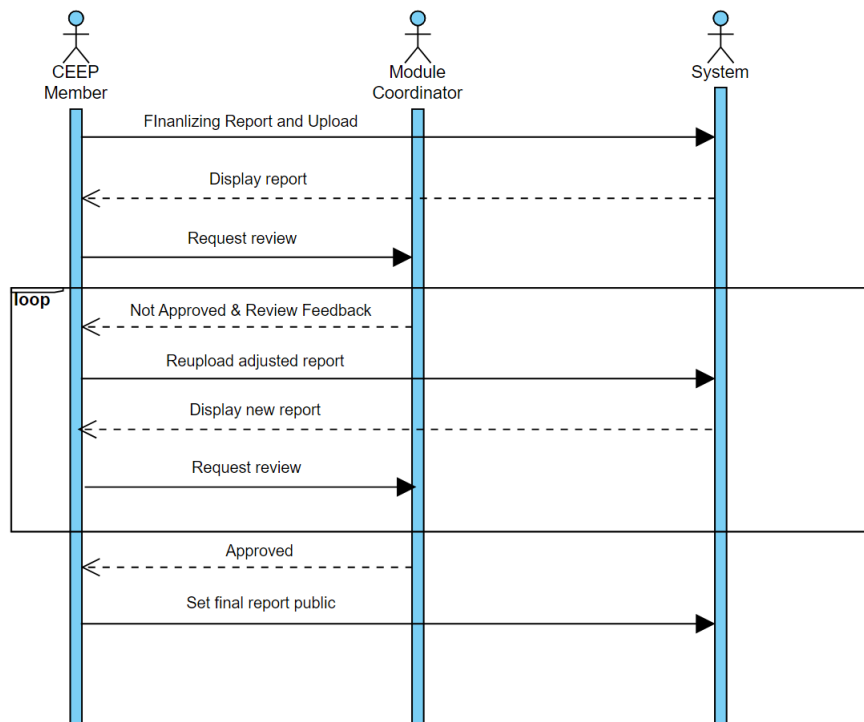
A.1.4 Report review

Figure A.4: Sequence diagram for the reviewing process of the report

A.2 Use Case Diagram

Figure A.5: Use Case Diagram

A.3 Authentication Functionality Framework
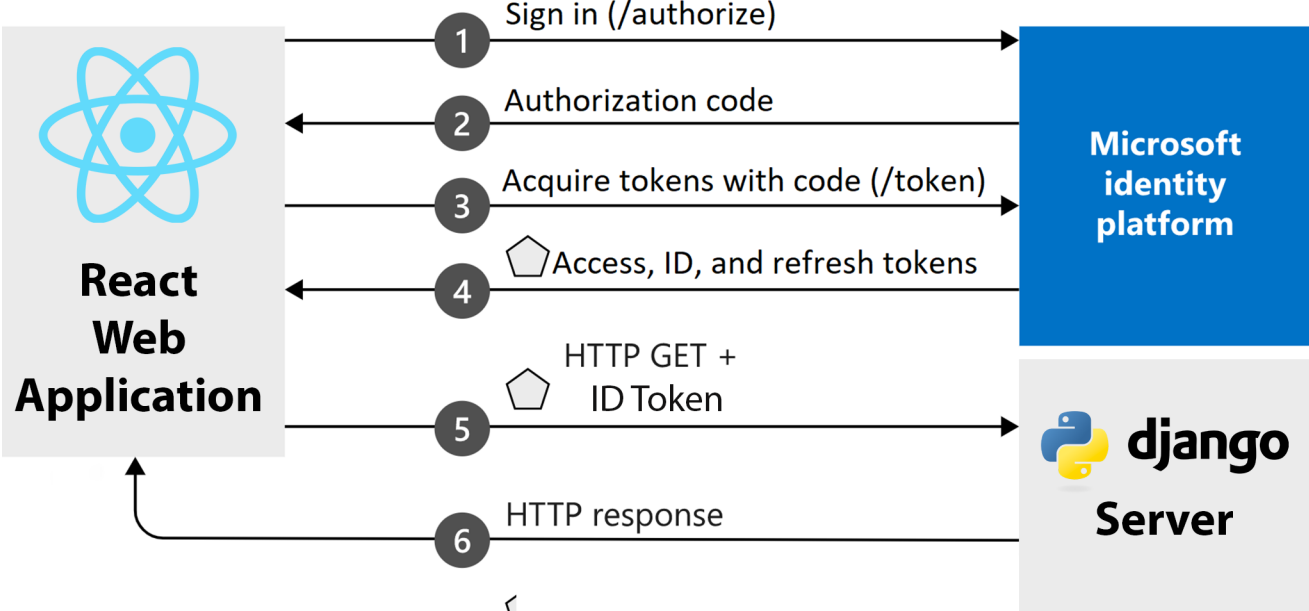
Figure A.6: Authentication system

A.4 Entity Diagram
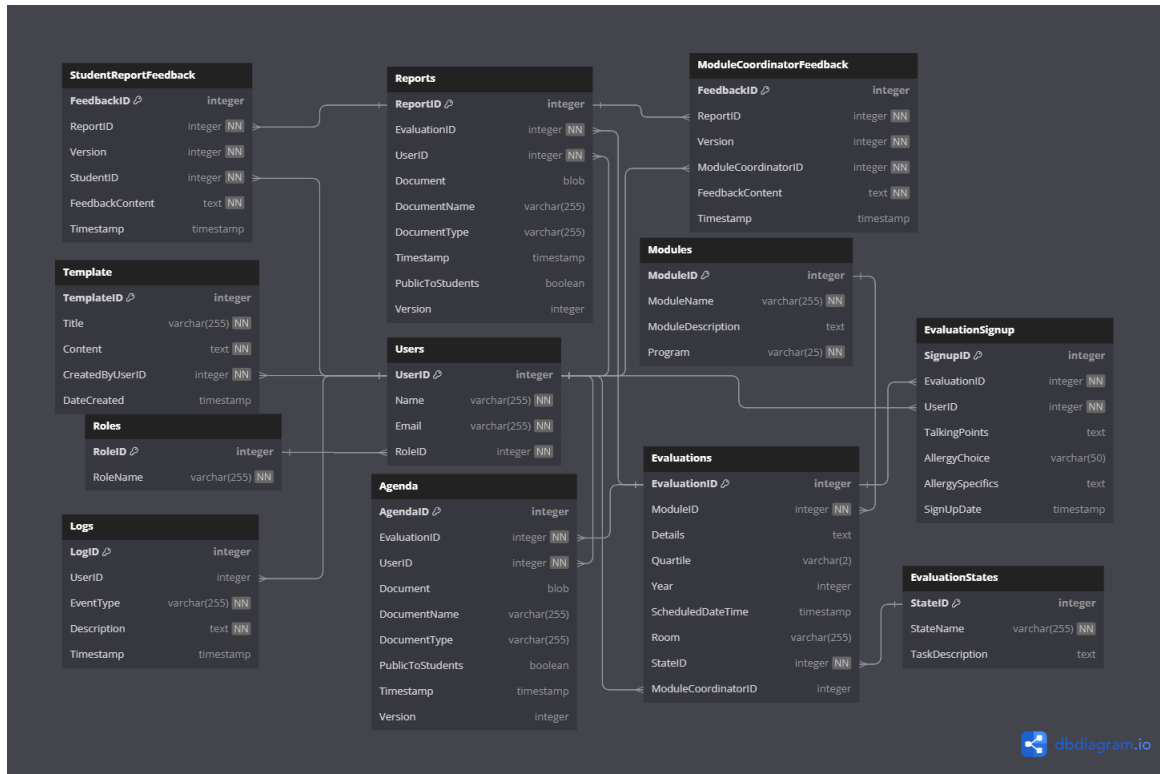
Figure A.7: Entity Relationship Diagram

B SYSTEM OVERVIEW
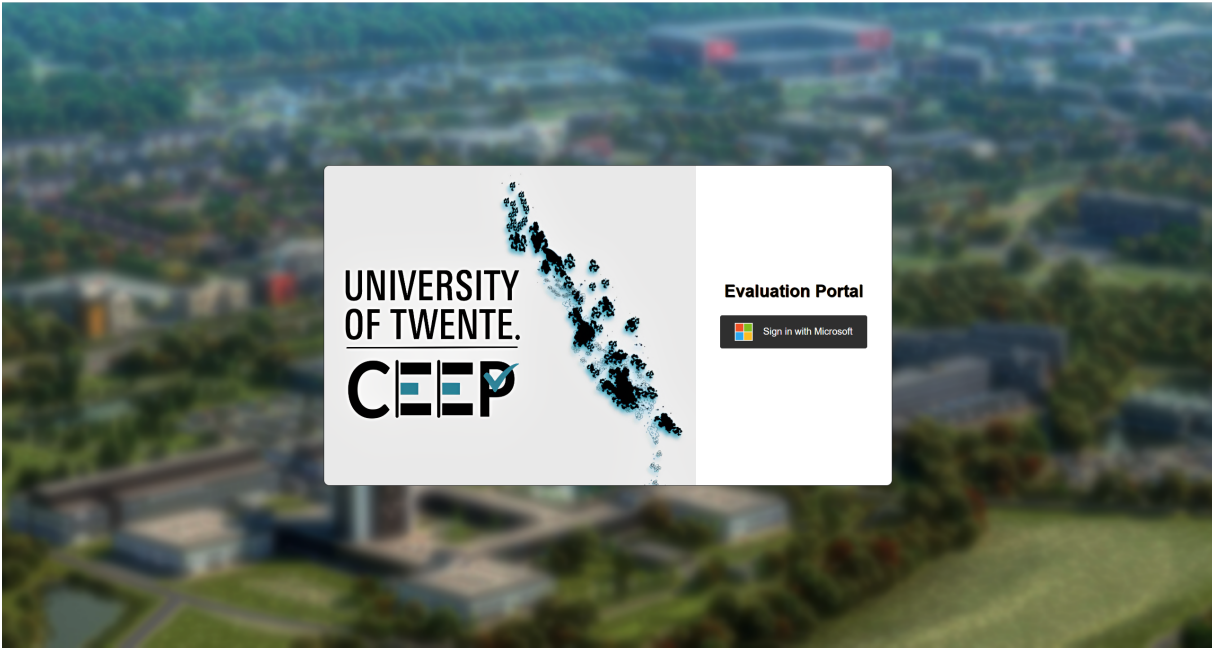
Figure B.1: CEEP Login Portal

Figure B.2: Evaluation Creation Form

Pearls of Computer Science Module, 2024 (Q1B)

Scheduled Date Time:
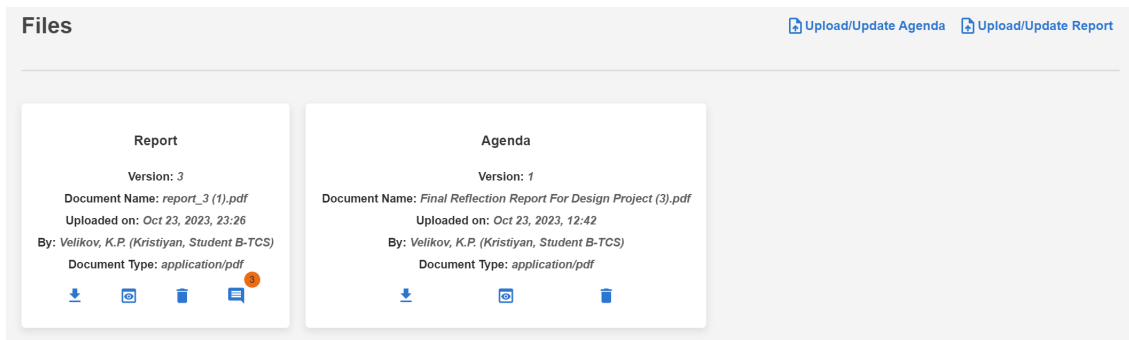Nov 1, 2023, 18:37

Room:
NH 211

Module Coordinator:
Velikov, K.P. (Kristiyan, Student B-TCS)

Agenda: 👁 Preview ⬇ Download

Report: 👁 Preview ⬇ Download

Details:
This is a high level description of the evaluation

Figure B.3: Evaluation Details

Files ⬆ Upload/Update Agenda ⬆ Upload/Update Report

Report	Agenda
Version: 3	Version: 1
Document Name: report_3 (1).pdf	Document Name: Final Reflection Report For Design Project (3).pdf
Uploaded on: Oct 23, 2023, 23:26	Uploaded on: Oct 23, 2023, 12:42
By: Velikov, K.P. (Kristiyan, Student B-TCS)	By: Velikov, K.P. (Kristiyan, Student B-TCS)
Document Type: application/pdf	Document Type: application/pdf

Figure B.4: Upload Agenda and Report View

Signed-up Students / Teachers

ID	Name	Role	Name	Talking Points	Allergies	Signup Date ↓
				bla		
7	a.onofrei@student.utwente.nl	Admin	Onofrei, A. (Andrea, Stude...		None	Oct 2, 2023, 22:06

1 row selected   Rows per page: 100 ▾   1–1 of 1 < >

Allergy Statistics

None: 1	Lactose Intolerance: 0	Gluten Intolerance: 0
Nut Allergies: 0	Seafood Allergies: 0	Egg Allergies: 0
	Other: 0	
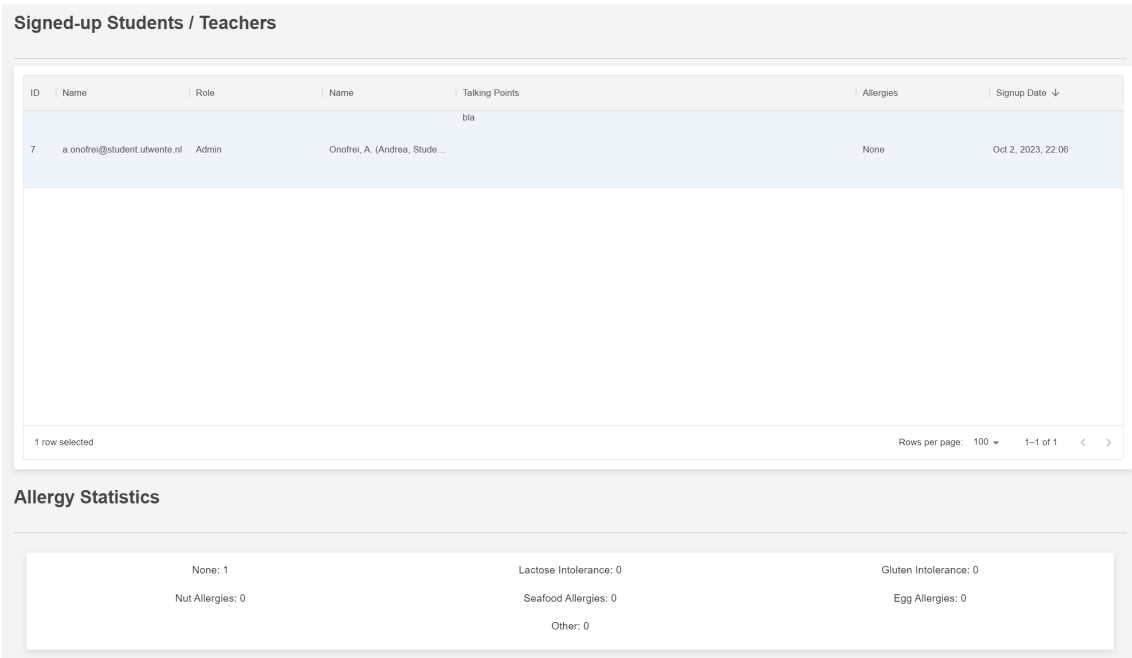
Figure B.5: CEEP Member Sign Up Details View

ALL TASKS   MY TASKS

Evaluation	State	TODO ↑	Assigned CEEP Members
Data Science & Artificial Intelligence Module, 2023 (Q1A)	Meeting Concluded	Start creating the report and upload the minutes for the evaluation. You can then change the state to "Awaiting Students Feedback" when done.	Velikov, K.P. (Kristiyan, Student B-TCS), Pan, A. (An, Student B-TCS,M-CS)
Pearls of Computer Science Module, 2024 (Q1B)	Finished	The evaluation has been concluded. Nothing has to be done. Students can view the evaluation in the past evaluations tab.	Velikov, K.P. (Kristiyan, Student B-TCS), Pan, A. (An, Student B-TCS,M-CS)
Computer Systems Module, 2023 (Q1A)	Awaiting Students Feedback	Wait for students to add feedback to the report. Revise and upload the new report if needed based on the feedback.	Onofrei, A. (Andrea, Student B-TCS), Velikov, K.P. (Kristiyan, Student B-TCS)
Computer Systems Module, 2023 (Q2B)	Requested Report Changes	Upload the revised report based on the coordinator's feedback. Change the state to "Awaiting Module Coordinator Review" when done.	Velikov, K.P. (Kristiyan, Student B-TCS), Bozea, D.A. (Diana-Alexandra, Student B-TCS)

1–4 of 4 < >

Figure B.6: Tasks Overview Page

Figure B.7: My Tasks Overview Page

Figure B.8: Report Feedback Request to Students
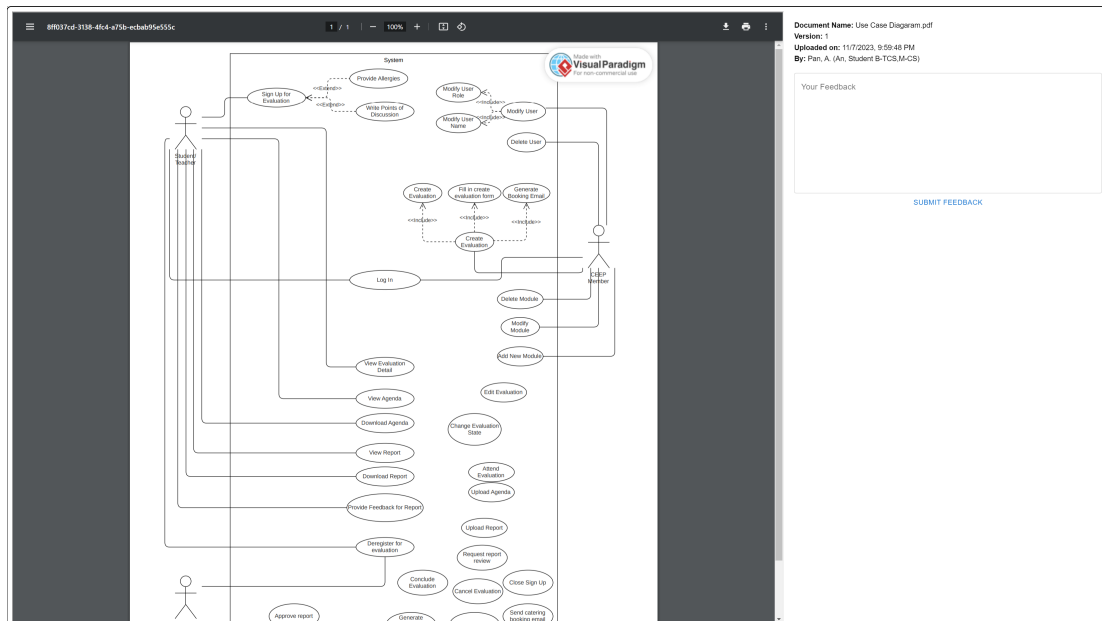
Figure B.9: Student Report Feedback View

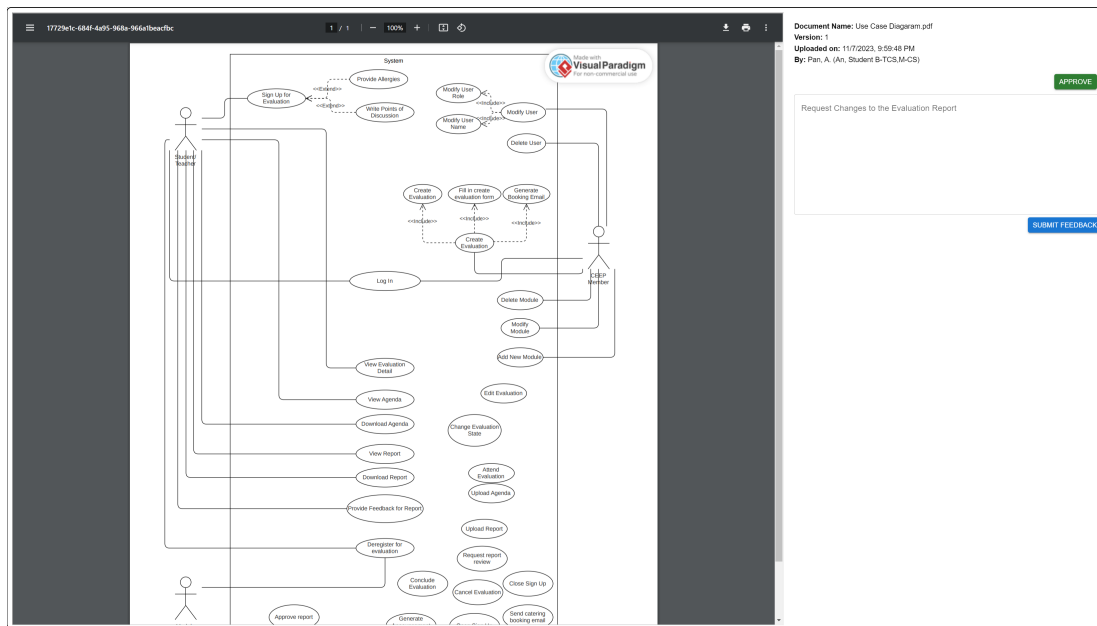Figure B.10: Report Feedback Request to Students

Figure B.11: Student Report Feedback View

Module Coordinator Feedback

Report Version: 1	11/7/2023, 8:15:56 PM	Bozea, D.A. (Diana-Alexandra, Student B-TCS)
This report needs reformatting.		

Student Feedback

Figure B.12: Report Feedback Overview

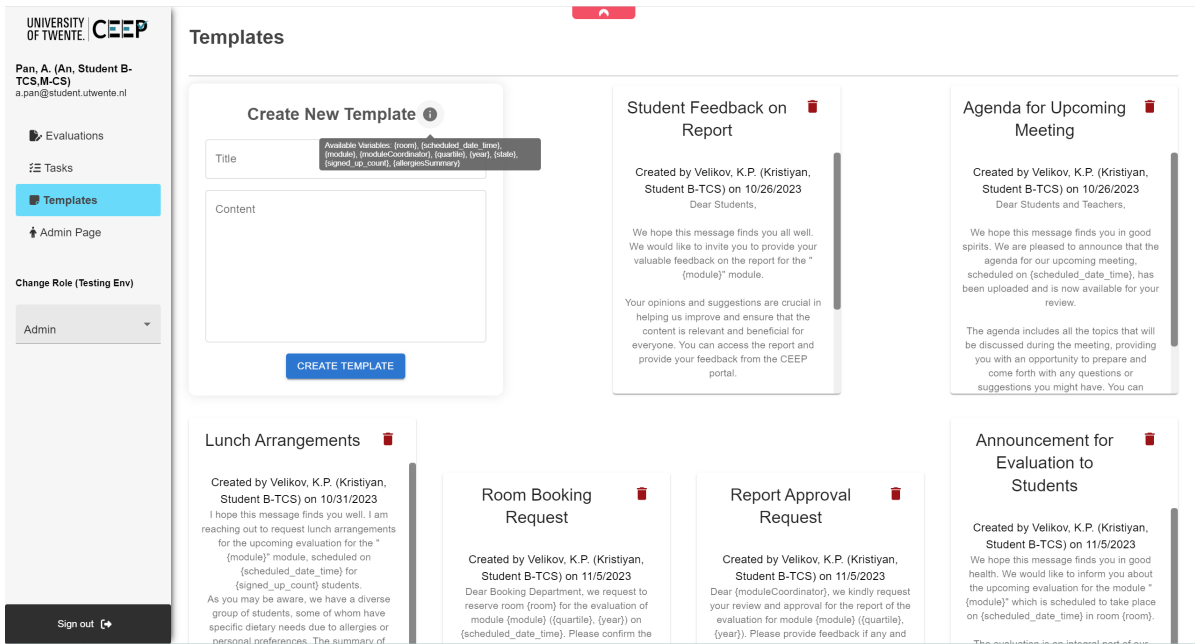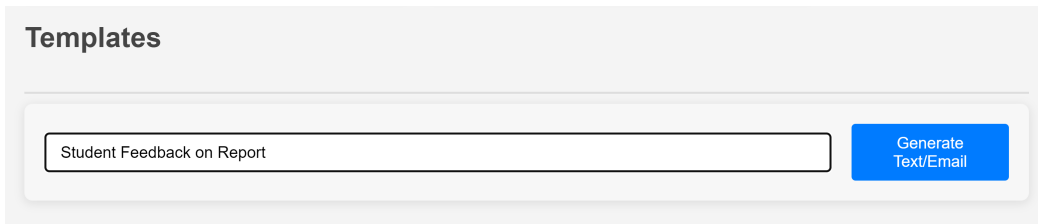Figure B.13: Template Page Overview

Figure B.14: Generate Template Functionality

C STAKEHOLDERS

C.1 Stakeholder Onion