



# CS319 Object-Oriented Software Engineering

*Internship Management System*

## Project Analysis Report

Ahmet Alperen Yılmazyıldız 22002712

Ahmet Berke Gökmen 22002105

Erdem Eren Çağlar 22002359

Mahmut Mert Gençtürk 22003506

Cem Apaydin 21802270

Instructor: Eray Tüzün

Teaching Assistant(s): Yahya Elnouby, Muhammad Umair Ahmed and Tolga Özgün

Second Iteration

May 21, 2023

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the CS319

# **Contents**

|  |           |
|--|-----------|
| <b>1 Introduction</b>  | <b>3</b>  |
| <b>2 Current System</b>  | <b>3</b>  |
| <b>3 Proposed System</b>   | <b>5</b>  |
| 3.1 Overview   | 5         |
| 3.2 Actors   | 6         |
| 3.3 Non-functional Requirements                                    | 7         |
| Usability  | 7         |
| Reliability  | 7         |
| Security / Safety  | 8         |
| Performance  | 8         |
| Maintainability  | 9         |
| 3.4 Pseudo Requirements  | 9         |
| 3.5 System Models  | 9         |
| 3.5.1 Use-Case Model   | 9         |
| 3.5.2 Object and Class Model                                       | 31        |
| 3.5.3 Dynamic Models   | 33        |
| 3.5.3.1 Activity Diagrams  | 33        |
| 3.5.3.1.1 Activity Diagram for Assigning Evaluators                | 33        |
| 3.5.3.1.2 Activity Diagram for Assigning Assistants                | 34        |
| 3.5.3.1.3 Activity Diagram for Part A Evaluation                   | 35        |
| 3.5.3.1.4 Activity Diagram for Part B Evaluation                   | 36        |
| 3.5.3.2 State Diagrams   | 36        |
| 3.5.3.2.1 State Diagram for Semester Initialization & Deactivation | 36        |
| 3.5.3.2.2 State Diagram for Evaluation of the Part A               | 37        |
| 3.5.3.2.3 State Diagram for Evaluation of the Part B and C         | 37        |
| 3.5.3.2.4 State Diagram for Final Course Grade                     | 38        |
| 3.5.4 User Interface   | 39        |
| <b>4 Improvement Summary</b>                                       | <b>53</b> |
| <b>5 References</b>  | <b>54</b> |

# Project Analysis Report

*Internship Management System*

## 1 Introduction

Our project constitutes an internship management system that will allow the students, secretaries, teaching assistants, and evaluators to conduct their necessary operations easier than the currently used system. We, as the team of this application, want to provide an application that the users will spend less time and effort on throughout this process. We endeavor to expedite the current system by eliminating any unnecessary by-hand operations.

The system will ease the operations related to internships by providing an application that allows the students to track their submission's status and submit their (revisioned) reports and comments about what they have changed via the application. The students won't need any more agent or effort to find out their submission's situation since the status check we offer will inform the student about any revisions and their submission's progress until completion. The evaluators and the teaching assistants will be able to track the forms that are assigned to them. The teaching assistants will be able to check the reports before the evaluators and request changes. The evaluators will have more authority over these operations compared to the assistants. The evaluators will be able to grade the students' reports, request revisions, add comments to the reports, and assign and change submission deadlines. The admins or secretaries will have the most authority over the application. They will be able to track each operation managed through this application and manipulate the process by adjusting the system's management, such as the deadlines, assigned evaluators and teaching assistants, semester initialization, and deactivation.

As detailed in the following section, the current system requires many redundant exertions that each individual taking part in this process has been going through for many years. This internship management system will enable the existing mechanism to operate more efficiently and spare all individuals from this burden, especially the secretaries.

## 2 Current System

One of the main stakeholders of this project and one of the clients of this project, Selim Aksoy, presented how the internship reports are handled in the current system. In the current system, the internship reports are managed in a decentralized way. There is no inclusive platform that contains all of the internship management features. The current internship management is done in 3 different platforms: Moodle, Google Drive, and lastly, e-mail. The main steps of internship management are listed as this:

1. Students are required to upload their internship reports to Moodle until an established deadline.
2. Summer training evaluation forms filled by companies collected by the faculty secretary.
3. After this, the internship reports are pre-reviewed by the teaching assistants and according to their feedback, students may need to submit their reports again to Moodle.
4. Faculty secretary collects these reports from Moodle and uploads them to Google Drive for faculty members' evaluation alongside the company evaluation forms of the students.
5. Faculty members are assigned to the reports as evaluators by the faculty secretary. Assignments of faculty members are done according to the ranking of the student's name and surname, as Selim Aksoy indicated in the presentation. Also, the faculty secretary can arrange the assignment of evaluators according to how many students a faculty member will evaluate. Assigned faculty members find which student's internship report to evaluate from the Google Drive folders that the faculty secretary creates in the 4th step.
6. After the evaluation, faculty members may require revision from the student by indicating which changes they should make before proceeding to the report's final version. Faculty members may require multiple revisions until they find the report satisfactory for final assessment. Evaluators send an email explaining what changes students should make, or students can view the feedback in Google Drive. Google Drive links are made available to the students on a website specifically designed to announce the grades. The website contains a simple table with student IDs and corresponding letter grades. If the faculty member requires revision, they put the Google Drive link in the grade section. Since this website is public, students can see not only their grades or report status but also other students' grades and report status. Moreover, they can reach other students' feedback by clicking the Google Drive link, which is causing privacy issues on a large scale. Also, it is impractical for the students to find their internship report status in the pile of student IDs.
7. Students re-submit their report with an explanation of the changes they made by sending an e-mail to their evaluators by the established deadline.
8. Faculty members assess the final version of the reports by filling out a form about the quality of the internship report. By combining the grade of the form with the company evaluation form, the evaluator grades the overall internship report as satisfactory or unsatisfactory.
9. The grades are announced on a website, which is mentioned in the 6th step. As mentioned before, in the 6th step, students need to find their

student ID and grades from the table, which is not a practical way of learning grades. Moreover, students can also see other students' grades which violates the privacy of the students.

As can be seen, these processes are made on independent platforms. Since there is no built-in connection between these platforms, keeping track of the progress may take a lot of work for all parties. For example, faculty members cannot track how many internship reports are left to evaluate, and the faculty secretary needs to send the summer internship reports by hand. Students cannot see the progress of their internship reports properly. Also, revision requirements made by faculty members and re-submission of the students were made manually through email. These issues cause unnecessary delays in the process, making internship management difficult for all parties.

## **3 Proposed System**

### **3.1 Overview**

The current system uses manual methods, which needs to be more practical and results in excessive time loss for students and academic staff. Manual paperwork should take little to no time. Therefore we propose to create a browser-based system that will benefit all parties involved in the internship report evaluation cycle.

Our system will automate the processes currently done by hand (see section 2). Initially, the users of the application will be able to log in to the system by using the credentials that are automatically created by the super admin upon each semester start. Then, depending on the user's role, they will be shown a dashboard to access their related work. For instance, a student will be able to see their applications, track their reports' statuses, and resubmit their revised report if they receive a change request from either assistants or evaluators. Evaluators will see what reports they need to evaluate, will be able to request changes by explaining what is missing or needs to be changed, or approve the report and assign grades to each evaluation criterion. Assistants, just like evaluators, will be able to request changes by providing what needs to be changed on the reports.

Most importantly, the system will automate the process of semester and class tracking; that is, the secretaries (admins) will no longer have to deal with each student separately. They will get an excel sheet exported from STARS containing who is taking a specific internship course this semester. Then, upon uploading the excel file to the system, all the users' accounts will be automatically created (if they have not taken any course before), and they will get a notification email saying that "Your account has been registered to X semester, Y course. You can view your applications on your dashboard". Moreover, the admins will be able to add individuals to each semester course, such as assistants who joined later or evaluators. Furthermore, the secretaries will no longer have to direct company evaluation forms to evaluators by hand.

The system will send an automatic email to the student's supervisor with a one-time link, and the supervisor will be able to provide feedback using the system. However, if they want, secretaries can still get the grades in an enclosed letter. Then, they can upload the evaluation report to the system so that it can be viewed by faculty members while grading the work. Lastly, they will be able to set deadlines, send email notifications to students, and change which evaluator and assistant is assigned to which report if required. At the end of each semester, the admins will be able to deactivate a semester course and export all students' grades alongside their required information as an excel sheet.

In summary, our web-based application provides a new automated way of dealing with internship reports and combines everything into one place. It offers solutions for all parties and aims to reduce the time they spend during the internship report evaluation process.

## 3.2 Actors

As a result of the requirement analysis, we have identified four types of users that will be using the internship management system.

- **User:** User is the most basic type of user in the system. It encapsulates all other user roles and provides basic access to the system. Such as logging in & out, resetting passwords, requesting password reset link and getting notifications about new updates.
- **Student:** Student users are created by admins (secretaries) upon semester initialization. They cannot register to the system on their own; their accounts are automatically created when the admin starts the semester with a specific course (e.g., 2022-2023-spring CS299). If a student account is already created, their account will automatically be registered to the newly created semester (this feature allows students to take CS299 and CS399 simultaneously). Students in the system can view their registered semesters and internship report status. Moreover, they can create submissions in which they provide the internship report and related comments if needed for revisions. They can view all the annotated pdf or comment feedback from assistants and evaluators. Then, according to the assistant's and evaluator's response to the report, they upload a revised version of their report.
- **Assistant:** Assistants are responsible for pre-checks (grammar, spelling, Turnitin) on reports before evaluators can view them. They can request changes in the report and approve them.
- **Evaluator:** Evaluators are responsible for evaluating the work and reports in a submission. They can request changes in the report and approve them. Furthermore, they can assign specific points to each field in the work evaluation form, thus grading the report.

- **Secretary:** Secretary's capabilities are broad. They can initialize a semester with a course and create accounts for each user. After initialization, if required, they can add additional people (TAs, evaluators, other students) to an existing semester so that they can start working on reports. Moreover, they can register courses (such as CS299 and CS399), allowing for different courses to be dynamically recognized by the system. Furthermore, they have the required role of viewing all the reports, submissions, and semesters in the system. If needed, they can search for users by using their names to get all the information about them. Furthermore, they can get an excel report of students and corresponding grades alongside the statistics of the semester. At the end of each semester, admins can deactivate that specific semester, so that no further action can be taken on submissions.
- **Department Chair:** The department chair has the ability to do everything that secretaries can do. On top of that, they can create secretary accounts.

### **3.3 Non-functional Requirements**

#### ***Usability***

As the project is designed to ease the report evaluation procedure, it has to provide a clear, unique, and easy-to-interact interface. It should provide easy access to all the frequently used features, such as showing reports and their statuses for students, reports waiting to be graded to assistants and evaluators, and lastly, initializing a semester with a group of students should be as easy as possible. All these features should be at most five clicks away from the dashboard.

- Uploading, downloading, and evaluating a report will be as simple as clicking a button.
- The colors of the reports will match their status. Red if it is rejected and requires changes, orange if it's waiting for review, and green if approved.
- Responsive design approaches will be used; the system should easily be used via any device. The website should be accessible and easily operated through a mobile phone with a small screen which means no two components go over another in any screen.

#### ***Reliability***

Regarding reliability, the system should ensure that it will not lose any data in case of an exception and, no reports will get stuck meaning they can neither be evaluated nor re-uploaded. The system's functionalities should be solid

as a rock and function as expected. All the end user expectations from the system should be satisfied.

- Since all of the components of the system (Database, backend, frontend, file storage, etc.) will be deployed to a cloud provider such as AWS, it will be ensured that the system will be up and running 99% of the time.
- Since the system must not lose any historical data for legal purposes, automatic database archiving will be performed monthly.

## ***Security / Safety***

Since the system will store sensitive user information (email, name, internship data, etc.), it should be protected against database attacks such as No-SQL injections. Also, the system should ensure that no student can see another's reports, status, or progress. Moreover, some people should only be able to access resources or view pages if they have the required roles.

- To protect privacy, only limited information about the students will be visible to other parties (students, assistants, evaluators).
- The passwords of all the users will not be saved as plain text in the database. They will be hashed using BCrypt [2], and the hashed version will be saved into the database. The hashed version will be compared against the plain text password on each login request to ensure secure transactions between server and client.
- There will be no register option; only admins can create student profiles upon the semester beginning and use the related excel file exported from STARS. This ensures that no unnecessary user is registered into the system.
- One student can only have one account, which will be created and tied to their Bilkent id and email.

## ***Performance***

The system should ensure that no operation takes too much time, that is more than one second to load. All stakeholders should be pleased with the system and not get frustrated due to slowness.

- Navigation between pages will be almost instant since the system will use dynamic page generation, it is expected to take at most one second navigating between the pages.
- Some frequently accessed resources (such as announcements) will be cached using an in-memory database for faster response times, such as 50 milliseconds.

## **Maintainability**

With the design patterns and state-of-the-art techniques used in the system, it will be easily extendable and improved.

- Since the system will be built on top of OOP (Object Oriented Programming) principles, it will be easy for developers to utilize features like encapsulation and polymorphism.
  - All backend endpoints' documentation will be available through the Swagger API documentation tool to ease testing and frontend development & maintenance.

## **3.4 Pseudo Requirements**

A numbered list is below:

1. The project must be a web based application.
  2. The project must be implemented in an Object-Oriented programming language and use an Object-Oriented design pattern.

## 3.5 System Models

### **3.5.1 Use-Case Model**

High resolution version of the use case diagram can be seen [here](#).

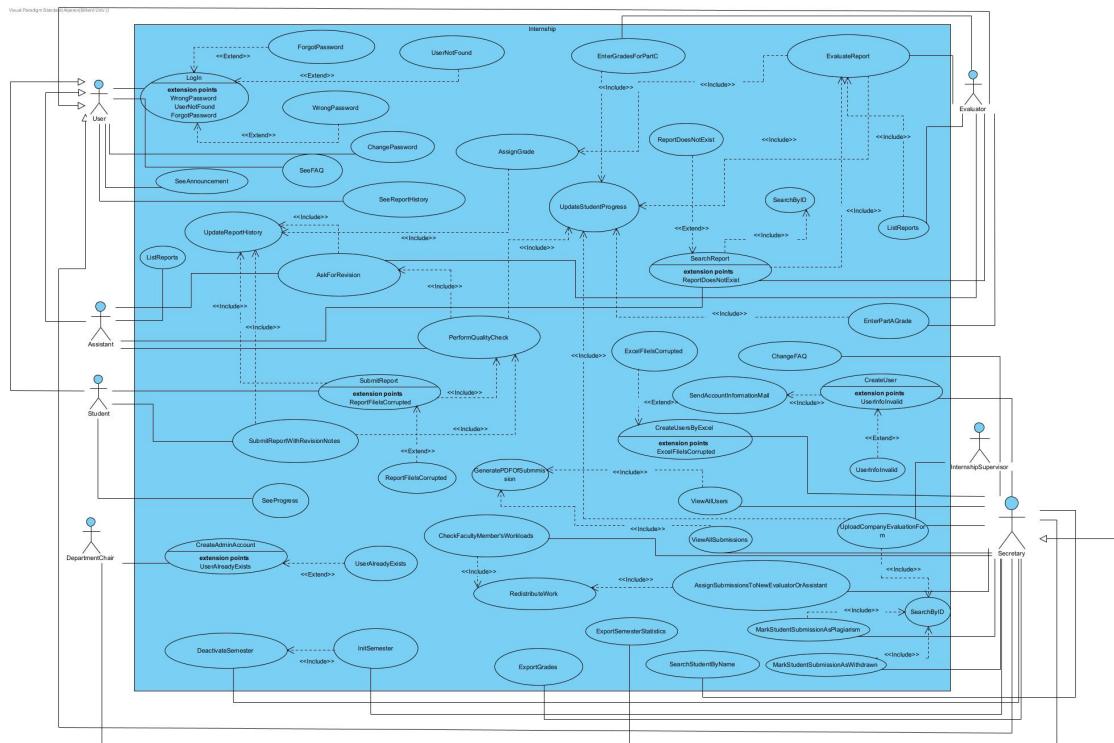


Figure 1: Use Case Diagram for the Internship Management System

**Use case name:** LogIn

**Participating actors:** Initiated by User

**Flow of events:**

1. The User enters their account information (Bilkent ID, password) to the login form of the system.
2. The User activates the login procedure by clicking the sign in button below the form.
3. IMS takes the information from the form to evaluate the login credentials.

**Entry condition:** The user does not have a valid token.

**Exit condition:** . The User granted with an valid token; **OR**

- . The process was canceled because of an error.
- 

**Use case name:** ForgotPassword

**Participating actors:** Initiated by User

**Flow of events:**

1. The User sends a request for the change his/her password by clicking forgotPassword button and entering their email.
2. IMS sends verification code to the requested account's email to verify the request's validation.
3. User clicks the link generated by the system that was sent to his/her email, to the system.
4. IMS verifies the code and generates a form for the new password of the user.
5. The User enters the new password which will be used as their new password in the system. Then confirm it and activate the changePassword function by clicking the submit button.

**Entry condition:** . Entered id by the user must be present in the system and link with an active account, OR

- . The code in the link should be valid UUID and should not be expired or used.

**Exit condition:** . The User's password successfully changed, **OR**

- . Process exterminated because of an error at the process of code verification. The code may already be used or expired.
- 

**Use case name:** WrongPassword

**Participating actors:** Initiated by User

**Flow of events:**

1. User enters his/her account information wrong five times in a row.
2. IMS applies a lock on the IP address so that the User cannot submit the login form again and has to wait one minute.

**Entry condition:** . User's id was found in the system however the entered password was not correct.

**Exit condition:** . User waits one minute.

---

**Use case name:** UserNotFound

**Participating actors:** Initiated by User

**Flow of events:**

1. User enters its account information to the login form
2. IMS could not find the entered id in the system database five times in a row.
3. IMS applies a lock on the IP address so that the User cannot submit the login form again and has to wait one minute.

**Entry condition:** . User tried to enter the system with an id that does not exist in the database.

**Exit condition:** . Successful login or lock on the IP address expires.

---

**Use case name:** ChangePassword

**Participating actors:** Initiated by User

**Flow of events:**

1. User clicks the change password button to generate the passwordChange function on her/his terminal.
2. User enters a new password. After that User clicks the submit button.
3. IMS takes the password information and changes it with the old one in the database after the request.

**Entry condition:** . User should successfully login to the system and has a valid token.

- . Users should enter the account Settings page (Change password page).

**Exit condition:** . User successfully changes their password.

---

**Use case name:** SubmitReport

**Participating actors:** Communicates with Student

**Flow of events:**

1. Student upload his/her internship report to the system
2. IMS checks for the file size limit then activates the submit button.
3. Student clicks the submit button and activates the uploadReport function on his/her end.
4. IMS saves the uploaded report to the database and links it to Student's submission then activates updateReportHistory use case.

**Entry condition:** . User should successfully login to the system and has a valid token throughout the process **AND** submission should be in a valid state to upload another report.

**Exit condition:** . Student successfully uploaded his/her report to the system **OR**

- . The process was canceled because the uploaded file exceeded the limit.
- 

**Use case name:** SubmitReportWithRevisionNotes

**Participating actors:** Communicates with Student

**Flow of events:**

- 1.** Student uploads revised internship report to the corresponding field.
- 2.** Student uploads revision notes to the corresponding field
- 3.** IMS checks for the file size limit then activates the submit button.
- 4.** Student clicks the submit button and activates the uploadReportWithRevision function on his/her terminal.
- 5.** IMS saves the uploaded revised report and revision notes to the database and links it to Student's submission then activates updateReportHistory use case.

**Entry condition:** . User should successfully login to the system and has a valid token throughout the process **AND** submission should be in a valid state to upload another report.

. The previous version of the report should not get approval from the Evaluator or Assistant.

**Exit condition:** . Student successfully uploaded his/her report and revision notes to the system **OR**

. The process was canceled because the uploaded file exceeded the file size limit or another unexpected error occurred.

---

**Use case name:** SeeProgress

**Participating actors:** Communicates with User

**Flow of events:**

- 1.** Student navigates to the dashboard and activates the progress function of his/her submission on his/her terminal.
- 2.** IMS shows the current state of the submission, state history of the submission with dates, and next state submission should be achieved to get approval.

**Entry condition:** . Student should have successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Student sees his/her submission's progress.

---

**Use case name:** SeeFAQ

**Participating actors:** Communicates with User

**Flow of events:**

1. User clicks the seeFAQ button on his/her dashboard and activates the FAQ function on his/her terminal.
2. IMS shows the current FAQ and their answers written by the Admin.

**Entry condition:** . User should have successfully logged in to the system and have a valid token throughout the process.

**Exit condition:** . User successfully sees the FAQ.

---

**Use case name:** SeeAnnouncement

**Participating actors:** Communicates with User

**Flow of events:**

1. User clicks the seeAnnouncement button on his/her dashboard and activates the Announcement function on his/her terminal.
2. IMS shows the current Announcements written by the Admin.

**Entry condition:** . User should have successfully logged in to the system and have a valid token throughout the process.

**Exit condition:** . User successfully sees the Announcements.

---

**Use case name:** QualityCheck

**Participating actors:** Communicates with Assistant

**Flow of events:**

1. Assistant downloads the assigned Student's report by the system.
2. IMS activates the updateReportHistory function to change the report's state.
3. IMS creates a form for evaluating the report.
4. In the case of an unsuccessful report Assistant activates askForRevision use case by requesting changes.

**5.** IMS activates the updateReportHistory use case to change the report's state.

**Entry condition:** . There should be a submitted report from a Student with a state of under assistant evaluation. This use case is initiated by the system.

**Exit condition:** . Assistant successfully evaluates the Student's report and changes its state in the system.

---

**Use case name:** AskForRevision

**Participating actors:** Communicates with Evaluator and Assistant

**Flow of events:**

**1.** Assistant or Evaluator upload feedback notes to the system to indicate what should be changed in the revised version of the Student's internship report.

**2.** IMS checks the size limit of the uploaded file and activates the submit report.

**3.** Assistant or Evaluator clicks submit button and activates giveFeedback function on his/her terminal.

**4.** IMS saves the feedback notes to the system and links it to the Student's report.

**5.** IMS activates UpdateReportHistory use case to change the report's state.

**Entry condition:** . This use case **includes** the qualityCheck and evaluateReport use cases. Therefore, it requires a submitted report from the Student and is initiated by the system.

**Exit condition:** . Assistant or Evaluator successfully gives feedback to the Student's report and change its state to "waiting student revision", **OR**

. The process was canceled because the uploaded file exceeded the size limit.

---

**Use case name:** EnterCompanyEvaluationForm

**Participating actors:** Communicates with InternshipSupervisor

**Flow of events:**

**1.** InternshipSupervisor clicks the enterGradeForInternship button to generate company evaluation form function on his/her terminal.

**2.** IMS generates a company evaluation form on the page and a notification popup to inform InternshipSupervisor about Bilkent's passing grade policy.

**3.** InternshipSupervisor enters the grade of the associated Students internship.

**4.** After all the inputs are filled, IMS activates the submit button.

**5.** InternshipSupervisor clicks the submit button.

**6.** IMS generates reports from entered grades in PDF format and sends it to Admin.

**Entry condition:** . This use case is activated after the EnterTheSystemByOneTimeLink use case. This use case is initiated by the system.

**Exit condition:** . InternshipSupervisor successfully filled the company evaluation form for the associated student.

---

**Use case name:** SearchUserByName

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

**1.** The Secretary/Department Chair searches a Student from the search label by name.

**Entry condition:** . The Secretary/Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . The Secretary/Department Chair sees the Student's information.

---

**Use case name:** EnterGradesForPartC

**Participating actors:** Communicates with Evaluator

**Flow of events:**

**1.** Evaluator sees the part B grade of the selected Students report assigned to him/her.

- 2.** Evaluator enters the grades for the part C to the form.
- 3.** IMS checks if any field remains blank and activates the submit button.
- 4.** Evaluator clicks to submit button to generate passingNotification function on his/her terminal.
- 5.** IMS generates a notification box to inform the Evaluator about bilkent passing grades and activates the "I Approve" button for the entered grades.
- 6.** Evaluator clicks the "I Approve" button to generate the enterPartC function on the system.
- 7.** IMS saves the entered grades to the associated Students submission and generates updateStudentProgress to change the submission's state.

**Entry condition:** . Evaluator should successfully login to the system and have a valid token throughout the process **AND** the submission should be in waiting grading state.

**Exit condition:** . Evaluator successfully entered part C grades of the associated Student, **OR**

. The process is canceled and filled form saves as an "ongoing evaluation" but does not update the Students process because Evaluator's exit the system without clicking the "I Approve" button.

---

**Use case name:** SearchByID

**Participating actors:** Initiated by Evaluator and Assistant and Secretary and Department Chair

**Flow of events:**

- 1.** User types in the ID of the desired user.
- 2.** User presses the search button.
- 3.** IMS returns the information associated with the user and depending on the including use case.

**Entry condition:** . This use case is included by MarkStudentSubmissionAsPlagiarism, MarkStudentSubmissionAsWithdrawn, SearchReport and UploadCompanyEvaluationForm.

**Exit condition:** . User sees the corresponding information depending on the including use case.

---

**Use case name:** SearchByName

**Participating actors:** Initiated by Secretary and Department Chair

**Flow of events:**

1. Secretary or Department Chair types in the desired name in the search box.
3. IMS returns the list of users with the matching name.

**Entry condition:** . The Secretary or Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . The secretary or Department Chair receives the list of users.

---

**Use case name:** SearchReport

**Participating actors:** Initiated by Evaluator and Assistant

**Flow of events:**

1. Evaluator or Assistant clicks the see assigned submissions button to generate a listReports function on his/her terminal.
2. IMS lists all the reports that are assigned to the assistant or evaluator.
5. Evaluator selects a report and clicks the evaluate button to generate EvaluateReport function on his/her terminal or Assistant clicks the evaluate button to generate PerformQualityCheck on his/her terminal.
6. IMS redirects the Evaluator to the evaluation page of the selected report.

**Entry condition:** . Evaluator or Assistant should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Evaluator or Assistant successfully enters the target Student report's evaluation page.

---

**Use case name:** EvaluateReport

**Participating actors:** Communicates with Evaluator

**Flow of events:**

- 1.** Evaluator can download the Student's internship report and if it is second or more iteration, revision notes and the report history.
- 2.** IMS shows the criteria of the evaluation after the downloading process ends.
- 3.** After the Evaluator finishes his evaluation process she/he can click the assignGrade button or askRevision button to generate assignGrade use case or askForRevision use case.
- 4.** According to the decision of the Evaluator IMS redirect the Evaluator to askForRevision or enterGrades page and generate updateStudentProgress use case to change the Student's submission history.

**Entry condition:** . Evaluator or Assistant should successfully login to the system and have a valid token throughout the process and the Student should have a submission in the system which is not completed.

**Exit condition:** . Evaluator successfully evaluates the selected Student's report and redirects to another page according to his/her decision about the report, **OR**

- . The process was canceled because the Evaluator exit from the system without making a decision about the Student's internship report and Student' process remains unchanged.
- 

**Use case name:** AssignGrade

**Participating actors:** Communicates with Evaluator

**Flow of events:**

- 1.** Evaluator enters the grades according to corresponding part's criteria to the form generated by the system.
- 2.** IMS checks no input remains blank and generates the submit button.
- 3.** Evaluator clicks the submit button and generates enterPartAGrade, enterPartBGrade or enterPartCGrade function on his/her terminal.
- 4.** IMS creates a notification box about Bilkent's grade policy and generates an "I Approve" button to get confirmation from the Evaluator.
- 5.** Evaluator clicks to the "I Approve" button.
- 6.** If the entered grades are below the benchmark IMS creates a notification box to inform the Evaluator that Student will get "U" grade from the registered lesson and generates a "I Approve" button.

**7.** Evaluator clicks the “I Approve” button.

**8.** IMS saves the entered grades to the associated Student’s report and generates updateReportHistory and updateStudentProgress use cases and redirect Evaluator to the dashboard page.

**Entry condition:** . This use case **is included by** the EvaluateReport use case. This use case is initiated by the system.

**Exit condition:** . Evaluator successfully entered grades for the selected Student’s report, **OR**

. The process is canceled because Evaluator terminates the process without clicking the “I Approve” button and gives confirmation about grade policy.

---

**Use case name:** ListReports

**Participating actors:** Initiated by Evaluator or Assistant

**Flow of events:**

**1.** Evaluator or Assistant navigates to the assigned reports page.

**2.** IMS lists all the reports that are assigned to Evaluator or Assistant.

**Entry condition:** . Evaluator or Assistant should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Evaluator or Assistant successfully sees all the reports assigned to him and that are pending.

---

**Use case name:** ChangeFAQ

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

**1.** Secretary/Department Chair clicks the FAQ button from his dashboard and generates a toFAQ function on his/her terminal.

**2.** IMS redirects the Secretary/Department Chair to the FAQ page.

**3.** Secretary/Department Chair sees the current FAQ for the internship report process for the Students and clicks the changeFAQ button.

- 4.** IMS generates a box to show the Secretary/Department Chair an input to enter new questions and answers.
- 5.** Secretary/Department Chair enters the new question and its answer to the system.
- 6.** IMS generates the saveChanges button.
- 7.** Secretary/Department Chair clicks the saveChanges button and generates saveNewFAQ function on his/her terminal.
- 8.** IMS saves the new uploaded FAQ to the system.

**Entry condition:** . Secretary/Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Secretary/Department Chair successfully uploaded the new question and its answer to the system

---

**Use case name:** CreateUser

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

- 1.** Secretary/Department Chair clicks the createUser button in her/his dashboard to generate the createUser function on his/her terminal.
- 2.** IMS redirect Secretary/Department Chair to a page that has fields representing necessary information for creating a user like email address, Bilkent Id, name and type and another "uploadExcel" button to activate the CreateUsersByExcel use case.
- 3.** Secretary/Department Chair enters User's information to the generated form.
- 4.** IMS checks if the no form field remains blank and activates the submit button.
- 5.** Secretary/Department Chair clicks the "Submit" button to activate saveUser function on his/her terminal.
- 6.** IMS creates a new User into the system by the given information.

**Entry condition:** . Secretary/Department Chair should successfully login to the system and have a valid token throughout the process, **AND**

. At the time the system should have an ongoing registered semester.

**Exit condition:** . Secretary/Department Chair successfully creates the User into the system.

---

**Use case name:** CreateUsersByExcel

**Participating actors:** Communicates with Secretary/Department Chair

**Flow of events:**

**1.** Secretary/Department Chair uploads his/her file in Excel format which contains necessary columns about the User, to the corresponding field.

**2.** IMS checks the Excel file for the necessary information columns and if the file contains all the prerequisite information for all the users, it generates a "Submit" button for the Secretary/Department Chair.

**3.** Secretary/Department Chair clicks the "Submit" button.

**4.** IMS creates a new User according to type information in the file for each row of the uploaded file and saves them into the database.

**Entry condition:** . This use case **extends** from the CreateUser use case. It is initiated by the system.

**Exit condition:** . Secretary/Department Chair creates User accounts according to the informations at the uploaded Excel file and saves them into the system, **OR**

. IMS stops the process because the uploaded Excel file does not contain one or more necessary information columns.

---

**Use case name:** AssignSubmissionToNewEvaluatorOrAssistant

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

**1.** Secretary/Department Chair navigates to assign submissions page.

**2.** IMS lists all of the submissions made for that semester.

**3.** Secretary/Department Chair chooses the submission/s and selects which type of user will be assigned to that submission. After that, Admin will enter the id of the user that will be assigned.

**4.** IMS will save the response and add the submission to the submission to be evaluated list of the evaluator or the assistant.

**Entry condition:** Secretary/Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** Secretary/Department Chair assigns submission to the evaluator or assistant successfully.

---

**Use case name:** UploadCompanyEvaluationForm

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

**1.** Secretary/Department Chair navigates to Part A file page of the selected Student.

**2.** IMS redirects Secretary/Department Chair to the new page with a label which Admin could upload the company evaluation form file to the system in PDF format.

**3.** Secretary/Department Chair uploads the company evaluation form to the corresponding field.

**4.** IMS checks the uploaded file size if it does not exceed the size limit and activates the "submit" report.

**5.** Secretary/Department Chair clicks the "submit" button to generate the "enterCompanyEvaluationForm" function on his/her terminal.

**6.** IMS saves the uploaded file to the system and links it to the relative Student's submission.

**Entry condition:** . Secretary/Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Secretary/Department Chair successfully uploads the company evaluation file to the system.

---

**Use case name:** ExportGrades

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

- 1.** Secretary/Department Chair clicks the “exportGrades” button from the selected semester’s page and generates the “transformGradesToExcel” function on his/her terminal.
- 2.** IMS creates a file to be downloaded by the Secretary/Department Chair which contains registered Student’s ID, name, and their current grade and semester information.
- 3.** IMS generates an “download” button for the Secretary/Department Chair to download the generated file.
- 4.** Secretary/Department Chair clicks the “download” button and downloads the generated file.

**Entry condition:** . Secretary/Department Chair should successfully login to the system and have a valid token throughout the process, **AND**

. At the time the system should have an ongoing registered semester.

**Exit condition:** . Secretary/Department Chair successfully downloads the “Grades” PDF.

---

**Use case name:** InitSemester

**Participating actors:** Initiated by Secretary/Department Chair

**Flow of events:**

- 1.** Secretary/Department Chair clicks the “InitSemester” button from his/her dashboard to generate the “StartSemester” function on his/her terminal.
- 2.** IMS redirects the Secretary/Department Chair to a new page which contains input fields for semester name, course name, and a file upload field for users’ information sheet (Excel file).
- 3.** Secretary/Department Chair fills the fields and uploads the file.
- 4.** IMS activates the “Start” button after it checks that no field remains blank and the file is valid.
- 5.** Secretary/Department Chair clicks the “Start” button to generate a new semester object in the system.
- 6.** IMS creates a new semester and saves it to the system as a semester. Then, the system creates users who are not previously enrolled into the system according to the information in the uploaded sheet, sends semester registration mails to all users and sends account information mails to the new users.

**Entry condition:** . Secretary/Department Chair should successfully login to the system and have a valid token throughout the process and have a valid Excel file that contains needed information for all users.

**Exit condition:** . Secretary/Department Chair successfully ends a new semester and saves it to the system.

---

**Use case name:** DeactivateSemester

**Participating actors:** Communicates with Secretary/Department Chair

**Flow of events:**

1. Secretary/Department Chair clicks the "DeactivateSemester" button from the current semester's page and generates the "DeactivateSemester" function on his/her terminal.
2. IMS generates a confirmation dialog to get confirmation from the Admin to end the semester and activates the "Deactivate" button.
3. Secretary/Department Chair clicks the deactivate button.
4. IMS marks the semester as deactivated and makes it available for exporting grades.

**Entry condition:** . This use case extends from the StartSemester use case. It is initiated by the system.

**Exit condition:** . Secretary/Department Chair successfully ends a semester of the system.

---

**Use case name:** CheckFacultyMember'sWork

**Participating actors:** Communicates with Secretary/Department Chair

**Flow of events:**

1. Secretary/Department Chair can see the number of reports assigned to each Evaluator and Assistant and may change the distribution of the submissions as he/she desires.
2. IMS will show a list of the evaluators' and assistants' assigned submissions and their statuses.

**Entry condition:** Secretary/Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** Secretary/Department Chair successfully sees or changes the assignments.

---

**Use case name:** RedistributeWork

**Participating actors:** Communicates with Secretary/Department Chair

**Flow of events:**

1. Secretary/Department Chair can see which reports are assigned to who and change their assigned evaluators and assistants.
2. IMS will show the list of reports and information of these reports' evaluators and a button for each report to change the evaluator.
3. Secretary/Department Chair clicks change buttons besides the reports and types in the Bilkent ID of the requested User.
4. IMS will assign the selected report to the selected evaluator.

**Entry condition:** Secretary/Department Chair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** Secretary/Department Chair successfully sees or changes the assignments.

---

**Use case name:** SeeReportHistory

**Participating actors:** Communicates with Secretary/Department Chair and Evaluator and Assistant

**Flow of events:**

1. Secretary/Department Chair, Evaluator or Assistant clicks the "see ReportHistory" button of the selected Student.
2. IMS generates a label for every report uploaded by the Student' submission, and every revision request by the Assistant and Evaluator.
3. Secretary/Department Chair, Assistant or Evaluator can see and download every uploaded report and revision requests.

**Entry condition:** . This use case extends from the CheckFacultyMemeber'sWork use case. This use case is initiated by the system.

**Exit condition:** . Secretary/Department Chair, Assistant or Evaluator successfully sees the report history of the selected Student.

---

**Use case name:** MarkStudentSubmissionAsPlagiarism

**Participating actors:** Communicates with Secretary/Department Chair

**Flow of events:**

1. Secretary/DepartmentChair navigates to mark submission as plagiarism or withdrawn page.
2. Secretary/DepartmentChair types in Student's id and selects which submission to handle.
3. Secretary/DepartmentChair clicks the Plagiarised button.
4. IMS marks the report as plagiarized and stops the evaluation process.

**Entry condition:** . Secretary/DepartmentChair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Secretary/DepartmentChair , marks the submission as Plagiarized.

---

**Use case name:** MarkStudentSubmissionAsWithdrawn

**Participating actors:** Communicates with Secretary/Department Chair

**Flow of events:**

1. Secretary/DepartmentChair navigates to mark submission as plagiarism or withdrawn page.
2. Secretary/DepartmentChair types in Student's id and selects which submission to handle.
3. Secretary/DepartmentChair clicks the Withdrawn button.
4. IMS marks the report as withdrawn and stops the evaluation process.

**Entry condition:** . Secretary/DepartmentChair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** . Secretary/DepartmentChair , marks the submission as Withdrawn.

---

**Use case name:** ExcelFileIsCorrupted

**Participating actors:** Communicates with Secretary/DepartmentChair

**Flow of events:**

1. Secretary/DepartmentChair navigates to initialize a semester page.
2. Secretary/DepartmentChair uploads the Users file and submits the form.
3. IMS rejects the file due to one of the several reasons. File is not an excel file, file does not contain required fields or file is empty.

**Entry condition:** . Uploaded file was corrupted.

**Exit condition:** . IMS rejects the file and tells the Secretary/DepartmentChair to upload a valid file via a pop up.

---

**Use case name:** ReportFileIsCorrupted

**Participating actors:** Communicates with Student

**Flow of events:**

1. Student navigates to create a submission page.
2. Student uploads the report file and submits the form.
3. IMS rejects the file due to one of the several reasons. File is not a PDF file, or the file is not readable.

**Entry condition:** . Uploaded file was corrupted.

**Exit condition:** . IMS rejects the file and tells the Student to upload a valid file via a pop up.

---

**Use case name:** UserInfoInvalid

**Participating actors:** Communicates with Secretary/DepartmentChair

**Flow of events:**

1. Secretary/DepartmentChair navigates to create a User page.

- 2.** Secretary/DepartmentChair enters the information of the User (ID, email, role, department, etc.)
- 3.** Secretary/DepartmentChair submits the form.
- 4.** IMS rejects the creation of the user since the given information is invalid.

**Entry condition:** . Given User info is invalid.

**Exit condition:** . IMS does not create the user and creates a pop saying which fields are invalid and need to change.

---

**Use case name:** ViewAllUsers

**Participating actors:** Communicates with Secretary/DepartmentChair

**Flow of events:**

- 1.** Secretary/DepartmentChair clicks the "ListUsers" button and generates a list function on his/her terminal.
- 2.** IMS lists all the existing users with their type and their registered semester.

**Entry condition:** Secretary/DepartmentChair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** IMS successfully lists all the registered users to the system.

---

**Use case name:** ReportDoesNotExist

**Participating actors:** Communicates with Evaluator/Assistant

**Flow of events:**

- 1.** Evaluator/Assistant tries to check the report of the student by searching the student by his/her ID.
- 2.** IMS does not forward the report since it does not exist.

**Entry condition:** Student has not uploaded any report.

**Exit condition:** IMS does not forward the report and creates a pop saying that the report does not exist.

---

**Use case name:** ViewAllSubmissions

**Participating actors:** Communicates with Secretary/DepartmentChair

**Flow of events:**

**1.** Secretary/DepartmentChair clicks the “ListSubmisions” button and generates a listSubmission function on his/her terminal.

**2.** IMS lists all the submissions of the students who registered for the current semester.

**Entry condition:** Secretary/DepartmentChair should successfully login to the system and have a valid token throughout the process.

**Exit condition:** IMS successfully lists all the submissions at the system.

---

**Use case name:** GeneratePDFOfSubmission

**Participating actors:** Communicates with Secretary/DepartmentChair

**Flow of events:**

**1.** Secretary/DepartmentChair selects one submission from the user list or submission list.

**2.** Secretary/DepartmentChair clicks the “createPDF” button to generate the createPDF function on his/her terminal.

**3.** IMS creates the latest version of the submission as a PDF file.

**Entry condition:** This use case **is included by** ViewAllSubmissions and ViewAllUsers use cases.

**Exit condition:** IMS successfully creates a PDF file of the submission and the Secretary/DepartmentChair downloads it.

---

**Use case name:** CreateAdminAccount

**Participating actors:** Communicates with DepartmentChair

**Flow of events:**

**1.** DepartmentChair clicks the “createAdmin” button to generate createAdminForm to his/her terminal.

**2.** IMS redirects DepartmentChair to a new page which includes an input form for the admin creation.

**3.** DepartmentChair enters the admin accounts information to the corresponding input areas and after filling all the input labels clicks the “create” button.

**Entry condition:** Department Chair should successfully enter the system AND have a valid token throughout the process.

**Exit condition:** IMS successfully creates a new Admin account.

### ***3.5.2 Object and Class Model***

High resolution version of the class diagram can be seen [here](#).

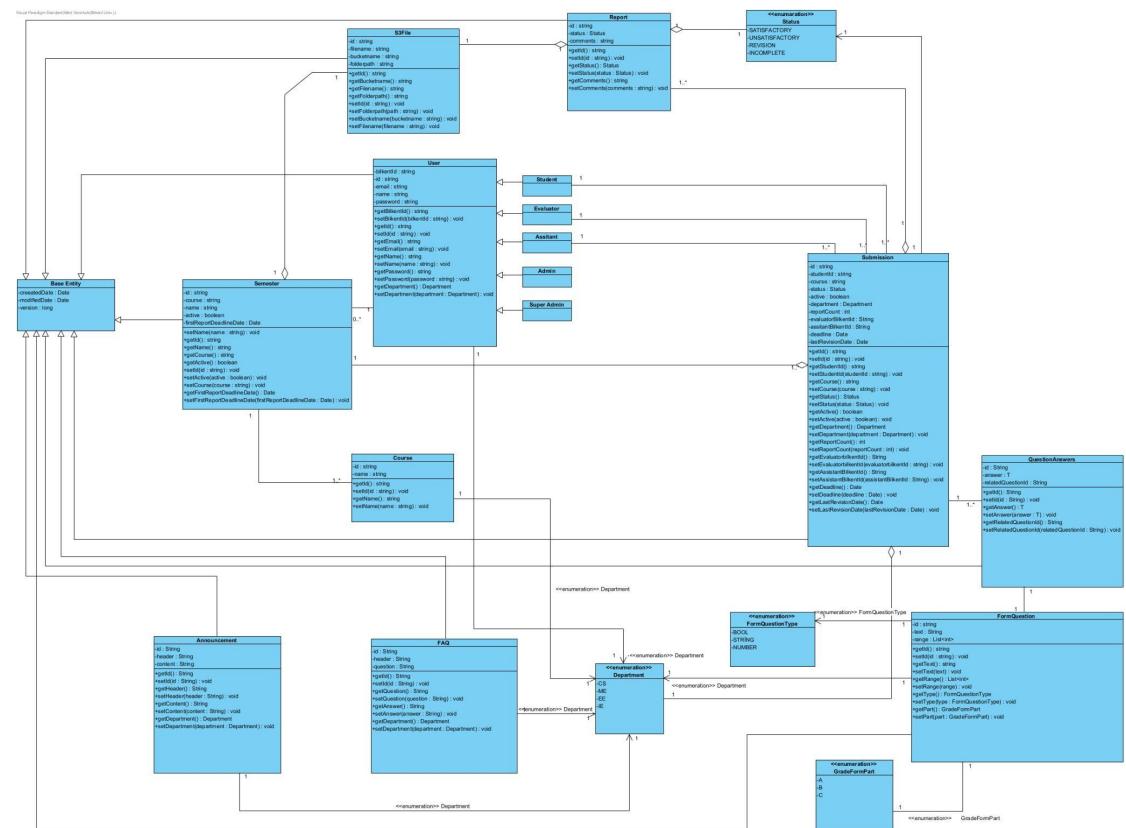


Figure 2: Object and Class models for the Internship Management System

**Base Entity:** A base supermodel that has creation date, modification date and version attributes. All the other models extend it.

**S3File:** A file model that is used to store the data of all kinds of files. Main purpose of this additional file class is to store a reference to the files in Amazon Simple Storage Service (S3). Since storing files in our main database would cost too much, we decided to use S3 and store related file id instead of files.

**Semester:** Semester model is used to store the course and semester information together, has a deadline for all the students to upload reports

**Course:** Course class is used to store course name and department information.

**User:** User model stores the data of all kinds of users. Different types of users extend the base user model. While the User model holds typical data for users like id, name and password, it also stores the semesters that the user has access to. For instance, if a student or evaluator has 2022-2023-spring/CS299 semester in their semester list, they will be able to do operations regarding that semester. Students will be able to upload reports, whereas the evaluator will be able to see assigned reports to them coming from that semester. Also, the Secretary and Department Chair can see semester statistics. Secretary and Department Chair users however, won't have any semester assigned to them, and they have control over semesters, all submissions and all users.

**Report:** Report model represents the single report file with its comments and status. Every file upload correlates to a report object.

**Submission:** Submission model represents the student's submission to the course. It stores the student's data, all the uploaded reports (before and after the reviews), the internship information, the course and semester information, and related evaluation form of their submission. It also shows the general status of students' reports. It is a general model for the evaluation process of the internship report. Moreover, it holds the assistant and evaluator information that is assigned to each submission.

**FormQuestion:** Form Question model is the model that stores the information of the final grading report's questions. It has different parts, since the report has different parts, and a type enumeration to specify what type of answer it needs. Questions also differ from department to department.

**QuestionAnswers:** Question Answer model is the model that stores the information of the answer of the related question. It has just the answer data and it only serves to create a grading form file and statistics file. Student's status of submission, satisfactory or unsatisfactory, will be decided according to these answers.

**FAQ:** Frequently Asked Questions is the information model that the Secretary and Department Chair create to inform other users. Questions can differ by department. It only holds the question and answer data.

**Announcement:** Announcement model is the information that the Secretary and Department Chair create to inform other users with an expiration date. It is similar to the frequently asked questions model, however, announcements have an expiration date.

### 3.5.3 Dynamic Models

#### 3.5.3.1 Activity Diagrams

##### 3.5.3.1.1 Activity Diagram for Assigning Evaluators

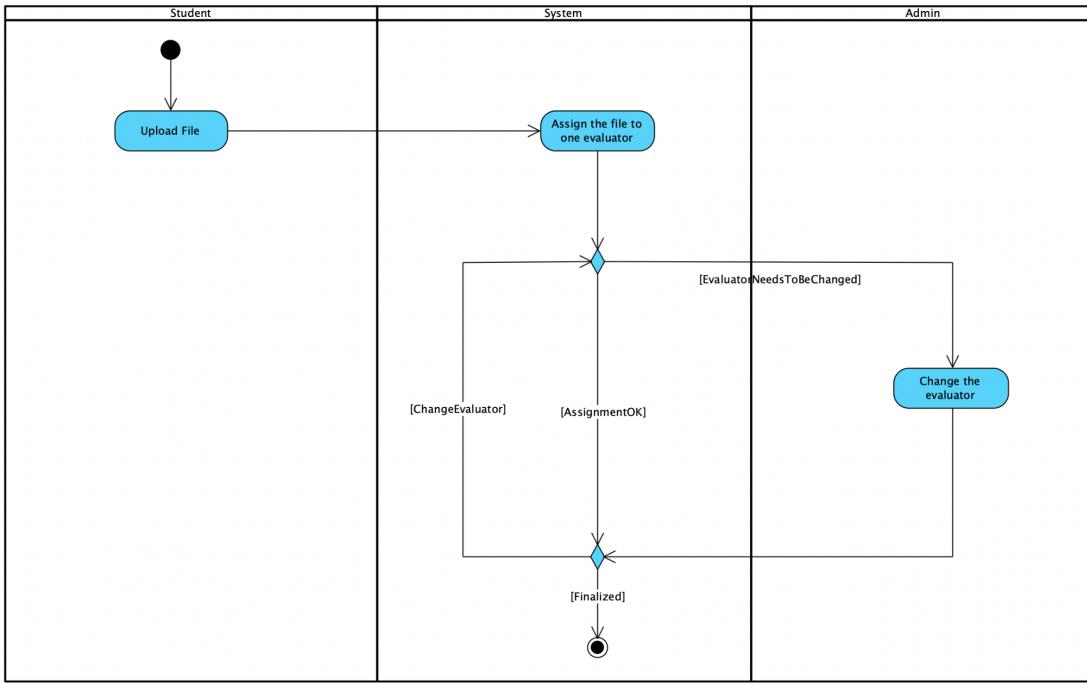


Figure 3: Changing Evaluator (Faculty Member) Activity Diagram for the Internship Management System

In the activity diagram for assigning teaching assistants, the student will upload his/her file. Then, the system will assign the file to a teaching assistant. If the assignment is convenient, there will be no changes to the assignment. If the teaching assistant needs to be changed, the admin will change the assigned teaching assistant. If any re-assignments need to be made after, the admin will be able to make this adjustment.

### 3.5.3.1.2 Activity Diagram for Assigning Assistants

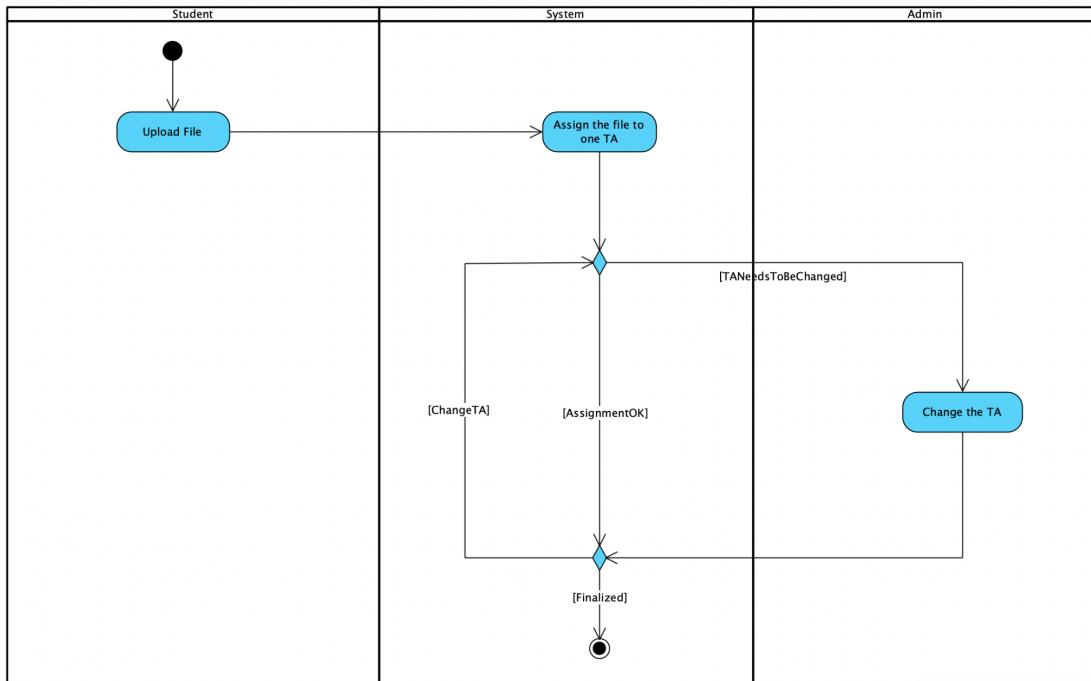


Figure 4: Activity Diagram of Assistant (TA) Assignment for the Internship Management System

In the activity diagram for assigning evaluators, the student will upload his/her file. Then, the system will assign the file to an evaluator. If the assignment is convenient, there will be no changes to the assignment. If the evaluator needs to be changed, the admin will change the assigned evaluator. If any re-assignments need to be made after, the admin will be able to make this adjustment.

### 3.5.3.1.3 Activity Diagram for Part A Evaluation

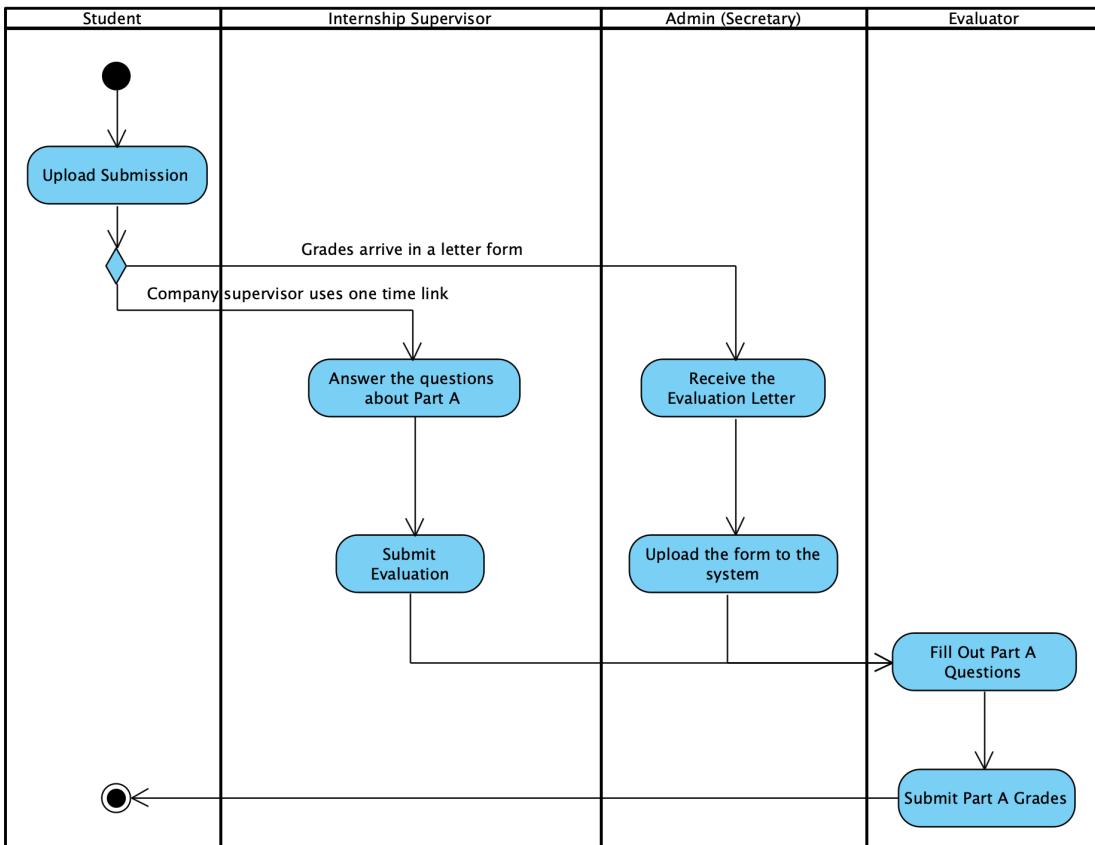


Figure 5: Part A Evaluation Activity Diagram for the Internship Management System

This diagram explains how the evaluation of part A is done. First, after the student uploads their submission, they choose how the company evaluation form will be delivered. If it'll be online via the link created by the system, it's only done once by the supervisor and the rest is handled by the system. If it'll arrive in letter form, then one of the department secretaries (admin) needs to upload that document to the system. Lastly, the faculty member (evaluator) looks at the grades and submits the overall evaluation for part A.

#### 3.5.3.1.4 Activity Diagram for Part B Evaluation

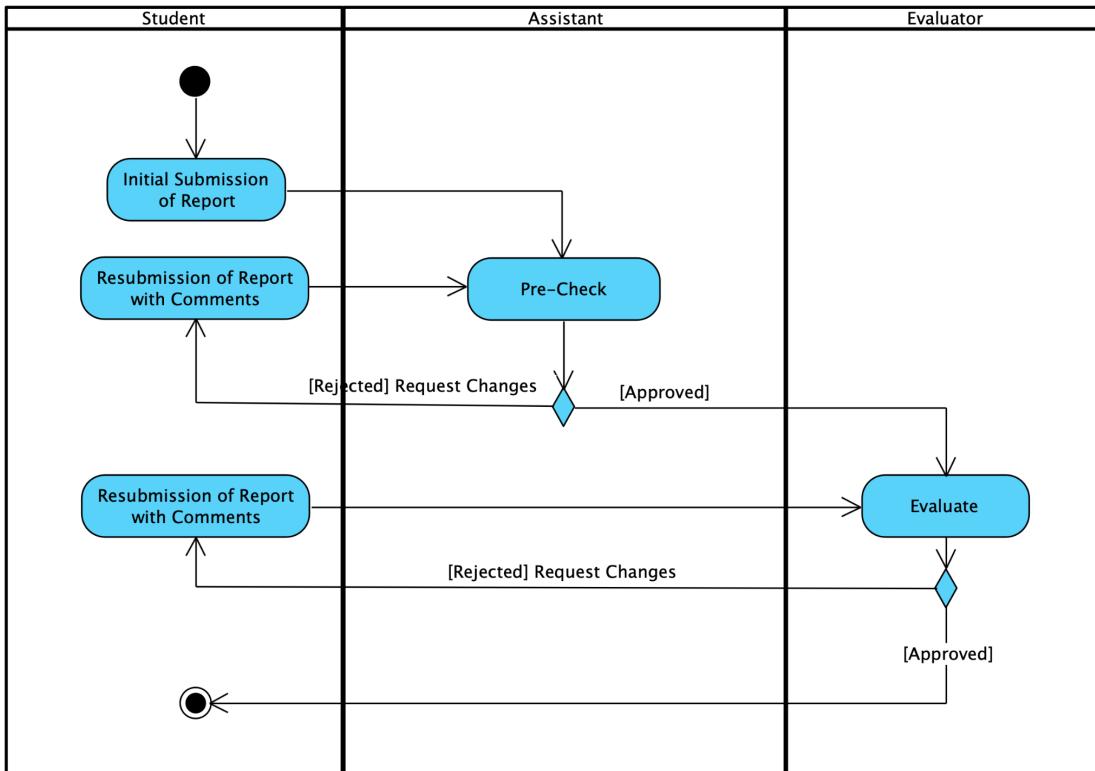


Figure 6: Part B Evaluation Activity Diagram for the Internship Management System

This diagram explains how the evaluation of part B is done. First, after the student submits their initial report, it's pre-checked by an assistant and re-uploaded by the student if requested. Then, it goes through the same steps with the evaluator, that is, it gets re-uploaded upon the evaluator's requests as many times as needed. Lastly, when the report no longer requires re-upload, the evaluator marks it as satisfactory.

#### 3.5.3.2 State Diagrams

##### 3.5.3.2.1 State Diagram for Semester Initialization & Deactivation



Figure 7: State Diagram for Semester Initialization and Deactivation

In the state diagram for semester initialization and deactivation, the admin of the internship management system will initialize the semester. The semester will be deactivated when the admin deactivates or the semester ends.

### 3.5.3.2.2 State Diagram for Evaluation of the Part A

High resolution version of the diagram can be seen [here](#).

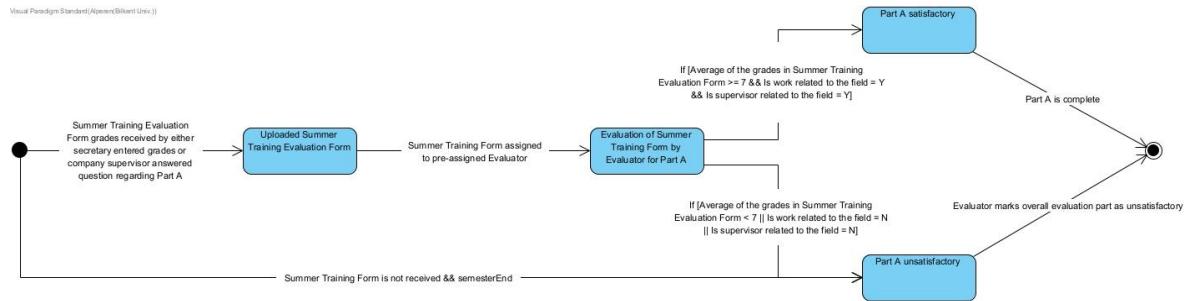


Fig. 8: Evaluation of the Part A State Diagram for the Internship Management System

This diagram shows how Part A of the internship evaluation form is evaluated.

### 3.5.3.2.3 State Diagram for Evaluation of the Part B and C

High resolution version of the diagram can be seen [here](#).

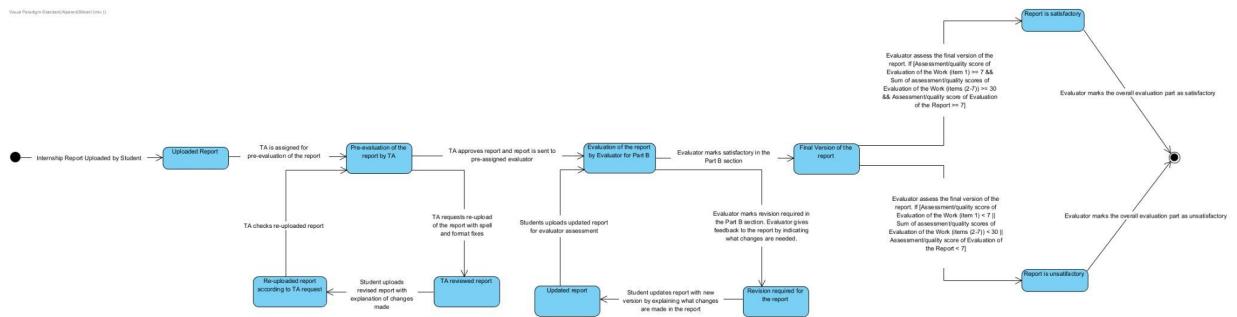


Fig. 9: Evaluation of Part B and C State Diagram for the Internship Management System

This diagram shows how the Part B and C of the internship evaluation form is evaluated.

### 3.5.3.2.4 State Diagram for Final Course Grade

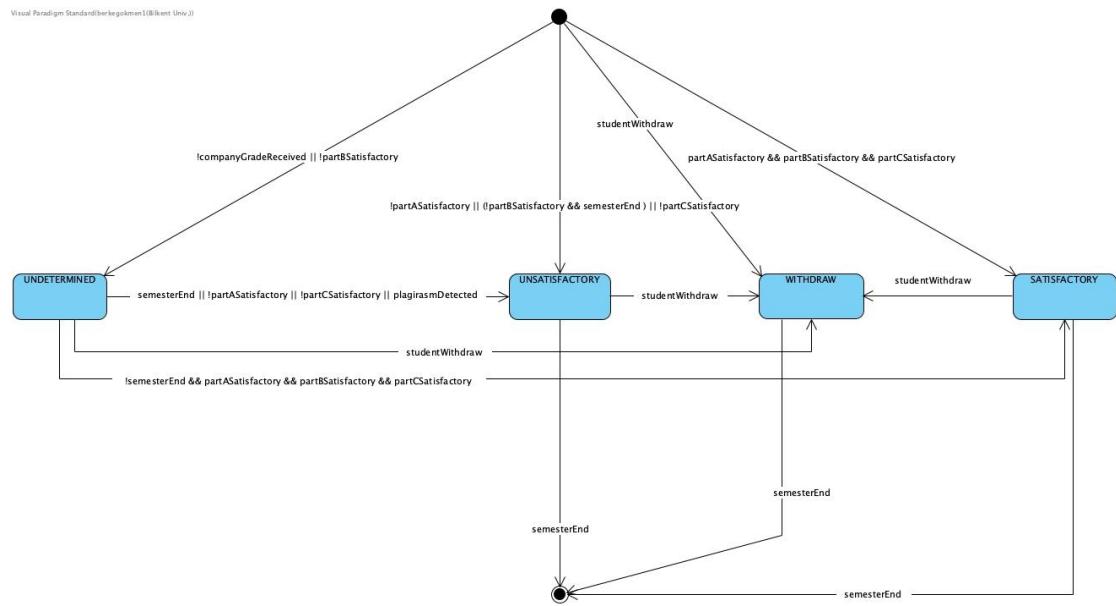


Figure 8: Course Grade State Diagram for the Internship Management System

This diagram explains how the overall grade of a student is determined. All the conditions depend on individual parts of the evaluation being completed. The state diagram of part A can be seen in 3.5.3.2.1, part B and C in 3.5.3.2..

### **3.5.4 User Interface**

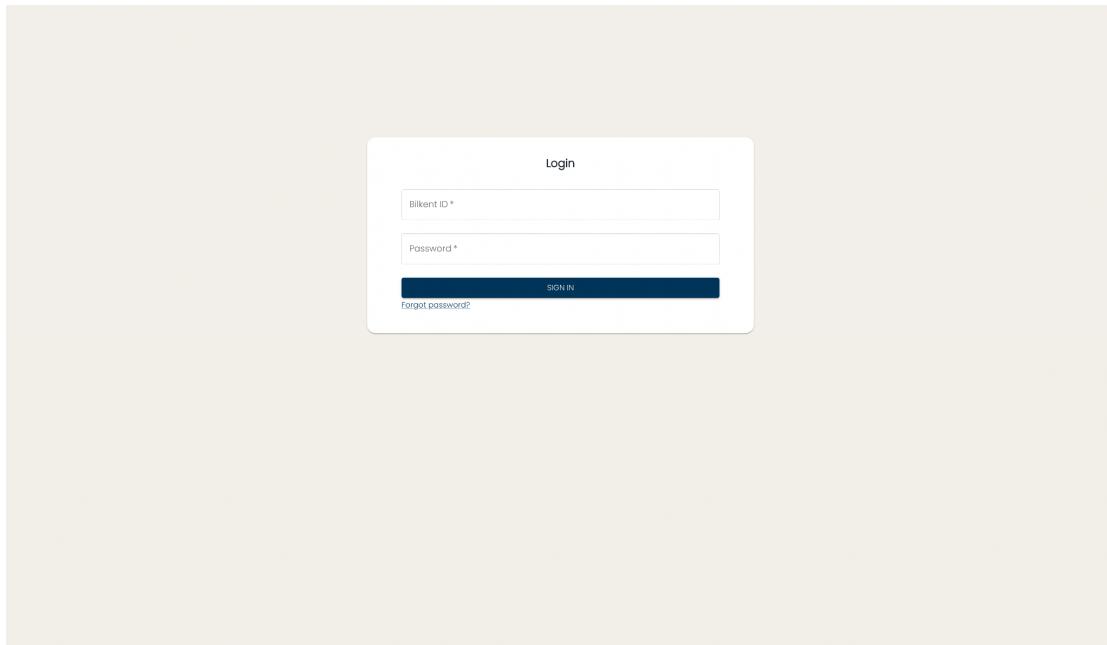


Figure 9: Login Page

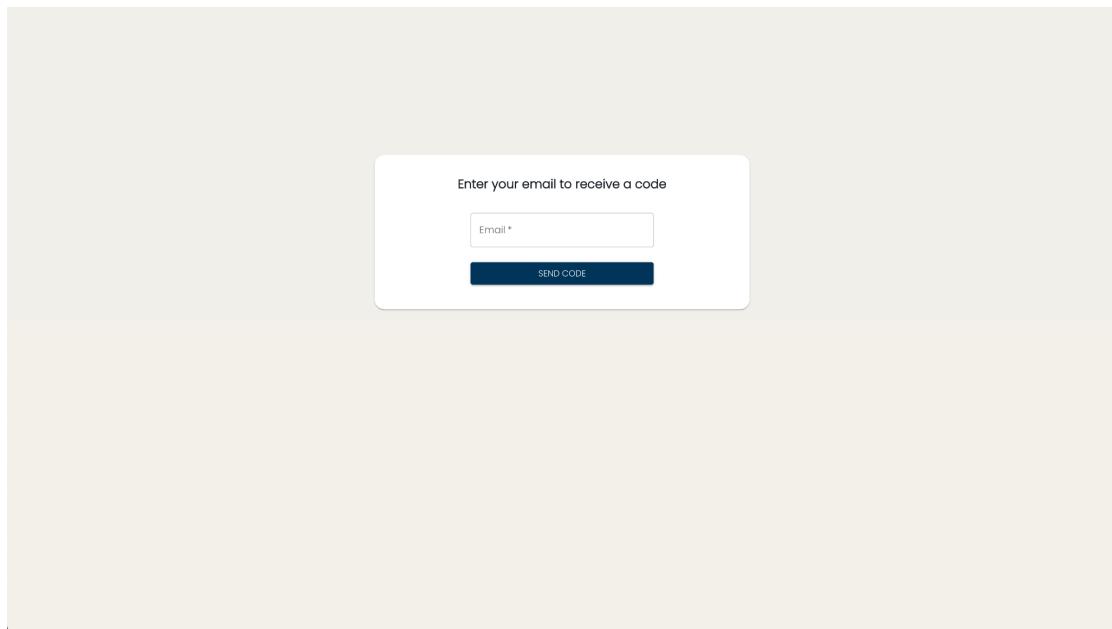


Figure 10: Request Password Reset Email Page

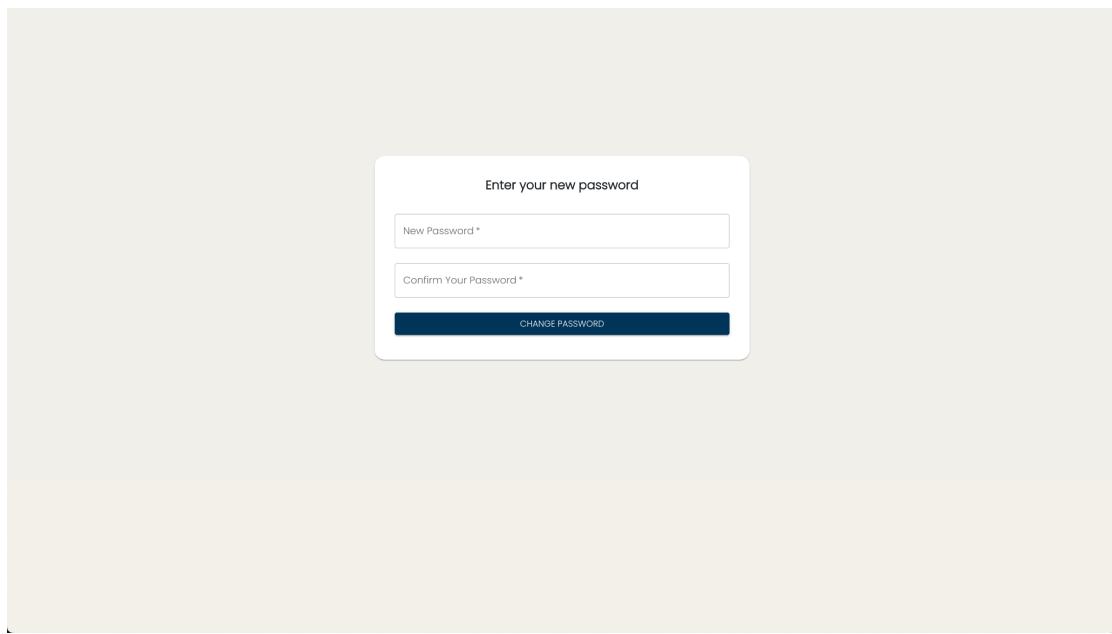


Figure 11: Change Password With Given Link Page

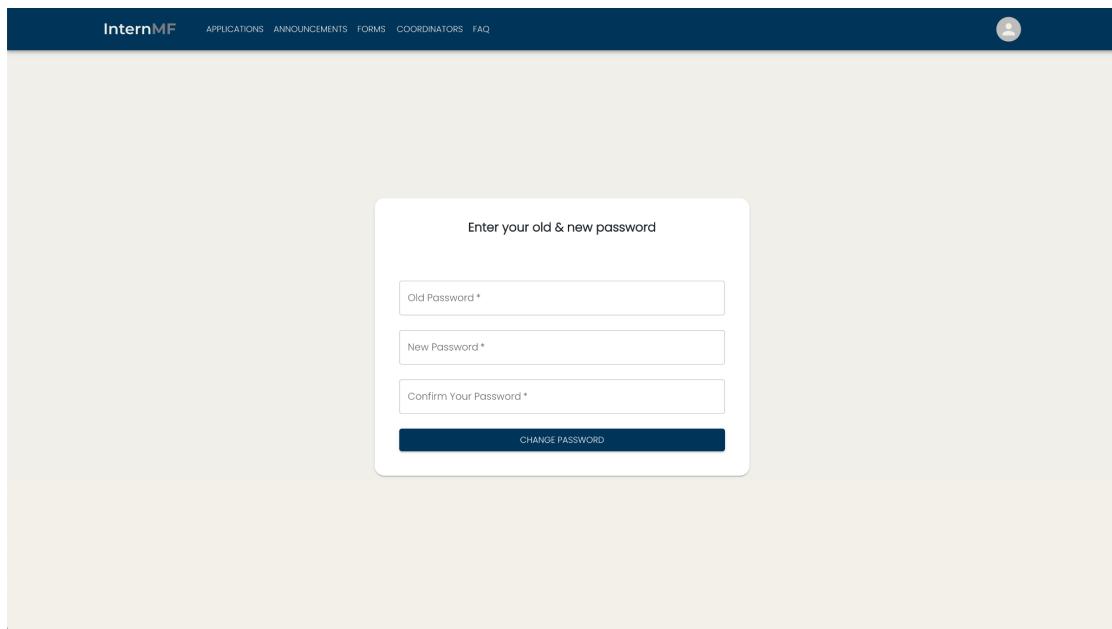


Figure 12: Change Password Using Existing Password Page

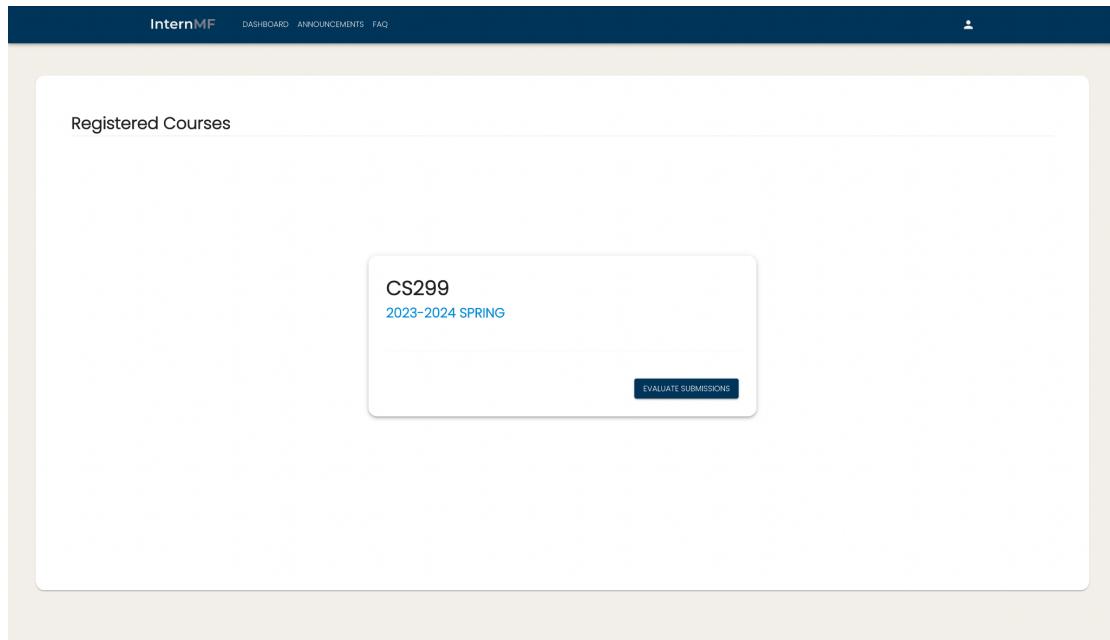


Figure 13: Evaluator Dashboard Page

The screenshot shows the Evaluator View Assigned Submissions page. At the top, there is a dark header bar with the InternMF logo, a user icon, and navigation links for DASHBOARD, ANNOUNCEMENTS, and FAQ. Below the header, a table lists student submissions. The table has columns for Student ID, Student Name, Course, Evaluator, and Status. Each row includes a "SEE EVALUATION" button. The table is currently filtered to show "PASSED TA REVIEW" submissions. There is also a "ALL" link at the top right of the table.

Figure 14: Evaluator View Assigned Submissions Page

| Course | Number of students | Semester         | EXPORT LIST | DEACTIVATE |
|--------|--------------------|------------------|-------------|------------|
| CS299  | 120                | 2023-2024 Fall   |             |            |
| CS399  | 136                | 2023-2024 Fall   |             |            |
| EE299  | 128                | 2023-2024 Fall   |             |            |
| EE399  | 119                | 2023-2024 Fall   |             |            |
| ME299  | 99                 | 2023-2024 Fall   |             |            |
| ME399  | 116                | 2023-2024 Fall   |             |            |
| IE299  | 131                | 2023-2024 Fall   |             |            |
| IE399  | 139                | 2023-2024 Fall   |             |            |
| CS299  | 101                | 2022-2023 Spring |             |            |
| CS399  | 124                | 2022-2023 Spring |             |            |
| EE299  | 111                | 2022-2023 Spring |             |            |
| EE399  | 102                | 2022-2023 Spring |             |            |
| ME299  | 88                 | 2022-2023 Spring |             |            |

Figure 15: Export Submission Page

**Evaluate**

**Student Information**

Name: Student  
Bilkent ID: 22000004  
Email: erdemeren02@gmail.com  
Course: CS299

**Evaluate Student's Internship Report**

DOWNLOAD INTERNSHIP REPORT

Part A

COMPANY EVALUATION REPORT

SATISFACTORY

**Old versions of Student's Internship Report**

| From:       | Date                       | VIEW |
|-------------|----------------------------|------|
| Student     | May 20th 2023, 4:27:12 pm  |      |
| Student     | May 20th 2023, 4:26:32 pm  |      |
| Student     | May 20th 2023, 4:24:15 pm  |      |
| Student     | May 20th 2023, 4:23:42 pm  |      |
| Student     | May 20th 2023, 4:21:14 pm  |      |
| Evaluator 2 | May 20th 2023, 11:59:51 pm |      |
| Evaluator 2 | May 19th 2023, 8:56:29 pm  |      |
| Evaluator 2 | May 19th 2023, 8:36:52 pm  |      |
| Evaluator 2 | May 19th 2023, 8:36:52 pm  |      |
| Evaluator 2 | May 19th 2023, 8:36:52 pm  |      |

Figure 16: Evaluate Submission Page when the evaluation process is finished

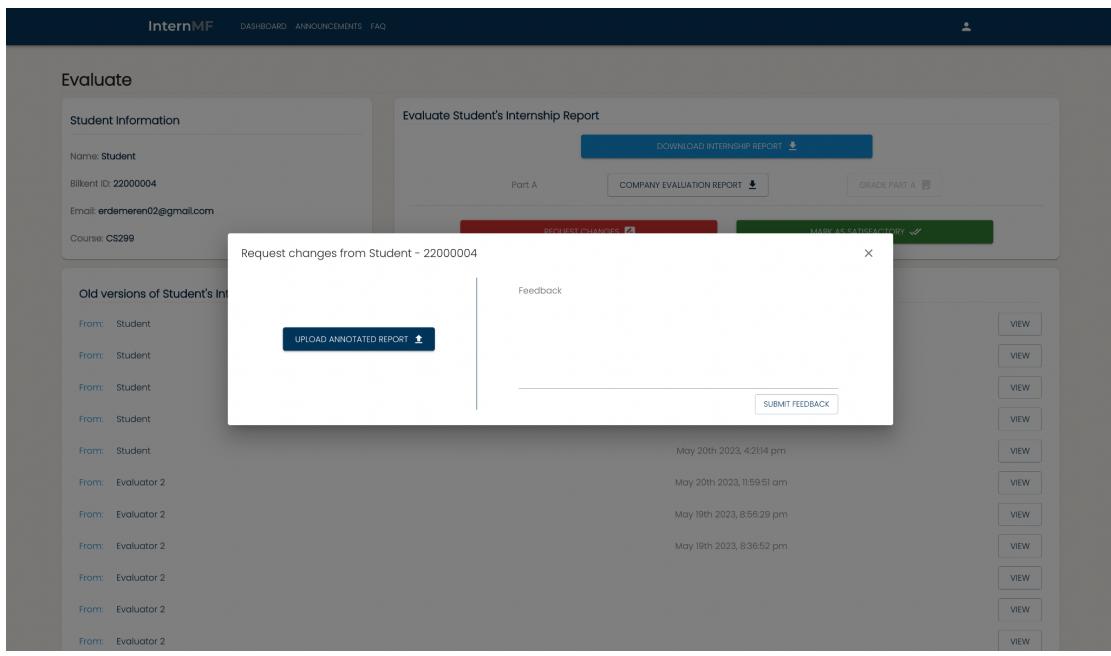


Figure 17: Give Feedback to Report Page for Evaluator

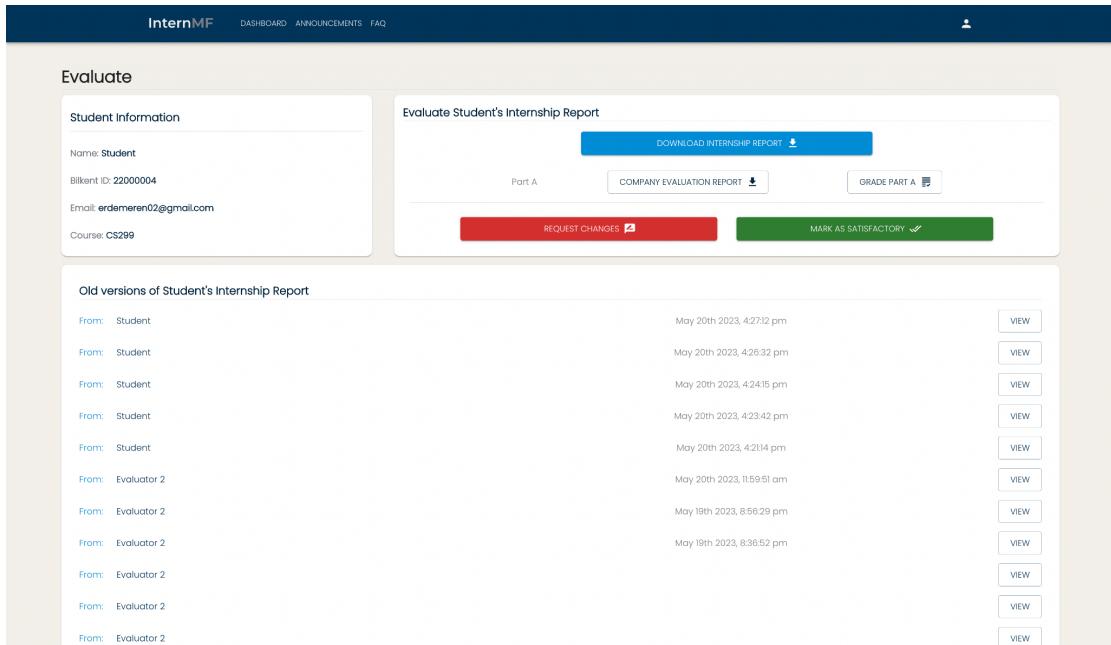


Figure 18: Evaluate Submission Page when the evaluation process is ongoing (Requesting changes & giving feedback)

The screenshot shows the InternMF platform's 'Evaluate' section for a student's internship report. On the left, a sidebar displays 'Student Information' with fields: Name: Student, Bilkent ID: 22000004, Email: erdemeren02@gmail.com, and Course: CS299. The main area is titled 'Evaluate Student's Internship Report' and contains a blue button 'DOWNLOAD INTERNSHIP REPORT'. Below it, a section 'Old versions of Student's Internship Report' lists several attachments from different senders (Student, Evaluator 2) with their respective dates and 'VIEW' buttons. A green message box at the bottom states 'Submission successfully marked as satisfactory!'. On the right, there are buttons for 'GRADE PART A' and 'GRADE PART C', with 'GRADE PART C' being highlighted in green.

Figure 19: Evaluate Submission Page when the evaluation process is ongoing (Grading the submission)

This screenshot shows the 'Grade Part A' dialog box overlaid on the evaluate page. The dialog has a header 'Grade Part A' with a close button. It displays 'Student Name: Student' and 'Bilkent ID: 22000004'. Below this, a section 'Questions' asks two questions with checkboxes: 'Is the work done related to computer engineering? [Y/N]' and 'Is the supervisor a computer engineer or has a similar engineering background? [Y/N]'. A note above the questions says 'Average of the grades on the Summer Training Evaluation Form (Staj Değerlendirme Formu) filled by the employer' and 'Passing grade is 7'. At the bottom of the dialog is a 'SUBMIT' button. The background of the page shows the same 'Evaluate' interface as Figure 19, with the 'GRADE PART A' button visible.

Figure 20: Grade Part A of Submission Page

**Evaluate**

**Student Information**

Name: Student  
Bilkent ID: 2200004  
Email: erdemeren02@gmail.com  
Course: CS299

**Evaluate Student's Internship Report**

**Grade Part C**

Student Name: Student  
Bilkent ID: 2200004

**Questions**

Sum of the Assessment/quality scores of Evaluation of the Work - items (2)-(7)  
(*Passing grade is 30*)

Assessment/quality score of Evaluation of the Work - item (1)  
(*Passing grade is 7*)

The Assessment/quality score of Evaluation of the Report  
(*Passing grade is 7*)

**SUBMIT**

From: Student May 19th 2023, 8:56:29 pm  
From: Student May 19th 2023, 8:56:52 pm  
From: Student  
From: Evaluator 2  
From: Evaluator 2  
From: Evaluator 2  
From: Evaluator 2  
From: Evaluator 2

Figure 21: Grade Part C of Submission Page

**Evaluate**

**Student Information**

Name: Student  
Bilkent ID: 2200004  
Email: erdemeren02@gmail.com  
Course: CS299

**Evaluate Student's Internship Report**

Part A

**GRADE PART C**

See the comments of Student

**Comments**

Note: If this is your first submission, you can send this text as comment. However, on the next submission(s), if your TA or evaluator requested changes, you need to use this section to explain your changes that you made. Thus, please delete this section and add your own comments.

**DOWNLOAD REPORT**

From: Student May 20th 2023, 4:23:42 pm  
From: Student May 20th 2023, 4:21:14 pm  
From: Student May 20th 2023, 11:59:51 pm  
From: Evaluator 2 May 19th 2023, 8:56:29 pm  
From: Evaluator 2 May 19th 2023, 8:56:52 pm  
From: Evaluator 2  
From: Evaluator 2  
From: Evaluator 2  
From: Evaluator 2

Figure 22: See Previous Feedback About The Report

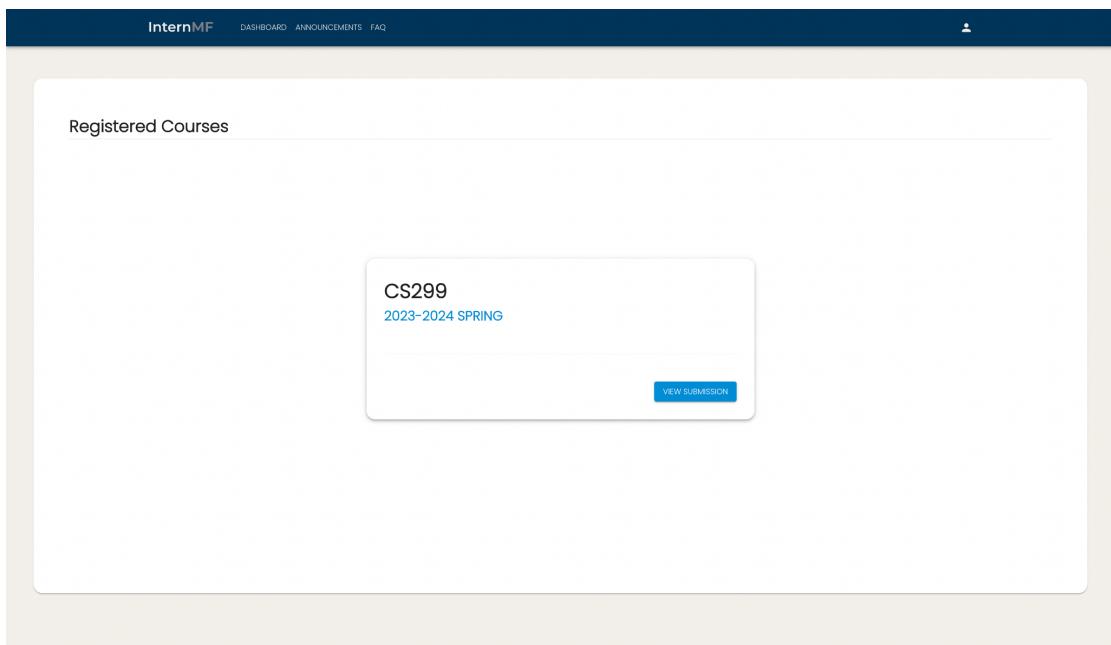


Figure 23: Student Dashboard

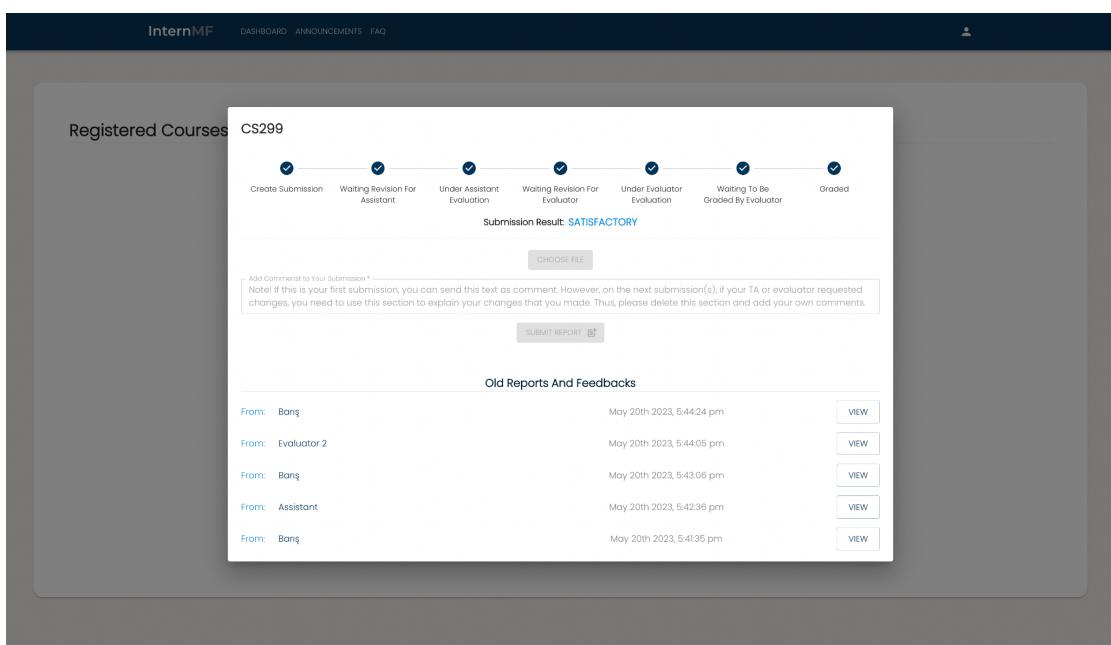


Figure 24: Submission Detail for Student

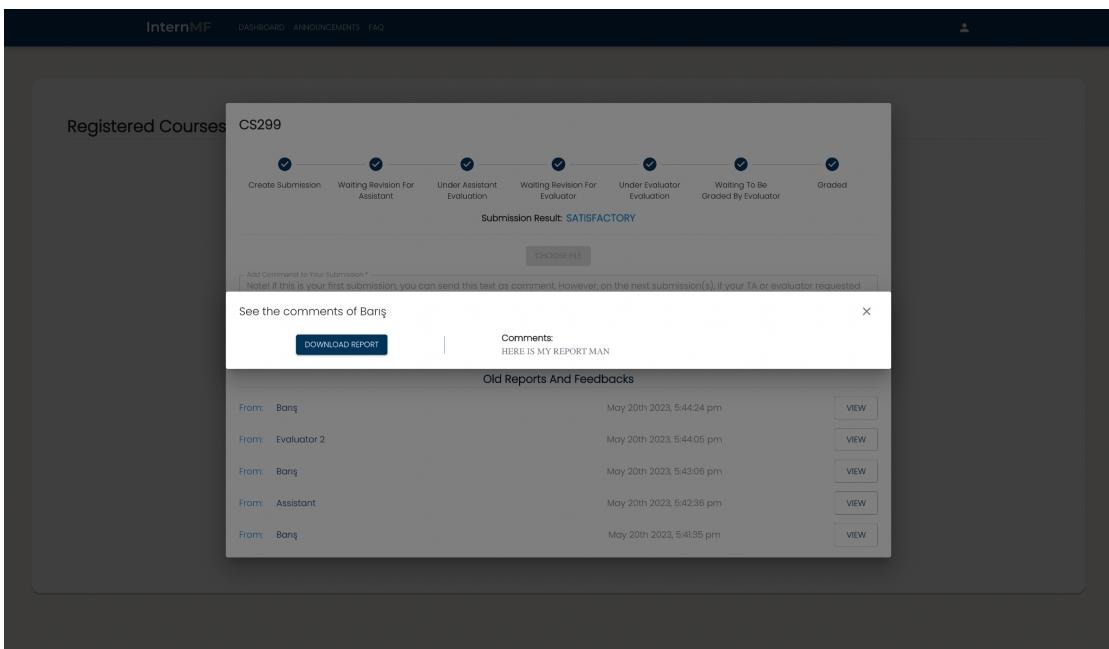


Figure 25: Student Seeing Previous Feedback

The screenshot shows the InternMF assistant view assigned submissions page. At the top, there's a navigation bar with links for DASHBOARD, ANNOUNCEMENTS, and FAQ. Below the navigation, a section titled "Submissions to be evaluated" displays a table of assignments. The table has columns for "Student ID", "Student Name", "Course", "Evaluator", and "Status". There are three rows:

| Student ID | Student Name | Course | Evaluator    | Status                       |
|------------|--------------|--------|--------------|------------------------------|
| 22000004   | Student      | CS299  | Evaluator 2  | WAITING_ORADING_BY_EVALUATOR |
| 22000007   | Student 3    | CS299  | Not assigned | UNDER_EVALUATOR_EVALUATION   |
| 2190234    | Bang         | CS299  | Evaluator 2  | SATISFACTORY                 |

Each row has an "EVALUATE" button with a file icon and a "SEE EVALUATION" button with a file icon. A small inset image in the bottom right corner shows a mobile device displaying the InternMF interface.

Figure 26: Assistant View Assigned Submissions Page

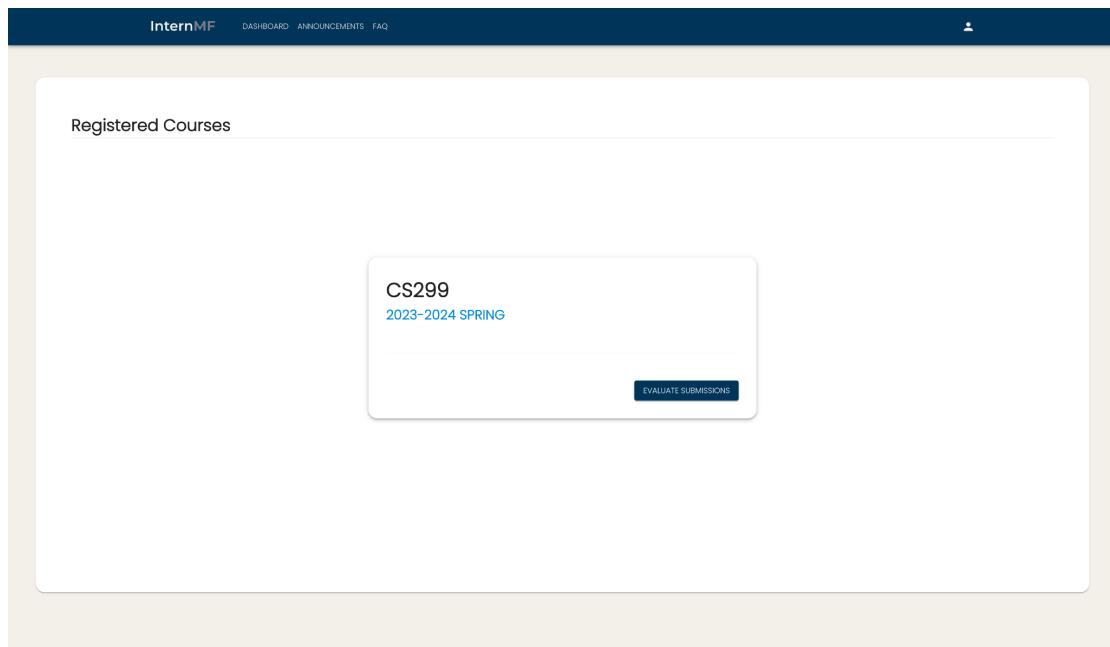


Figure 27: Assistant Dashboard

The screenshot shows the InternMF Evaluate Student Submission for Assistant interface. It has a dark header bar with the InternMF logo, a navigation menu (Dashboard, Announcements, FAQ), and a user profile icon. The main content area is divided into two sections: "Evaluate" on the left and "Pre-Evaluate Student's Internship Report" on the right. The "Evaluate" section contains a "Student Information" form with fields for Name (Student), Bilkent ID (22000004), Email (erdemeren02@gmail.com), and Course (CS299). The "Pre-Evaluate Student's Internship Report" section includes a "DOWNLOAD INTERNSHIP REPORT" button and a message stating "This submission has passed teaching assistant review stage!". Below these sections is a table titled "Old versions of Student's Internship Report" showing multiple email attachments from various evaluators. Each attachment has a "VIEW" button next to it.

Figure 28: Evaluate Student Submission for Assistant

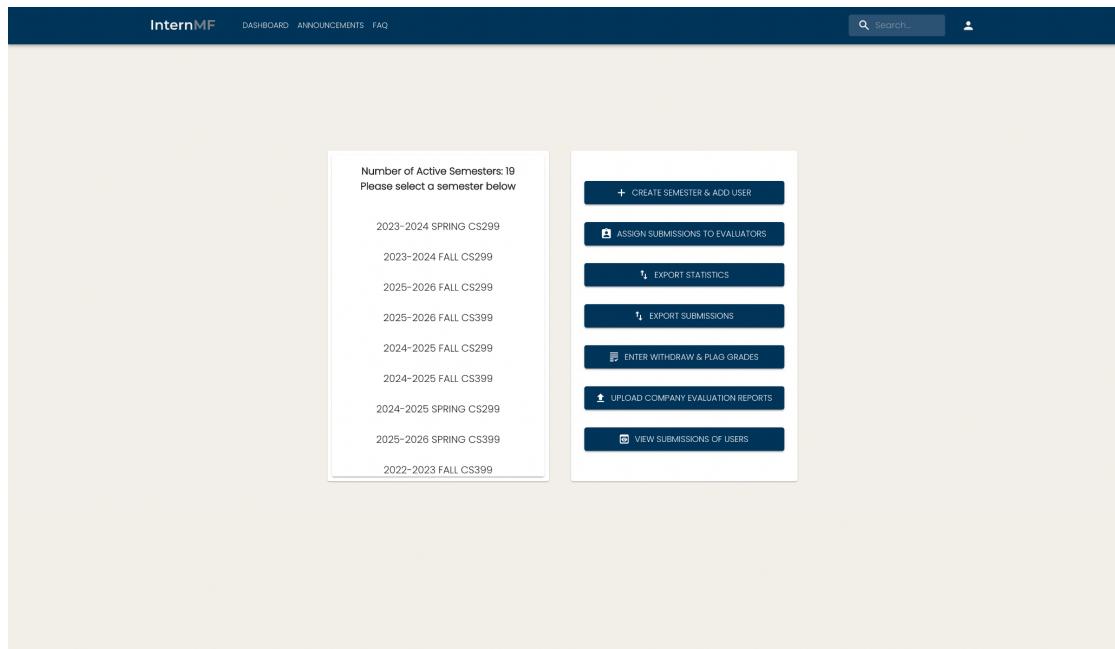


Figure 29: Secretary Dashboard

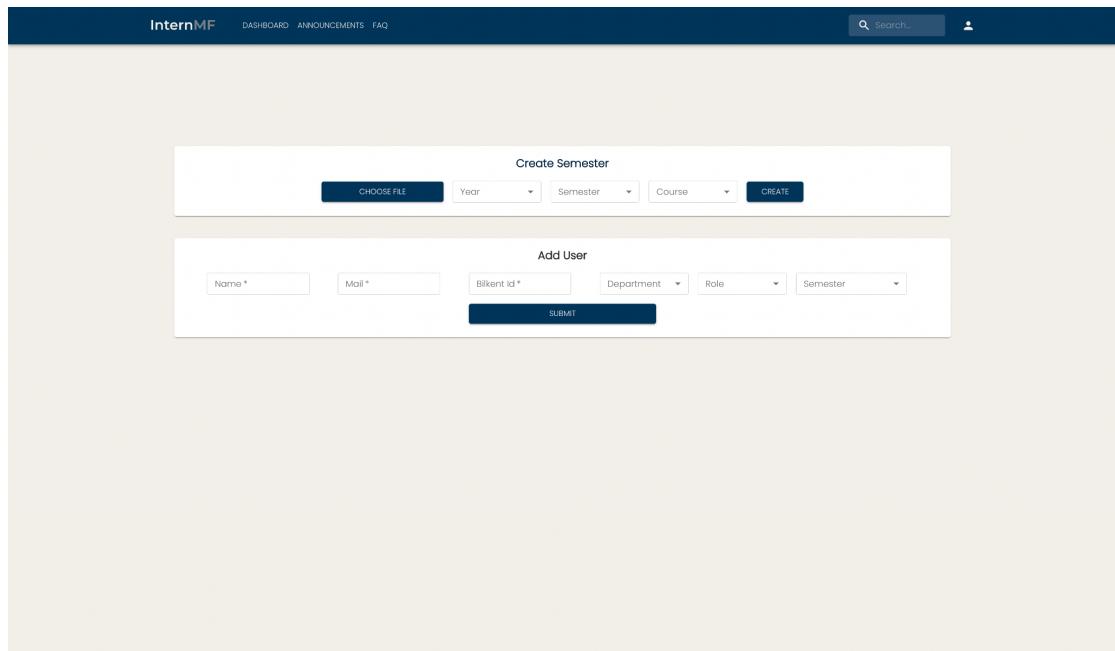
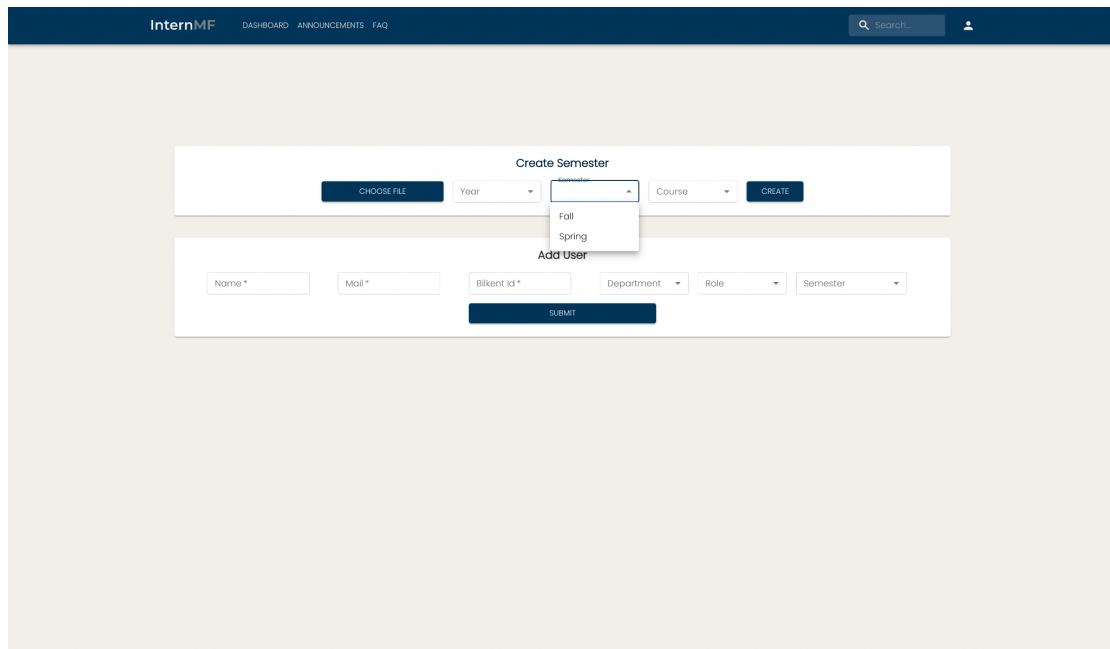
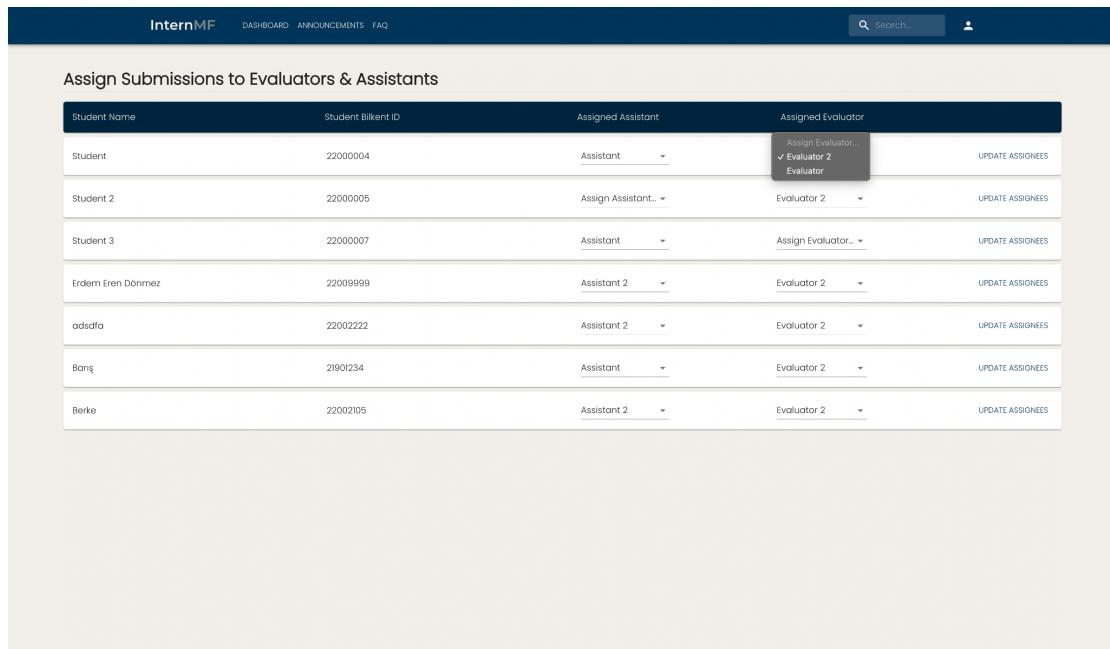


Figure 30: Secretary Initialize Semester Page



The screenshot shows the 'Create Semester' interface. At the top, there are fields for 'CHOOSE FILE', 'Year' (set to '2022'), 'Course' (dropdown menu open), and a 'CREATE' button. Below this, a dropdown menu for 'Semester' is open, showing 'Fall' and 'Spring'. The main form area is titled 'Add User' and contains fields for 'Name \*', 'Mail \*', 'Bilkent ID \*', 'Department', 'Role', and 'Semester'. A 'SUBMIT' button is at the bottom.

Figure 31: Secretary Initialize Semester Page



The screenshot shows the 'Assign Submissions to Evaluators & Assistants' page. The table has columns for 'Student Name', 'Student Bilkent ID', 'Assigned Assistant', 'Assigned Evaluator', and actions. The 'Assigned Evaluator' column includes a dropdown and a 'UPDATE ASSESSORES' button. A modal window titled 'Assign Evaluator...' is open, showing a dropdown with 'Evaluator 2' selected and a checked checkbox for 'Evaluator 2'.

| Student Name      | Student Bilkent ID | Assigned Assistant  | Assigned Evaluator                                |                   |
|-------------------|--------------------|---------------------|---|-------------------|
| Student           | 22000004           | Assistant           | Assign Evaluator...<br>✓ Evaluator 2<br>Evaluator | UPDATE ASSESSORES |
| Student 2         | 22000005           | Assign Assistant... | Evaluator 2                                       | UPDATE ASSESSORES |
| Student 3         | 22000007           | Assistant           | Assign Evaluator...                               | UPDATE ASSESSORES |
| Erdem Eren Dönmez | 22009999           | Assistant 2         | Evaluator 2                                       | UPDATE ASSESSORES |
| adsdfa            | 22002222           | Assistant 2         | Evaluator 2                                       | UPDATE ASSESSORES |
| Bang              | 21901234           | Assistant           | Evaluator 2                                       | UPDATE ASSESSORES |
| Berke             | 22002105           | Assistant 2         | Evaluator 2                                       | UPDATE ASSESSORES |

Figure 32: Secretary Assign Submissions Page

**Mark Student as Withdrawn or Plagiarised**

Student ID \*  SEARCH STUDENT SUBMISSION

|   |  |  |
|---|--|--|
| Semester: 2023-2024 FALL<br>Course: CS299<br>Student Name: Berké<br>Bilkent ID: 22002105<br>Submission Status: WITHDRAW             | <input type="button" value="WITHDRAWN"/> | <input type="button" value="PLAGIARISED"/> |
| Semester: 2023-2024 SPRING<br>Course: CS299<br>Student Name: Berké<br>Bilkent ID: 22002105<br>Submission Status: WAITING_SUBMISSION | <input type="button" value="WITHDRAWN"/> | <input type="button" value="PLAGIARISED"/> |
| Semester: 2023-2023 FALL<br>Course: CS299<br>Student Name: Berké<br>Bilkent ID: 22002105<br>Submission Status: PLAGIARISED          | <input type="button" value="WITHDRAWN"/> | <input type="button" value="PLAGIARISED"/> |



Figure 33: Mark Student Submission as Withdrawn or Plagiarised Page for Secretary

**Upload Company Report**

Student ID \*  SEARCH STUDENT SUBMISSIONS

|   |  |  |
|---|--|--|
| Semester: 2023-2028 SPRING<br>Course: CS299<br>Student Name: Berké<br>Bilkent ID: 22002105<br>Submission Status: UNDER_ASSISTANT_EVALUATION | <input type="button" value="DOWNLOAD REPORT"/>             | <input type="button" value="DELETE REPORT"/> |
| Semester: 2023-2024 FALL<br>Course: CS299<br>Student Name: Berké<br>Bilkent ID: 22002105<br>Submission Status: WITHDRAW                     | <input type="button" value="CHOOSE FILE"/>                 | <input type="button" value="UPLOAD"/>        |
| Semester: 2023-2024 SPRING<br>Course: CS299   | <div style="border: 1px solid #ccc; height: 100px;"></div> |  |

Figure 34: Company Report Upload Page For The Secretary

Admin Creation Panel

Name \*

Bilkent Mail \*

Bilkent ID \*

Department \*

Electric Electronic Engine...

**CREATE ADMIN**

Figure 35: Department Chair's Admin Account Creation Page

Search / View Student Submissions

Student ID \* 22002105

SEARCH STUDENT SUBMISSIONS

Semester: 2027-2028 SPRING  
Course: CS299  
Student Name: Berke  
Bilkent ID: 22002105  
Submission Status: UNDER\_ASSISTANT\_EVALUATION

Semester: 2023-2024 FALL  
Course: CS299  
Student Name: Berke  
Bilkent ID: 22002105  
Submission Status: WITHDRAW

Semester: 2023-2024 SPRING  
Course: CS299

Figure 36: Search Student Submission With Bilkent ID Page for Secretary & Department Chair

## **4 Improvement Summary**

### **General**

- Improved the grammar of the report.

### **3 Proposed System**

#### **3.2 Actors**

- Explained Secretary and Department Chair actors separately.

#### **3.3 Non-functional Requirements**

- Rewrote this section in order to provide a clearer overview of the non functional requirements and how they can be tested.

### **3.5 System Models**

#### **3.5.1 Use-Case Model**

- Revised the typo mistakes use case diagram
- Added new use cases called CreateAdminAccount, UserAlreadyExists, ExportSemesterStatistics, ViewAllUsers, ViewAllSubmissions.
- Use Case diagrams structure redesigned for improved readability.

#### **3.5.2 Object and Class Model**

- Removed internship class, removed the internship supervisor information.
- Removed GradeForm class and Part classes, separated the questions into FormQuestion class.
- Added evaluatorId and assistantId to submission class.
- Re-organized the class diagram.
  - Used child classes to show different types of users.
  - Added department information to the semester and course classes.
  - Added deadline information for semester and submission.
  - Revised the relationship between user and submission classes.
  - Revised FAQ and announcement classes, added department information.

#### **3.5.4 User Interface**

- Revised & updated the application's pages.

## 5 References

- [1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.
- [2] W. by: E. Paraschiv, "Password encoding with Spring," *Baeldung*, 26-Nov-2022. [Online]. Available: <https://www.baeldung.com/spring-security-registration-password-encoding-bcrypt>. [Accessed: 04-May-2023].