
Software Requirements Specification and Design Description

for

A Software for Summer Internship

Umutcan Celik 2526200

Mert Damburacı 2453108

Ali Fırat Ozdemir 2453447

Efekan Uysal 2585461

Middle East Technical University Northern Cyprus Campus

Computer Engineering

İdil Candan

26.10.2024

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope.....	2
1.3	Related Work	3
1.4	Product Overview	4
1.4.1	Product perspective	4
1.4.2	Product functions.....	9
1.4.3	Identified stakeholders and design concerns	10
1.4.4	User characteristics	10
1.4.5	Limitations.....	11
1.4.6	Assumptions and dependencies	12
2	Specific requirements.....	13
2.1	External interfaces.....	13
2.2	Functions.....	13
2.2.1	Use Case Diagram	13
2.2.2	Use-Case Description	14
2.3	Usability Requirements	19
2.4	Performance requirements	20
2.5	Logical database requirements	23
2.5.1	Relational Database Diagram	23
2.5.2	Relational Database Diagram Assumptions.....	24
2.6	Software system attributes.....	25
2.7	Supporting information	25
3	Software Estimation	26
4	Architectural Views	29
4.1	Logical View.....	29
4.1.1	Class Diagram	29
4.1.2	Class Diagram Assumptions	30
4.2	Process View.....	31
4.2.1	Activity Diagram	31
4.2.2	Sequence Diagrams	39
4.2.3	Data Flow Diagrams	44
4.3	Development View	57
4.3.1	Component and Deployment Diagram.....	57
5	Project Scheduling	58
5.1	Milestones and Tasks	58
5.2	Gantt Chart.....	60
6	Conclusion	61
7	References	62
8	Appendices	63
8.1	Acronyms and abbreviations.....	63
8.2	Glossary.....	63

1 Introduction

1.1 Purpose

Our project has five key objectives:

First, it enables students to easily find internship opportunities in their areas of interest. Instead of manually searching for internships or waiting for connections, the system provides a matching mechanism that aligns specific student profiles with relevant opportunities based on their skills, preferences, and qualifications. This feature allows students to access a wide range of suitable opportunities from different companies through the system, ensuring they can quickly identify and apply to internships of interest. In addition, the system allows students to apply directly to companies through the platform. This eliminates the need for email-based applications or visiting different websites, simplifying the application process for students. Moreover, the platform provides students the opportunity to connect with individuals who have previously completed internships at the same companies. This allows them to gather insights about the internship experience, the company environment, and expectations, helping them make more informed decisions about where to apply. Additionally, students can enter details about new internships they are undertaking, serving as references for other students who may be interested in similar opportunities. This feature encourages collaboration and shared knowledge within the student community. As a result, this project offers students the opportunity to find internships easily, apply directly through the platform, connect with peers to gain valuable insights, and contribute as references for future internships. At the same time, companies can manage their internship programs in one centralized location, making the process more efficient and accessible for both parties.

1.2 Scope

Goal of our project is to enhance the existing internship system at METU NCC for computer engineering students. We aim to improve the current platform by adding targeted features that streamline the internship experience for both the Compulsory Non-Credit course (CNG 300/400) and optional internships. Our system focuses on refining user-friendliness for students, companies, student affairs, coordinators, and instructors.

Purpose:

Our main aim is to improve the current system by simplifying the internship process. By integrating features like registration, tracking, evaluations, and notifications, we will address the challenges users face. This enhancement ensures a more straightforward, transparent, and efficient way of managing internships.

Objectives:

- **Convenience:** Simplify the user experience when filling out registration forms and starting internships. Our system will help companies and coordinators save time on approving and assessing internships.
- **Notifications and Alerts:** Automate email notifications for students, coordinators, instructors, student affairs, and companies at various stages of the internship process. This includes reminders for form submissions, approval notifications, and updates during the evaluation phase.
- **Report Evaluation and Updates:** Enable students to upload reports for evaluations while instructors can review and provide feedback without requiring physical meetings.
- **Statistics and Monitoring:** Provide coordinators with tools to monitor internship registrations, approvals, and evaluations effectively.

- **Internship Cancellation and Other Functions:** Allow students to cancel internships before starting, ensuring flexibility and better class alignment.

Overall, the enhanced system will offer improved functionality as a web application, accessible across computers, tablets, and smartphones, ensuring ease of use for all stakeholders.

1.3 Related Work

Internship management systems are essential tools for streamlining the process of managing internships for both students and educational institutions. They enable the coordination of students, faculty, and external company supervisors, ensuring that all necessary steps are completed efficiently. Existing systems, such as METU NCC's current internship management system, allow students to fill out internship forms, submit reports, and receive feedback. These systems also provide coordinators with the ability to approve internship placements and monitor student progress (Budić et al., 2021). You can see the existing system example in Figure 1.

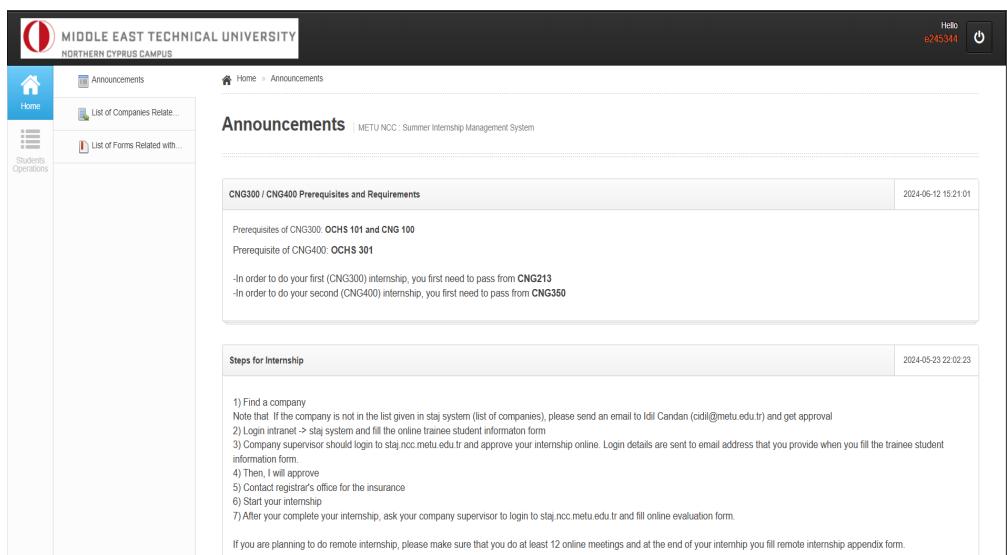


Figure 1: Example of existing system

In our case, we have extended the existing functionality of the internship management system by introducing new features and improving user interactions. For example:

Students can now browse internship opportunities, check companies based on state or country, submit resumes, fill trainee information forms, and apply for internships. They can also manage their internship history, receive notifications, and submit reports.

Coordinator have enhanced role in managing announcements, assigning instructors, setting deadlines, and evaluating reports.

Company supervisors can assign supervisors, evaluate students, and give feedback directly through the system.

Student Affairs personnel handle tasks related to approving internships, verifying health insurance, and ensuring the administrative steps are completed

In contrast to traditional systems (Ismaili, 2018), which typically rely on manual follow-ups and offer minimal real-time interaction, our system streamlines essential processes by automating tasks and incorporating notification services (e.g., email alerts). Furthermore, it provides advanced features such as evaluation feedback and the capability to revise reports directly within the platform, fostering continuous improvement and efficient communication.

1.4 Product Overview

1.4.1 Product perspective

The Summer Internship System is designed as a standalone platform that integrates with external service, such as Google Mail, to enhance its functionality. It serves as a centralized interface for students, coordinators, and companies involved in the internship process.

For Students: The system provides tools to browse internship opportunities, submit applications, and track application statuses seamlessly.

For Coordinators: Coordinators can manage announcements, approve or reject applications, and oversee the internship workflows efficiently.

For Companies: Companies can post internship positions and evaluate trainee forms through the system.

While the platform operates independently, it interacts with external mail service, such as Google Mail, to provide automated email notifications. These notifications ensure timely updates regarding applications, approvals, rejections, and form submissions, enhancing communication for all users. The system architecture and interaction with the external mail service are visually depicted in Figure 2.

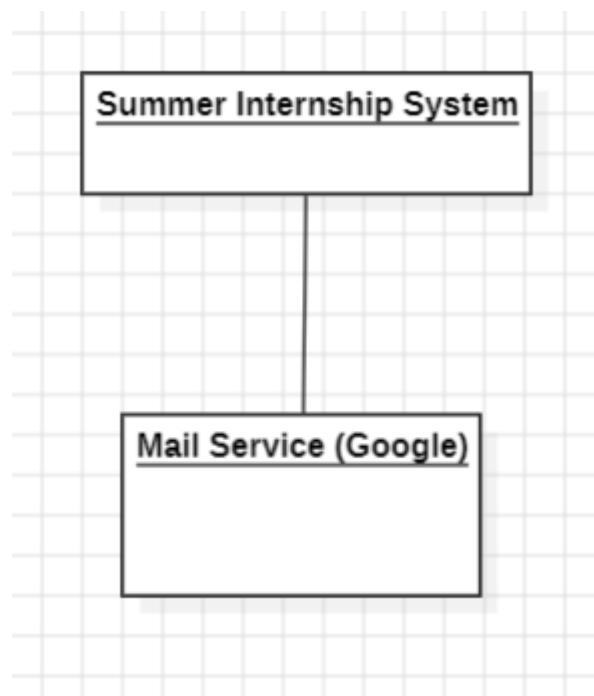


Figure 2: Product Perspective Block Diagram

1.4.1.1 System Interfaces

1. Google Interface

Summer Internship System interfaces with the external Google Mail Service to manage email notifications. This interface enables the system to send automated emails, such as reminders, application confirmations, and important updates related to internships. By using Google's mail service, the system ensures secure and reliable email delivery, which is essential for maintaining effective communication with users and enhancing their overall experience within the platform.

1.4.1.2 User interfaces

1. Welcome Page:

Welcome Page is the homepage of the Internship System, offering students easy access to key sections such as Announcements, My Internships, Forms, and Notifications. The dashboard provides a clear layout for students to quickly navigate to their desired section, creating a cohesive and user-friendly entry point to the internship management system. You can see the welcome page in Figure 3

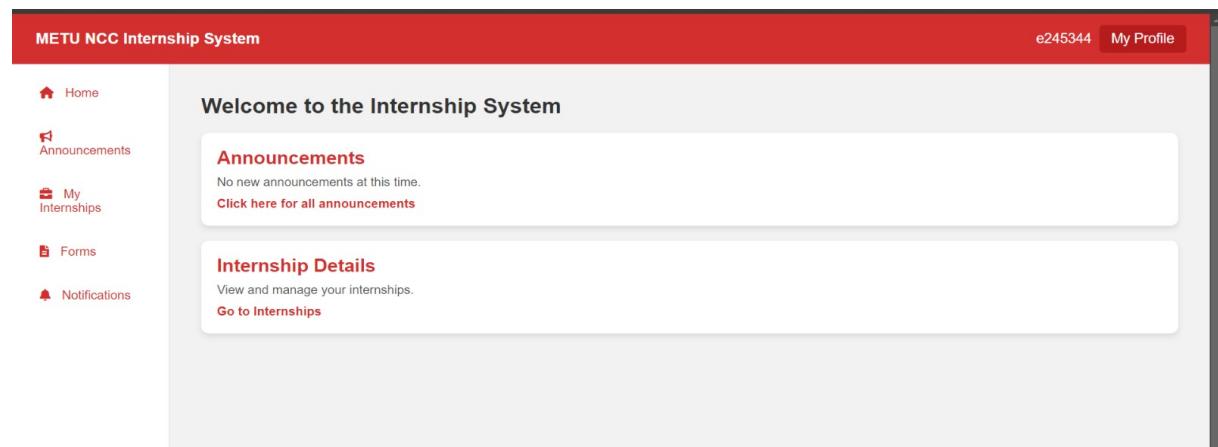


Figure 3: Welcome Page

2. My Internships Page:

My Internships Interface displays a summary of the student's completed internships, along with overall statistics such as total internships and average approval time. Each internship entry includes details like the company name, year, supervisor, and completion status. This layout provides students with quick access to download their reports and view the outcomes, facilitating efficient internship tracking. You can see from Figure 4.

The screenshot shows the 'My Internships' page of the METU NCC Internship System. At the top, there is a red header bar with the text 'METU NCC Internship System' on the left and 'e245344 My Profile' on the right. Below the header, the main content area has a white background. It features a title 'My Internships' at the top center. Underneath it is a box titled 'Internship Statistics' containing the following data:
Total Internships: 2
Average Approval Time: 10 days
Completed Internships: 2

Below the statistics, there are two separate boxes, each representing a completed internship:
Company: XYZ Corporation
Year: 2023
Course: CNG400
Supervisor: John Doe
Status: Completed
Report Status: Approved
Outcome: Successful
A green button indicates '100% Completed'. A red 'Download Report' button is below it.
Company: ABC Ltd.
Year: 2022
Course: CNG300
Supervisor: Jane Smith
Status: Completed
Report Status: Approved
Outcome: Needs Improvement
A green button indicates '100% Completed'. A red 'Download Report' button is below it.
At the bottom of the page, a message states: 'Report Status: Your report for CNG400 has been approved!'

Figure 4 : My Internships Page

3. Former Internships Page:

Former Internships Interface provides students with an overview of previous internship placements. Each entry displays information like the company name, country, contact email, and rating. Action buttons allow students to apply to the company, add it to favourites, or view the company profile, ensuring that students can make informed decisions based on their interests and past internship experience. You can see from Figure 5.

The screenshot shows the 'Former Internships' page of the METU NCC Internship System. At the top, there is a red header bar with the text 'METU NCC Internship System' on the left and 'e245344 My Profile' on the right. Below the header, the main content area has a white background. It features a title 'Former Internships' at the top center. There is a dropdown menu labeled 'Filter by Country:' with 'All Countries' selected. Below the title, there is a table with the following columns: Student Name, Company Name, Country, State, Address, Email, Rating, and Actions. The table contains three rows of data:
Ali Özdemir XYZ Corporation Turkey Istanbul Levent Mah. Plaza Sok. No: 12, Istanbul ali.ozdemir@xyzcorp.com ★★★★☆ Apply ▾ Favorite Company Profile Application Status: Pending
Ayşe Demir ABC Technologies Turkey Ankara Atatürk Bulvarı, No: 45, Ankara ayse.demir@abctech.com ★★★☆☆ Apply ▾ Favorite Company Profile Application Status: Not Submitted
John Smith DEF Group USA California Silicon Valley, CA john.smith@defgroup.com ★★★★★ Apply ▾ Favorite Company Profile Application Status: Submitted

Figure 5: Former Internships Page

4. Resume Upload Page:

Resume Upload Interface enables students to add their resume to their profile in PDF format. Users can drag and drop their file or browse to select it, making the process flexible and user-friendly. This interface is streamlined for easy resume submission, ensuring that students can upload their documents efficiently. You can see from Figure 6.

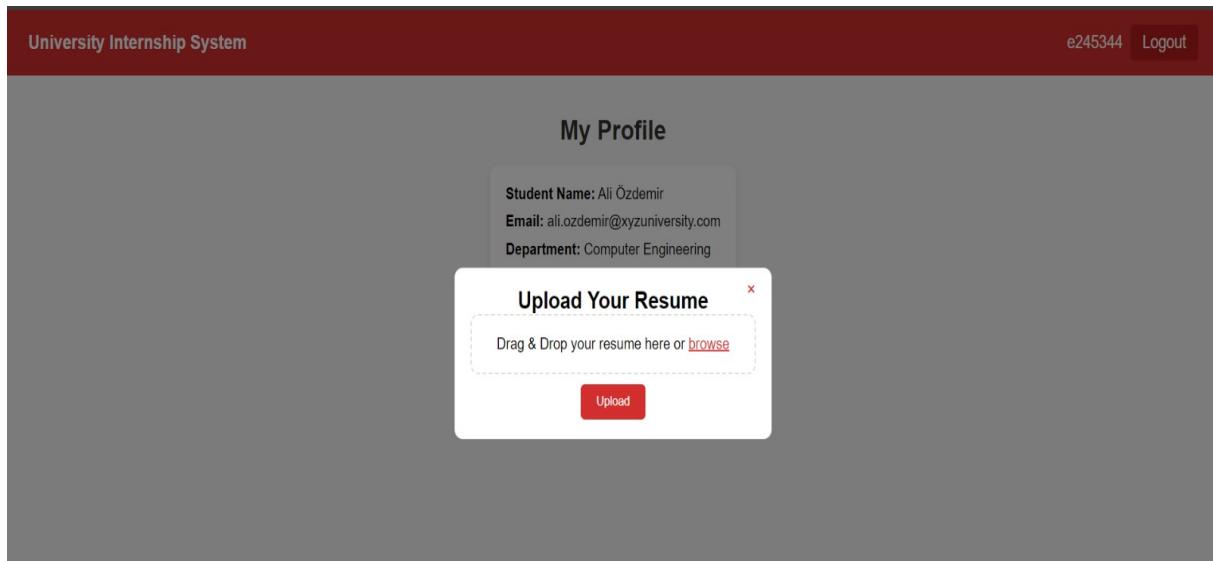


Figure 6: Resume Upload Page

5. List of Forms Page:

Forms Interface in the Internship System allows students to access, filter, and download relevant documents easily. Users can search for forms by title or filter them by category, making navigation straightforward. Additionally, each form entry includes options to download, add to favourites, and view details, enhancing the user experience by providing direct access to essential resources. You can see from Figure 7.

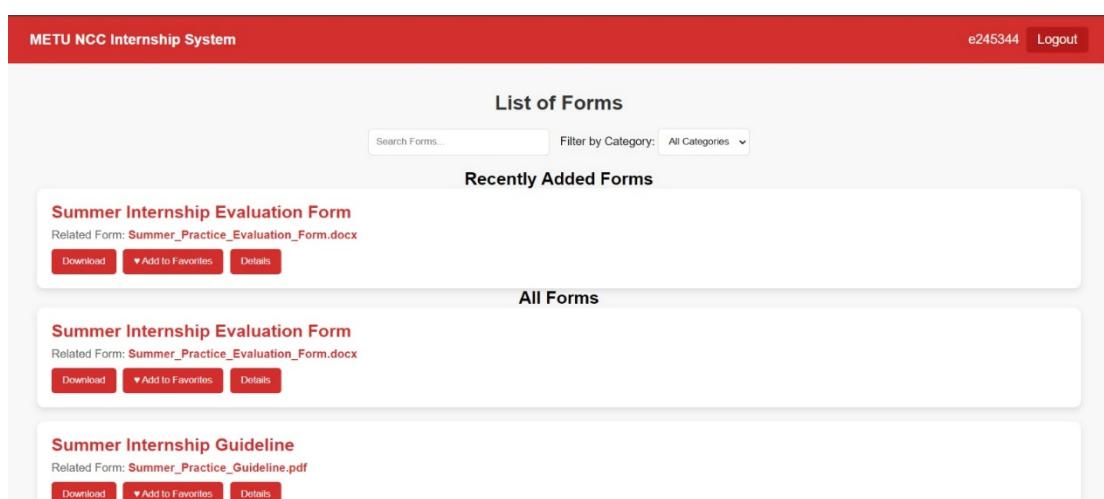


Figure 7: List of Forms Page

6. Company Page

Company interface is the page companies see when they login to the system. Here the company representative can see the details of the students who are on internship in their organization and can approve or reject their internship. In addition, if they want to check or update the supervisor information of the approved users later, they can use the Supervisor button. They can also evaluate the students accepted for internship with the Evaluate button. You can see from Figure 8.

The screenshot shows the 'Internships' section of the METU NCC Internship System. At the top, there is a red header bar with the text 'METU NCC Internship System' on the left and 'Company A' with a dropdown arrow on the right. Below the header is a search bar labeled 'Search...'. The main content area has a title 'Internships' and a table with the following data:

Student Name	Email	Start Date	Finish Date	Health Insurance	Action
Ali Fırat Özdemir	2453447@metu.edu.tr	01/08/2024	30/08/2024	No	<button>Accept</button> <button>Reject</button>
Mert Damburacı	2453108@metu.edu.tr	01/06/2024	30/06/2024	Yes	<button>Supervisor</button> <button>Evaluate</button>
Efekan Uysal	2585461@metu.edu.tr	01/07/2023	03/08/2024	Yes	<button>Supervisor</button> <button>Evaluate</button>
Umutcan Çelik	2526200@metu.edu.tr	30/06/2024	28/07/2024	Yes	<button>Supervisor</button> <button>Evaluate</button>

Below the table are navigation arrows: «, 1, 2, 3, ». At the bottom of the page, there is a footer with the text 'METU NCC Internship System', 'Copyright © 2024. All Rights Reserved.', and 'If you experience any problems, please contact us at example@edu.tr'.

Figure 8: Company Page

7. Intern Acceptance Form Window

If companies click on the Accept button to accept a student's internship, a form will appear where some important information about the internship, including Supervisor information, must be entered to accept the intern. You can see from Figure 9.

The screenshot shows the 'Trainee Supervisor Details for Ali Fırat Özdemir' form. At the top, there is a red header bar with the text 'METU NCC Internship System' on the left and 'Company A' with a dropdown arrow on the right. Below the header is a search bar labeled 'Search...'. The main content area has a title 'Trainee Supervisor Details for Ali Fırat Özdemir' and several input fields:

Trainee Supervisor's Name	Action
<input type="text"/>	<button>Accept</button> <button>Reject</button>
Trainee Supervisor's Surname	
Trainee Supervisor's E-mail	
Trainee Supervisor's Telephone	
Trainee Supervisor's Fax	

Below the table are navigation arrows: «, 1, 2, 3, ». At the bottom of the form, there are two buttons: 'Accept' and 'Cancel'.

Figure 9: Intern Acceptance Form

8. Student Affairs Page

Student Affairs Interface is the interface that authorized personnel in student affairs see. Here, staff can see students' details, the company they will be doing their internship with, and the approvals they have received for their internship. Staff can select the internships they want from the checkboxes on the left and create a report using the Generate Excel Report button. Additionally, they can mark the students whose health report has been approved as approved with the Approve button. You can see from Figure 10.

■	Student No	Name	Surname	Course	ID no	Department	Company	Company Status	Coordinator Status	Health Insurance Undertaking
<input checked="" type="checkbox"/>	2453447	Ali Fırat	Özdemir	CNG300	12345678901	Computer Engineering	A Soft	APPROVED	APPROVED	<button>Approve</button>
<input checked="" type="checkbox"/>	2453108	Mert	Damburacı	CNG300	12345678902	Computer Engineering	B Ltd.	APPROVED	APPROVED	<button>Approve</button>
<input type="checkbox"/>	2585461	Efekan	Uysal	CNG400	12345678903	Computer Engineering	C San. ve Tic. A.Ş.	APPROVED	APPROVED	APPROVED
<input type="checkbox"/>	2526200	Umutcan	Çelik	CNG400	12345678904	Computer Engineering	D A.Ş.	APPROVED	APPROVED	APPROVED

Figure 10: Student Affairs Page

1.4.2 Product functions

1. The system shall allow students or coordinator to view the list of internship companies, see the details of companies and internships done there and enable students to apply for internship using their resume.
2. The system shall allow students to fill out Trainee Information Forms and edit them until they have been approved by a coordinator.
3. The system shall allow coordinators to approve, reject or request a re-edit on Trainee Information Forms with their comments in cases of rejection or requested re-edits.
4. The system shall allow students to view their past internships, see their details, upload reports and update their reports.
5. The system shall allow coordinators to add summer training forms.
6. The system shall allow students and coordinators to view summer training forms.
7. The system shall allow students to add and update their resume.
8. The system shall allow coordinators to add announcements.
9. The system shall allow students, instructors, coordinators and student affairs to see announcements.
10. The system shall allow coordinators to set deadline dates.
11. The system shall allow coordinators to assign instructors.
12. The system shall allow instructors to evaluate reports, view company evaluations, ask for a re-edit on reports with comments or grade them.
13. The system shall allow companies to evaluate internships conducted there.

14. The system shall allow companies to assign supervisors for internships conducted there.
15. The system shall allow student affairs to view the list of approved internships and approve or reject a health insurance while checking Trainee Information Form.

1.4.3 Identified stakeholders and design concerns

1. **Students:** Students interact with the system to apply for internships, upload resumes and reports and track their internship progress. Their primary concern is a user-friendly, accessible platform that provides accurate and timely information about internships and evaluations.
2. **Coordinators:** Coordinators oversee the entire internship process, including managing applications, assigning instructors, approving internships, and sending announcements. They need a reliable and efficient system that allows them to manage these responsibilities seamlessly.
3. **Instructors:** Instructors use the system to evaluate student reports and provide feedback. Their main concern is having a straightforward interface that allows them to access and evaluate reports efficiently.
4. **Companies:** Companies view and manage internship applications, evaluate students' performance, and provide feedback. Their primary concern is having secure access to applications and an intuitive system to manage evaluations.
5. **Student Affairs:** Student Affairs plays a critical role in validating health insurance and ensuring compliance with university regulations. Their concern is having a system that integrates health insurance evaluation and provides an efficient way to process these tasks.

1.4.4 User characteristics

Our project targets distinct groups of users:

- **Students:** Those completing their internships and submitting reports.
- **Companies:** Providing internship opportunities and monitoring progress.
- **Coordinators:** Overseeing the internship process and managing approvals.
- **Instructors:** Evaluating internship reports and giving feedback.

Our system is designed to be intuitive and user-friendly, requiring no advanced technical skills. Users with basic internet browsing knowledge can navigate the platform with ease. Accessibility is a priority; the application adheres to accessibility standards to ensure a comfortable experience for all users, including those with visual impairments or other disabilities. Our platform is responsive and accessible on any internet-connected device, including smartphones, tablets, and computers.

1.4.5 Limitations

1. User Errors:

Inaccurate or incomplete inputs, such as incorrect data in the Trainee Information Forms, could impact the work of coordinators and delay processes.

2. Technical Constraints:

Factors like internet speed, server response times, and limited server capacity may affect the website's overall performance and responsiveness.

3. Resource Limitations:

Limited project budget and time constraints may lead to delays or omission of certain features planned for the system.

4. Compliance with Regulations:

Adhering to data privacy regulations is essential. Any mishandling or unauthorized access to personal information could result in legal issues and harm user trust.

5. Dependency on External Services:

The system relies on external services, such as Google Mail, for sending notifications. Any downtime, connectivity issues, or disruptions in these services could delay critical updates and communication with users.

1.4.6 Assumptions and dependencies

- 1. Service Provided:** Students can browse available internships or job positions and perform searches using keywords such as field, company name, or location to find relevant opportunities.
- 2. Coordinator Authorization:** Coordinators do not need to register through the system. Instead, they can manually approve new coordinators for the admin panel. The system will generate and send the necessary login credentials to these new administrators.
- 3. Registered Users Only:** Only registered students can apply for internships or jobs. Students must log in to submit applications. Once an application is submitted, the system sends confirmation emails and status reminders through integrated email services like Gmail.
- 4. Reporting Issues:** Only students and companies can report issues on the system. Coordinators and instructors can view these reports along with the detailed information provided by the users.
- 5. Posting Internship Positions:** When coordinators post a new internship position, they can review and modify the post details before final submission. The system ensures that all necessary fields are completed and that no inappropriate content is included through an automated validation process.
- 6. Editing Permissions:** Only coordinators have the authority to edit job or internship postings. Active students currently enrolled in the system can apply for available internship positions.
- 7. Data Security Compliance:** The system ensures the secure handling of sensitive information, such as companies contact details and other critical data. All data will be stored securely and will comply with data privacy regulations, requiring explicit user permission for storage and use.
- 8. Search Functionality:** The system provides keyword-based search functionality for both students and coordinators. Users can easily filter through internship positions by criteria such as duration, required qualifications, and company name.
- 9. Automatic Notifications:** The system automatically sends notifications to both students and companies via email. These notifications include updates such as application status changes, new internship opportunities, and upcoming deadlines.

2 Specific requirements

2.1 External interfaces

System Interface	Functionality	Input	Output
Mail Service	Send Notification Email	Mailing address to be notified Email content	Confirmation Result Code: 0: success 1:unknown problem

2.2 Functions

2.2.1 Use Case Diagram

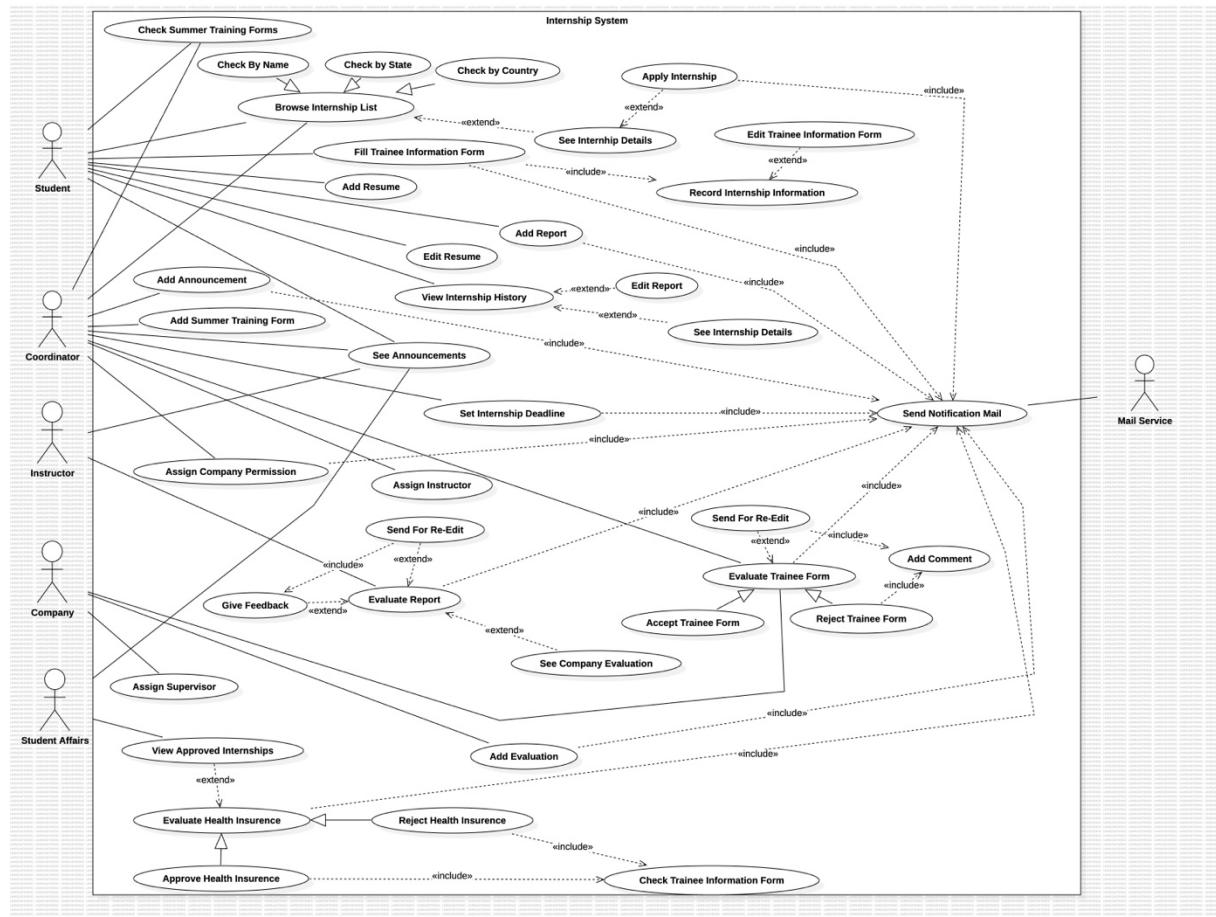


Figure 11: Use Case Diagram

Design Rationale: The Internship System Use Case Diagram illustrates the interactions among different actors and the functionalities provided by the system. It is designed to represent the relationships between users and the system's core features in a clear and comprehensive manner. The rationale behind the design is as follows:

Student:

Represents the primary users of the system who interact with functionalities related to applying for internships, managing personal internship-related documents, and tracking progress.

Activities such as applying for internships, viewing internship details, and uploading resumes are included to simplify the application process.

Coordinator:

Manages administrative aspects such as adding announcements, setting internship deadlines, and overseeing student internships.

The inclusion of advanced permissions like assigning instructors and company permissions reflects their pivotal role in the system.

Instructor:

Focuses on reviewing and evaluating internship reports, as well as providing feedback to students.

Their role ensures academic oversight and quality assurance.

Company:

Engages with the system to assign supervisors and give feedback on students' performance.

This supports collaboration between educational institutions and external organizations.

Student Affairs:

Handles critical administrative processes such as health insurance approval/rejection, ensuring students meet institutional requirements before starting their internships.

Mail Service:

An external system integrated to send notification emails, ensuring smooth communication among stakeholders.

2.2.2 Use-Case Description

You can see the use-case descriptions in the photos below.

Use case	Add Resume
Actors	Student
Cross references	N/A
Typical Course of Events	
Actor Intentions	System Responsibility
1. Student logs into the website.	
2. Student goes to My Resume page.	
	3. System displays student's resume.
4. Student clicks "add" button for add a new resume.	
	5. System opens a new file upload location.
6. Student uploads new resume pdf file.	
	7. System shows a confirmation message.
Alternative Courses	
N/A	

Figure 12: Use-case details of Add Resume

Use case	See Announcements
Actors	Student, Coordinator, Instructor, Student Affairs
Cross references	N/A
Typical Course of Events	
Actor Intentions	System Responsibility
1. Student, Coordinator, Instructor or Student Affairs logs into the website.	
2. Student, Coordinator, Instructor or Student Affairs goes to the announcements page.	
	3. System displays all announcements.
Alternative Courses	
N/A	

Figure 13: Use-case details of See Announcements

Use case	Evaluate Intern
Actors	Company, Mail Service
Cross references	N/A
Typical Course of Events	
Actor Intentions	System Responsibility
1. Company logs into the website.	
2. Company goes to the evaluate form page.	
	3. System shows an evaluation form.
4. Company enters the intern's performance details.	
	5. System checks the informations.
	6. System shows a confirmation message.
	7. System sends a notification e-mail to the student.
Alternative Courses	
Step 5: Empty information (such as working day), so the evaluate cannot be added to the database. Return step 4.	

Figure 14: Use-case details of Evaluate Intern

Use case	Check Summer Training Forms
Actors	Student, Coordinator
Cross references	N/A
Typical Course of Events	
Actor Intentions	System Responsibility
1. Student or coordinator logs into the website.	
2. Student or coordinator goes to the forms page.	
	3. System displays all forms.
4. Student or coordinator selects a form.	
	5. System displays the details of the selected form.
Alternative Courses	
N/A	

Figure 15: Use-case details of Check Summer Training Forms

Use case	Add Summer Training Form
Actors	Coordinator
Cross references	
Typical Course of Events	
Actor Intentions	System Responsibility
1. Coordinator logs in to website.	
2.Coordinator clicks the forms page.	
	3.System will display forms page.
4. Coordinator clicks the "Add" button.	
	5.System will display an upload box.
6.Coordinator uploads necessary forms.	
	7.System will display confirmation message.
Alternative Courses	

Figure 16: Use-case details of Add Summer Training Forms

Use case	Add Report
Actors	Student, Mail Service
Cross references	N/A
Typical Course of Events	
Actor Intentions	System Responsibility
1.Student logs in to the website.	
2.Student goes to My Internships page.	
	3. System shows the students profile page.
4. Student selects the finished internship and clicks the "Add" button	
	5. System will display an upload box.
6.Student uploads the report pdf.	
	7.System will display confirmation message.
	8.System will send notification mail to the Coordinator.
Alternative Courses	
Step 6: Other than PDF files will not be accepted. If so system will display error message and will go	

Figure 17: Use-case details of Add Report

Use case	Accept Trainee Form
Actors	Coordinator, Company, Mail Service
Cross references	Send for Re-Edit
Typical Course of Events	
Actor Intentions	System Responsibility
1. Coordinator or Company logs into the website.	
2. Coordinator or Company goes to the Trainee Form page.	
	3. System shows Trainee Forms.
4. Coordinator or Company checks Trainee Forms.	
5. Coordinator or Company accepts trainee form	
	6. System shows confirmation message.
	7. System sends notification e-mail to the student.
Alternative Courses	
Step 4: If there is incorrect information (such as the start date of the internship), the Coordinator or the Company will send the form to the student for re-edit.	

Figure 18: Use-case details of Accept Trainee Form

Use case	Fill Trainee Information Form
Actors	Student,Mail Service
Cross references	Edit Trainee Information Form
Typical Course of Events	
Actor Intentions	System Responsibility
1.Student logs into internship portal.	
2.Student enter the details for new Internship of the company.	
	3.System checks that it is on the database.
	4.System notify the user company added on internship database.
	5.System send notification to coordinator new company added on internship database.
Alternative Courses	
Step3: If is that in database system doesn't allow to user to enter name for the company. Return step:2.	

Figure 19: Use-case details of Fill Trainee Information Form

Use case	Assign Company Permission
Actors	Coordinator,Mail Service
Cross references	N/A
Typical Course of Events	
Actor Intentions	System Responsibility
1.Coordinator check the internship on the list.	
	2.System checks whether(exist on) this login credentials is in the system.
3.Coordinator approve company information.	
	4.System add and give credentials in database.
	5.System sends login information to the company
Alternative Courses	
Step 2 : If the system detects that these credentials already exist in the database, the following steps will be taken coordinator is informed with a message These login credentials are already registered in the system. System give options to modify existing company information. Return Step:1.	
Step 4 : If there is an issue with the database connection, the system will display an error message and prompt the coordinator to retry the operation. Return Step: 1.	

Figure 20: Use-case details of Assign Company Permissions

Use case	Evaluate Report
Actors	Instructor,Student,Mail Service
Cross references	Send for Re-Edit,Give Feedback
Typical Course of Events	
Actor Intentions	System Responsibility
1.Instructor receives the internship report from the student and opens it for evaluation.	
	2.System displays the submitted report to the instructor for review.
3.Instructor evaluates the content of the report and determines whether it meets the required criteria.	
	4.System asks the instructor to confirm their decision (Accept, Reject, Send for Re-Edit).
5.Instructor approves and directs.	
	6.System sends an email to notify the student about the evaluation result (accepted, rejected, or sent for re-edit).
Alternative Courses	
Step 6: If the instructor selects Accept:If confirmed, the system updates the report status to accepted and sends an email to the student indicating that the report has been approved.	
Step 6: If the instructor selects Reject:If confirmed, the system marks the report as rejected and sends an email to the student providing feedback and reasons for rejection.	
Step 6: If the instructor selects Send for Re-Edit:If confirmed, the system sends an email to the student with detailed feedback, asking them to revise and resubmit the report.	
Step 6: Instructor give grade for the internship (passing or fail) conditions.	
Step 6: Instructor can see company evolution form.	

Figure 21 : Use-case details of Evaluate Report

Use case	View Approved Internships
Actors	Student Affairs
Cross references	Approve Health Insurance, Reject Health Insurance
Typical Course of Events	
Actor Intentions	System Responsibility
1.Student Affairs logs into the system to view the approved internships.	
	2.System verifies and displays the list of approved internships.
3.Student Affairs reviews the details of the approved internships.	
	4.System provides the officer with detailed information about the internships with options.
5.Student Affairs evaluates the health insurance status of the student.	
	6.System logs the evaluation in the database.
	7.System notifies the student about the evaluation of their health insurance status.
Alternative Courses	
Step 6: If there is an issue with saving the evaluation in the database, the system will display an error message to the Student Affairs officer, indicating that the evaluation could not be logged. The system will then prompt the officer to retry the operation. Return step : 5.	
Step 5: If student affairs approve or reject to report add the database and notify to student. Return Step:7	

Figure 22: Use-case details of View Approved Internships

2.3 Usability Requirements

1- Success Score

- **Requirement:** The system should achieve a success score of at least 90% during usability testing.
- **Justification:** This score ensures that users (students, coordinators, and companies) can successfully complete tasks such as logging in, applying for internships, reviewing applications, and submitting reports without errors. Achieving a high success score ensures a smooth user experience and increases satisfaction. (Brooke, J. "SUS: A quick and dirty usability scale." *Usability Evaluation in Industry* (1996)).

2- Error Rate

- **Requirement:** The average error rate must be below 2 errors per task during operations such as form submissions, report uploads, and application management.
- **Justification:** Minimizing errors in core functionalities (e.g., internship applications, report evaluations) ensures that users can operate the system efficiently, reducing frustration and improving trust in the platform.

3-Task Completion Time

- **Requirement:** The average time to complete key operations, such as applying for internships, submitting forms, or viewing application statuses, should not exceed 1 minute.
- **Justification:** Faster task completion ensures user engagement, reduces time spent on repetitive operations, and allows coordinators and students to manage their processes more efficiently.

4- System Usability Scale (SUS) Score

- **Requirement:** The system should achieve a System Usability Scale (SUS) score of at least 75, indicating above-average usability.
- **Justification:** A high SUS score reflects that users can intuitively navigate the platform for tasks like searching internships, uploading resumes, and submitting reports. This ensures the system aligns with user expectations for modern web applications. (Nielsen, Jakob. *Usability engineering*. Morgan Kaufmann, 1994.)

5- Real-Time Tracking and Updates

- **Requirement:** The system must display real-time updates regarding application statuses, form submissions, and approvals with a refresh rate of under 10 seconds.
- **Justification:** Providing real-time tracking allows students, coordinators, and companies to stay informed about updates without the need for manual follow-ups, enhancing overall efficiency.

6- Accessibility and Inclusivity

- **Requirement:** The platform must comply with WCAG 2.1 AA accessibility standards, ensuring features like screen reader compatibility, high-contrast options, and intuitive navigation for users with disabilities.
- **Justification:** Ensuring accessibility makes the system inclusive and usable by a broader range of users, including those with visual or motor impairments.

7- Search and Filtering Efficiency

- **Requirement:** The search functionality must allow users to filter internships by field, location, company name, and academic background, returning results within 5 seconds.
- **Justification:** Efficient search and filtering capabilities enable students to quickly identify relevant opportunities without unnecessary delays, enhancing user productivity.

2.4 Performance requirements

The performance of the internship management system is crucial for ensuring a smooth and efficient experience for all users, including students, coordinators, instructors, company supervisors, and student affairs personnel. The following are the key performance requirements for the system:

- **Response Time:**

The system must provide a response time of under 3 seconds for all user interactions (e.g., form submissions, page loading, viewing announcements, and uploading reports) under normal load conditions (500 concurrent users) (Hoxmeier & DiCesare, 2000).

Justification: Slow response times can frustrate users and cause delays in critical tasks like internship approvals and report submissions. A quick response time ensures a smooth and effective user experience, especially when managing time-sensitive operations such as internship deadlines and feedback loops.

- **Scalability:**

The system should be able to scale up to handle up to 1000 concurrent users without any significant performance degradation.

Justification: The system will be used by many students, coordinators, and companies, especially during peak times like the start and end of semesters when internship applications, evaluations, and reports are submitted en masse. Scalability ensures the system can manage these peak loads without crashing or slowing down (Fikayo Adepoju Oreoluwa, 2021).

- **Throughput:**

The system must process at least 50 transactions per minute, including form submissions, report uploads, and notifications.

Justification: Throughput is important to ensure that the system can handle multiple simultaneous user actions, such as students submitting internship forms while companies submit evaluations. This ensures that the system is responsive during periods of high activity and can process large amounts of data efficiently (Bhaval Patel, 2024).

- **System Availability:**

The system must maintain an uptime of 99.9%, which translates to no more than 8 hours and 45 minutes of downtime per year.

Justification: The system will be used by students and companies from various time zones, meaning it must be available around the clock. High availability ensures that users can access the system when needed, particularly for critical tasks like meeting submission deadlines or responding to time-sensitive alerts (Ramotion, 2023).

- **Data Processing Time:**

Bulk data operations, such as generating reports or processing multiple student applications, should not exceed 30 seconds.

Justification: Coordinators and instructors may need to generate reports that summarize student data, internship statuses, or evaluation outcomes. Ensuring quick data processing reduces delays in these tasks and improves overall system efficiency (Paul).

- **Notification Latency:**

Notifications (email, SMS, etc.) should be delivered within 5 minutes of the triggering event (e.g., form approval, deadline reminders, report submission).

Justification: Timely notifications are essential to ensure that all stakeholders (students, supervisors, coordinators) are kept informed about critical tasks and deadlines. Delayed notifications could result in missed deadlines or bottlenecks in the approval process (SoftSages, 2024).

- **Load Testing:**

The system should undergo periodic load testing to ensure that it can handle peak usage scenarios without a drop in performance.

Justification: Ensuring optimal performance under high load conditions is essential for maintaining system reliability, especially during the internship application and evaluation periods, when the number of active users can spike (Ana Crudu & MoldStud, 2024).

By meeting these performance requirements, the internship management system will ensure efficient operations, minimal downtime, and a seamless user experience across all stages of the internship process. These requirements are based on the expected volume of users and system transactions, and they are designed to prevent performance bottlenecks or user frustration.

2.5 Logical database requirements

2.5.1 Relational Database Diagram

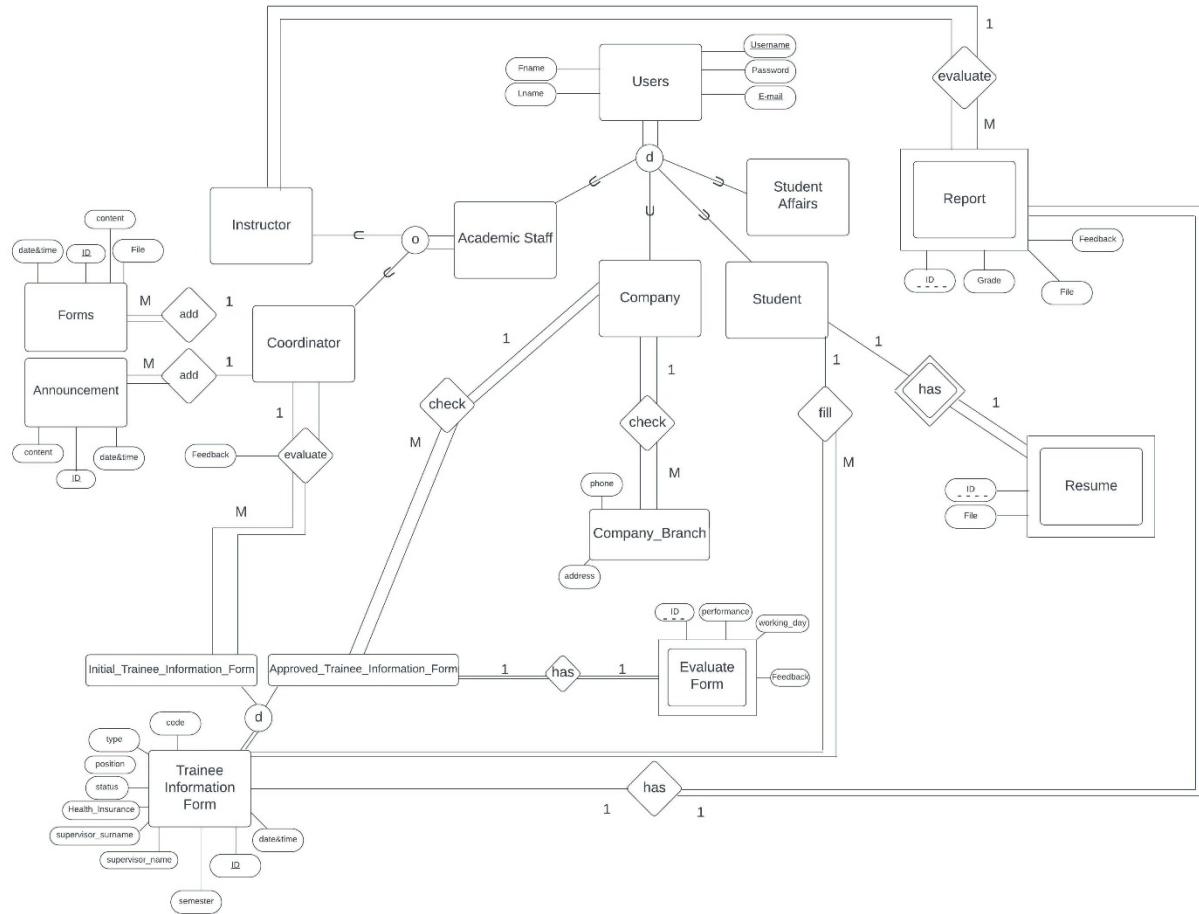


Figure 23: EER Diagram of Project

You can examine the EER Diagram from Figure 23 Design Requirements of EER Diagram are explained below.

Evaluate Form as a Weak Entity:

The Evaluate Form is a weak entity because it depends on the Trainee Information Form for its identification. This ensures that evaluations are tied to specific trainees, emphasizing their dependency and purpose in assessing internship performance.

Users and Roles:

The Users entity is generalized into multiple roles: Student, Coordinator, Instructor, Academic Staff, and Student Affairs. This structure allows for shared attributes like username, email, and password, while also accommodating role-specific functionalities, ensuring efficient user management.

Trainee Information Form:

This form captures detailed internship data, including supervisor information, health insurance, and semester. It serves as the foundation for monitoring and evaluating student internships.

Branch-Level Design:

The separation of Company and Company branch ensures scalability and precise tracking of internship locations, enabling companies to have multiple branches while maintaining clear records.

Approval Process:

Reports uploaded by students undergo a structured approval process, with statuses tracked as either approved or rejected. This ensures transparency, accountability, and structured feedback for students.

Forms and Announcements:

Coordinators have the flexibility to add timestamped forms and announcements, ensuring the system remains dynamic and adaptable to updates and communication needs.

Resume and Report Integration:

Resumes and reports are assigned unique IDs and are directly tied to evaluation processes, facilitating structured feedback and supporting students' academic and professional growth.

2.5.2 Relational Database Diagram Assumptions

1. We assume that each user has a unique username for easy identification within the system.
2. We believe that the coordinator can add multiple forms and announcements. These are recorded with date and time for tracking purposes.
3. Forms and reports submitted by students are assumed to be subject to review by instructors.
4. We ensure that every report has a unique ID and includes a grade as part of its evaluation.
5. Each evaluation form is designed to include details about the trainee's working days, performance, and feedback from relevant parties.
6. We assume that all announcements and forms are logged in with accurate date and time stamps for consistent record-keeping.
7. Each resume uploaded to the system must have a unique ID, ensuring proper distinction between different uploads.
8. The relationship between students and the companies they intern with is recorded in the system. A student can intern with multiple companies but can only intern at the same company once.
9. All reports and forms uploaded by students go through an approval process, where they are either approved or rejected based on evaluations, with feedback provided for clarity.
10. Company branches have unique identifiers to track their address and other details, ensuring accurate connections with students and their forms.
11. The system supports linking the initial trainee information forms to approved trainee information forms, reflecting the progression from submission to approval.

2.6 Software system attributes

Reliability:

1. The system will regularly apply software updates to ensure it stays compatible with new technologies and maintains optimal performance.
2. In case of potential errors, the system will prevent data loss by securely storing critical internship and user information.
3. The system will handle high traffic efficiently, ensuring no performance issues arise, even during periods with many active users and multiple ongoing transactions.
4. When incorrect data is entered, the system will allow users to easily correct and re-submit their information.

• Availability:

1. If the system undergoes maintenance, users will be notified, and access will be temporarily restricted. Once maintenance is complete, users can resume their activities without issues.
2. In the event of a system error, a clear and descriptive error message will be displayed to the user, explaining the nature of the issue.
3. After system failures, users will be able to retrieve their previously saved information without any data loss, ensuring continuity through robust data recording mechanisms.
4. The system will aim to minimize downtime, limiting unavailability to no more than 2% annually, to maintain a high level of service availability.

• Security:

1. The system will ensure that users' personal and internship-related data is encrypted and securely stored to prevent unauthorized access.
2. Sensitive actions within the system, such as updating personal details or submitting internship applications, will require an additional layer of confirmation, such as sending a verification code via email.
3. The system will regularly monitor and log user activity, identifying and responding to suspicious behaviour, such as repeated failed actions or unusual access patterns.
4. Users will be periodically prompted to review and update their contact and security information to ensure the system remains up to date and secure.

• Maintainability:

1. Code written for the system will be well-documented with clear comments, ensuring that future developers can easily understand and update the software.
2. The system's modular design will include intermediate safeguards so that if one component fails, the rest of the system will continue to function without disruption.
3. Version control and regular code reviews will be implemented to ensure that updates can be made smoothly, preventing system downtime or malfunction.
4. Developers will adhere to clean coding practices, ensuring that the software is easy to maintain and update over time.

2.7 Supporting information

N/A

3 Software Estimation

General System Characteristics	Degree of Influence (DI)
Data communications	5
Distributed processing	4
Performance	4
Heavily used configuration	3
Transaction rates	5
Online data entry	2
End-user efficiency	3
Online updates	3
Complex processing	4
Reusability	4
Installation ease	3
Operational ease	3
Multiple sites	2
Facilitate change	1
Total of Degree of Influence:	46
Value Adjustment Factor (VAF):	1,11

VAF = ((TDI*0.01) + 0.65) We used this formula to calculate VAF.

Program Characteristics	Low Complexity	Medium Complexity	High Complexity	Total
Inputs	1	2	0	10
Outputs	2	3	1	29
Inquiries	1	3	1	20
Logical Internal Files	2	2	0	14
External Interface Files	1	1	0	6

We summed up the totals for all program characteristics to obtain the **Unadjusted Total Function Points (UFP)**, which is **79**.

Unadjusted Total of Function Points (ATFP): 79

Finally, we calculated the **Adjusted Total Function Points (ATFP)** using the formula:
ATFP = UFP × VAF.

Adjusted Total of Function Points: **84.53**

KDSI Calculation:

Programming Language	Percentage	Language Unit Size	Total KDSI
Java	50%	153	26
TypeScript	20%	80	10
SQL	15%	72	5.4
Angular	15%	65	4.2

Then formula for calculating KDSI is:

$$\text{KDSI} = (\text{ATFP} * \text{Language Unit Size}) / 1000$$

1. **Java:**

$$\text{KDSI} = (153 \times 50) / 1000 = 26$$

2. **TypeScript:**

$$\text{KDSI} = (80 \times 20) / 1000 = 10$$

3. **SQL:**

$$\text{KDSI} = (72 \times 15) / 1000 = 5.4$$

4. **Angular:**

$$\text{KDSI} = (65 \times 15) / 1000 = 4.2$$

Development Type	Semi-Detached
Estimated Effort in Man-Months	17.54
Estimated Development Time (Months)	6.75
Estimated Team Size	3

Basic COCOMO Parameters:

$$a = 3.0, b = 1.12, c = 0.35 \text{ our KDSI } 45.6$$

Effort in Man-Months (MM):

$$\text{MM} = 3.0 \times (45.6)^{1.12} \text{ MM } \approx 17.54$$

Development Time (in Months):

$$\text{Time} = 2.5 \times (17.54)^{0.35} \text{ Time } \approx 6.75$$

Estimated Team Size:

$$\text{Team Size} = (17.54 / 6.75) \text{ Team Size } \approx 3$$

3.2 Jones' Estimation

Adjusted Total of Function Points (ATFP)	84.53
Kind of Software	Systems
Software Organization's Skills/Abilities	Average
Estimated Effort in Man-Months	17.54
Estimated Development Time in Months	6.75
Estimated Team Size	4

Our project software kind is systems because software components. We plan to integrate advanced backend systems with TypeScript, SQL, and Angular for web-based functionalities, alongside database interaction. We conducted calculations using both COCOMO and Jones' estimation methods, and the results were consistent. The estimations demonstrate a reasonable level of accuracy, validating our approach.

4 Architectural Views

In this report, the architecture of software systems will be described based on the Logical View, Process View, Development View, and Deployment View, as suggested by Kruchten (1995)¹

4.1 Logical View

4.1.1 Class Diagram

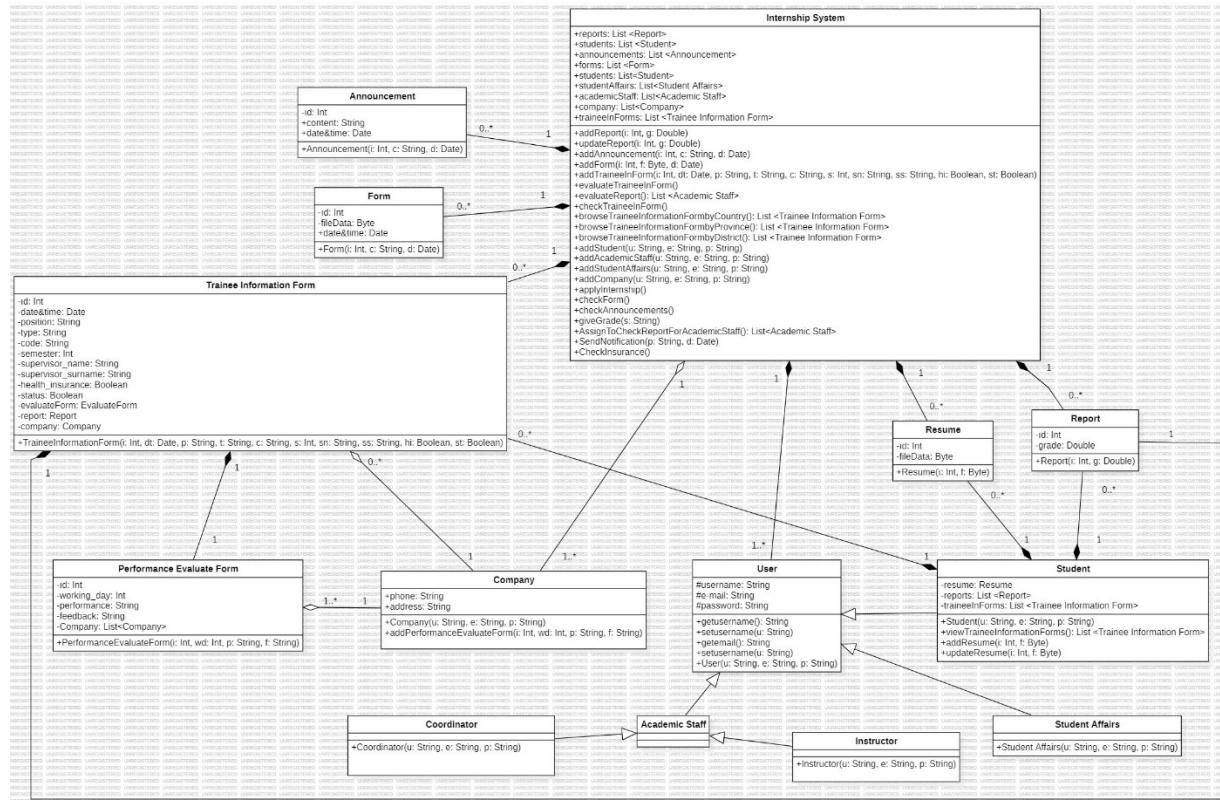


Figure 24: Class Diagram of Project

You can see the Class Diagram of Project from Figure 24. You can also examine Design Rationale below.

¹ P. B. Kruchten, "The 4+1 View Model of architecture," in IEEE Software, vol. 12, no. 6, pp. 42-50, Nov. 1995, doi: 10.1109/52.469759.

User Class Design:

The User class is the base for all user roles (Student, Coordinator, Instructor, Academic Staff, Student Affairs), allowing shared attributes like username, email, and password. This inheritance structure simplifies the addition of role-specific methods, such as evaluateReport() for instructors.

Trainee Information Form Dependency:

The Trainee Information Form class connects students with their internship details. It is associated with both companies and reports, ensuring that every internship is documented comprehensively.

Performance Evaluate Form as a Weak Entity:

The Performance Evaluate Form class is closely tied to the Trainee Information Form and depends on it for its identification. This highlights the subordinate nature of evaluations to internship records.

Resume and Report Integration:

The Resume and Report classes are explicitly linked to students, ensuring that all submissions are traceable to the respective user. Reports include a grading mechanism for instructors, while resumes focus on student profiles.

Coordinator Responsibilities:

The Coordinator class includes methods for adding forms, announcements, and managing internship records. This role ensures system updates and administrative control.

Announcement and Form Accessibility:

Announcements and forms are designed to be dynamic, with timestamps and visibility across user roles, ensuring effective communication.

Company Association:

The Company class includes attributes like address and phone for detailed information. It is linked to both Performance Evaluate Form and Trainee Information Form, ensuring a comprehensive record of internships.

Scalability and Maintainability:

The class diagram is structured to allow future expansion, such as adding new user roles or extending functionalities like advanced feedback systems. Each component is modular and maintainable.

4.1.2 Class Diagram Assumptions

1. Each user has a unique username and password to ensure secure and individual access to the system.
2. Students can upload multiple resumes and reports, but each resume and report are uniquely associated with one student.
3. Announcements are managed only by coordinators and are visible to all other users without restrictions.
4. Forms, such as Trainee Information Forms, include detailed attributes (e.g., supervisor name, position) to ensure comprehensive tracking of internship details.
5. Companies and their branches are recorded separately to enable detailed tracking of internships at specific locations.

6. Performance evaluation forms are directly tied to trainee information forms, ensuring evaluations are always linked to a specific internship.
7. Reports uploaded by students are graded by instructors or academic staff, reflecting academic performance during internships.
8. The system assumes all data flows between components (e.g., reports, resumes, trainee information forms) are consistent and traceable to maintain data integrity.
9. Students can only intern at the same company once, but they can intern at multiple different companies.

4.2 Process View

The process view shows how the system is composed of interacting processes.

4.2.1 Activity Diagram

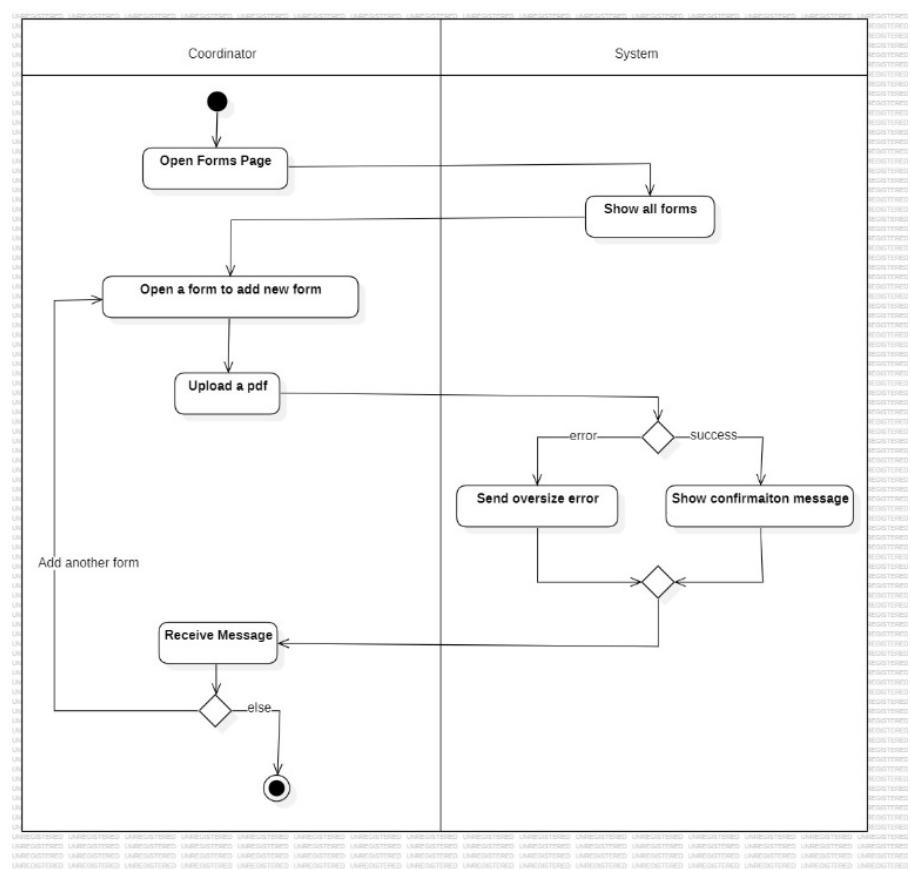


Figure 25: Add Form – Activity Diagram

Design Rationale for Add Form

In the "Add Form" process, we aimed to simplify how coordinators upload and manage forms in the system. Key considerations include:

Ease of Use: Coordinators can quickly access the Forms Page and follow a simple process to upload a new form in PDF format.

Error Handling: To prevent technical issues, the system checks the size of the uploaded PDF and notifies the coordinator if it exceeds the limit. This ensures the system remains efficient.

Feedback: After a successful upload, the system provides a confirmation message to reassure the coordinator that the form is accessible to users.

You can see the details of this process in figure 25.

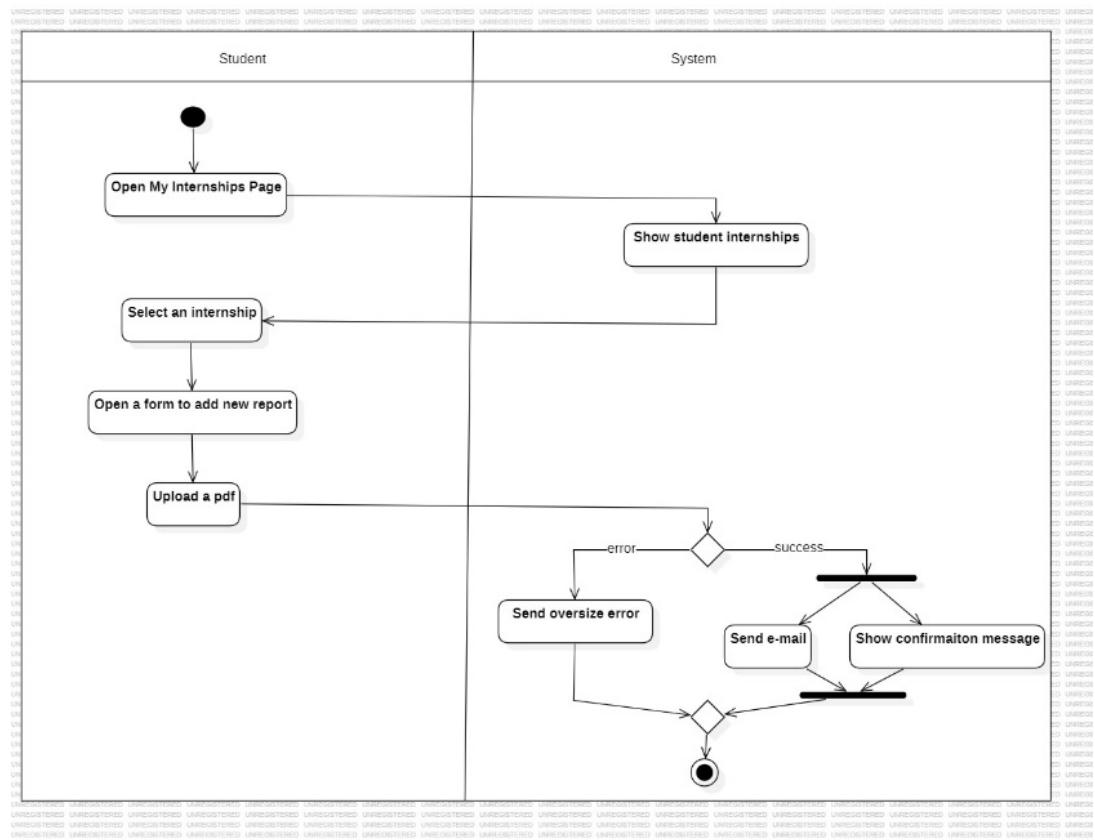


Figure 26: Add Report – Activity Diagram

Design Rationale for Add Report

For the "Add Report" process, we designed a clear and efficient workflow to enable students to upload their reports seamlessly. We considered the following:

Student-Centric Design: Students can access their internship details easily and select the relevant internship to upload their report.

Validation: The system checks the uploaded report for size issues and provides immediate feedback if an error occurs.

Notifications: Once the report is successfully uploaded, the system sends an email to the coordinator, ensuring they are informed about the new report submission.

Confirmation: A success message reassures students that their report has been added to the system.

You can see the details of this process in figure 26.

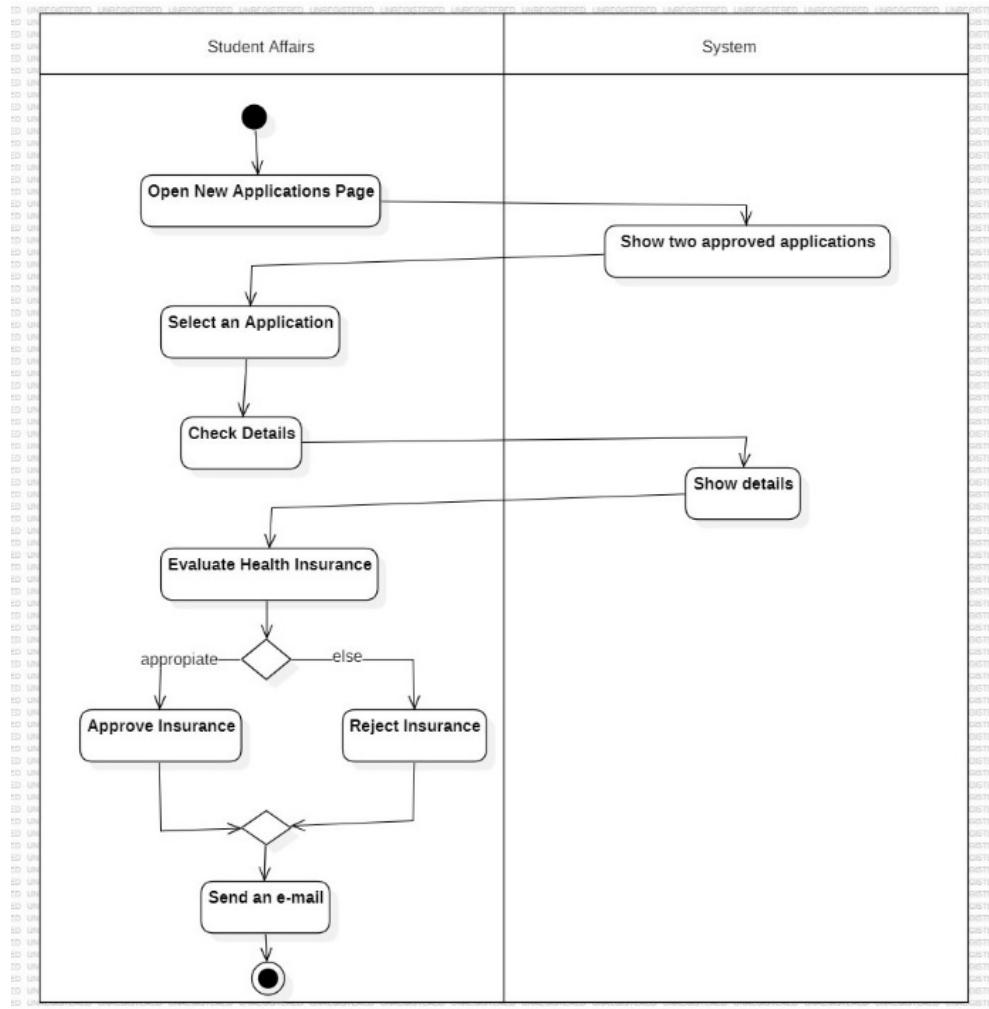


Figure 27: Evaluate Health Insurance – Activity Diagram

Design Rationale for Evaluate Health Insurance

The "Evaluate Health Insurance" process was designed to streamline how Student Affairs checks and approves insurance details. We focused on:

Transparency: Student Affairs officers can view both company and coordinator approvals before evaluating the application. This provides a clear context for decision-making.

Decision Workflow: Officers can approve or reject insurance details based on their review, ensuring accuracy and compliance.

Notifications: The system automatically informs students about the decision, maintaining effective communication.

You can see the details of this process in figure 27.

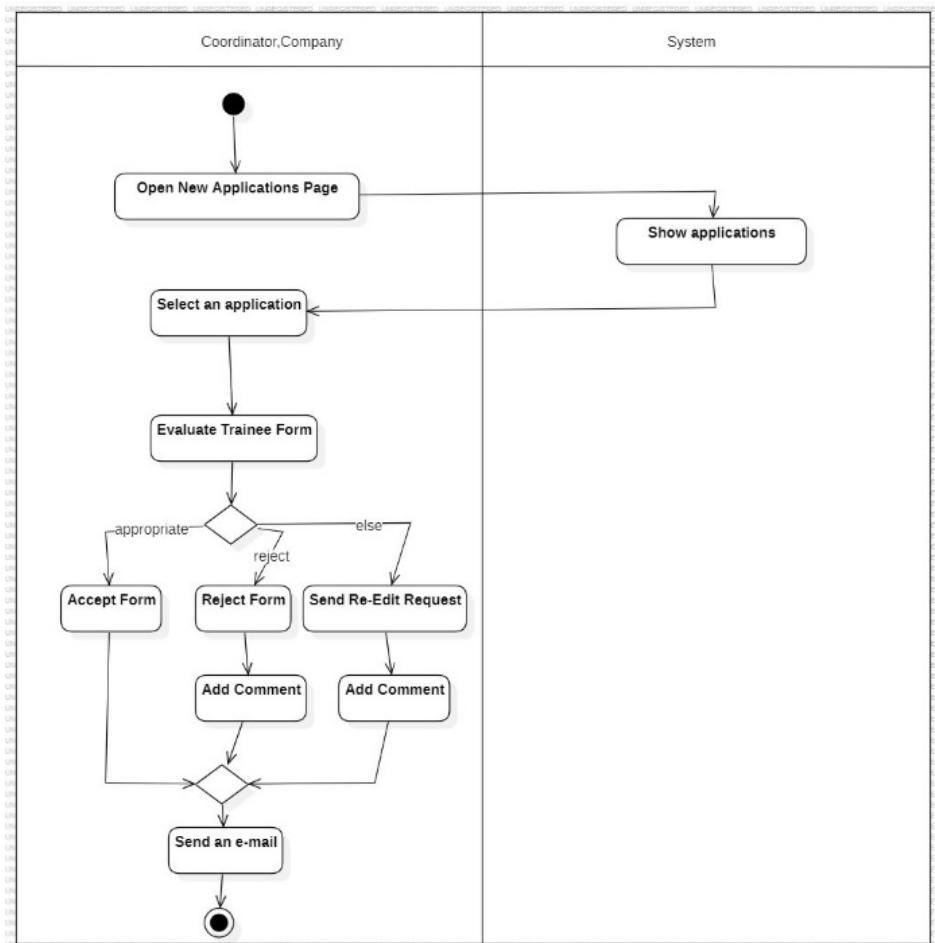


Figure 28: Evaluate Trainee Form – Activity Diagram

Design Rationale for Evaluate Trainee Form

In the "Evaluate Trainee Form" process, we aimed to facilitate how coordinators and companies review applications. Key considerations include:

Efficient Review: Coordinators and companies can access all applications in one place and check trainee forms systematically.

Error Handling: If a form is incomplete or incorrect, users can reject it or request changes, ensuring that only valid forms proceed.

Communication: Coordinators and companies can add comments during evaluation, which are sent to the student along with the result. This provides constructive feedback for improvements.

Automation: The system sends emails with results to students, ensuring they stay informed about the status of their applications.

You can see the details of this process in figure 28.

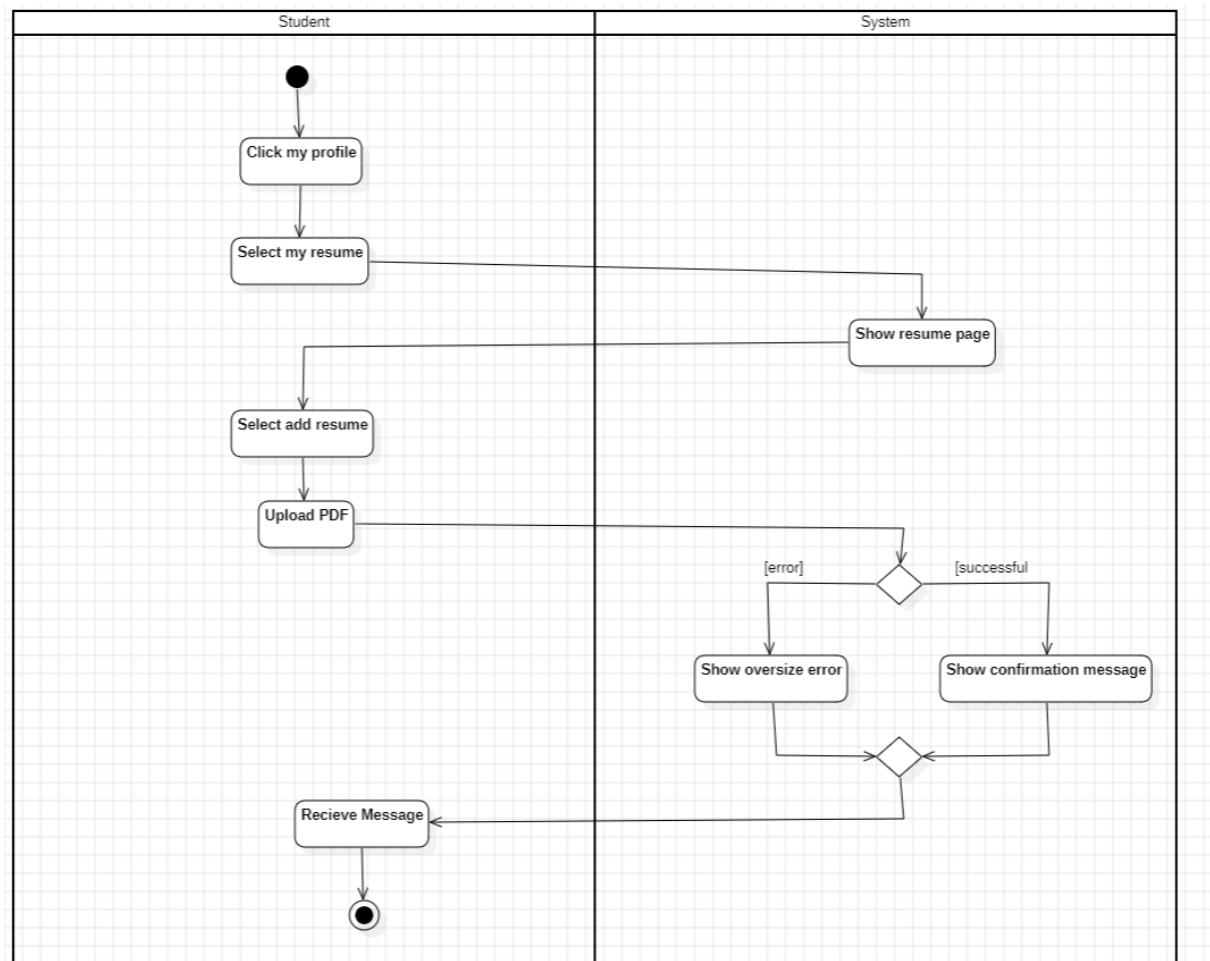


Figure 29: Add Resume – Activity Diagram

Design Rationale: Student clicks my profile button, then selects my resume page. System shows the page. Student selects add resume button and uploads resume as a PDF file. If PDF file is oversized then student faces error. If it is not it gets confirmation message. You can see from Figure 29.

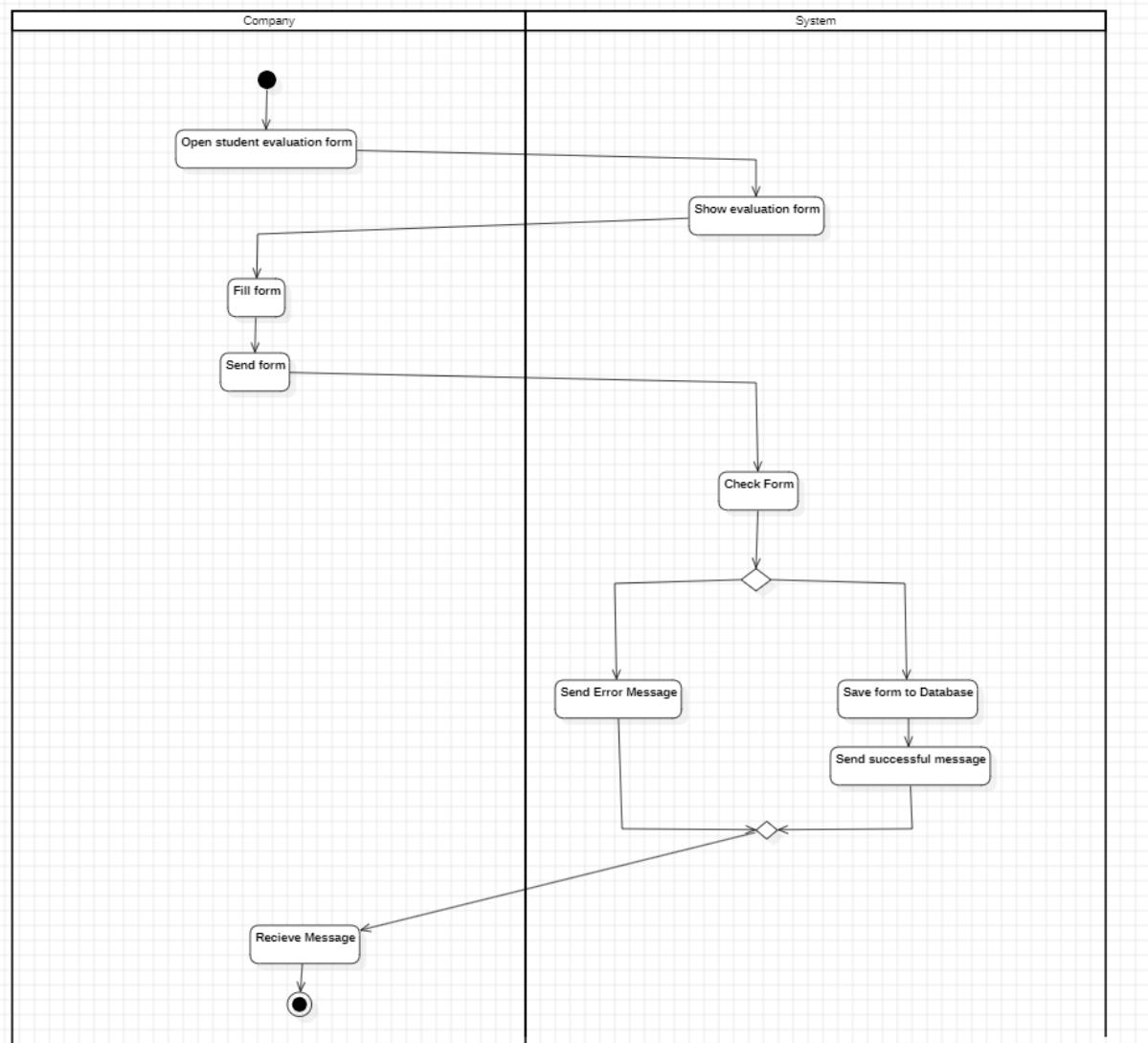


Figure 30: Add Evaluation – Activity Diagram

Design Rationale: After logging in the system Company opens student evaluation form. System shows it. Company evaluator fills and sends the form. System checks if it is error or not (such as dates are not same with trainee form) If it is error they face with error message. If it is not report will be saved and company will get successful message. You can see from Figure 30.

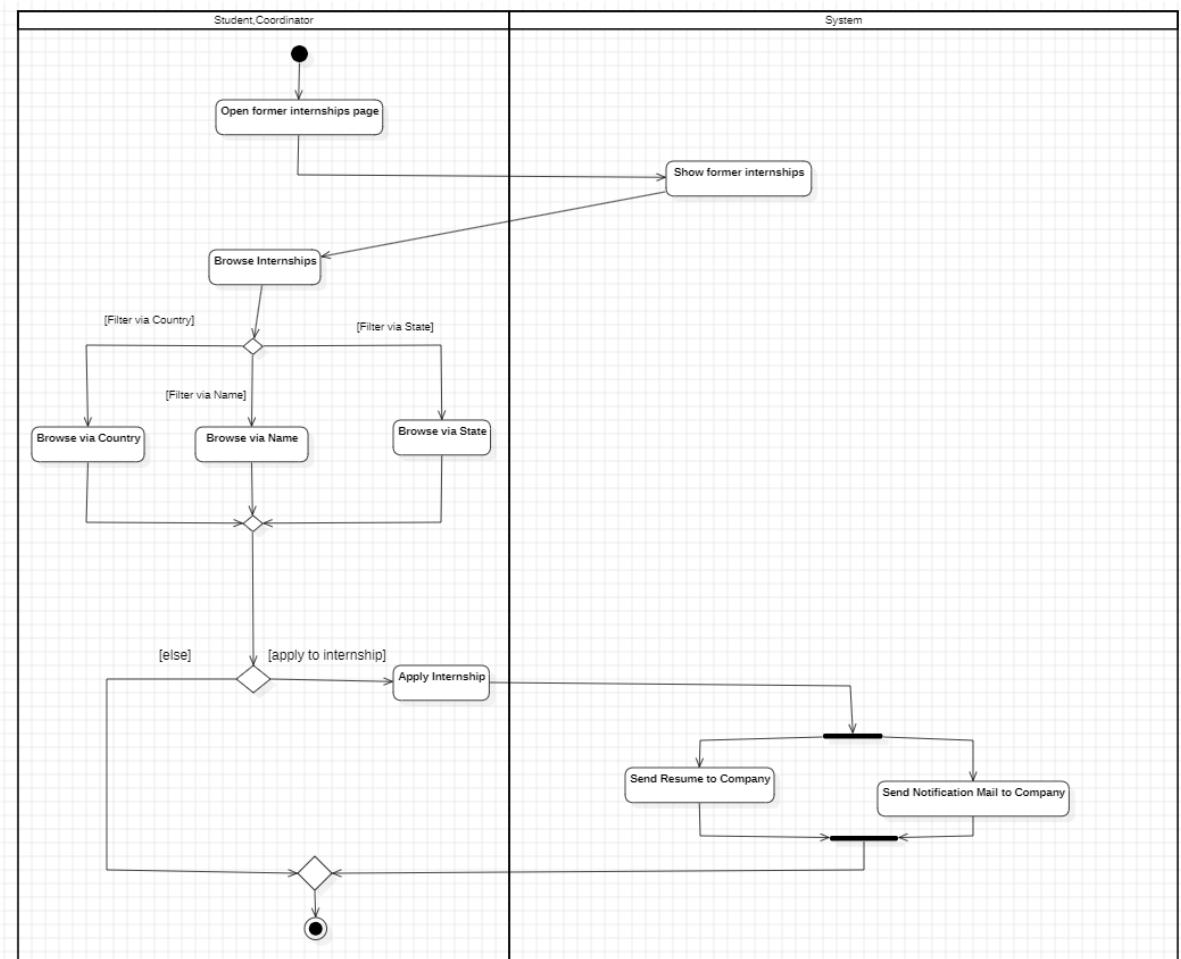


Figure 31: Browse Internships – Activity Diagram

Design Rationale: In this diagram browsing internships are shown. Students and Coordinator can both do it. First they need to go to former internships page, which system will provide. Then they can start to browse. After that they can use filtering such as via country, state and company name. Also students can apply the internships they see with apply button. If they do their already uploaded resume will be send to company interface and Company will be notified via mail. Of course if student does not have uploaded pdf they cannot click apply button. You can see from Figure 31.

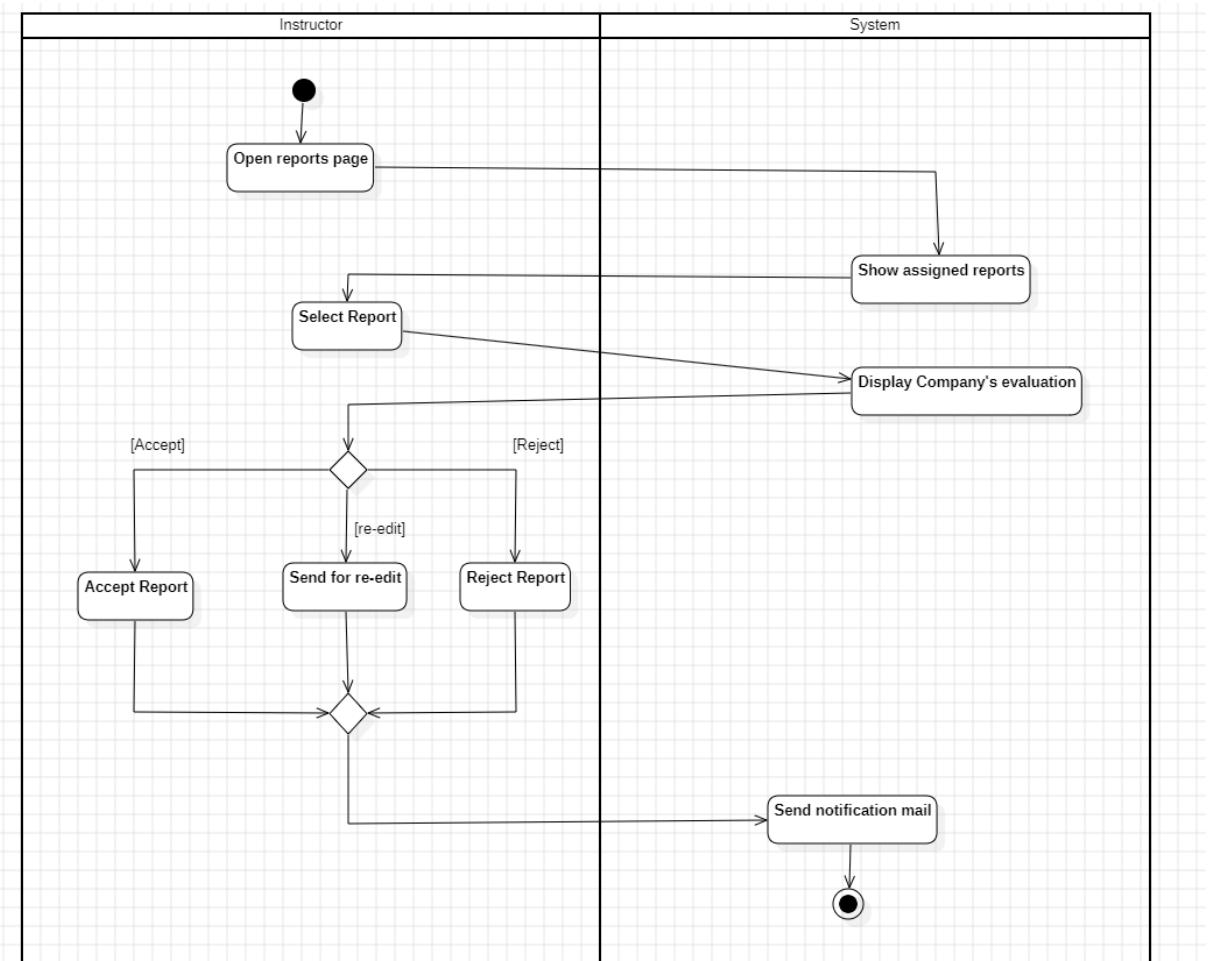


Figure 32: Evaluate Reports – Activity Diagram

Design Rationale: Evaluating reports are Instructors job. Instructors must open reports page. When they open they will see their assigned reports by Coordinator. Then they will select the report of any student and also system will provide Company's Evaluation then they start to evaluate it. They have 3 options after that. First they can Accept Report. Second they can send report back to student for re-editing purpose. Third they can reject the report. At the end notification mail will send to student after either taking feedback or report. You can see from Figure 32.

4.2.2 Sequence Diagrams

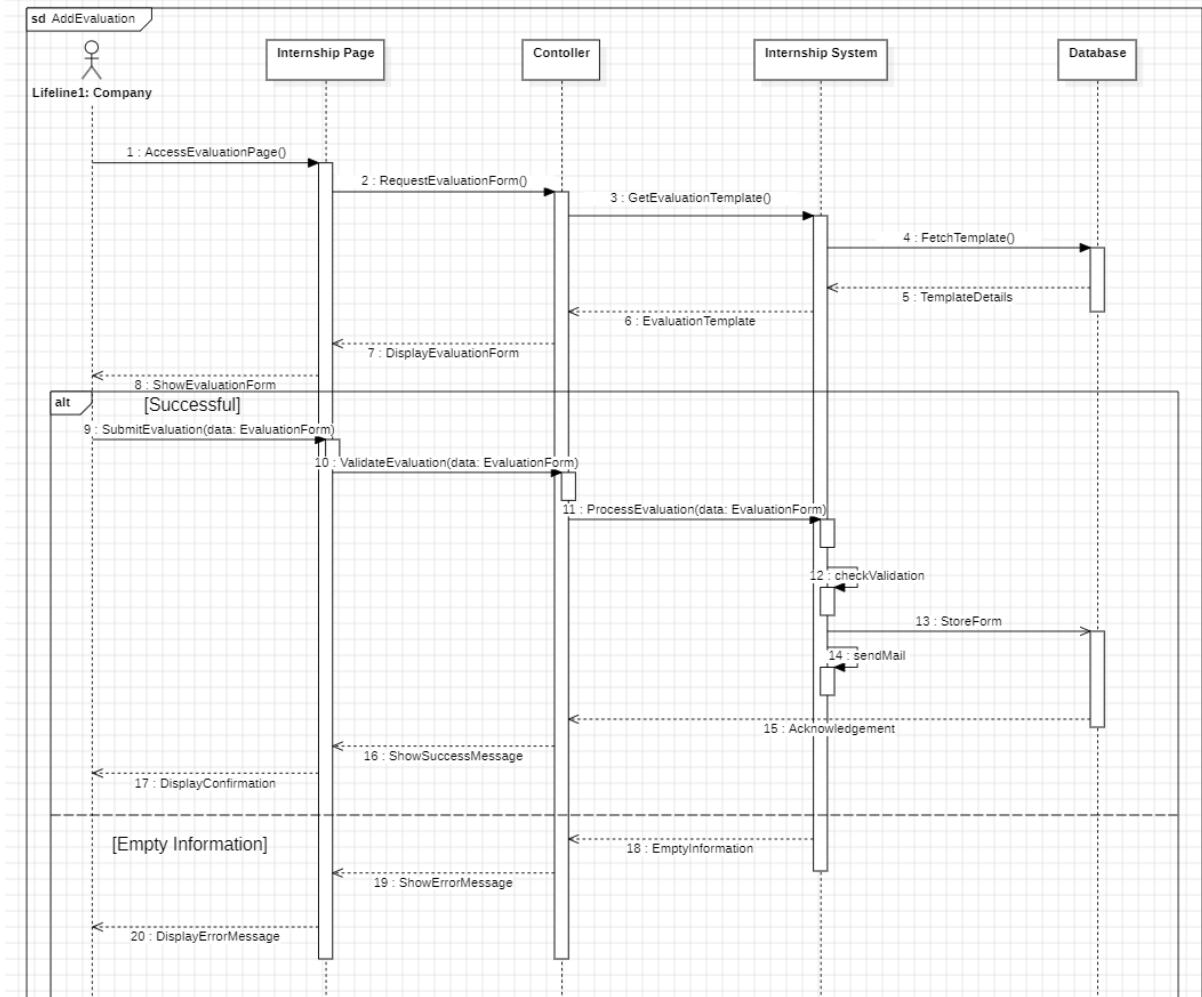


Figure 33: Sequence Diagram of AddEvaluation

Design Rationale: This is the company's evaluating a intern student process. At the end of the internship Company's must evaluate the student. First they need to access evaluating page. Since our project design is related with Model View Controller(MVC) pattern[6] , we have made our sequence diagrams related to this pattern. After accesing the evaluation form with controller and system. Company worker must fill the form with no empty information. If they try to submit with incorrect or missing information such as; wrong working days, they will be warned with error message. If they submit correctly, system will save the form to the database and send notification mail to student and coordinator. You can see from Figure 33.

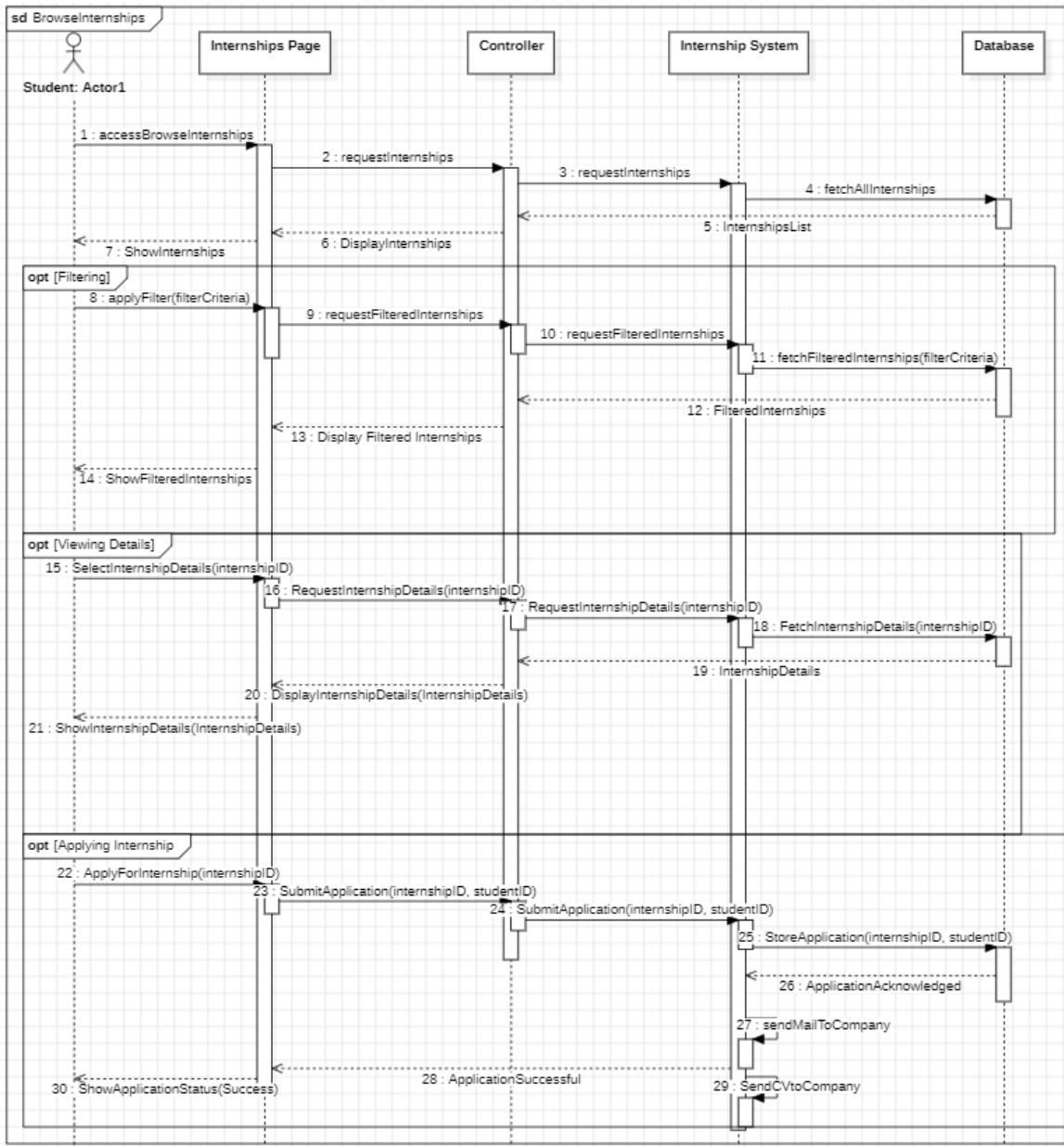


Figure 34: Sequence Diagram of Browse Internships

Design Rationale: This diagram is about browsing internships in our system. Students can access this via internships page. After that MVC pattern do his work to bring internships to student from database. After viewing internships student can use filtering with 3 options: Country, State and Name. Since all of the filtering methods are same and for sake of simplicity in our sequence diagram, we show them as one filtering. After that student have options such as viewing detail of the internships and applying an specific internship. If they view detail MVC pattern will do his work. If they apply for an internship system will send notification mail to company. Also system will send student's resume to company interface. You can see from Figure 34.

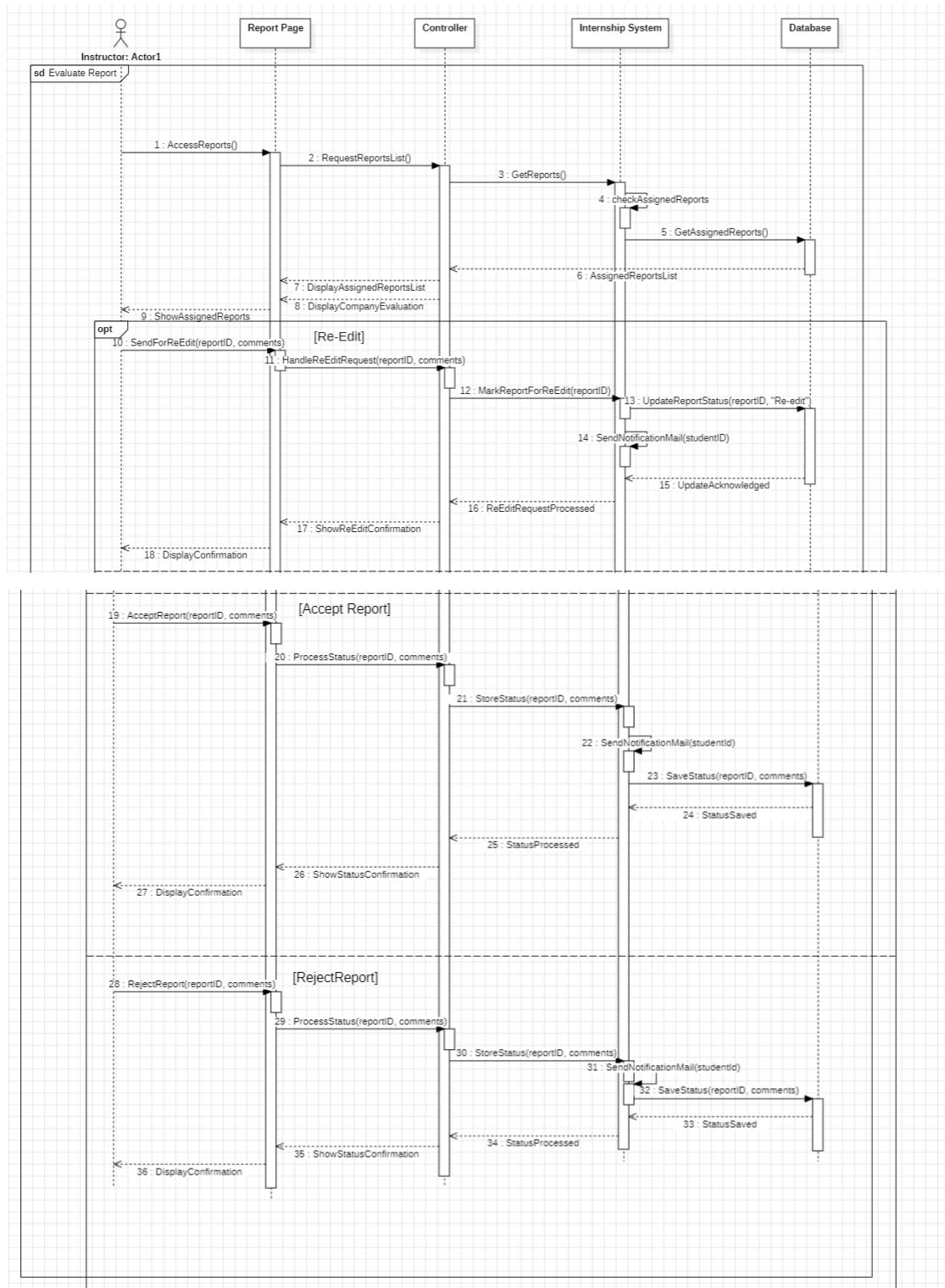


Figure 35: Sequence Diagram of Evaluate Report

Design Rationale: The sequence diagram above is about Evaluating Report of the students. Instructors are responsible for this. They first need to open the reports page, after that system will show assigned reports to the instructors. Then they have 3 choices which are: Send for Re-edit; they send report back to student and students will get notify in the process. Accept Report; they approve the report with grade feedback, and they can also see the Reject Report. You can see from Figure 35.

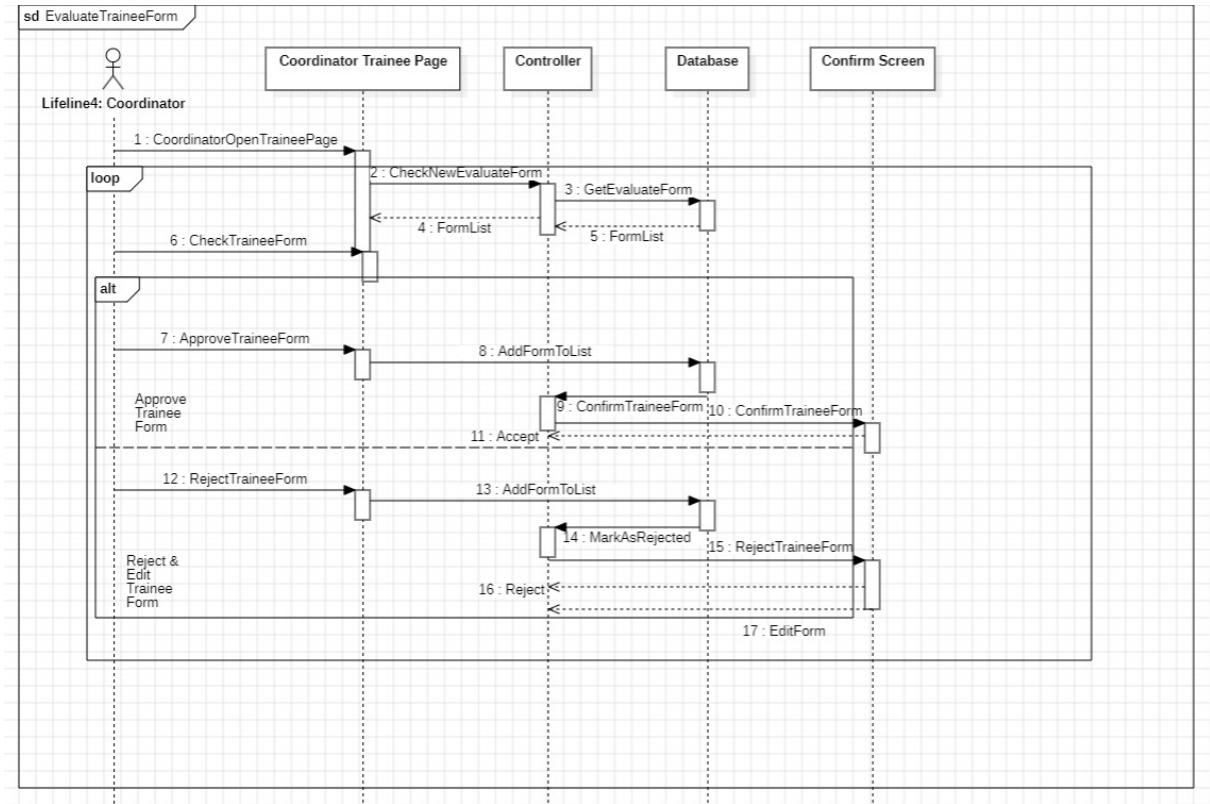


Figure 36: Sequence Diagram of Evaluate Trainee Form

Design Rationale: The sequence diagram describes how the coordinator evaluates trainee forms. The process involves the following steps:

Check New Forms: Coordinator checks for new evaluation forms. The system retrieves the list of forms from the database.

Approve Form: If a form is approved, it is added to the approved list, confirmed, and displayed on the confirmation screen.

Reject Form: If a form is rejected, it is marked as rejected in the database. Coordinator can edit the form before confirming the changes.

The system ensures proper validation and provides notifications for approval or rejection, streamlining the form evaluation process.

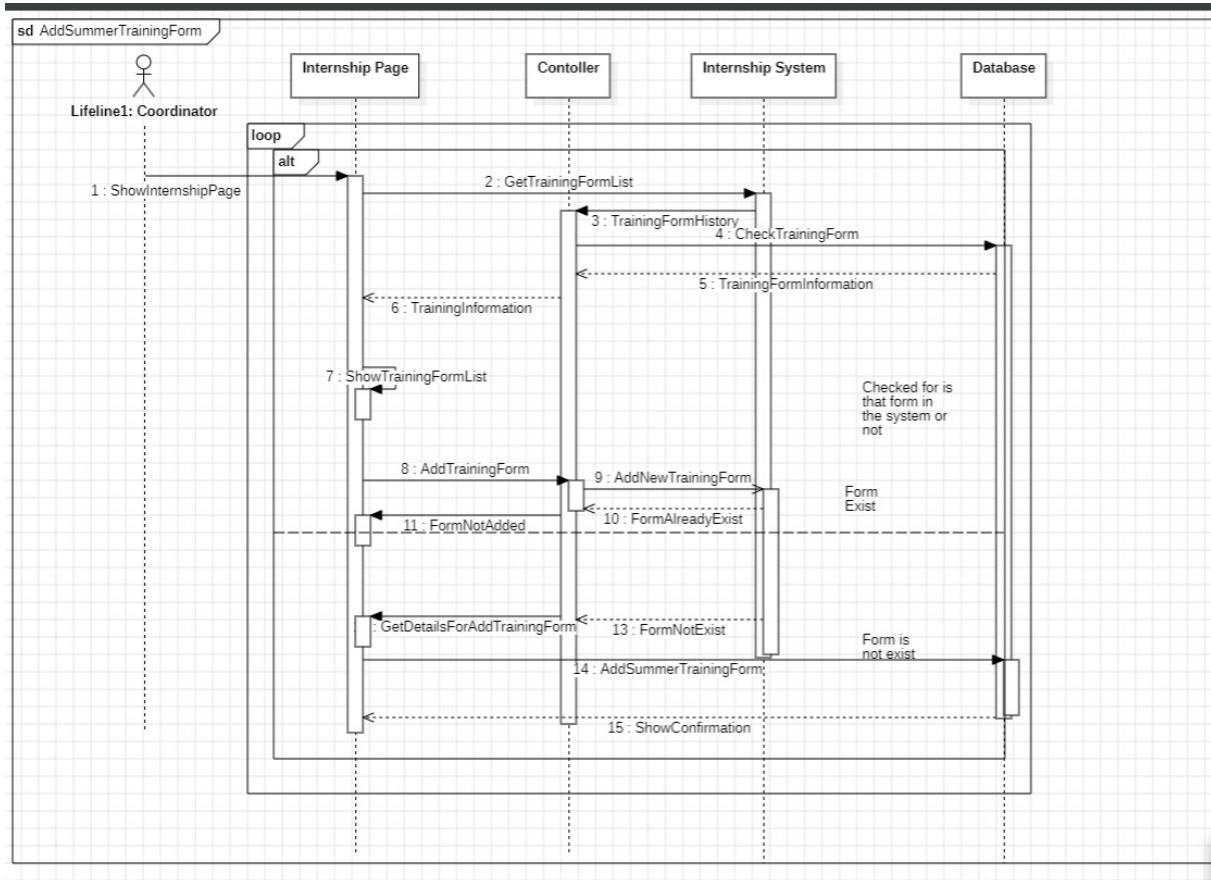


Figure 37: Sequence Diagram of Add Summer Training Form

DesignRationale:

The sequence diagram illustrates the Add Summer Training Form process initiated by coordinator. The process begins when coordinator navigates to the Internship Page and requests the Training FormList . The system then retrieves the list of training forms and checks for the existence of the requested form in the database. If the form already exists, the system displays a notification indicating the forms presence. If the form does not exist, the coordinator proceeds to add a new training form. Once the details of the new form are provided, the system validates the input and adds the Summer Training Form to the database. Finally, a confirmation message is displayed, ensuring the successful addition of the form. This process ensures that duplicate forms are avoided, and new forms are systematically added, maintaining data integrity within the system.

4.2.3 Data Flow Diagrams



Figure 38: Context Level Diagram – DFD

In this diagram, we showed the main users of our internship system and their tasks. As students, we can apply for internships, see announcements, and add or update our internship forms. Coordinators help us by adding announcements, approving our forms, and assigning instructors to guide us. Instructors check our reports and evaluate them, and they also review feedback from companies about our performance. Companies use the system to evaluate us and share their feedback. Student Affairs supports us by checking our forms and making sure our health insurance is valid. We also added the Mail Service, which sends emails to inform us about important updates in the system. You can see from Figure 38.

Design Rationale for Context Level Diagram

When designing the Context Level Diagram, we aimed to define the overall framework of the system and identify the main user groups. At this level:

Main Purpose of the System: Our goal was to streamline the internship process for all users, including students, coordinators, companies, instructors, and student affairs, ensuring a more efficient and user-friendly experience.

Simple and Inclusive Approach: We focused on showing the interactions between the system and its users in a general way, highlighting each user group's main tasks and connections with the system.

Clear Role Definitions: We ensured that user roles were clearly distinguished based on their responsibilities in the system. For example:

- Students can apply for internships, view announcements, and update their internship forms.
- Companies provide feedback and evaluate performance.
- Coordinators create announcements, approve forms, and assign instructors.

Defining Boundaries: We illustrated the system's boundaries and its interactions with external components like the Mail Service. This helped clarify how the system integrates with external processes.

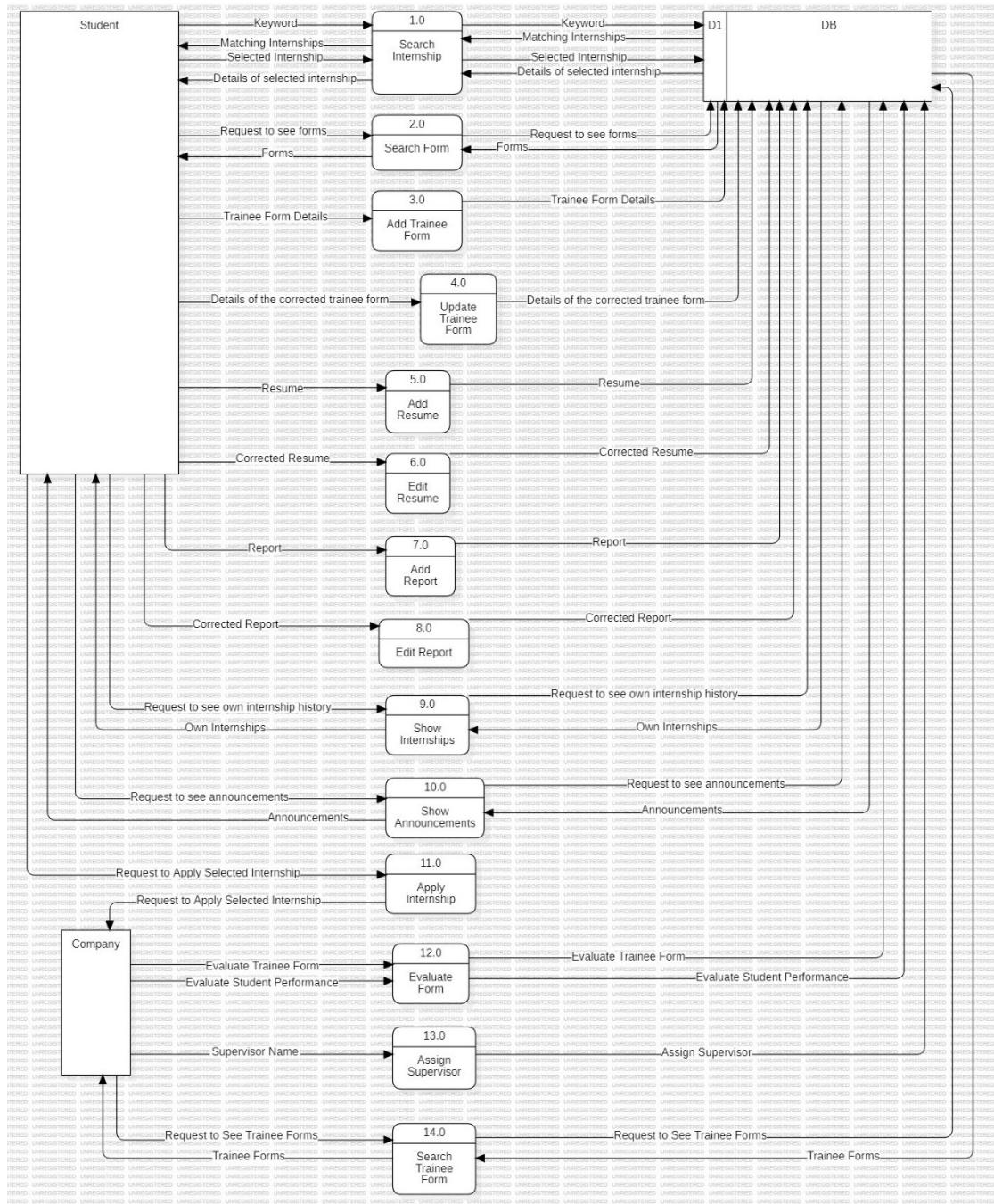


Figure 39: Level 0(1) – DFD

At this level, we explained the actions that both students and companies can do in our system. As students, we can search for available internships (1.0), add our trainee forms (3.0), or update them when needed (4.0). We can also see announcements about internships (10.0) and apply for internships that interest us (11.0). On the other side, companies evaluate us based on our internship performance and add this feedback to the system (6.0). All these actions are saved in the database, so everything stays up to date. For example, when we apply for an internship, the application is recorded in the system and can be seen by the coordinator. Similarly, companies' evaluations are saved and used by instructors to assess our overall performance. This helps create a smooth and organized internship process for both students and companies. You can see from Figure 39.

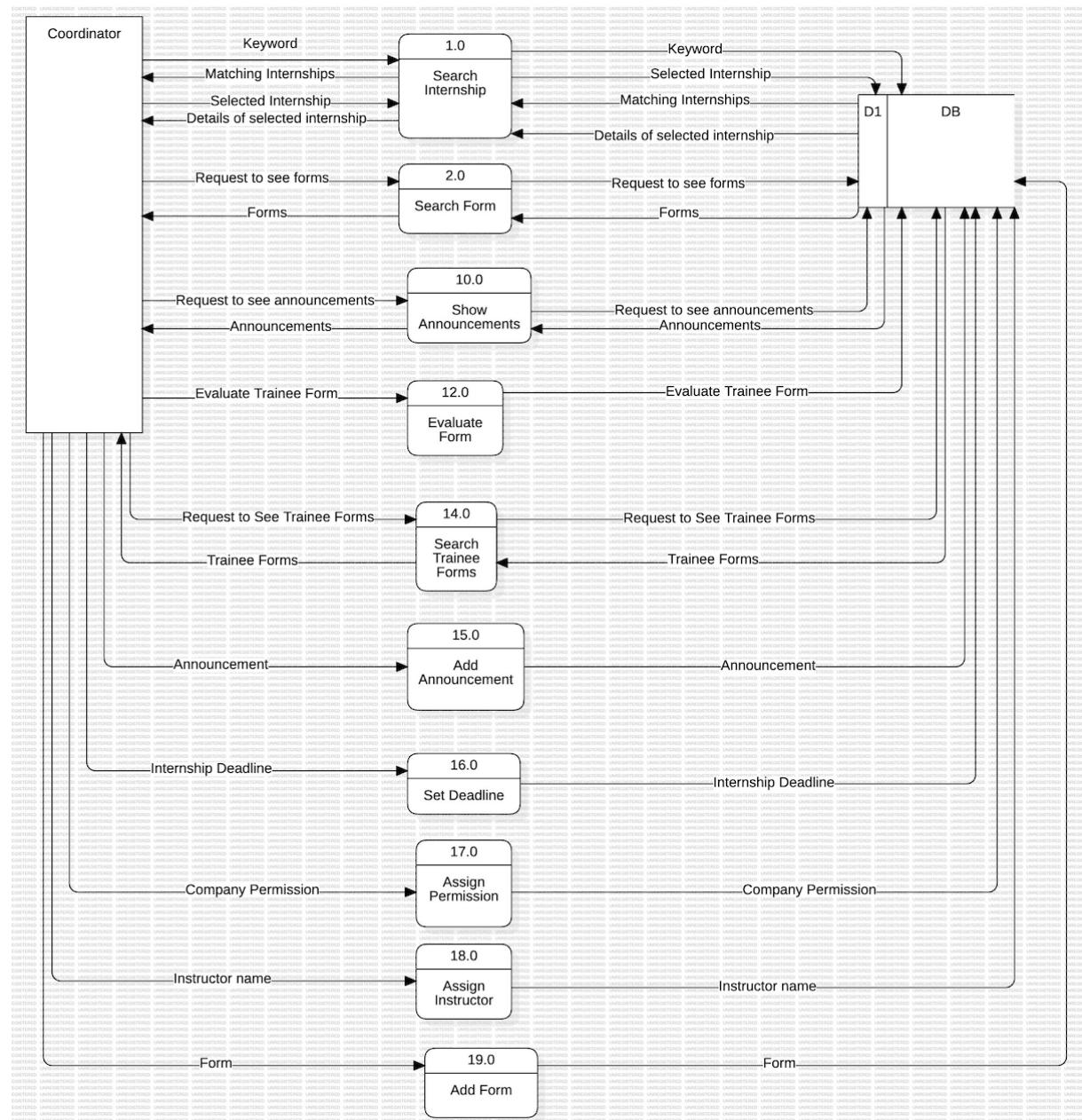


Figure 40: Level 0(2) – DFD

At this level, we focused on the coordinators' tasks. Coordinators can search for internships in the system (1.0) and add new forms to update information (19.0). They can also create announcements to share updates with students and instructors (14.0). Another task is assigning instructors to students for evaluating reports (17.0). For example, when a coordinator adds an announcement, it becomes visible to all students and instructors. These tasks help coordinators manage the internship process effectively. You can see from Figure 40.

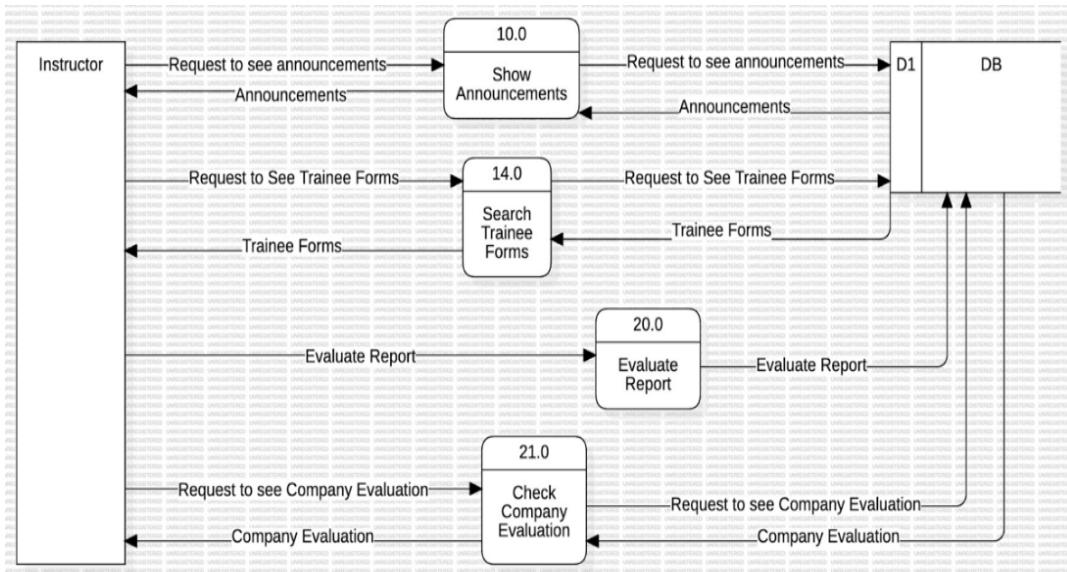


Figure 41: Level 0(3) – DFD

At this level, we explained the tasks of instructors. Instructors can review and evaluate students' internship reports (20.0). They can also check the feedback provided by companies about the interns (21.0). Additionally, they can see announcements shared by coordinators or other users in the system (10.0). For example, when an instructor evaluates a report, the feedback is saved in the system and students can see it. These tasks allow instructors to give useful feedback and assess students' performance during the internship. You can see from Figure 41.

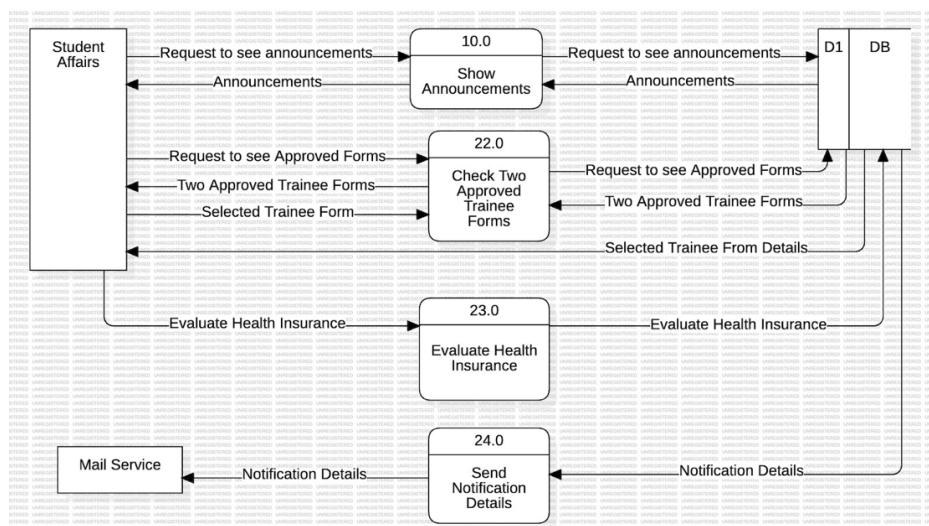


Figure 42: Level 0(4) – DFD

At this level, we described the tasks of Student Affairs. They check if the internship forms submitted by students are correct and approve them (22.0). They also evaluate if the students' health insurance meets the requirements for internships (23.0). For example, when a form is approved, this information

is saved in the system and can be viewed by both the student and the coordinator. Checking health insurance ensures that students are fully prepared for their internships. These tasks help complete all formal requirements for the internship process. You can see from Figure 42.

In our system, notifications are sent automatically after specific processes. These notifications help users stay updated. We decided not to show this process in the Data Flow Diagram (DFD) because it would make the diagram more complicated. Instead, you can refer to Figure 40. For the detailed table explaining how the notification process works.

Process	Notifications Sent To
Add Trainee Form, Update Trainee Form	Student, Coordinator, Company
Apply Internship	Student, Company
Add Student Report	Coordinator
Coordinator Announcement	Student, Instructor, Student Affairs
Coordinator sets Internship Deadline	Student
Coordinator assigns Company Permission	Company
Coordinator evaluates Trainee Form	Student
Coordinator evaluates Report	Student
Company adds Evaluation	Student
Student Affairs evaluates Health Insurance	Student

Figure 43: Notification Table

Design Rationale for Level 0 Diagram

For the Level 0 Diagram, we focused on breaking down the main tasks performed by each user group identified in the Context Level Diagram. At this level:

Breaking Down Main Tasks: We detailed the primary tasks for students, companies, coordinators, instructors, and student affairs. For instance:

- Students can view announcements (10.0), apply for internships (11.0), and update their forms (4.0).
- Companies provide feedback on internship performance (6.0).
- Coordinators create announcements (14.0) and assign instructors (17.0).

Illustrating Data Flows: We demonstrated the flow of information between different components of the system, linking user actions to their corresponding processes. This clarified how the database updates and how users stay informed about one another's actions.

Maintaining Consistency: We emphasized that every action in the system has a corresponding result saved in the database. For example:

- A student's application is visible to the coordinator for further action.
- Feedback provided by companies is accessible to instructors for evaluation purposes

Managing Complexity: We chose not to include detailed processes, like notifications, in the diagram to avoid overcomplication. Instead, we explained these processes in other sections of the document.

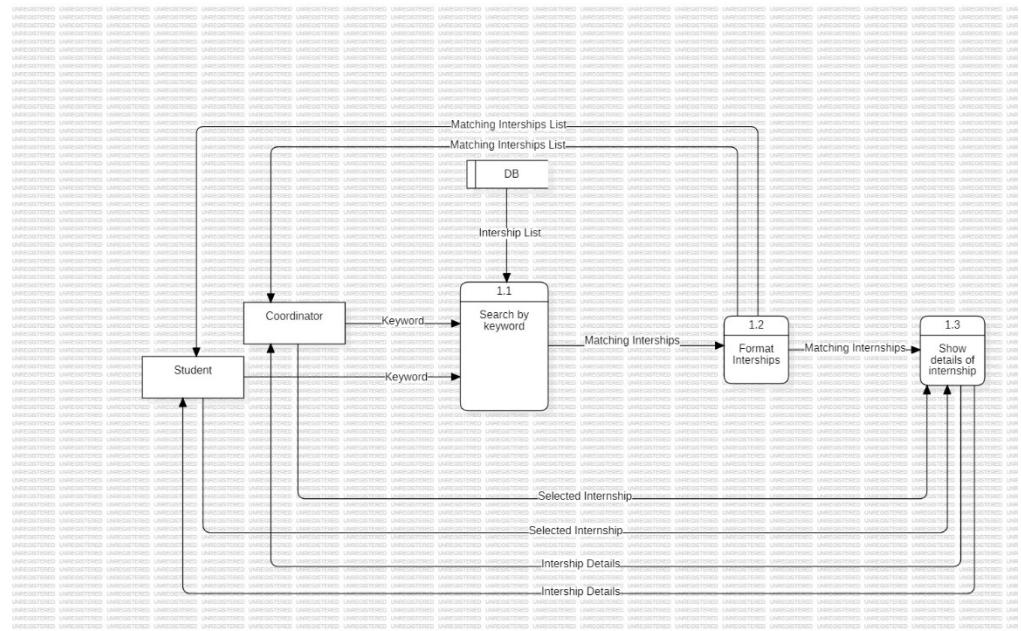


Figure 44: Level 1(1) – DFD

In the diagram in Figure 44, we have described the process by which students and coordinators can search for internships using keywords. The system fetches relevant internships from the database based on the input keyword. These internships are then formatted into a list of matches, which is displayed to the users. From this list, users can select an internship to view its detailed information.

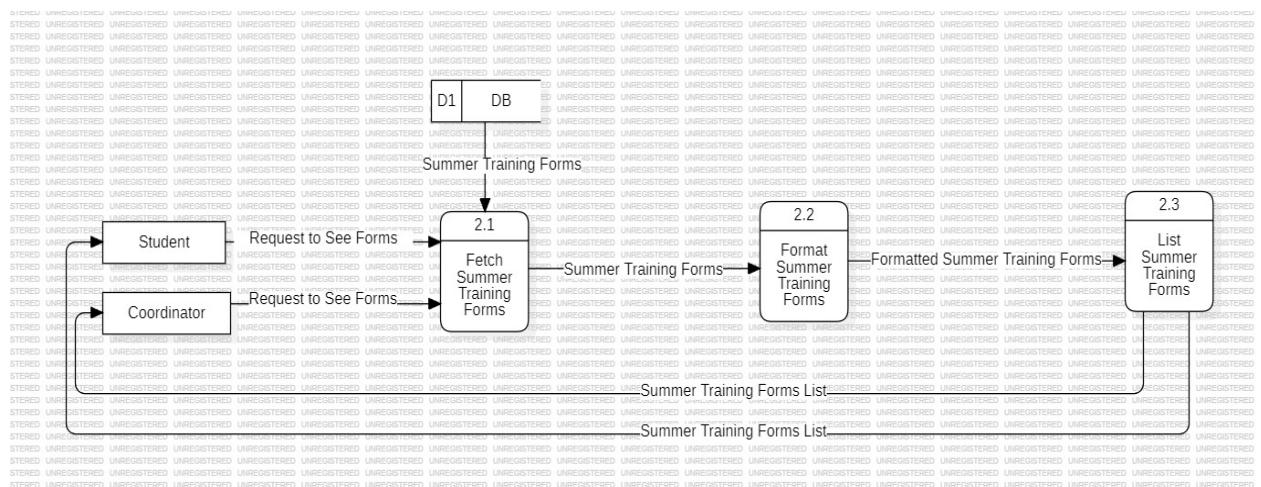


Figure 45: Level 1(2) – DFD

In this diagram, we have described how students and coordinators can view the summer training forms. The Coordinator or Student sends a request to fetch forms. The system retrieves the forms from the database. The form objects are formatted through the Format Summer Training Form process to make it suitable for display. The formatted forms are listed and sent to the Coordinator or Student. You can see from Figure 45.

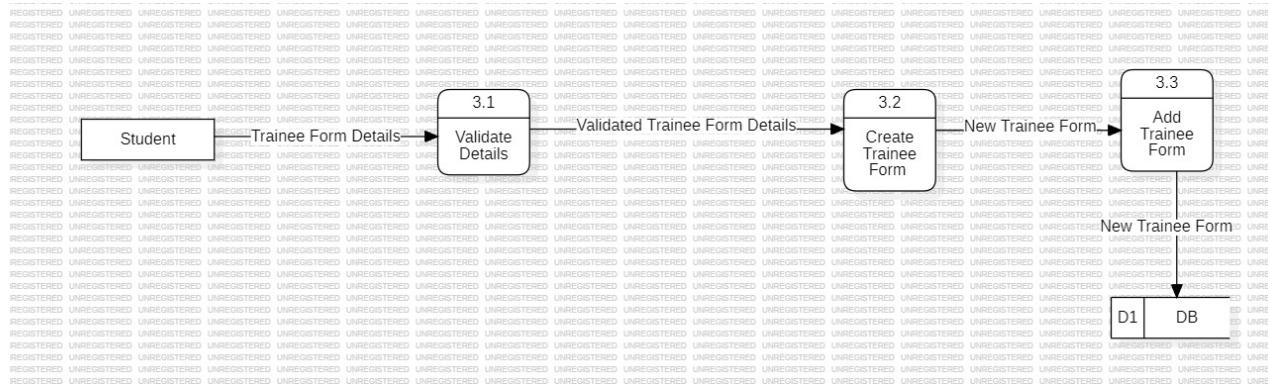


Figure 46: Level 1(3) – DFD

In this diagram in Figure 46, Student provides Trainee Form details, which are passed to the Validate Details process. This process ensures the input is valid before proceeding. The validated data is then sent to the Create Trainee Form process, which generates a trainee form object. The form is then passed to the Add Trainee Form process, which saves the information into the database.

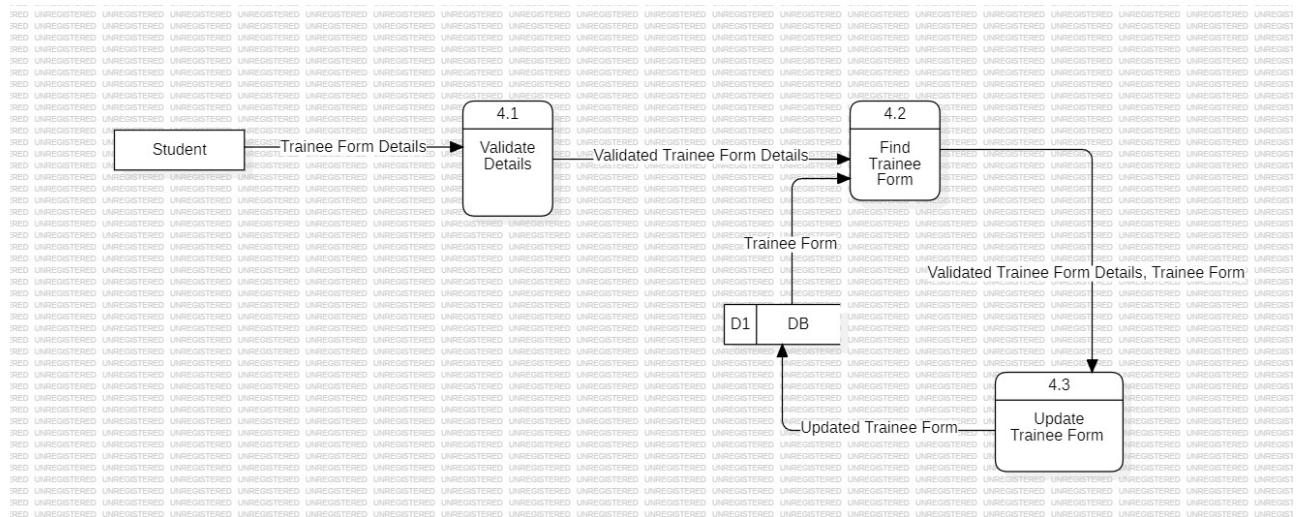


Figure 47: Level 1(4) – DFD

In this diagram, we have described how students can update their trainee forms. The process begins when the student provides their details to be validated via Validate Details process. Once validated, the system proceeds to Find Trainee Form process, which retrieves the trainee form to be updated from the database. Finally, Update Trainee Form process modifies the trainee form based on the updated information, and the updated form is saved back into the database. You can see from Figure 47.

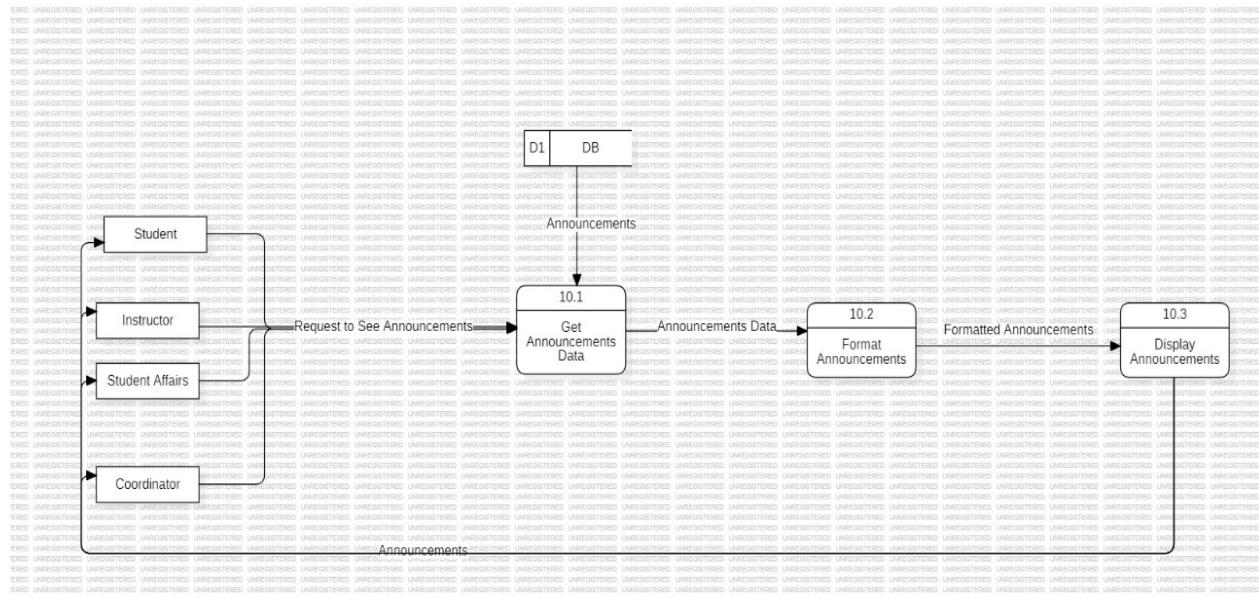


Figure 48: Level 1(10) – DFD

In this diagram, we have described how students, instructors, student affairs and coordinators can view the announcements. First, they interact with the system and Get Announcements Data process runs, which retrieves data from the database containing announcements. This data is then formatted through Format Announcements process, where the raw data is organized into a structured format suitable for display. Finally, the formatted internship details are presented back to the student through the Display Announcements process. You can see from figure 48.

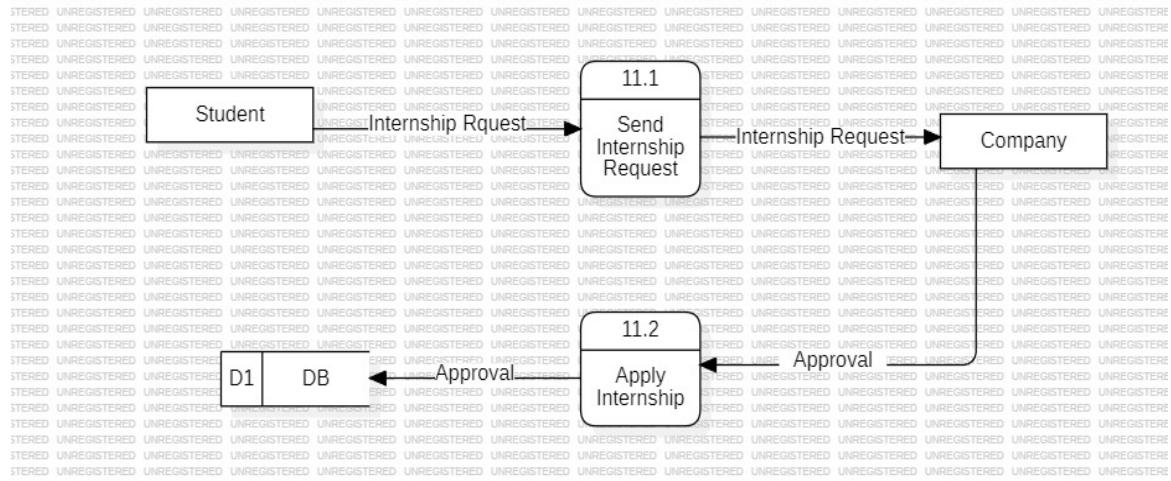


Figure 49: Level 1(11) – DFD

In this diagram in Figure 49, we have described how students can apply for an internship and how the company approve internships via the internship system. Student initiates the process by sending an internship request to a Company. The company processes this request and applies for the internship, storing the relevant data in the database.

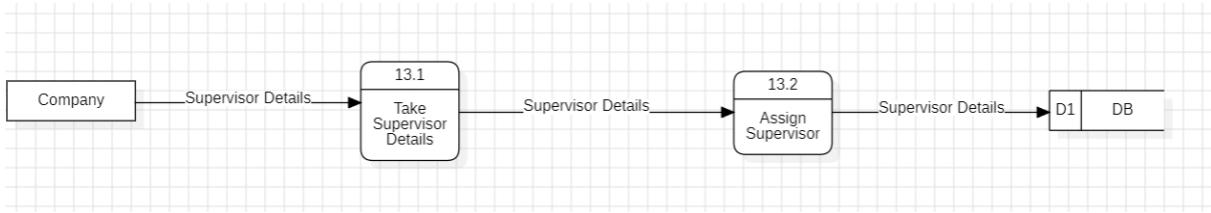


Figure 50: Level 1(13) – DFD

In this diagram, we have described how companies can fill supervisor details and assign a supervisor for an internship. The company provides the supervisor. Then this information flows to the Assign Supervisor process, where the supervisor details are being directed to the database. You can see from Figure 50.

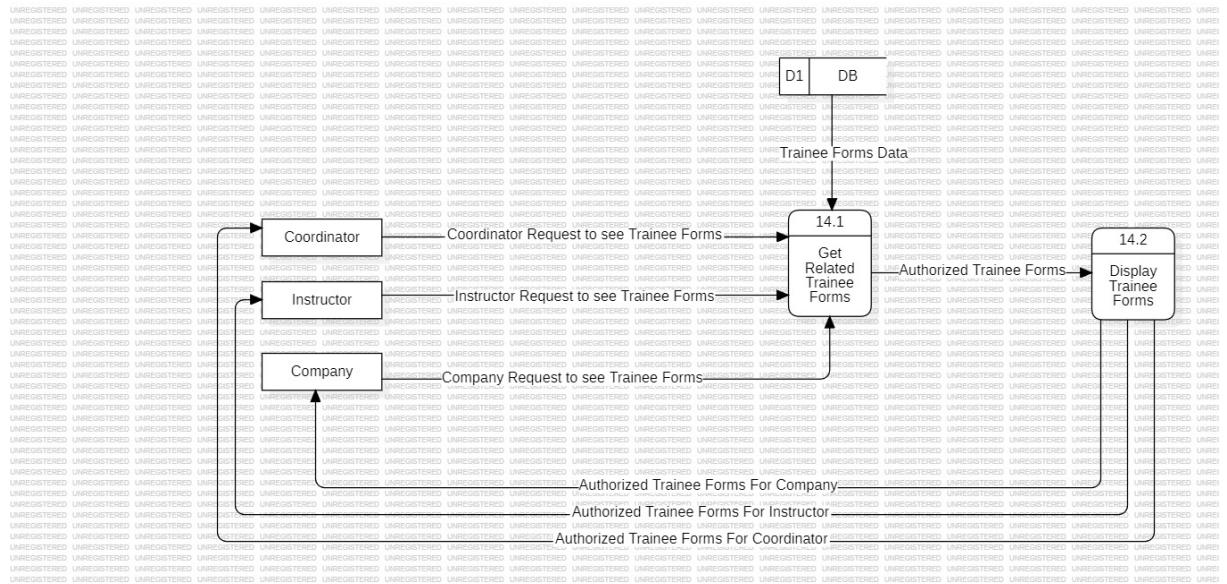


Figure 51: Level 1(14) – DFD

In this diagram in Figure 51, we have described how companies, instructors and coordinators can view trainee forms that they have access to. First, they send a request to reach their Trainee Forms. These forms are fetched by Get Related Trainee Forms process. Then these forms are directed to display trainee forms that displays the Authorized Trainee Forms for a specific company, instructor or coordinator.

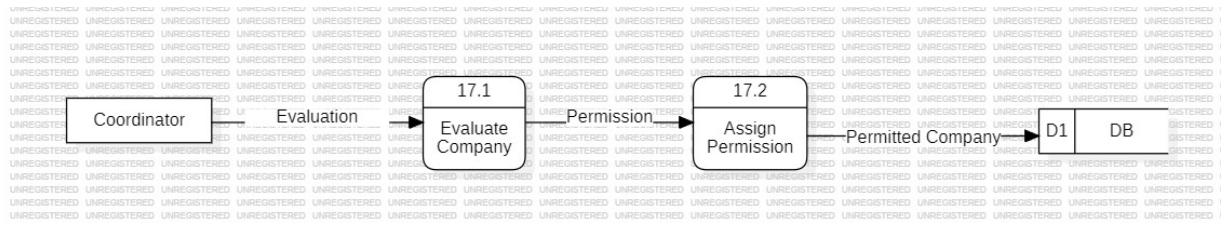


Figure 52: Level 1(17) – DFD

In this diagram in Figure 52, we have described how coordinators can evaluate and approve a company into the internship system. When coordinator use the system Evaluate Company process lets coordinator to evaluate the company. This process takes the evaluation input and directs permission to Assign Permission process. This process saves the company details as a permitted company in the database.

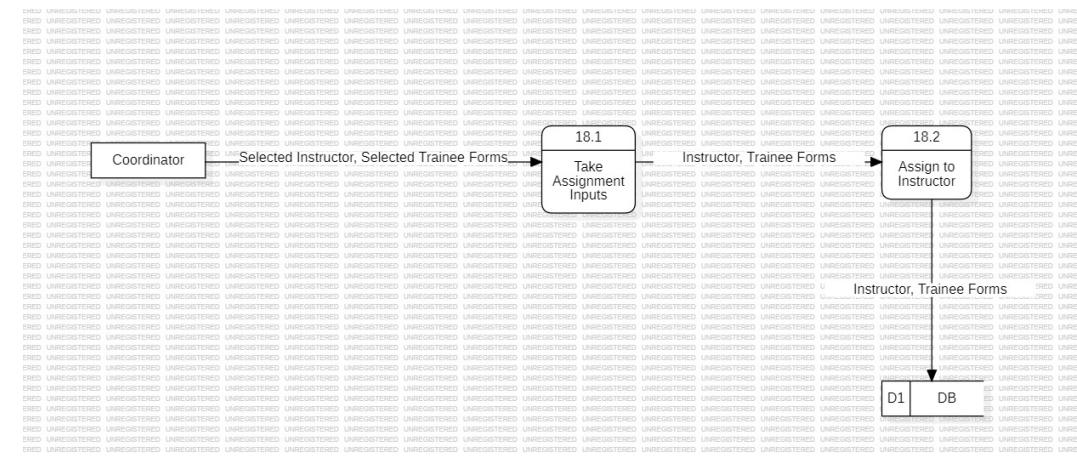


Figure 53: Level 1(18) – DFD

This diagram in Figure 53, shows how we have described how coordinators can assign instructors to trainee forms. Coordinator selects an instructor for a trainee form and Take Assignment Inputs process runs. Then Selected Instructor and Trainee Form being conveyed to the Assign to Instructor process to be saved in the database.

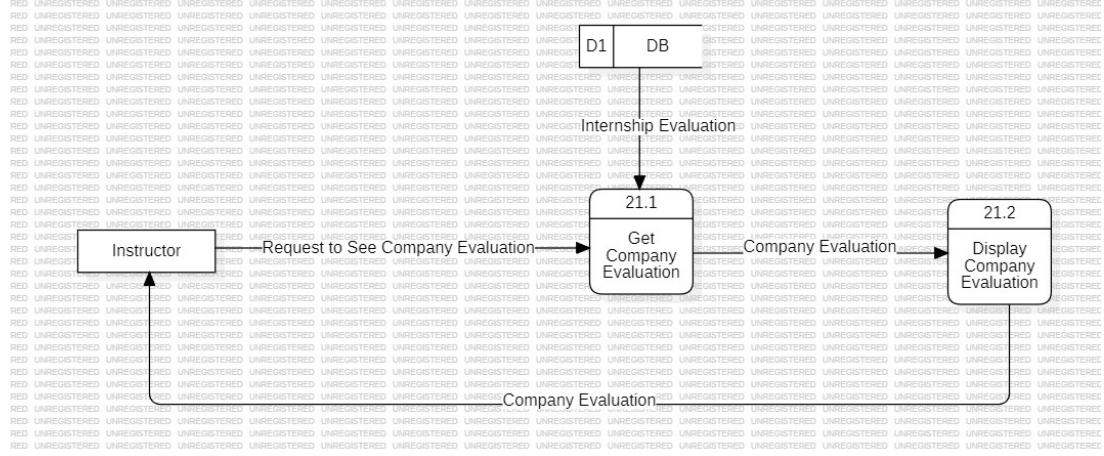


Figure 54: Level 1(21) – DFD

In this diagram, we have described how instructors can view company evaluations. The Instructor interacts with a system to obtain and view company evaluations. The process begins when the Instructor initiates a request to Get Company Evaluation, which involves fetching data from the database. Once the data is retrieved, it is passed to the Display Company Evaluation process, which formats and presents the evaluation to the Instructor. You can see from Figure 54.

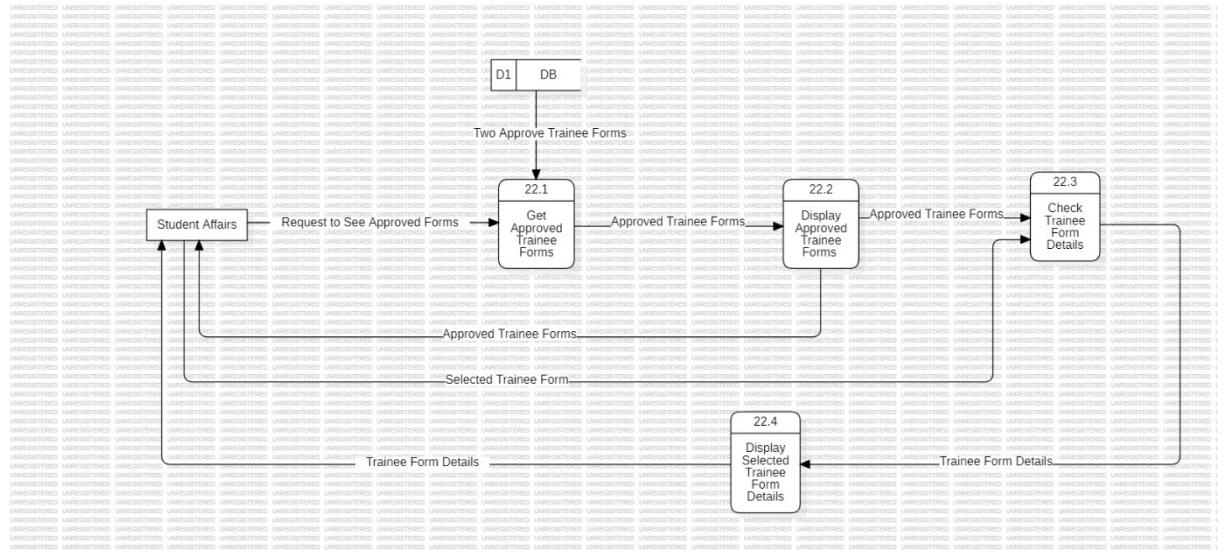


Figure 55: Level 1(22) – DFD

In the diagram in Figure 55, we have described how student affairs can view the approved trainee forms and select a trainee form to see its details. Student Affairs initiates the process by requesting forms from the Get Approved Trainee Forms process, which retrieves data from the Trainee Forms the database. The system then proceeds to Display Approved Trainee Forms, displaying approved trainee forms. Student Affairs can also select a trainee form to run Check Trainee Form Details process. The system concludes by sending Display Selected Trainee Form Details back to Student Affairs, providing the necessary details for review.

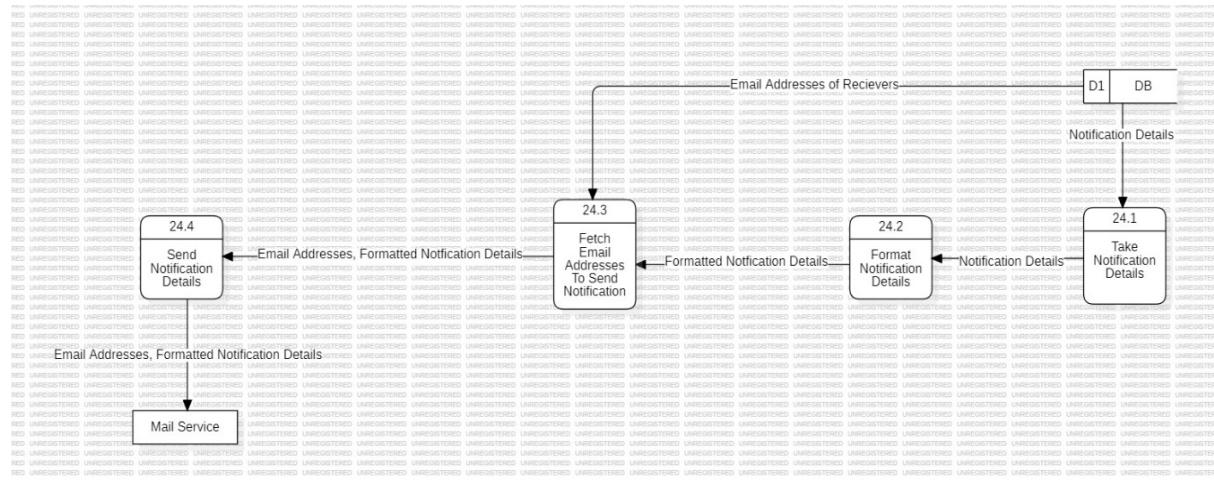


Figure 56: Level 1(24) – DFD

As shown in Figure 56, in this diagram, we have described how notifications are created and sent through the mail service with the details provided. When sending a notification, notification details is first retrieved from the database. These details are processed by the Take Notification Details process, which passes them to the Format Notification Details process for formatting. Once formatted, the Fetch Email Addresses to Send Email process retrieves email addresses to send email from the database. The formatted notification details, along with the fetched email addresses, are then passed to the Send Notification Details process. Finally, the notification details are forwarded to the Mail Service which is responsible for delivering the emails. This data flow diagram highlights how notifications are managed and sent systematically.

Design Rationale for Level 1 Diagram

For the Level 1 Diagram, we focused on breaking down the process in more detail than the Level 0. This level of DFD helps developers understand how the system works internally and provides a foundation for implementation. At this level:

Detailing Processes: We broke down the main process into smaller, manageable sub-processes.

Identifying Sub-Processes: We decomposed the high-level processes into its sub-processes.

Maintaining Context: We ensured it aligns with the system goals and the Level 0 DFD, providing consistency.

Facilitate Understanding: The diagram serves as a bridge for system analysis by providing more functions while avoiding overwhelming detail.

4.3 Development View

4.3.1 Component and Deployment Diagram

The Internship System stores the data in a database and accesses the data from there. Forms and announcements can be added to the internship system. There are also 5 different types of users to use the system: coordinator, student, company, instructor and student affairs. All of these users can be contacted via e-mail through another system, Mail Provider. Trainees can manage their reports, use Trainee Information Forms and upload and manage their Resume. Companies can evaluate students using Performance Evaluation Forms in the system.

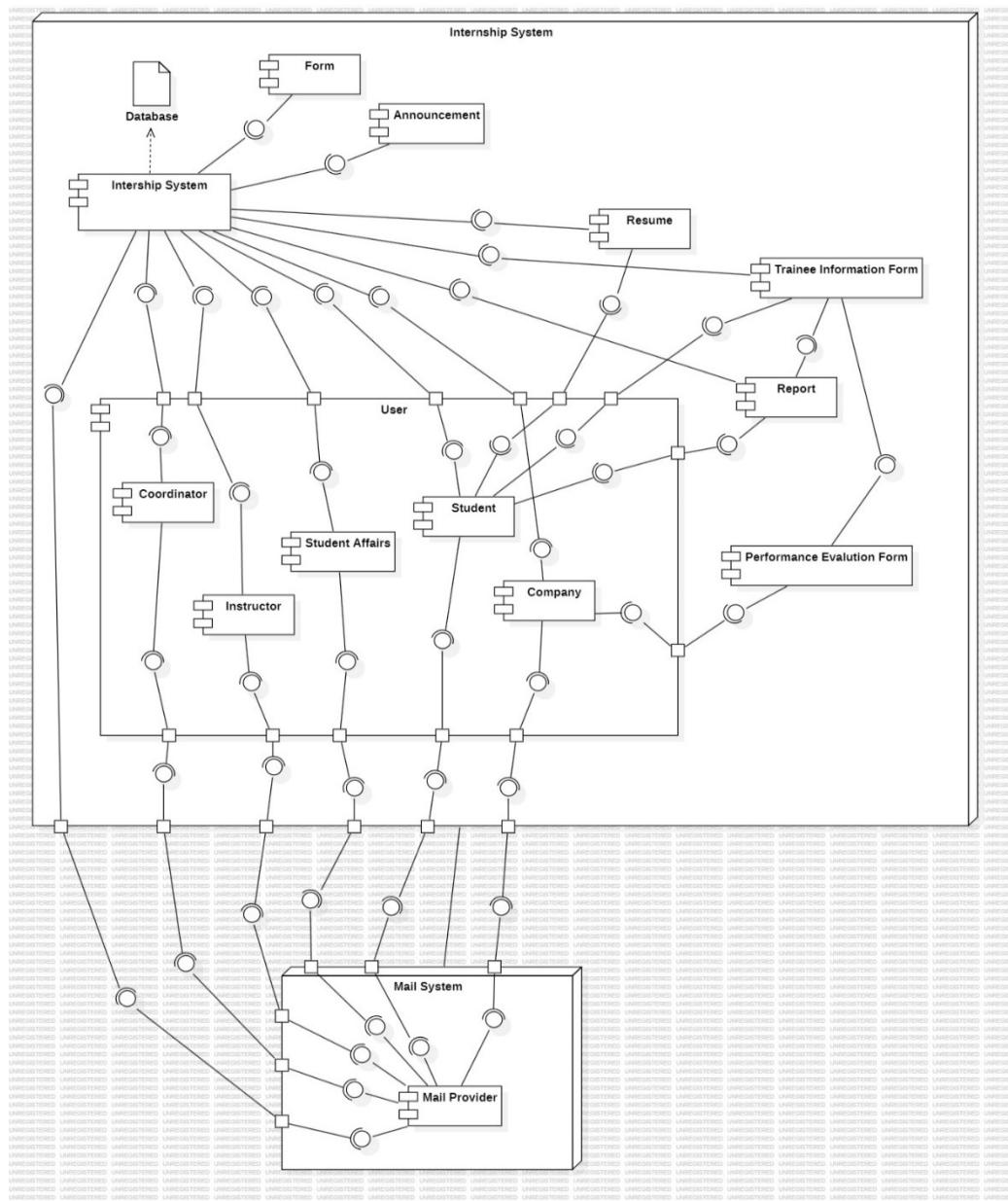


Figure 57: Component and Deployment Diagram

5 Project Scheduling

5.1 Milestones and Tasks

Complete SRS - Introduction (Week 2-6)

- Task 1: Purpose, Scope
- Responsible: Umutcan, Ali
- Task 2: Related Work, Product Overview, Definitions
- Responsible: Efekan, Mert
- Task Dependency: Task 1

Complete SRS - Specific Requirements (Week 2-6)

- Task 1: External Interfaces, Functions, Usability Requirements
- Responsible: Ali, Mert
- Task 2: Performance Requirements, Logical Database Requirements
- Responsible: Umutcan, Efekan
- Task 3: Software System Attributes, Supporting Information
- Responsible: Umutcan, Mert
- Task Dependency: Task 1

Complete SRS - Software Estimation (Week 2-6)

- Task 1: Cocomo
- Responsible: Mert

Complete SRS - Appendices (Week 2-6)

- Task 1: Acronyms and Abbreviations
- Responsible: Efekan
- Task 2: Sprint 1 Overview
- Responsible: Umutcan, Ali
- Task Dependency: Task 1(Software Estimation)

Complete SDD - Introduction (Week 6-10)

- Task 1: Summary, Identified Stakeholders, and Design Concerns
- Responsible: Efekan, Mert
- Task Dependency: Task 2(Appendices)(It means it should be started after SRS is over.)

Complete SDD - Architectural Views (Week 6-10)

- Task 1: Logical View, Class Diagram, Process View, Activity Diagram
- Responsible: Ali, Umutcan
- Task 2: Sequence Diagrams, Data Flow Diagrams, Development View, Physical View
- Responsible: Efekan, Ali
- Task Dependency: Task 1(Architectural Views)

Complete SDD - Appendices (Week 6-10)

- Task 1: Project Scheduling
- Responsible: Mert, Ali
- Task Dependency: Task 1(Architectural Views)
- Task 2: Sprint 2 Overview
- Responsible: Umutcan, Efekan
- Task Dependency: Task 1

Practical Development - Front-End Development (Week 10-13)

- Task 1: Angular (HTML, CSS)
- Responsible: Ali
- Task 2: Angular (TypeScript)
- Responsible: Mert
- Task Dependency: Task 1

Practical Development - Database Operations (Week 10-13)

- Task 1: Creating DB, Tables
- Responsible: Efekan
- Task 2: Adding Data
- Responsible: Umutcan
- Task Dependency: Task 1

Practical Development - Back-End Development (Week 10-13)

- Task 1: Develop Java
- Responsible: Umutcan, Ali

Practical Development - Research (Week 10-13)

- Task 1: Research Similar Projects
- Responsible: Mert

Final Report Preparation (Week 13-14)

- Task 1: Edit SRS
- Responsible: Mert, Ali
- Task 2: Edit SDD
- Responsible: Umutcan, Efekan

Final Presentation Preparations (Week 14-16)

- Task 1: Preparation of Slides and Video
- Responsible: Umutcan, Ali, Efekan, Mert
- Task Dependency: Task 2(Final Report Preparation)
- Task 2: Preparation for the Presentation
- Responsible: Umutcan, Ali, Efekan, Mert
- Task Dependency: Task 2(Final Report Preparation)

5.2 Gantt Chart

You can examine the Gantt chart from Figure 58 . Since it is large, if you want to examine it in detail, you can see it divided into two in Figures 59 and 60 . The meanings of the tasks written in the Gantt chart are explained in 5.1.1.

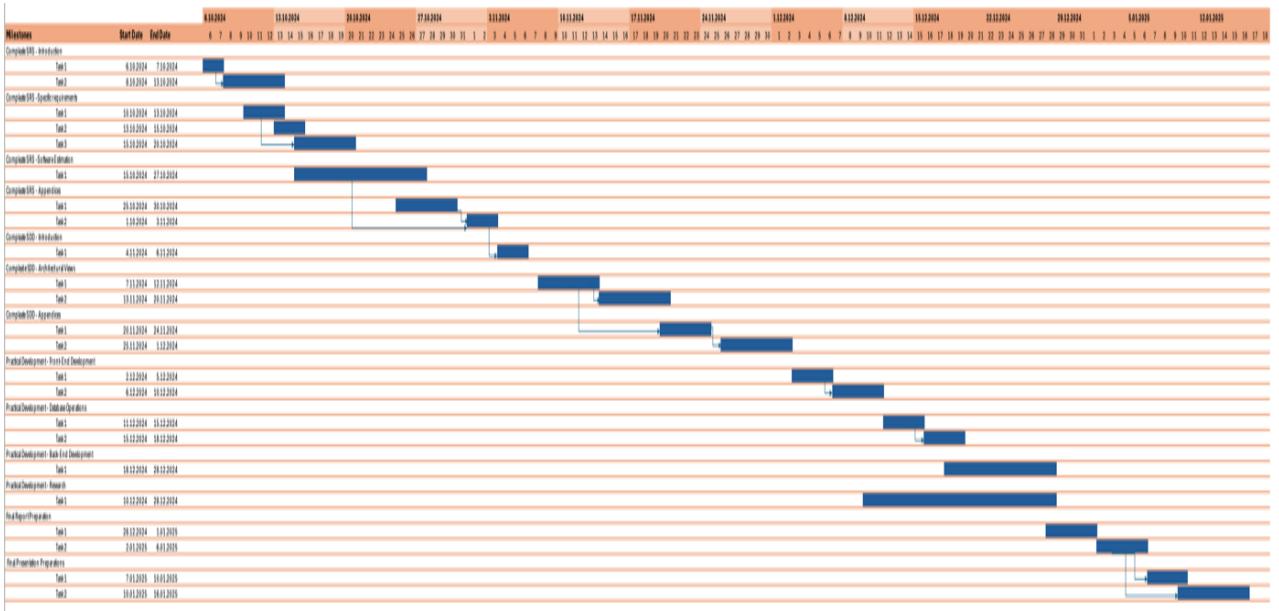


Figure 58: Gantt Chart-general

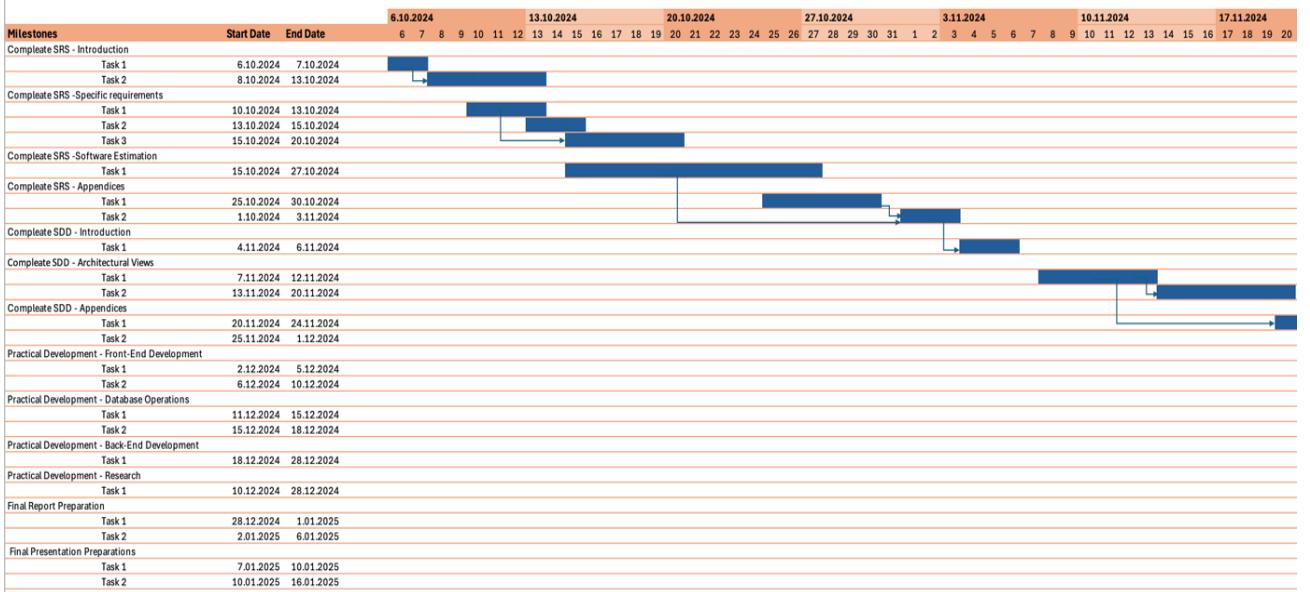


Figure 59: Gantt Chart-general

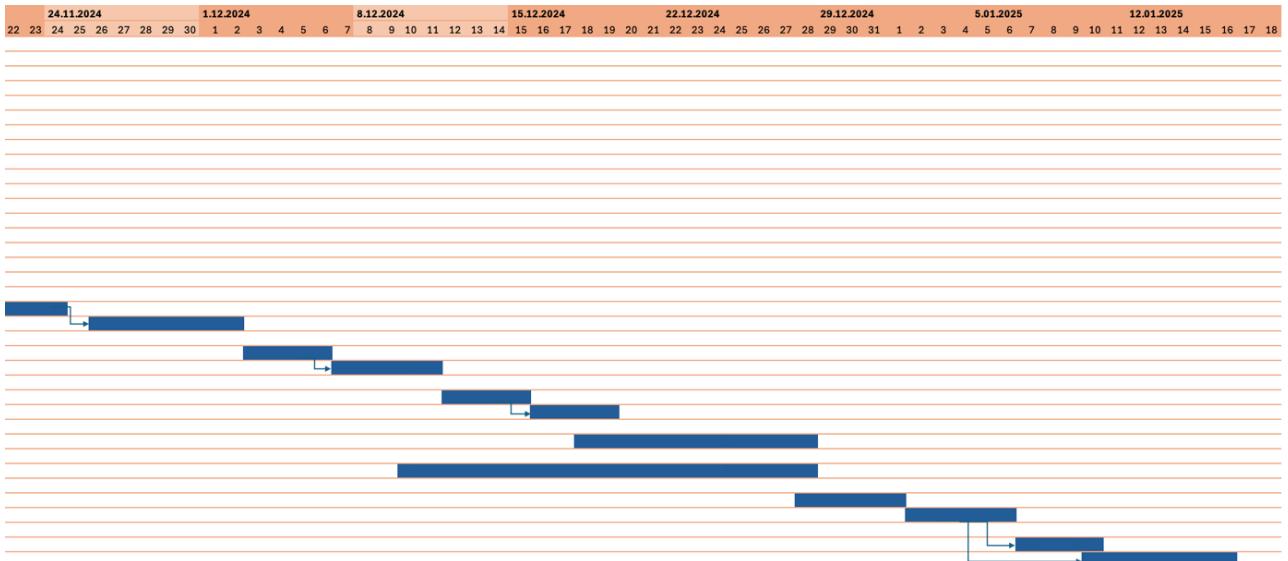


Figure 6o: Gantt Chart-general

6 Conclusion

Over the course of 14 weeks, we worked in a summer internship application project for three sprints. This provided us with hands-on experience in web development environments while working collaboratively as a team.

During this process, we focused on specifying software requirements, designing use cases and system functions, defining logical database requirements, performing software estimation, and creating detailed software design description that included a class diagram, sequence diagrams, activity diagrams, data flow diagrams, a component diagram, and a deployment diagram. Finally, we implemented a software prototype using PostgreSQL, Java Spring Boot, and TypeScript with Angular as the primary technologies.

While developing the system, we followed our supervisor İdil Candan's instructions to improve the existing METU NCC Internship System. In the new system, an enhanced email notification feature has been introduced for example: instructors are notified via email when a student uploads a report, and coordinators receive an email when a student completes the trainee information form, which was not available in the existing system. The website interface and actions were also simplified to reduce the system's learning curve and decrease complexity while retaining its full functionality.

We acquired foundational knowledge and practical experience in Angular, Java Spring Boot, and PostgreSQL starting from the basics. Through this process, we developed a good understanding of their working mechanisms and methodologies. Additionally, we enhanced our problem-solving and software development skills, particularly in software design, coding, testing database design, and website design areas.

In addition to enhancing our technical knowledge through this project, we also developed valuable skills in team communication and collaboration, which will be beneficial for future projects. Finally, we would like to express our gratitude to our supervisor, İdil Candan, and our course instructor, Enver Ever, for their invaluable support and feedback throughout this process.

7 References

1. Budić, H., Dužević, M., & Pecina, M. (2021). Developing an internship management system. ENTRENOVA - ENTerprise REsearch InNOVAtion Conference Proceedings, 7(1), 71-77. <https://hrcak.srce.hr/ojs/index.php/entrenova/article/view/13437>
2. Ismaili, I. (2018). Development of a web-based internship management system: A case study. South East European University Repository. https://repository.seeu.edu.mk/sites/thesis/ThesisSharedDocs/MA_128315.pdf
3. Hoxmeier, J. A., & DiCesare, C. (2000). System response time and user satisfaction: An experimental study of browser-based applications. AMCIS 2000 Proceedings, 347. Retrieved from <https://aisel.aisnet.org/amcis2000/347>
4. Oreoluwa, F. A. (2021). Scalability testing: Ensuring system performance for growing user bases. Digital Performance Journal. https://www.researchgate.net/publication/384826255_New_Model_for_Defining_and_Implementing_Performance_Tests5
5. Ramotion. (2023). High availability systems: Designing for 99.9% uptime. <https://www.ramotion.com/blog/scalable-web-applications/>
6. Paul, J. (n.d.). Efficient data processing in web systems: Design and implementation. Tech Digest. <https://repository.seeu.edu.mk>
7. SoftSages. (2024). Notification latency: Balancing timeliness and efficiency. Performance Matters. <https://www.softsages.com/blogs/scalable-and-high-performance-web-applications>
8. What is a UML Diagram? | Different Types and Benefits | Miro. (n.d.). <https://miro.com/diagramming/what-is-a-uml-diagram>
9. GeeksforGeeks. (2024, October 10). Sequence diagrams Unified Modeling Language (UML). GeeksforGeeks. <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams>

10. *What is a Data Flow Diagram.* (n.d.). Lucidchart. <https://www.lucidchart.com/pages/data-flow-diagram>
11. Wikipedia contributors. (2023, October 18). *Glossary of Unified Modeling Language terms*.
https://en.wikipedia.org/wiki/Glossary_of_Unified_Modeling_Language_terms
12. Bucanek, J. (2009). Model-view-controller pattern. *Learn Objective-C for Java Developers*, 353-402. https://link.springer.com/chapter/10.1007/978-1-4302-2370-2_20
13. Crudu, A., & MoldStud. (2024). The importance of load testing for system reliability. Mold Tech Journal. <https://moldstud.com/articles/p-importance-of-load-testing-for-ensuring-system-reliability-under-stress>
14. Nielsen, Jakob. *Usability engineering*. Morgan Kaufmann, 1994.
https://scholar.google.com/citations?view_op=view_citation&hl=tr&user=y5uL3wUAAAJ&citation_for_view=y5uL3wUAAAJ:WC9gN4BGCRCc
15. Brooke, J. "SUS: A quick and dirty usability scale." *Usability Evaluation in Industry* (1996).
<https://rickvanderzwet.nl/trac/personal/export/104/liacs/hci/docs/SUS-questionnaire.pdf>

8 Appendices

8.1 Acronyms and abbreviations

N/A

8.2 Glossary

N/A