Software Design Description

for

A Software for Summer Internship

Umutcan Celik 2526200

Mert Damburacı 2453108

Ali Fırat Ozdemir 2453447

Middle East Technical University Northern Cyprus Campus

Computer Engineering

Idil Candan

29.11.2024

Contents

1 Introduction

1.1 Summary

The Internship Management System is designed to simplify and automate the management of internship processes for universities. The system serves multiple stakeholders, including students, coordinator, instructors, companies, and student affairs, providing tailored functionalities to meet their specific needs. Students use the system to browse and apply for internships, submit required documents such as resumes and reports, and track their progress. Coordinators oversee the entire internship process, including managing forms, assigning instructors, and approving internships. Instructors evaluate student reports and provide feedback, while companies have exclusive access to view and manage student applications and evaluate their performance. Student Affairs is responsible for validating health insurance records and ensuring compliance with university policies. With built-in features for notifications, health insurance evaluation, and email alerts, the system ensures smooth communication and timely updates. By centralizing all internship-related activities in an efficient web-based platform, the system enhances productivity and minimizes manual effort for all involved parties.

1.2 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.

2 Glossary

Use Case Diagram: A visual representation of the functionalities of a system, showing the interactions between users (actors) and the system.[1]

Sequence Diagram: A type of UML diagram that represents the flow of messages between objects in a system for a specific process or interaction.[2]

Controller: A part of the software architecture that handles the interaction between the user interface and the backend systems, ensuring smooth operation.

Data Flow Diagram (DFD): A graphical representation of the flow of data within a system, illustrating how data is processed, stored, and moved between entities and processes.[3]

Class Diagram: A UML diagram that shows the static structure of a system, representing classes, their attributes, methods, and relationships such as associations, inheritances, and dependencies.[4]

Component Diagram: A UML diagram that illustrates the physical components (such as software modules, libraries, and databases) of a system and their relationships. It highlights how different parts of the system interact and depend on one another.[5]

3 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]

[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.

3.1 Logical View

3.1.1 Class Diagram

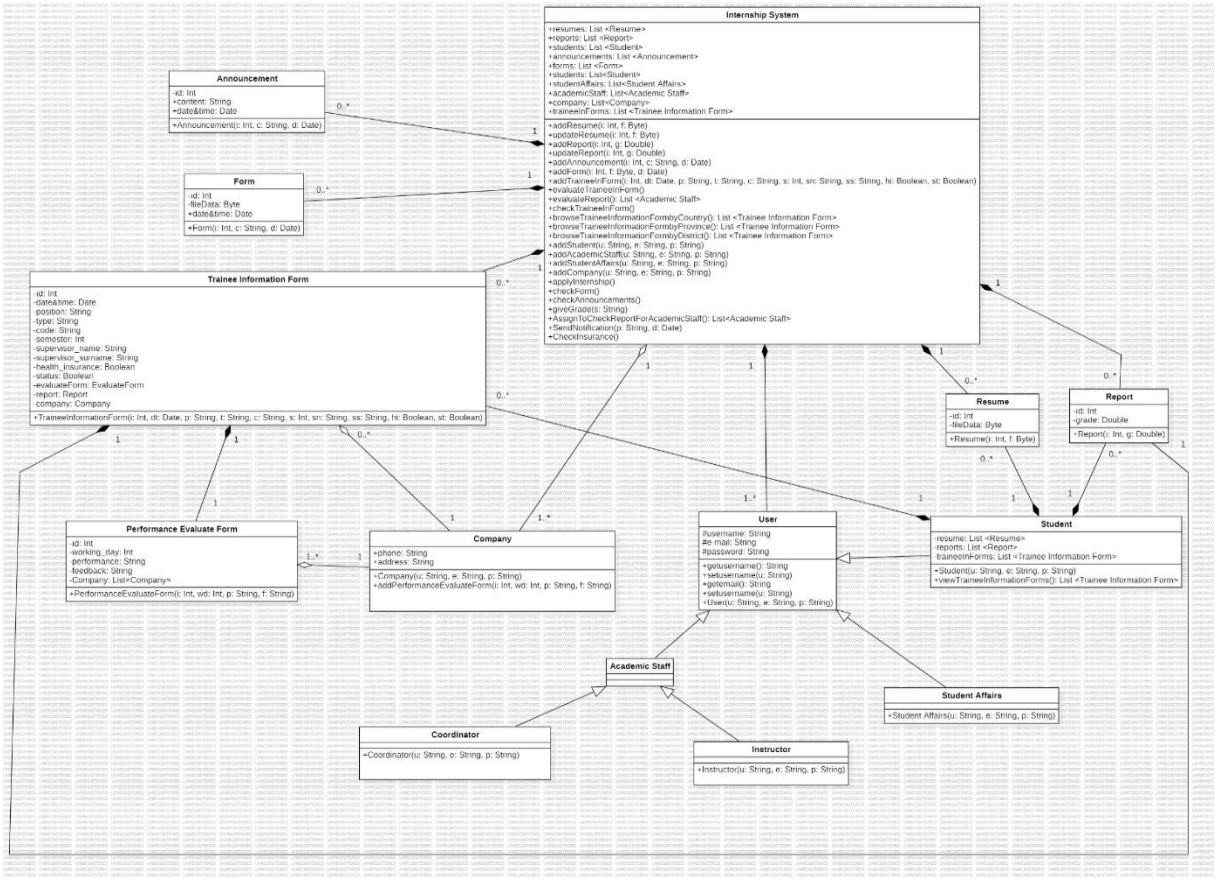

Figure 1: Class Diagram

Design Rationale:
1-Internship System:
Attributes:
Stores the overall system data, such as lists of resumes, reports, students, announcements, academic staff, and trainee information forms.
Methods:
Functions to manage and retrieve data for operations like adding forms, evaluating reports, sending announcements, and checking internships.

2-Trainee Information Form:
Attributes:
Contains details of an internship, such as date, deadline, supervisor name, semester, type of internship, and associated company. It also includes flags for form status and evaluation.
Methods:
Functions to create and update trainee information forms, evaluate forms, and generate reports for both students and companies.

3-Student:
Attributes:
Stores student-specific details like their resumes, reports, trainee forms, and user credentials.
Methods:
Functions to add resumes, view reports, and submit trainee information forms.

4-Company:
Attributes:
Includes company details such as name, address, phone, and performance evaluation forms related to interns.
Methods:
Functions to provide feedback, evaluate interns, and update company information.

5-Performance Evaluation Form:
Attributes:
Contains evaluation criteria such as working days, performance scores, and feedback from companies.
Methods:
Functions to add, view, and update evaluation results and link them with student records.

6-Announcement:
Attributes:
Holds announcement-specific information like ID, content, and creation date.
Methods:
Functions to create, update, and delete announcements for students and academic staff.

7-Form:
Attributes:
Generic attributes for forms, including form ID and form date.
Methods:
Functions to set and retrieve form details, ensuring reusability for various form types.

8-Resume:
Attributes:
Stores resume data as byte information uploaded files by students.
Methods:
Functions to update resumes.

9-Report:
Attributes:
Includes report details like report ID, grade, and additional file data.
Methods:
Functions to submit, update, and evaluate reports linked to internship processes.

10-User:
Attributes:
Stores user-specific details, such as username, password, email, and phone number.
Methods:
Functions to log in, update user information, and manage user credentials securely.

11-Coordinator:
Attributes:
Stores coordinator details like name, email, and assigned tasks.
Methods:
Functions to approve or reject forms, manage announcements, and monitor student progress.

12-Instructor:
Attributes:
Stores instructor details like name and contact information.
Methods:
Functions to evaluate student performance and provide feedback on reports.

13-Academic Staff:
Attributes:
Stores academic staff details and their roles in the internship system.
Methods:
Functions to review student progress and assist coordinators in managing internships.

14-Student Affairs:
Attributes:
Stores details related to student affairs tasks, such as their role in the internship system.
Methods:
Functions to facilitate communication between students, companies, and academic staff.

Assumptions:

User Roles and Access:

Users in the system are categorized into specific roles: Students, Coordinators, Instructors, Academic Staff, and Student Affairs.

Each role has distinct permissions and functionalities (coordinators manage forms, students submit resumes, instructors evaluate performance).

System Components:

The system manages core components like Trainee Information Forms, Resumes, Reports, and Performance Evaluation Forms.

These components are interlinked to allow data flow between related entities (reports and resumes linked to students).

Announcements and Forms:

Announcements are published by authorized roles (Coordinator)and are visible to all system users.

Forms are central to the system, with different types serving different purposes (Trainee Information Forms for internship details, Evaluation Forms for assessing students).

Company :

Companies are stored in the system and linked to Performance Evaluation Forms and Trainee Information Forms.

Companies can provide feedback on students' performance during their internships.

Data Relationships:

Students are associated with multiple forms such as resumes, reports, and trainee information forms.

Academic Staff and Instructors are linked to students for grading and feedback purposes.

3.2 Process View

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

3.2.1 Activity Diagram

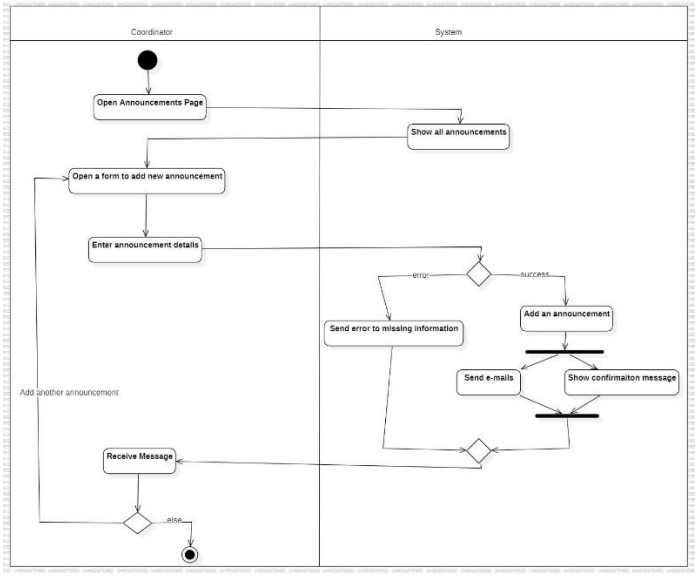

Figure 2 : Add Announcement – Activity Diagram

Coordinator opens the Announcements Page to see all announcements. They click to add a new announcement, fill out a form, and submit it. If there is a mistake, the system shows an error. If everything is correct, the system adds the announcement and sends an email to students, instructors, and student affairs. It also shows a success message to the coordinator. You can see from Figure 2.

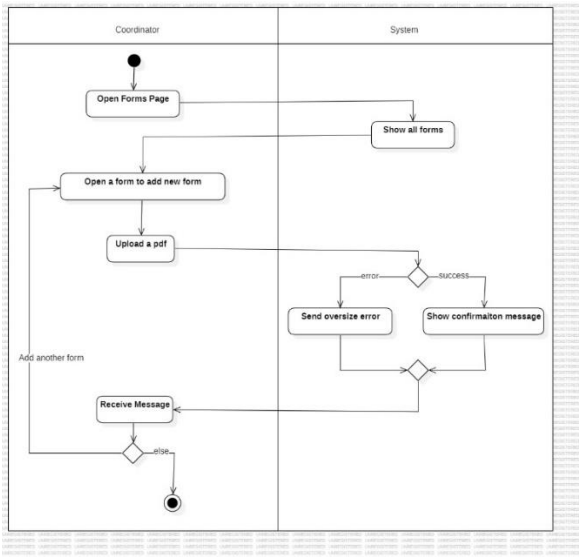

Figure 3 : Add Form – Activity Diagram

Coordinator opens the Forms Page to see all forms. They click to add a new form, upload a PDF, and submit it. If the PDF is too big, the system shows an error message. If it is the right size, the system uploads the form and shows a success message. You can see from Figure 3.

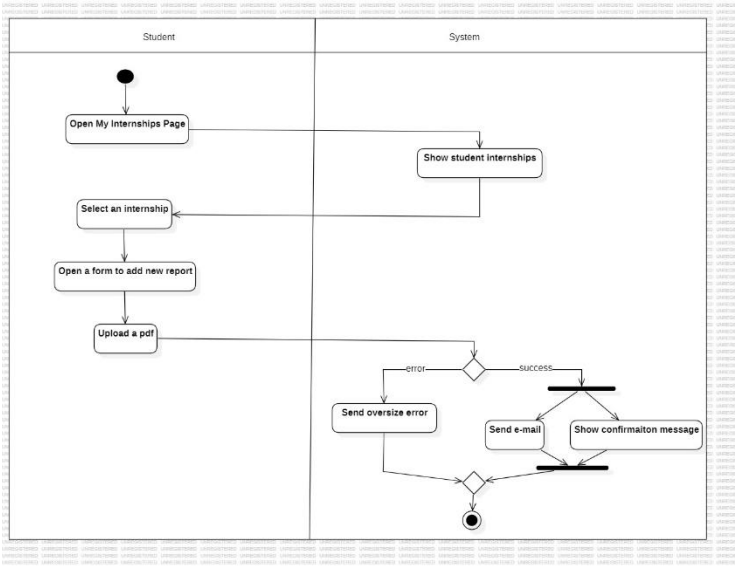

Figure 4 : Add Report – Activity Diagram

Student goes to the My Internships Page and sees their internships. They pick one internship, open a form to add a report, and upload a PDF. If the PDF is too big, the system shows an error. If the PDF is the correct size, the system uploads it, sends an email to the coordinator, and shows a success message to the student. You can see from Figure 4.

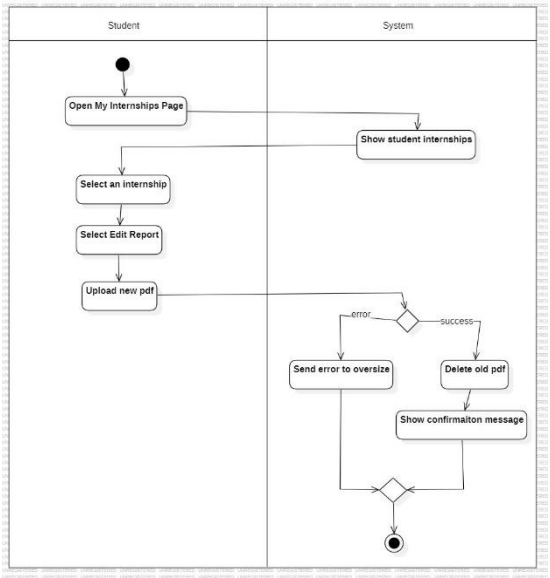

Figure 5 : Edit Report – Activity Diagram

Student opens the My Internships Page to see their internships. They select an internship, choose to edit the report, and upload a new PDF. If the PDF is too large, the system shows an error message. If it is the right size, the system deletes the old PDF, uploads the new one, and shows a success message. You can see from Figure 5.

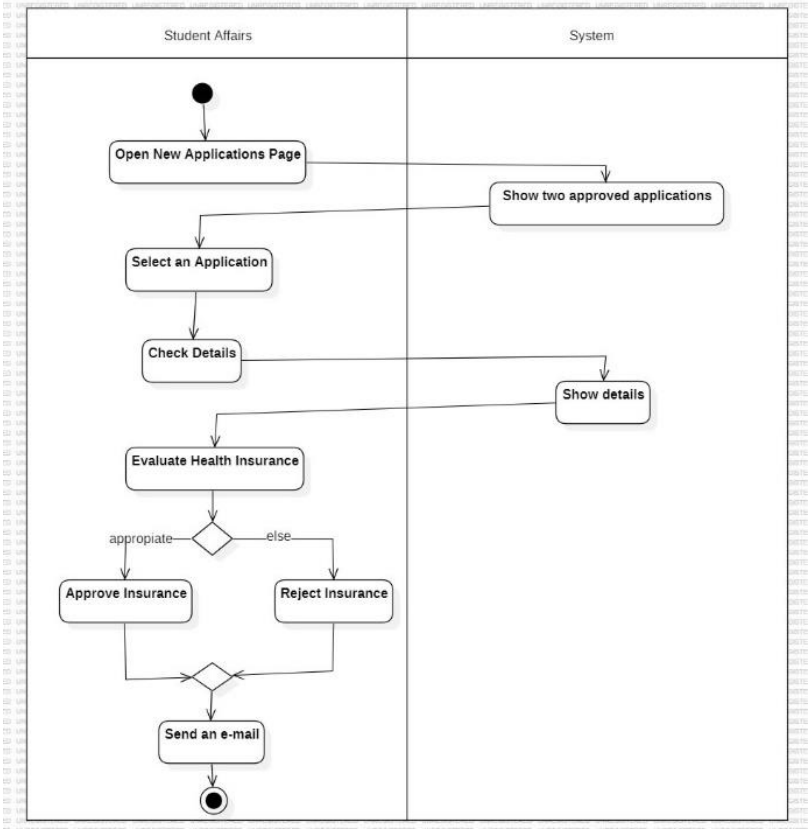

Figure 6 : Evaluate Health Insurance – Activity Diagram

Student Affairs officer opens the New Applications Page to see two approved applications. They pick an application, check its details, and decide if the insurance is okay or not. If it is okay, they approve it; if not, they reject it. The system sends an email to the student about the decision. You can see from Figure 6.

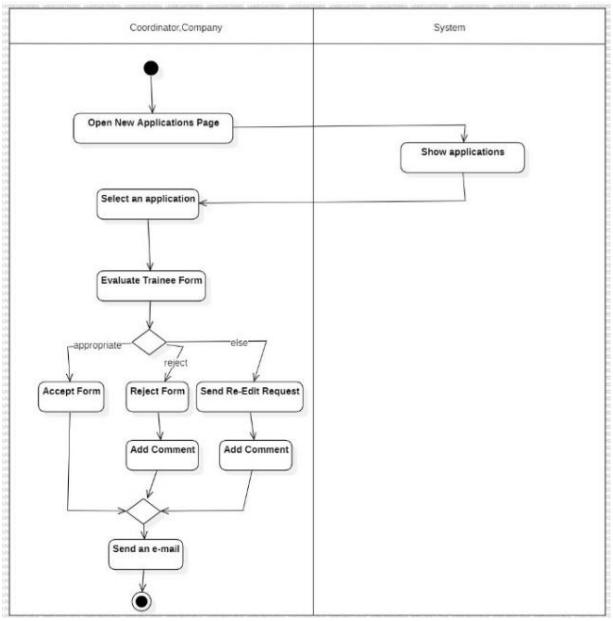

Figure 7 : Evaluate Trainee Form – Activity Diagram

Coordinator or company opens the New Applications Page to see all applications. They select an application and check the trainee form. If the form is good, they accept it. If not, they reject it or ask for changes. They can also add comments. The system then sends an email to the student with the result. You can see from Figure 7.

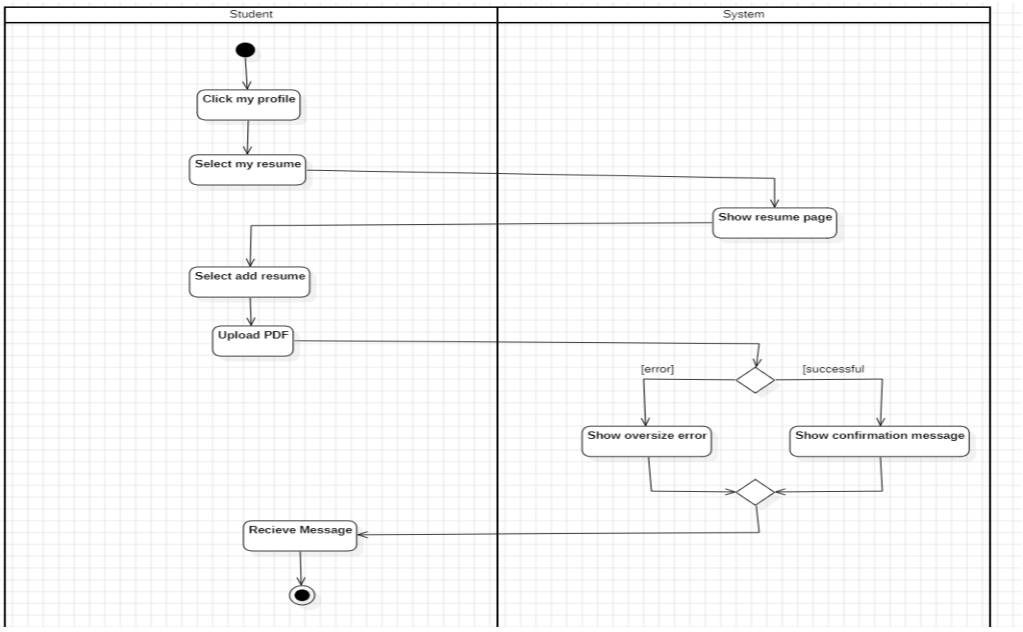

Figure 8: Add Resume – Activity Diagram

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 8.

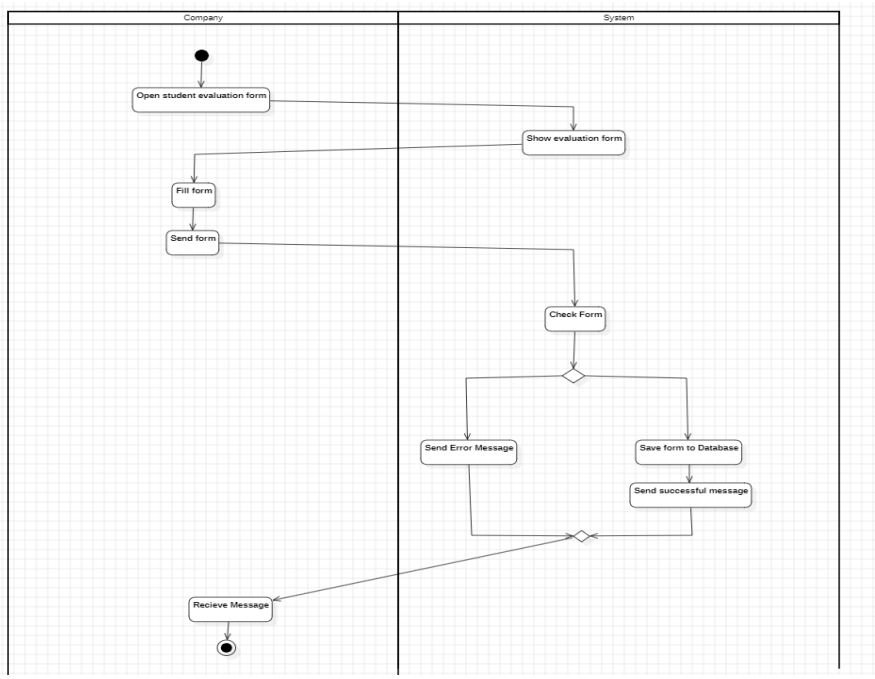

Figure 9 : Add Evaluation – Activity Diagram

After logining 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 9.

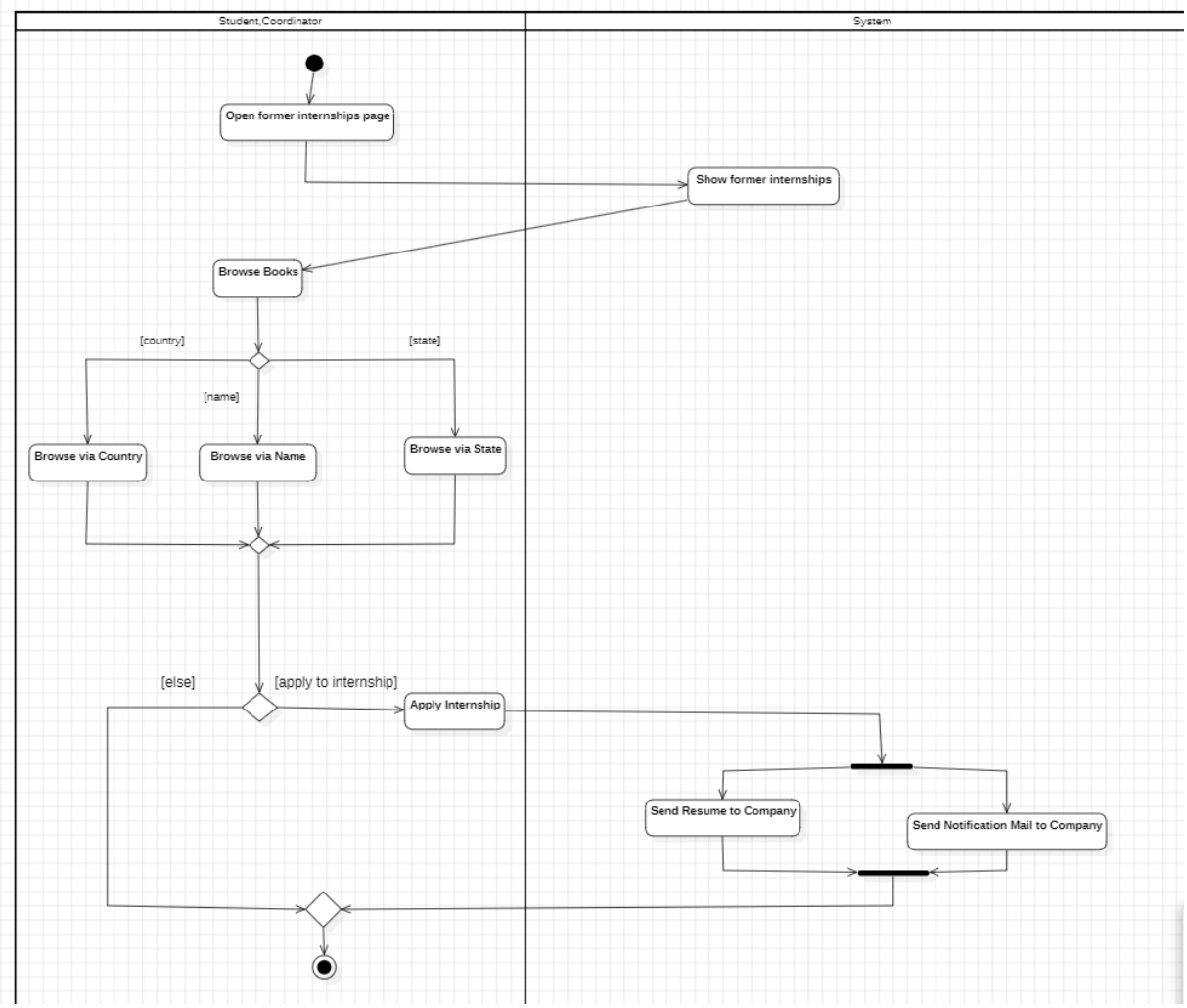

Figure 10 : Browse Internships – Activity Diagram

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 10.

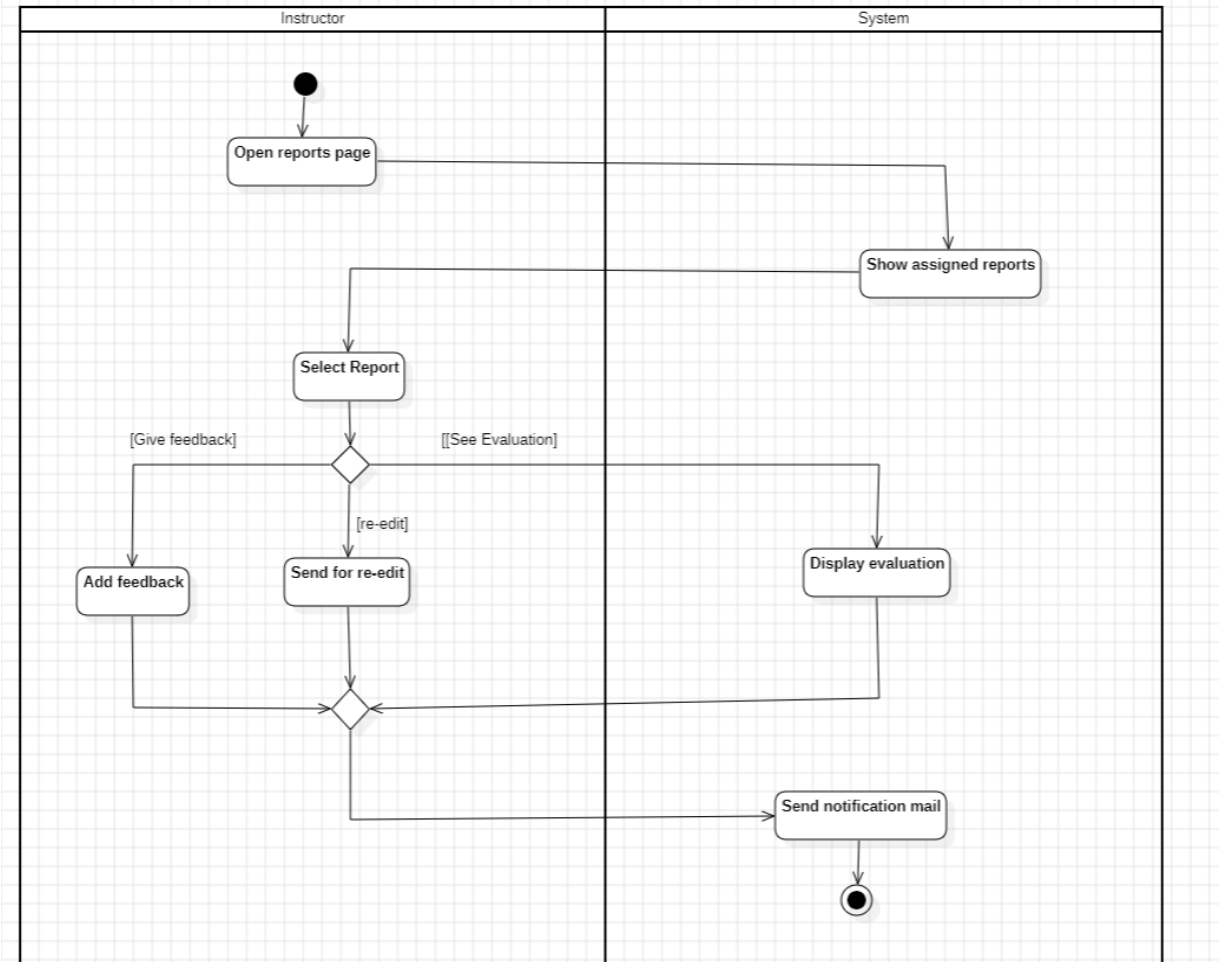

Figure 11 : Evaluate Reports – Activity Diagram

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 start to evaluate it. They have 3 options after that. First they can add feedback grade. Second they can send report back to student for re-editing purpose. Third they can take look at company evaluation form if they want to. At the end notification mail will send to student after either taking feedback or report. You can see from Figure 11.

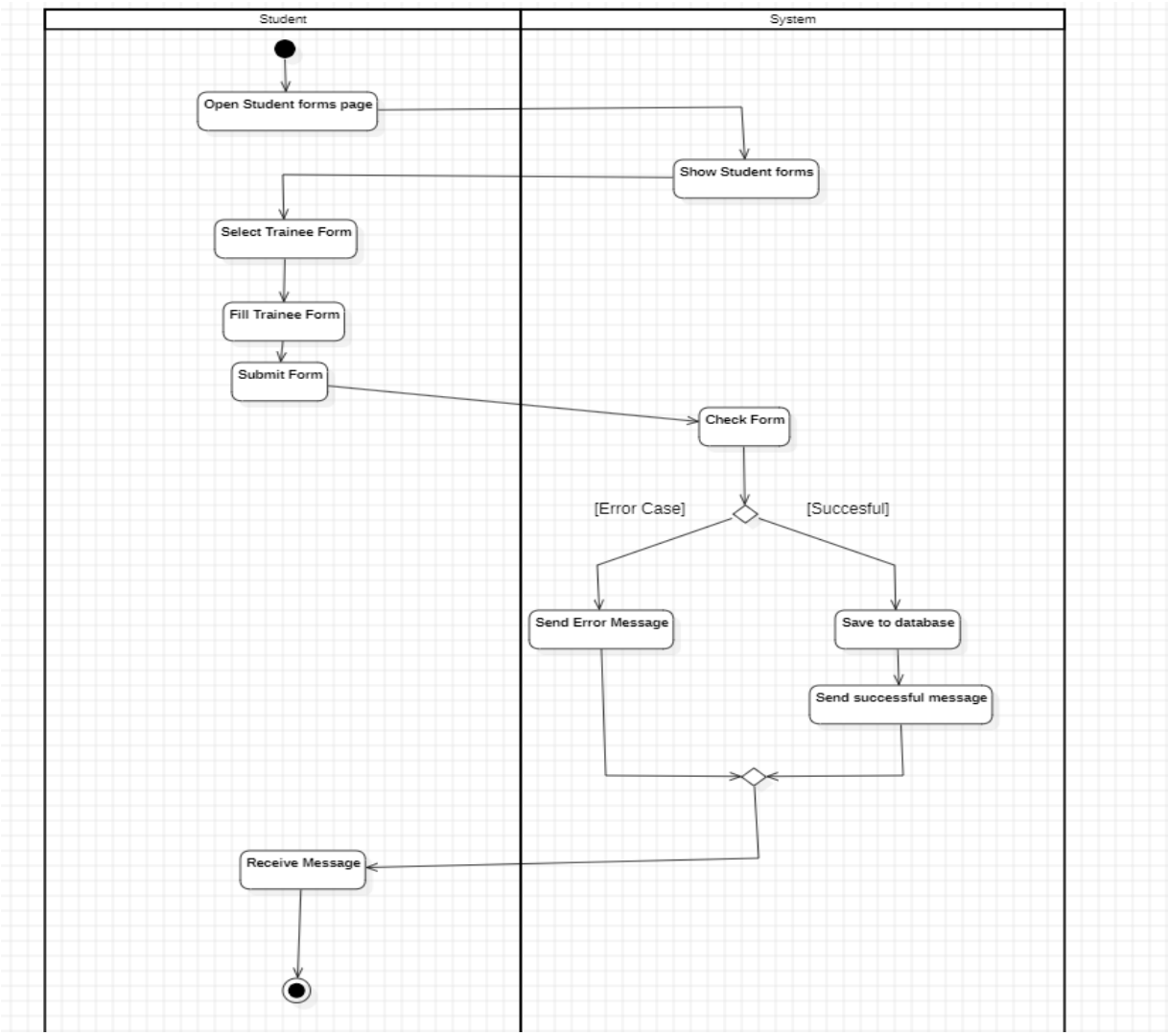

Figure 12 : Fill Trainee – Activity Diagram

At the beginning of the internships period, students must submit a Trainee Information Form. For that they must open forms page which will be displayed by our system, then they select the trainee form, fill it, send it. System will check if there is error such as misinformation or invalid dates are entered. If that is the case student will get error message. If it is valid, then they get successful message. You can see from Figure 12.

3.2.2 Sequence Diagrams

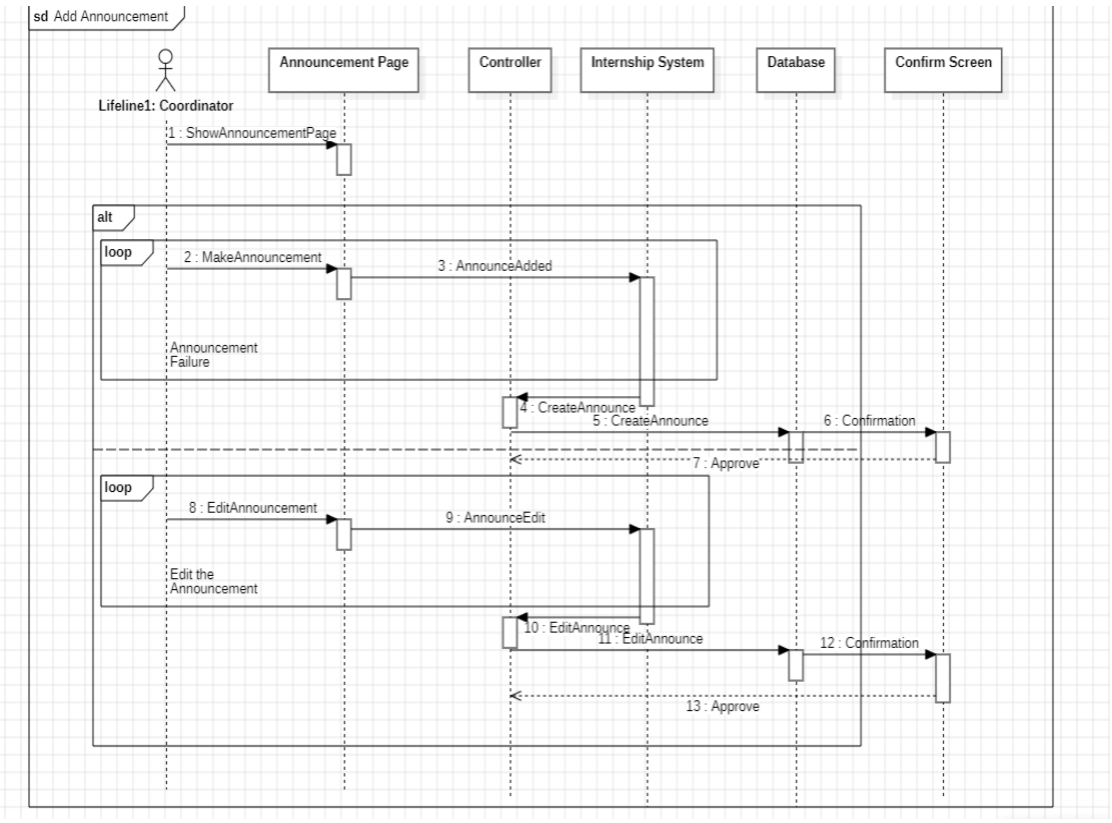

Figure 13: Sequence Diagram of Add Announcement

Add Announcement sequence diagram shows how the Coordinator creates and edits announcements. First, the coordinator accesses the announcement page using ShowAnnouncementPage. To create an announcement, the coordinator inputs details via MakeAnnouncement, which the Controller processes and sends to the Internship System for validation. The system stores the announcement in the Database, and a confirmation is sent back for approval. Similarly, for editing, the coordinator uses EditAnnouncement, and the updated details are validated and saved in the database. After a successful edit, a confirmation is returned for final approval. Additionally, a notification email is sent to relevant users to inform them about the new or updated announcement. Validation processes are assumed to be part of the CreateAnnouncement and EditAnnouncement methods for simplicity. You can see from Figure 13.

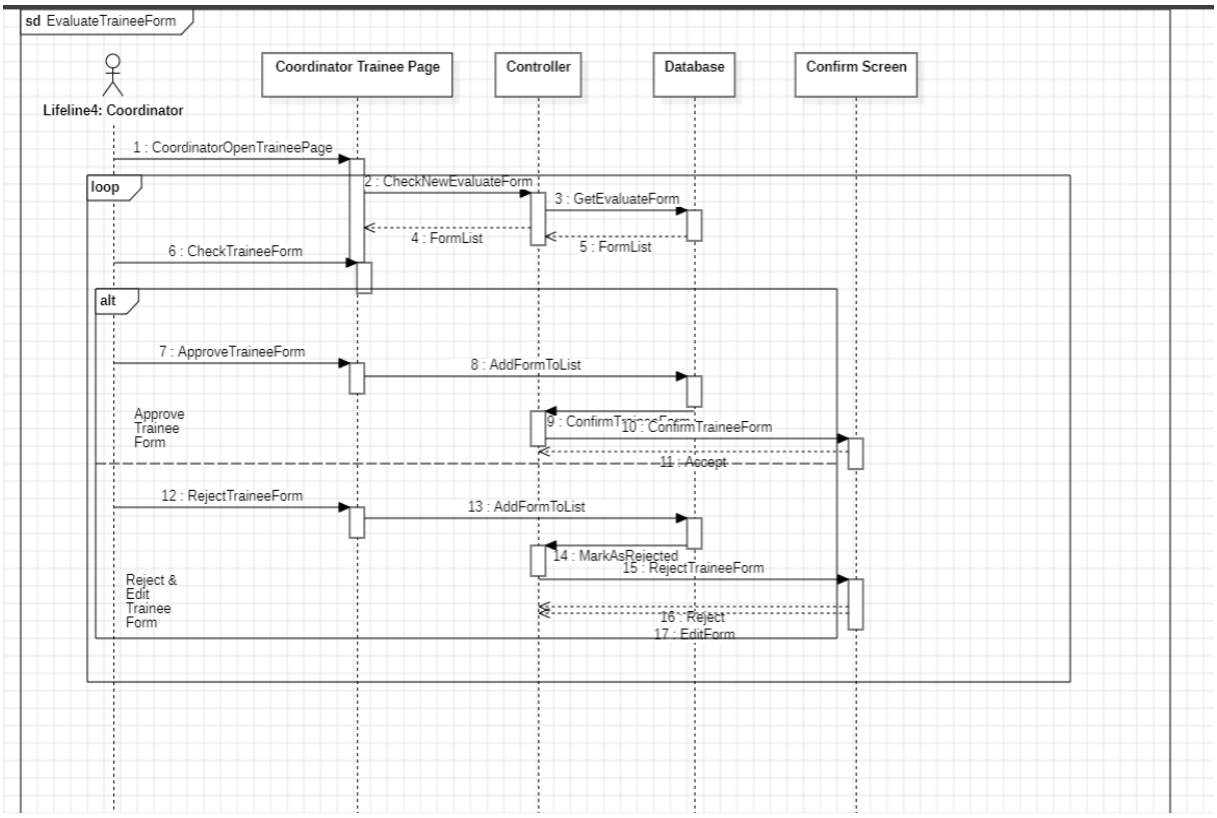

Figure 14: Sequence Diagram of Evaluate TraineeForm

Evaluate Trainee Form sequence diagram shows how the Coordinator reviews and manages trainee forms. First, the coordinator accesses the trainee page and views available forms. The system retrieves forms from the Database and presents them for evaluation. If a form is approved, it is added to the approved list, and the coordinator receives a confirmation. If the form is rejected, it is marked as rejected in the Database, and the coordinator has the option to edit it. Once the evaluation process is complete, a confirmation message is displayed. Additionally, after finalizing the action (approval or rejection), a notification email is sent to the trainee and relevant parties to inform them of the evaluation outcome. Validation processes are integrated into the form handling methods. You can see from Figure 14.

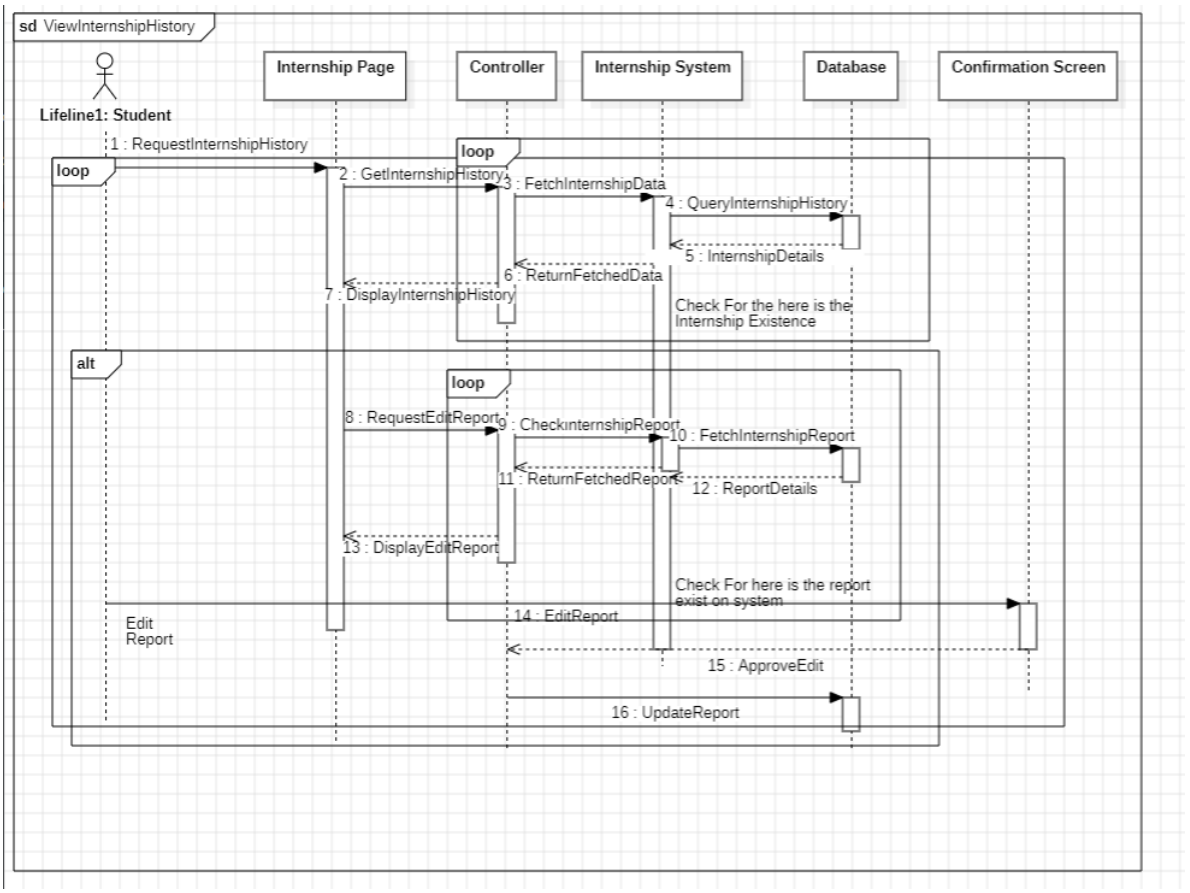

Figure 15: Sequence Diagram of ViewInternshipHistory

View Internship History sequence diagram shows how a Student views and manages their internship history. First, the student accesses the internship history page using RequestInternshipHistory. The system retrieves the internship data via GetInternshipHistory, which the Controller processes and fetches details from the Database. The internship details are validated and displayed to the student. If the student wishes to edit a report, they request this via RequestEditReport. The Controller processes the request and fetches the report details from the Database using FetchInternshipReport. After validation, the student can edit the report, and the system updates the report details and confirms the changes through the Confirmation Screen. Additionally, once the report is successfully edited, a notification email is sent to the coordinator and any relevant parties, informing them of the updated report. Validation processes are assumed to be part of the history and report handling methods. You can see from Figure 15.

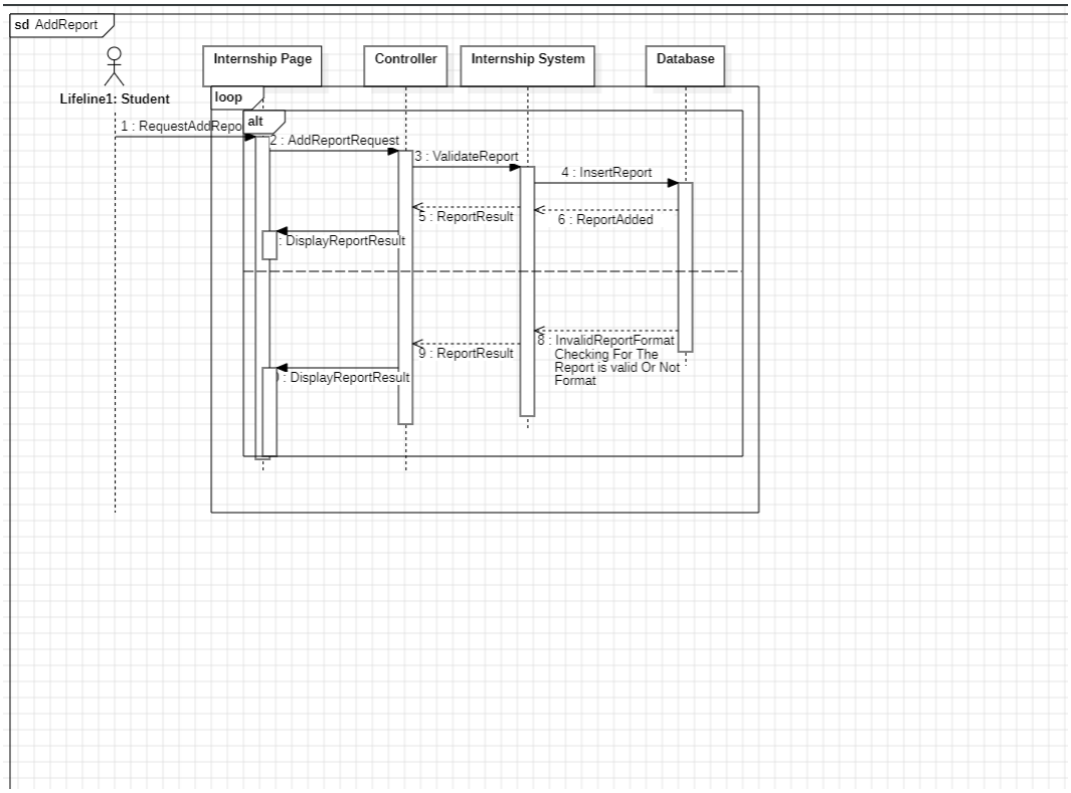

Figure 16: Sequence Diagram of AddReport

Add Report sequence diagram shows how a Student submits a report for their internship. First, the student initiates the process using RequestAddReport. The system processes the request via AddReportRequest, which is validated by the Internship System using ValidateReport. If the report is valid, the system sends the InsertReport command to the Database, and the ReportAdded confirmation is returned. The result is displayed to the student via DisplayReportResult. If the report format is invalid, an InvalidReportFormat response is returned, informing the student of the issue. Additionally, after the report is successfully added, a notification email is sent to the coordinator to inform them of the new report submission. Validation and formatting checks are assumed to be part of the ValidateReport process. You can see from Figure 16.

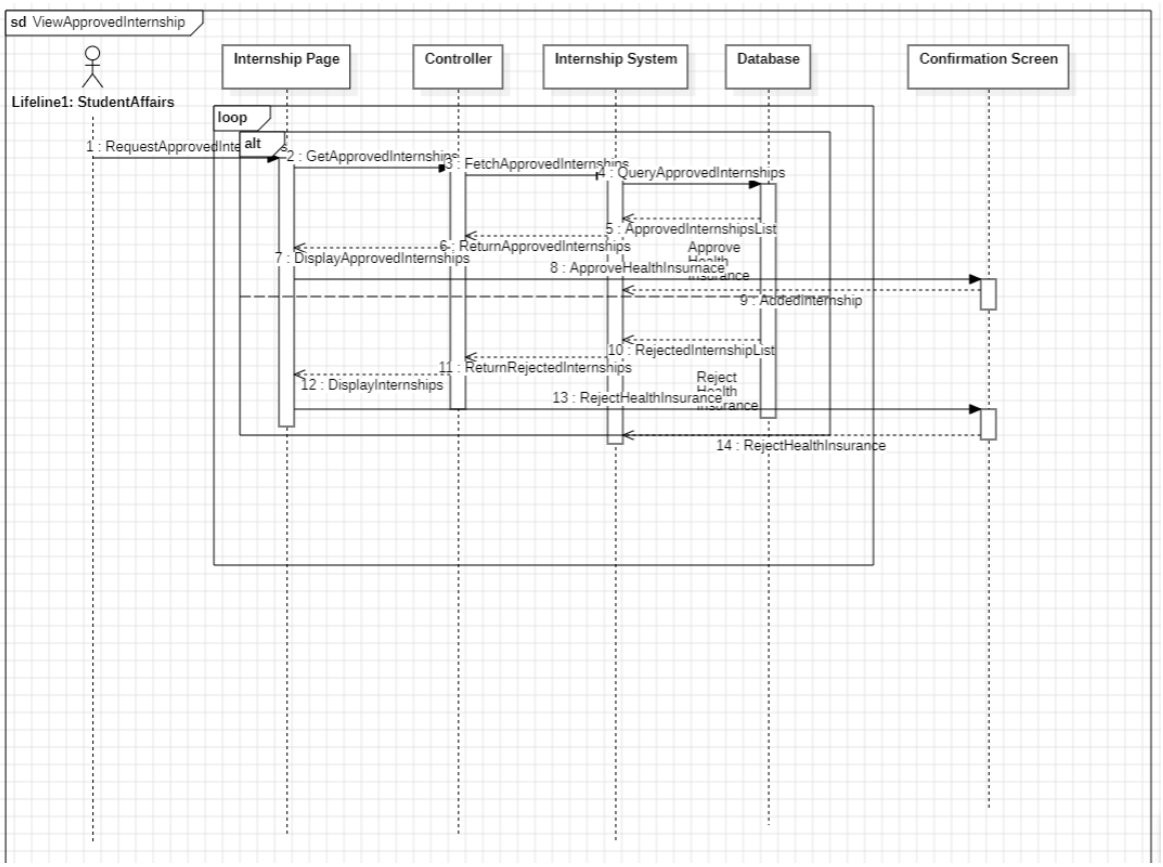

Figure 17: Sequence Diagram of ViewApprovedInternship

View Approved Internship sequence diagram shows how Student Affairs reviews and manages approved internships. First, Student Affairs accesses the approved internships using RequestApprovedInternship. The system retrieves the data via GetApprovedInternships, which the Controller processes and fetches from the Database through QueryApprovedInternships. The list of approved internships is validated and displayed to Student Affairs using DisplayApprovedInternships. If health insurance approval is required, ApproveHealthInsurance is triggered, and the internship is added using AddInternship. If health insurance is rejected, RejectHealthInsurance is called, marking the internship as rejected in the Database. The rejected internships are then displayed to Student Affairs using DisplayInternships. Validation processes are assumed to be part of the handling methods. You can see from Figure 17.

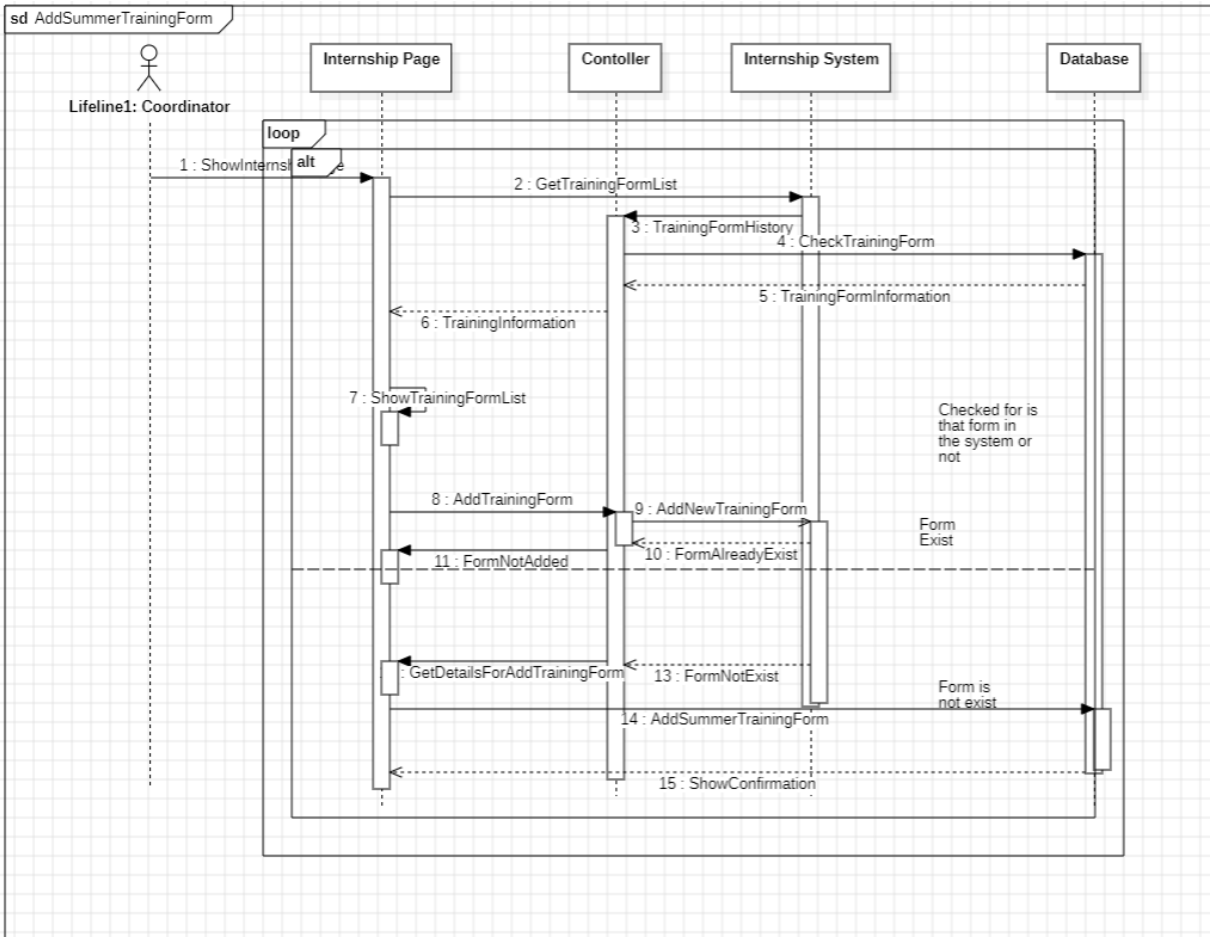

Figure 18: Sequence Diagram of AddSummerTrainingForm

Add Summer Training Form sequence diagram shows how the Coordinator manages and adds summer training forms. First, the coordinator accesses the training forms using ShowInterns. The system retrieves the list via GetTrainingFormList, which the Controller processes and fetches from the Database through TrainingFormHistory. The list of existing forms is validated and displayed to the coordinator using ShowTrainingFormList. If a new form is added, AddTrainingForm is triggered, and the system checks for duplicates using CheckTrainingForm. If the form already exists, FormAlreadyExist is returned, and the coordinator is notified. If the form does not exist, GetDetailsForAddTrainingForm is used to gather additional information, and the form is added using AddSummerTrainingForm. The addition is confirmed with a message displayed to the coordinator via ShowConfirmation. Validation processes are assumed to be part of the handling methods. You can see from Figure 18.

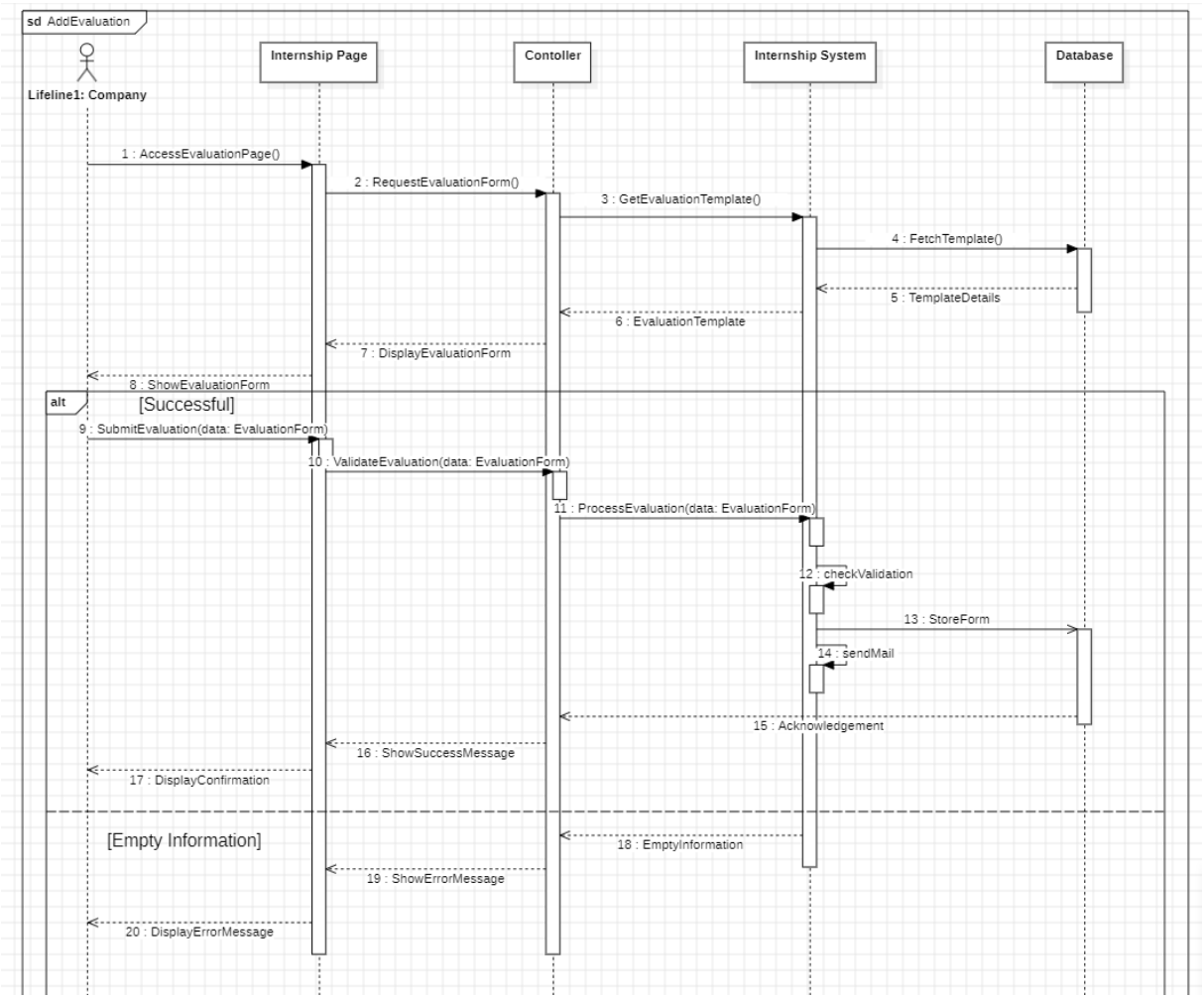

Figure 19: Sequence Diagram of AddEvaluation

This is the company's evaluating a intern student process. At the end of the internship Companys 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 19.

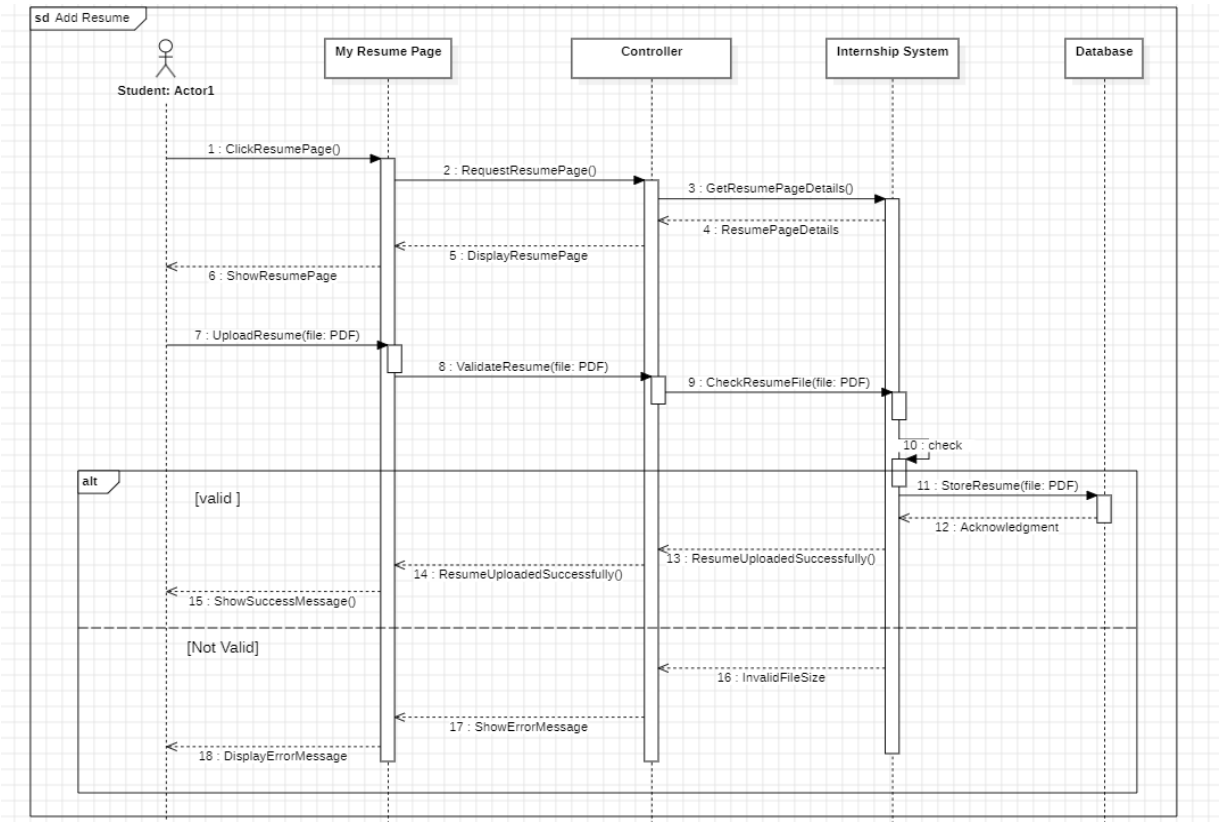

Figure 20: Sequence Diagram of Add Resume

As a new functionalty to the old system. This functionalty is about adding a resume to our system for further internship applications. First student clicks the my resume page from profile segment. After that controller and a system shows related information to the student. Student can add a resume pdf to our system. If the pdf size is too big they will get error, but if not they will receive success message. You can see from Figure 20.

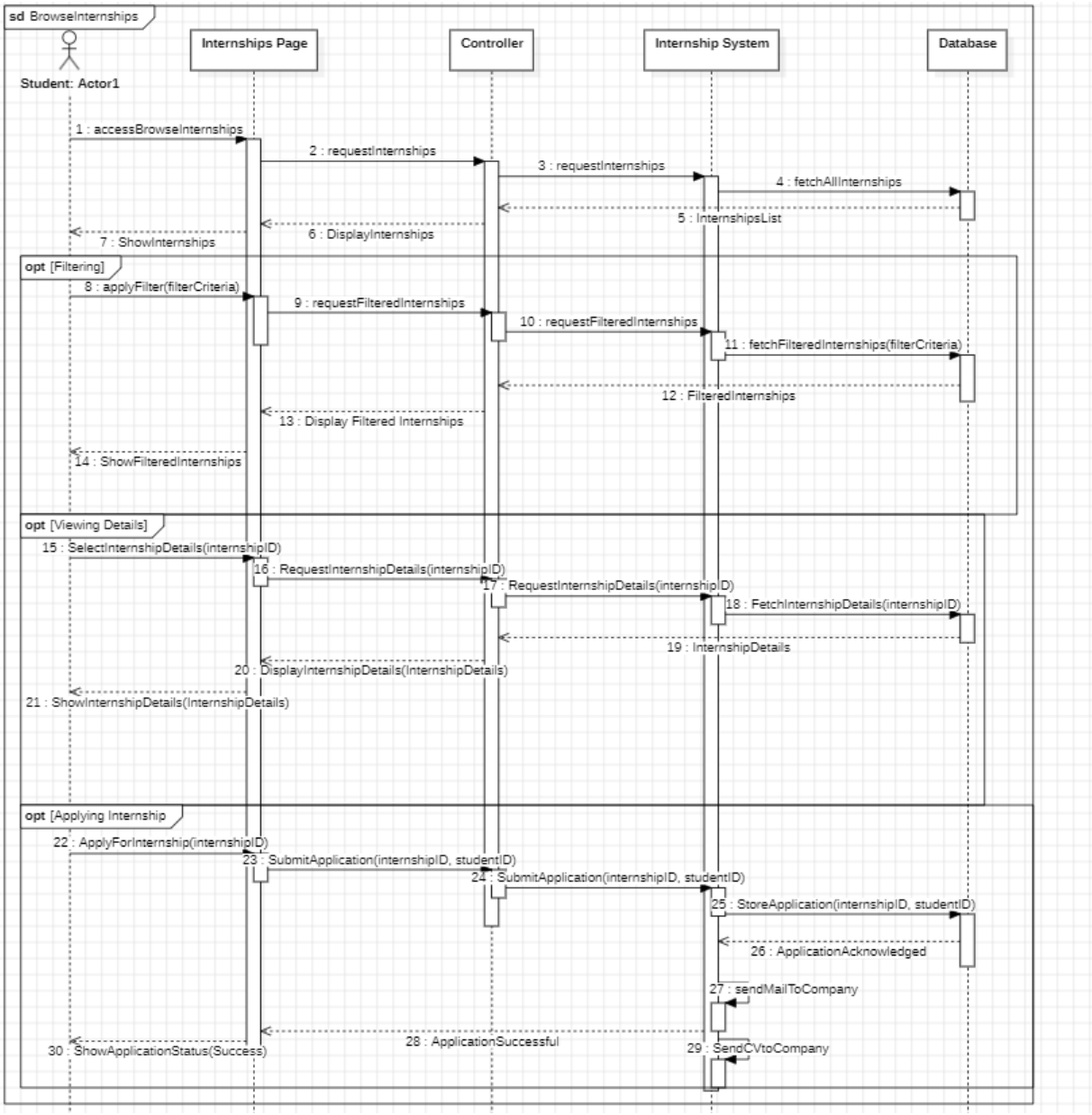

Figure 21: Sequence Diagram of Browse Internships

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 21.

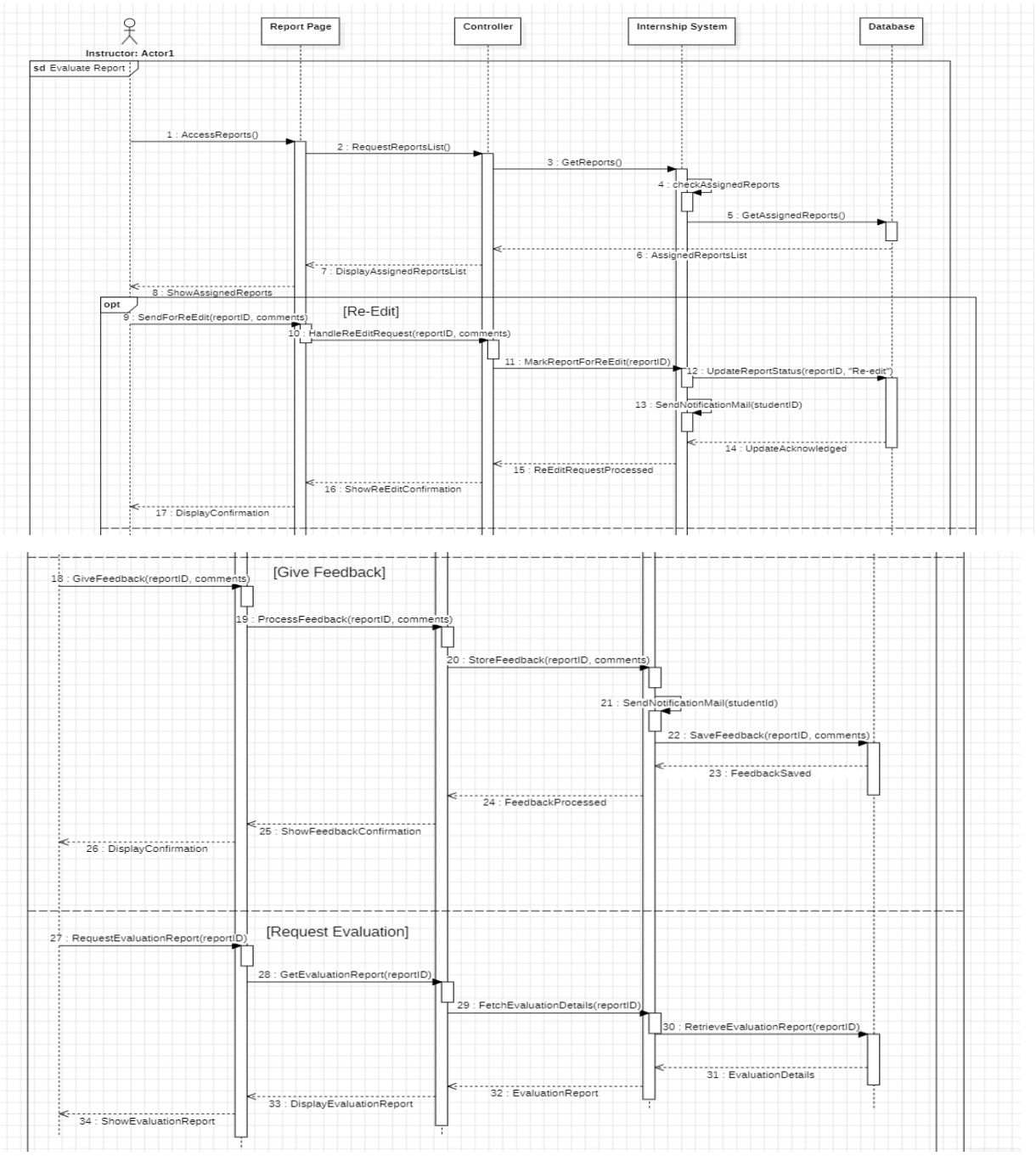

Figure 22: Sequence Diagram of Evaluate Report

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 also students will get notify in the process. Give feedback; they approve the report with a grade feedback, and they can also see the Evalution Report from Company. You can see from Figure 22.

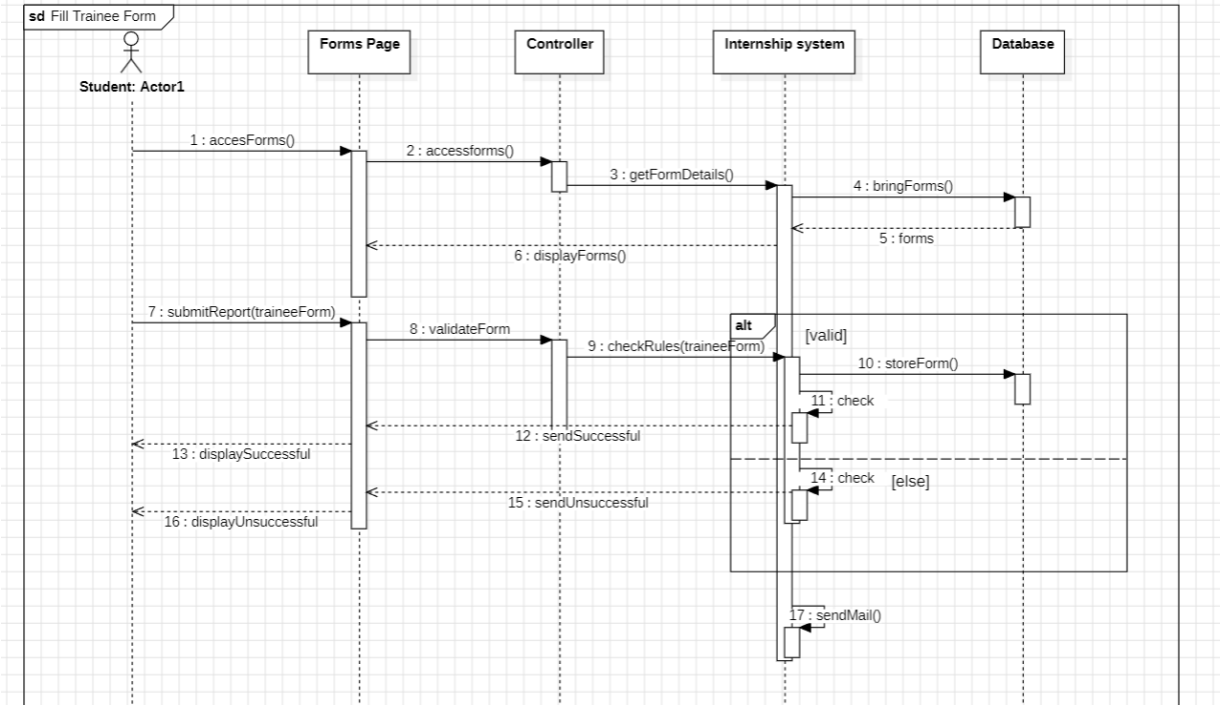

Figure 23: Sequence Diagram of Fill Trainee Form

At the early stages of internship periods students must fill a Trainee Form. For that they first need to access forms page after that system will provide all forms. Then when they fill and submit the form our system will check rules such as are dates of the internship is valid or not. If it is valid form it will be stored and coordinator will be notified via mail. If it is not then student will face an error message from system. You can see from Figure 23.

3.2.3 Data Flow Diagrams

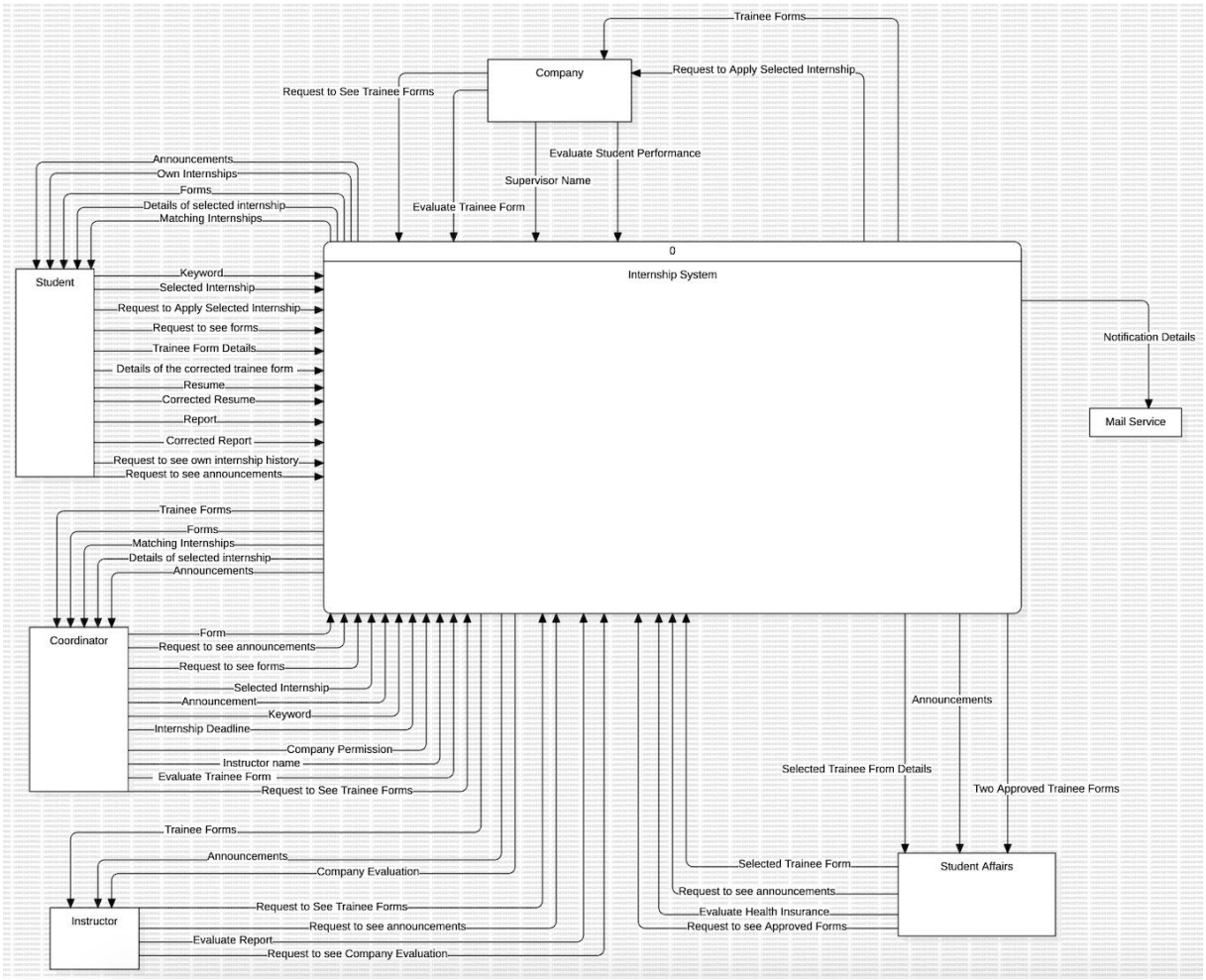

Figure 24: Context Level Diagram – DFD

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 24.

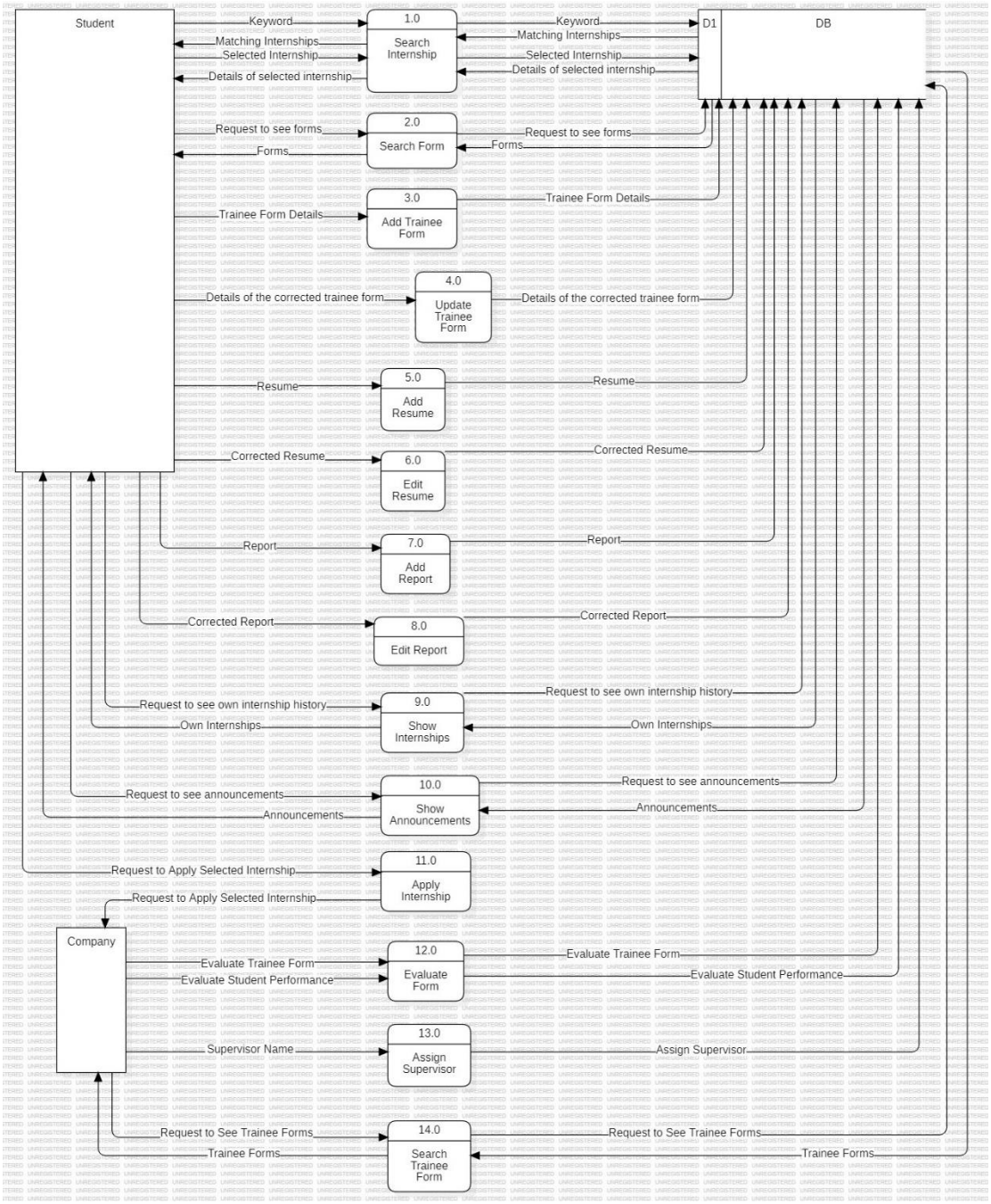

Figure 25: Level 0(1) – DFD

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 25.

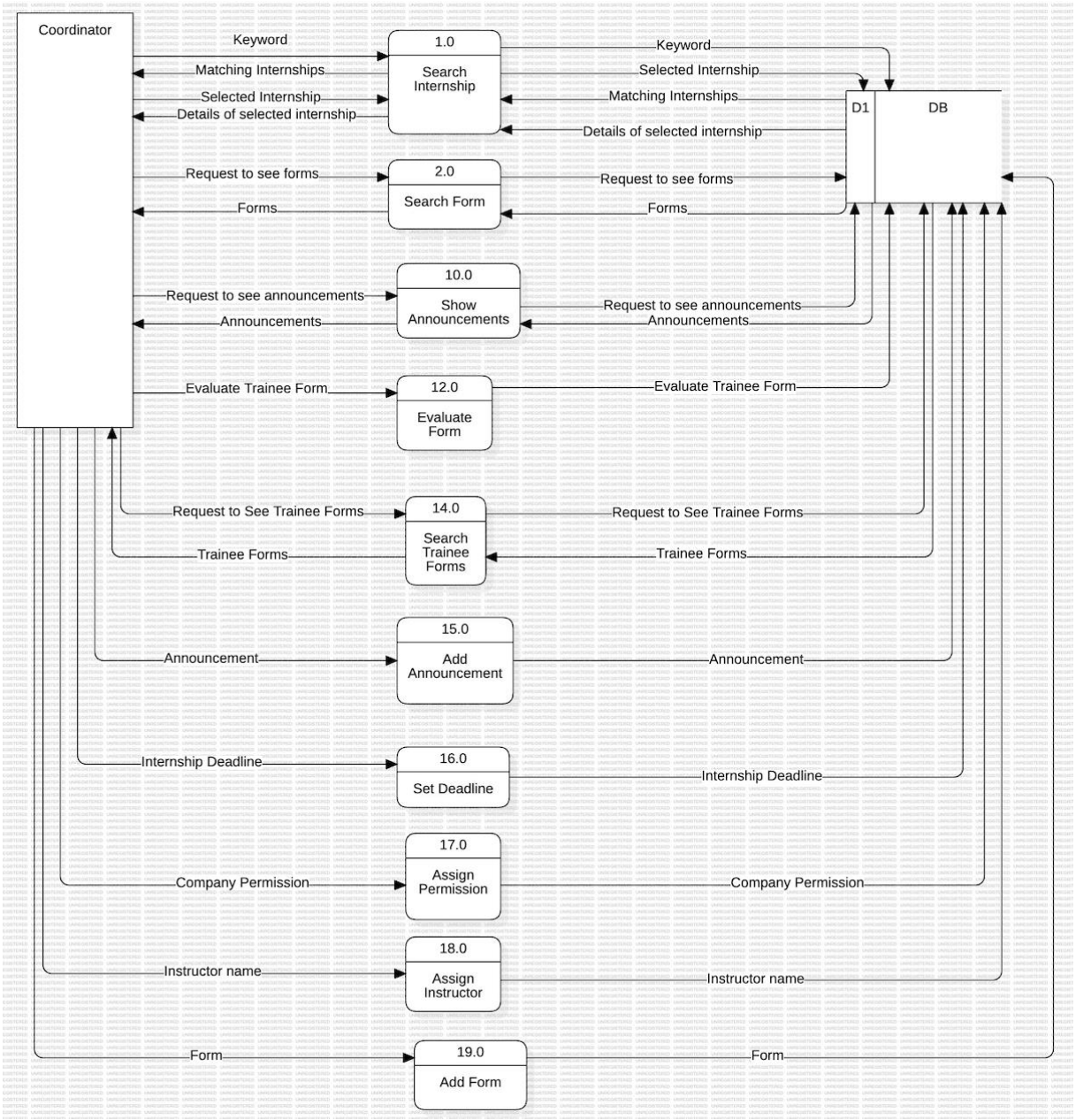

Figure 26: Level 0(2) – DFD

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 26.

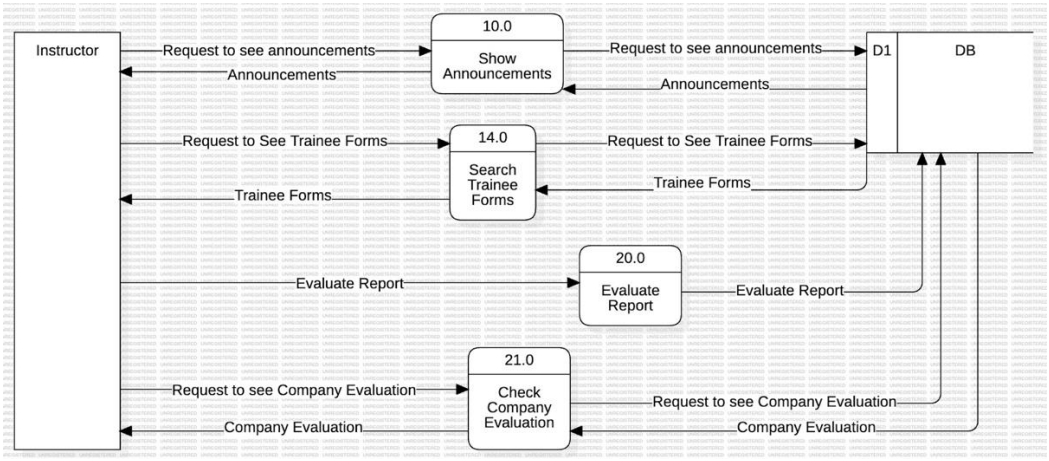

Figure 27: Level 0(3) – DFD

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 27.

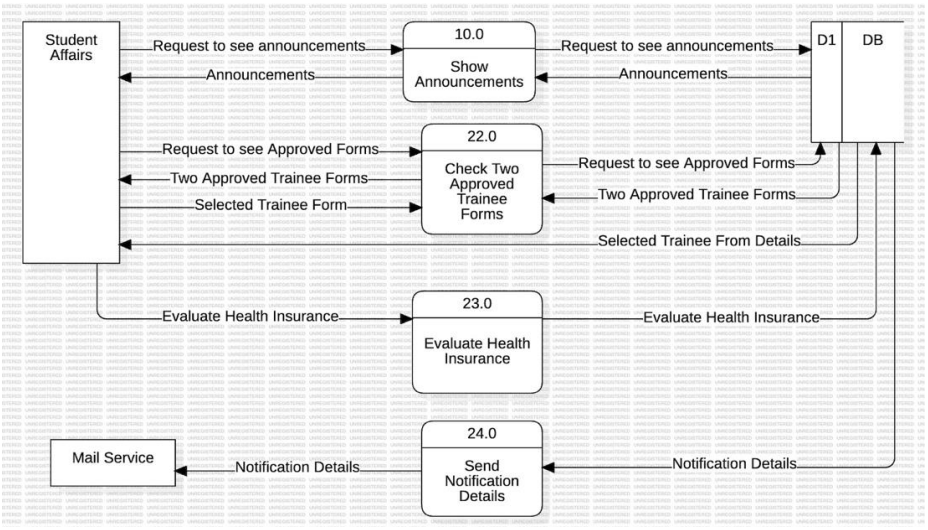

Figure 28: Level 0(4) – DFD

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 28.

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 29 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 29: Notification Table

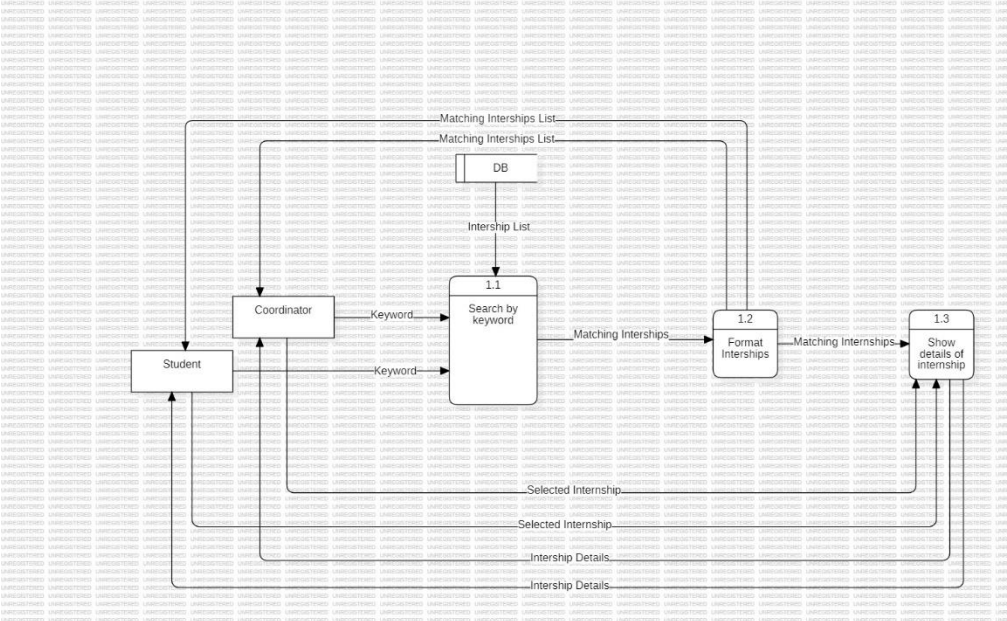

Figure 30: Level 1(1) – DFD

In the diagram in Figure 30, we have described how students and coordinators can search internsihp by keyword and view the details of an internship.

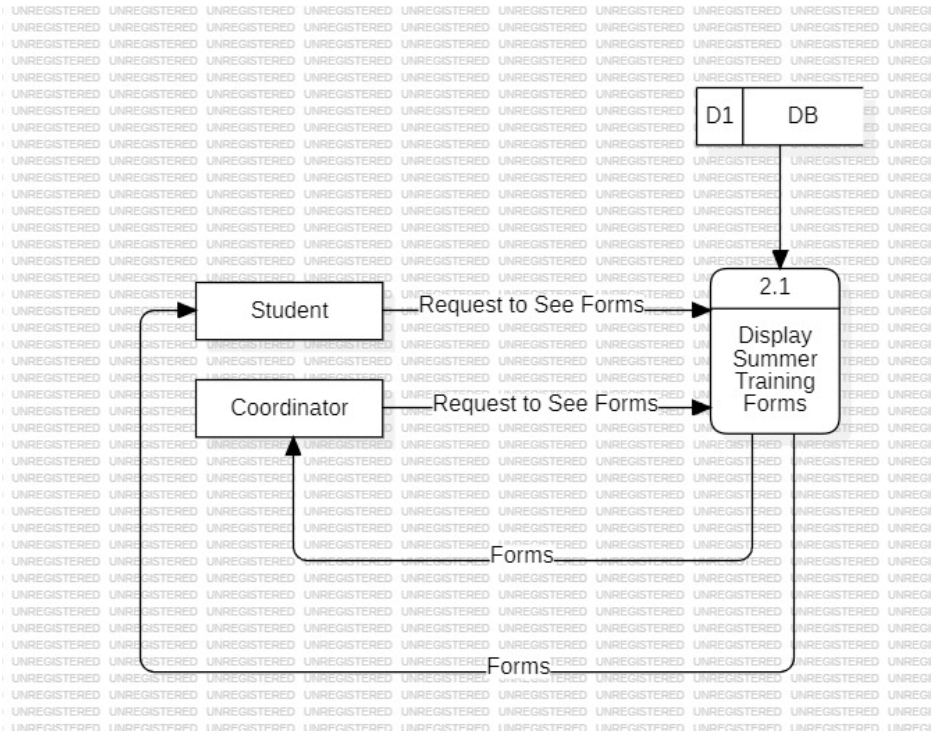

Figure 31: Level 1(2) – DFD

In this diagram, we have described how students and coordinators can view the summer training forms. You can see from Figure 31.

Figure 32: Level 1(3) – DFD

In this diagram in Figure 32, we have described how students can add trainee forms to the system.

Figure 33: Level 1(4) – DFD

In this diagram, we have described how students can update their trainee forms. You can see from Figure 33.

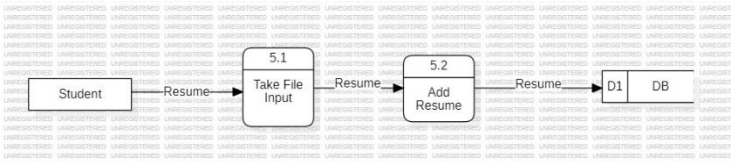

Figure 34: Level 1(5) – DFD

In this diagram in Figure 34, we have described how students can upload their resume to the system.

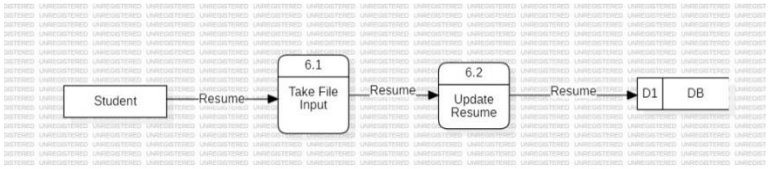

Figure 35: Level 1(6) – DFD

The diagram in Figure 35, shows how we have described how students can update their resume.

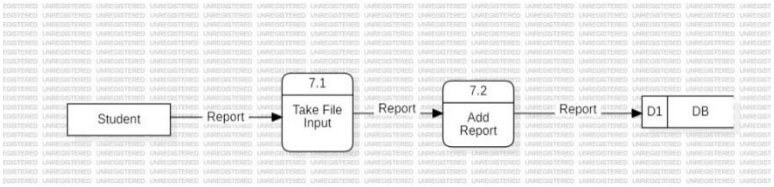

Figure 36: Level 1(7) – DFD

This diagram in Figure 36, explains how we have described how students can upload their internship reports.

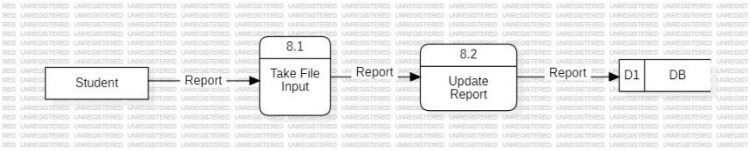

Figure 37: Level 1(8) – DFD

The diagram in Figure 37, shows how we have described how students can update their internship reports.

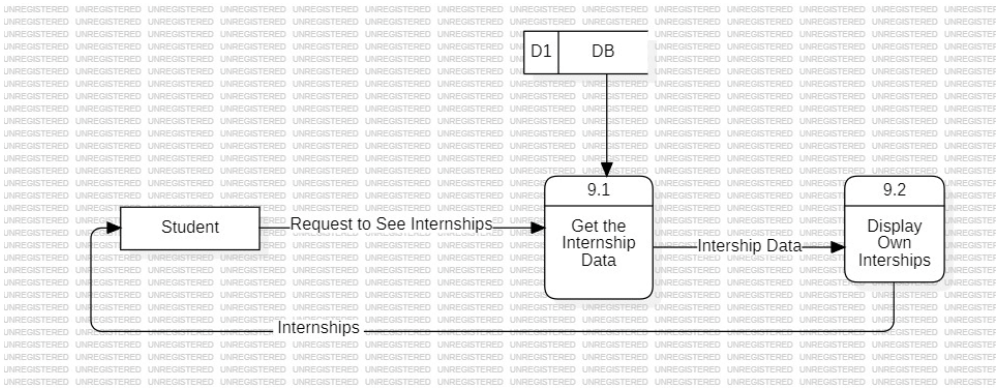

Figure 38: Level 1(9) – DFD

In this diagram in Figure 38, we have described how students can view their own internships.

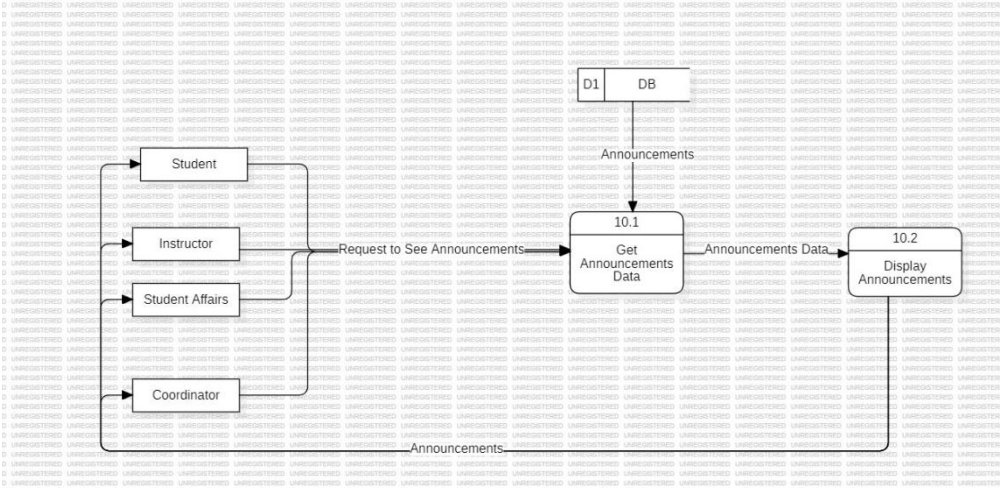

Figure 39: Level 1(10) – DFD

In this diagram, we have described how students, instructors, student affairs an coordinators can view the announcements. You can see from figure 39.

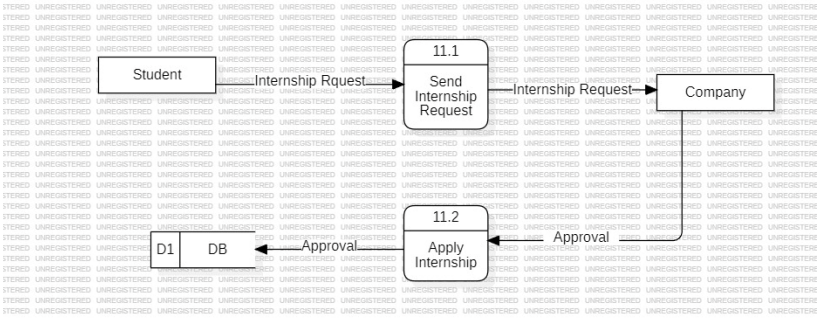

Figure 40: Level 1(11) – DFD

In this diagram in Figure 40, we have described how students can apply for an internship and how the company approve internships via the staj system.

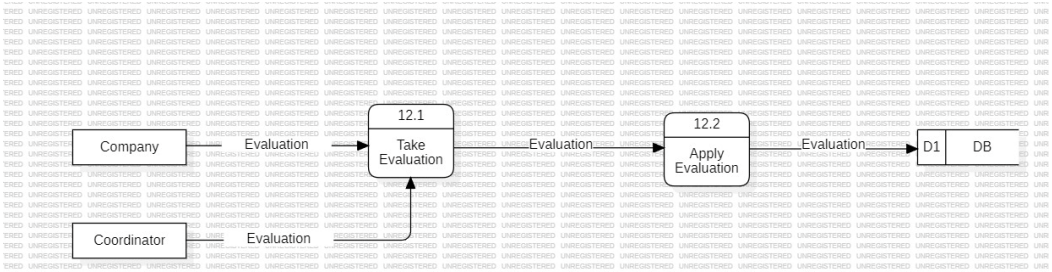

Figure 41: Level 1(12) – DFD

The diagram in Figure 41, demonstrates how we have described how companies and coordinators can evaluate internship requests via the staj system.

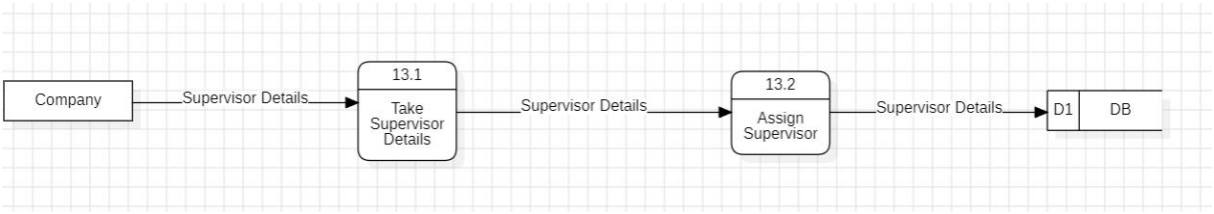

Figure 42: Level 1(13) – DFD

In this diagram, we have described how companies can fill supervisor details and assign a supervisor for an internship. You can see from Figure 42.

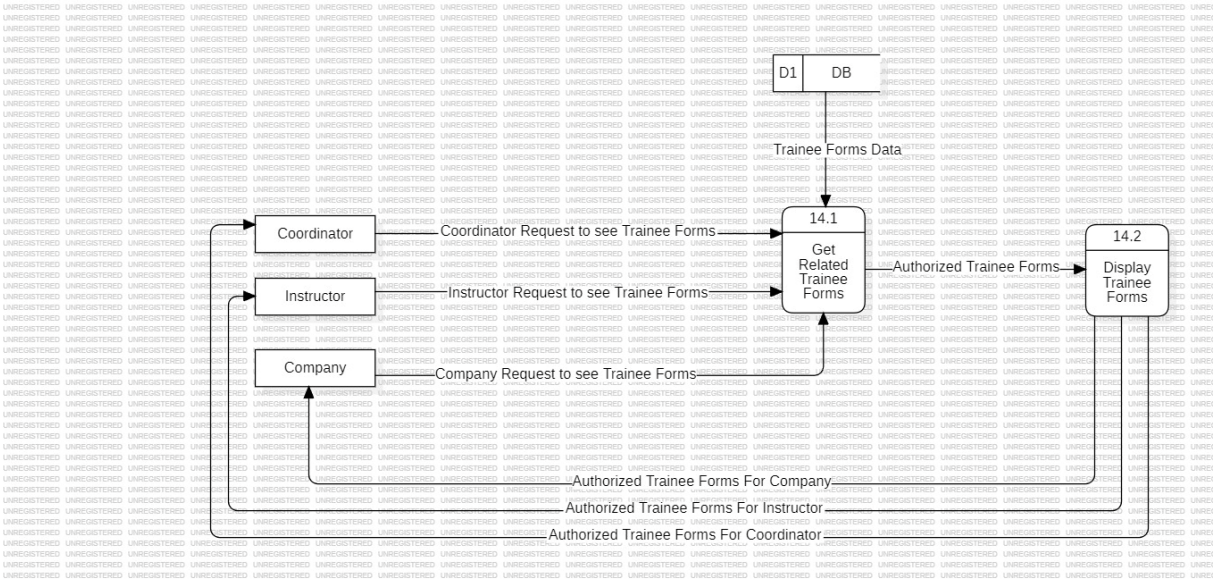

Figure 43: Level 1(14) – DFD

In this diagram in Figure 43, we have described how companies, instructors and coordinators can view trainee forms that they have access to.

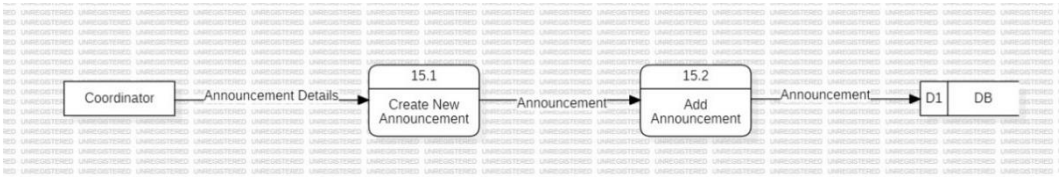

Figure 44: Level 1(15) – DFD

In this diagram, we have described how coordinators can add an announcement to the system. You can see from Figure 44.

Figure 45: Level 1(16) – DFD

The diagram in Figure 45, shows how we have described how coordinators set the deadline dates for internships.

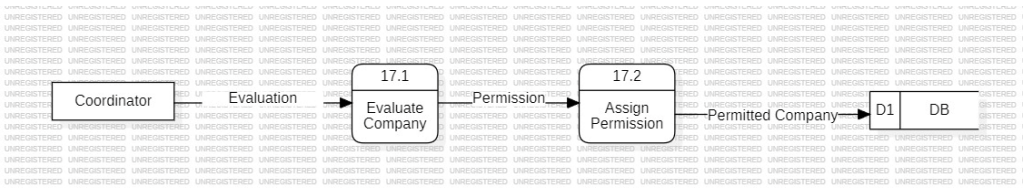

Figure 46: Level 1(17) – DFD

In this diagram in Figure 46, we have described how coordinators can evaluate and approve a company into the staj system.

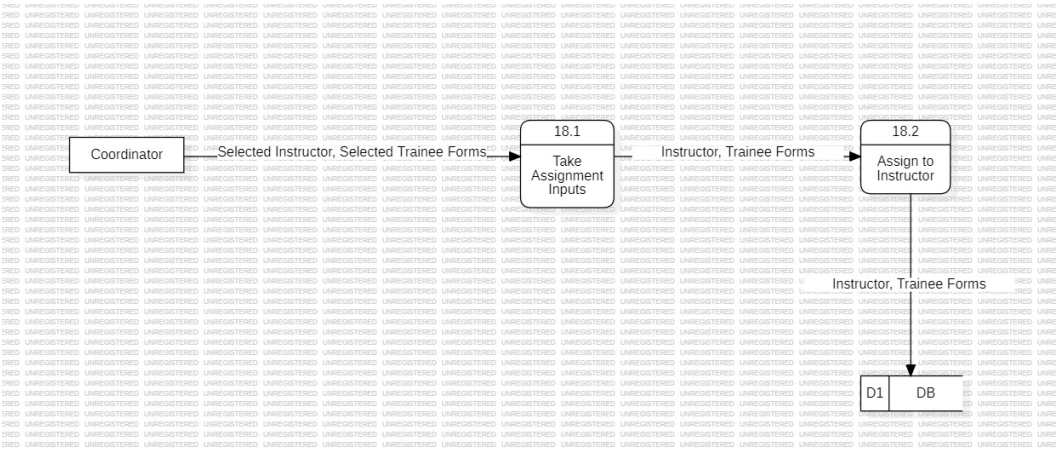

Figure 47: Level 1(18) – DFD

This diagram in Figure 47, shows how we have described how coordinators can assign instructors to trainee forms.

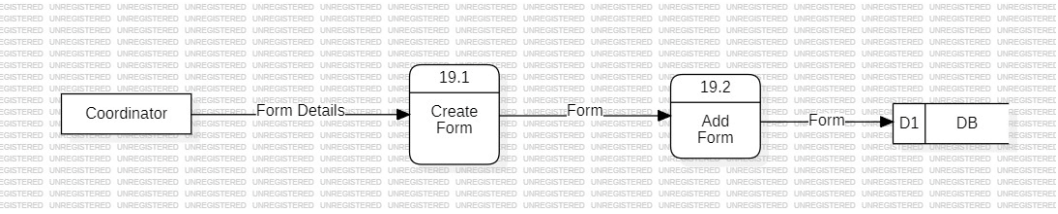

Figure 48: Level 1(19) – DFD

In this diagram, we have described how coordinators can add a form related with summer training. You can see it from Figure 48.

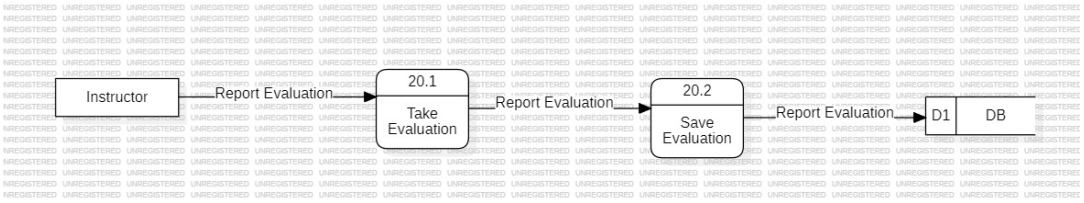

Figure 49: Level 1(20) – DFD

In this diagram, we have described how instructors can evaluate an internship report. You can refer to Figure 49 for a detailed overview.

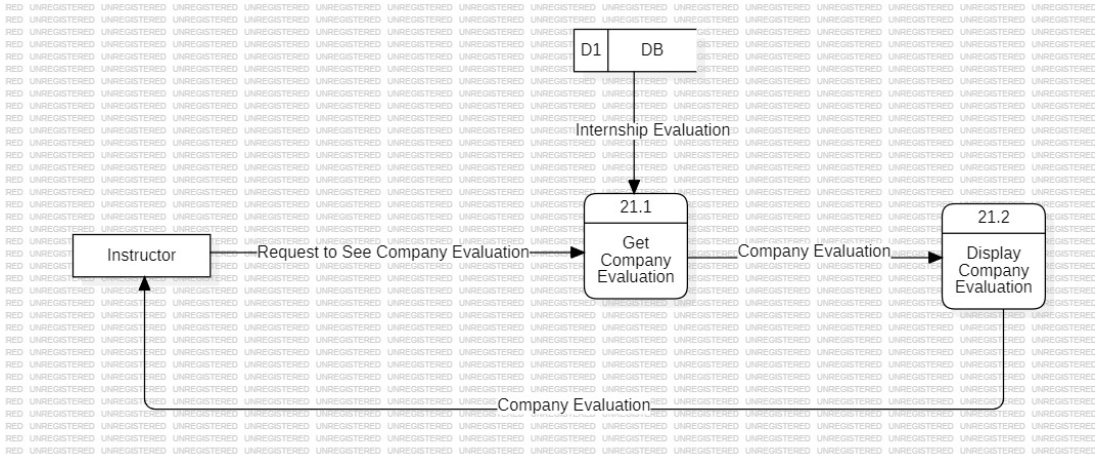

Figure 50: Level 1(21) – DFD

In this diagram, we have described how instructors can view company evaluations. You can see from Figure 50.

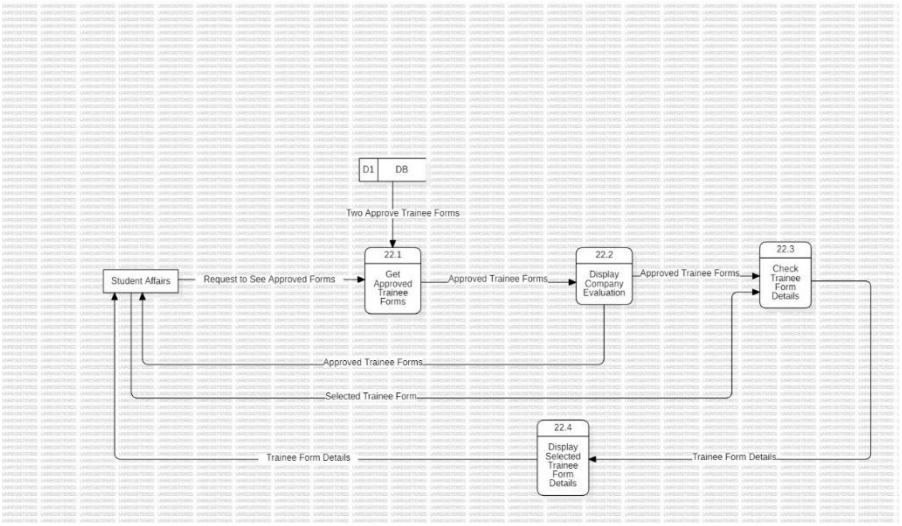

Figure 51: Level 1(23) – DFD

In the diagram in Figure 51, we have described how student affairs can view the approved trainee forms and select a trainee form to see its details.

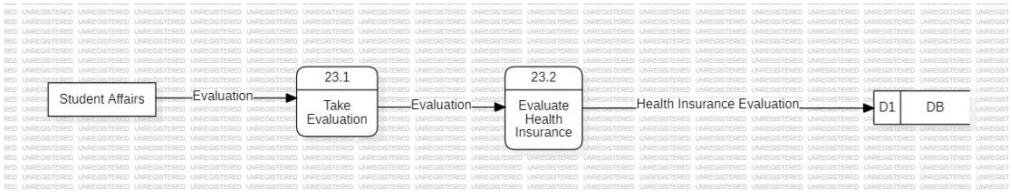

Figure 52: Level 1(23) – DFD

This diagram, explains how we have described how student affairs can evaluate health insurance for an internship. You can see from Figure 52.

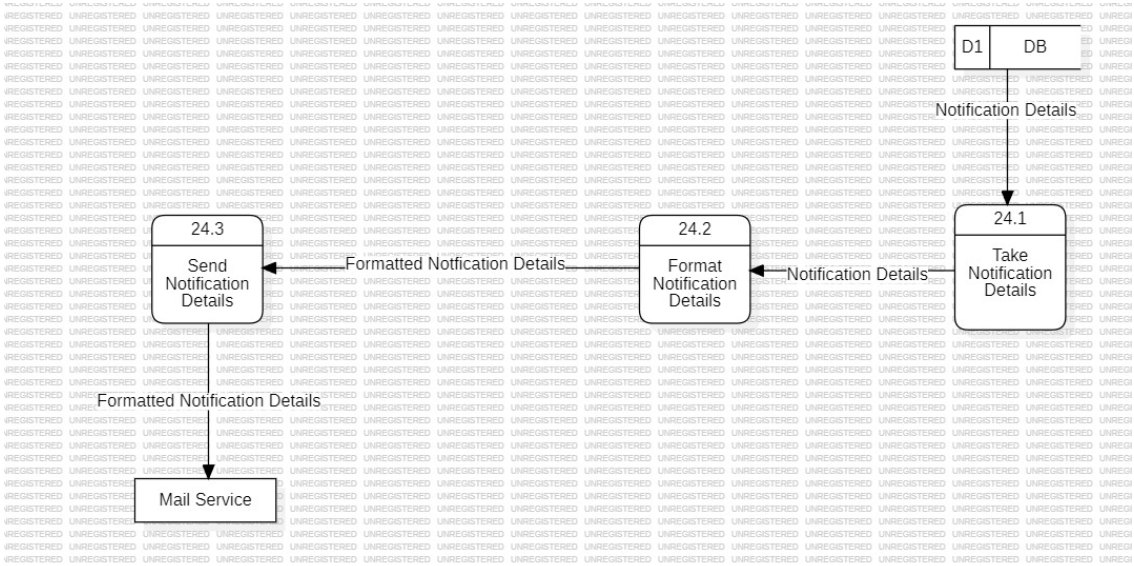

Figure 53: Level 1(24) – DFD

As shown in Figure 53, in this diagram, we have described how notifications are created and sent through the mail service with the details provided.

3.3 Development View

3.3.1 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 Performation Evaluation Forms in the system.

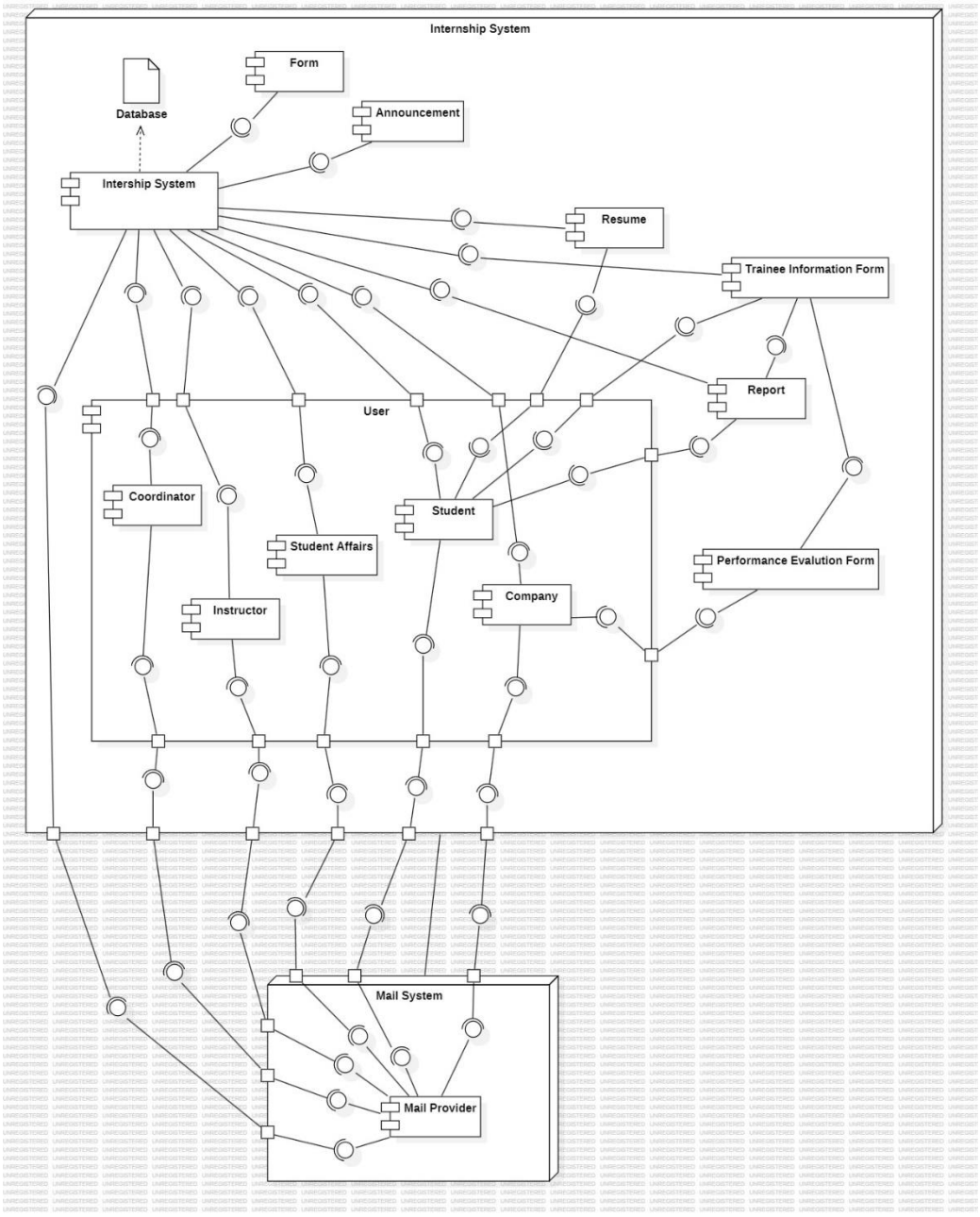

Figure 54: Component and Deployment Diagram

4 References

[1]: *What is a UML Diagram? | Different Types and Benefits | Miro*. (n.d.). https://miro.com/. https://miro.com/diagramming/what-is-a-uml-diagram

[2]: GeeksforGeeks. (2024, October 10). *Sequence diagrams Unified Modeling Language (UML)*. GeeksforGeeks. https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams

[3] *What is a Data Flow Diagram*. (n.d.). Lucidchart. https://www.lucidchart.com/pages/data-flow-diagram

[4], [5]: Wikipedia contributors. (2023, October 18). *Glossary of Unified Modeling Language terms*. Wikipedia. https://en.wikipedia.org/wiki/Glossary_of_Unified_Modeling_Language_terms

[6] 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

5 Appendices

5.1 Project Scheduling

5.1.1 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

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

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

Complete SDD - Introduction (Week 6-10)

- Task 1: Summary, Identified Stakeholders, and Design Concerns
- Responsible: Efekan, Mert

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

Complete SDD - Appendices (Week 6-10)

- Task 1: Project Scheduling
- Responsible: Mert, Ali
- Task 2: Sprint 2 Overview
- Responsible: Umutcan, Efekan

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

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

Practical Development - Database Operations (Week 10-13)

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

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 2: Preparation for the Presentation
- Responsible: Umutcan, Ali, Efekan, Mert

5.1.2 Gantt Chart

You can examine the Gantt chart from Figure 55. Since it is large, if you want to examine it in detail, you can see it divided into two in Figures 56 and 57. The meanings of the tasks written in the Gantt chart are explained in 5.1.1.

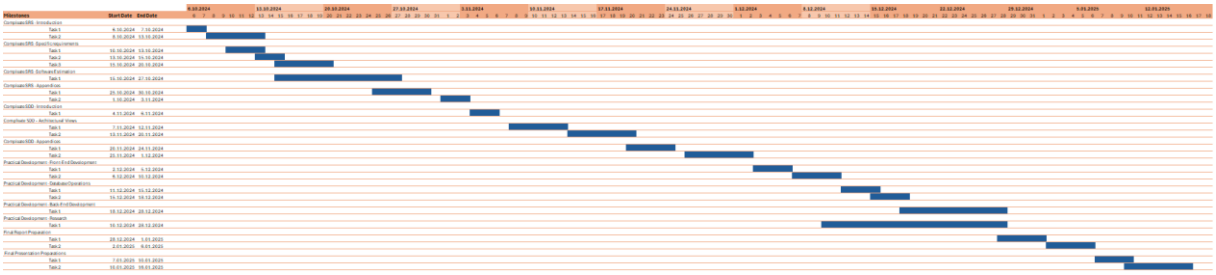

Figure 55: Gantt Chart-general

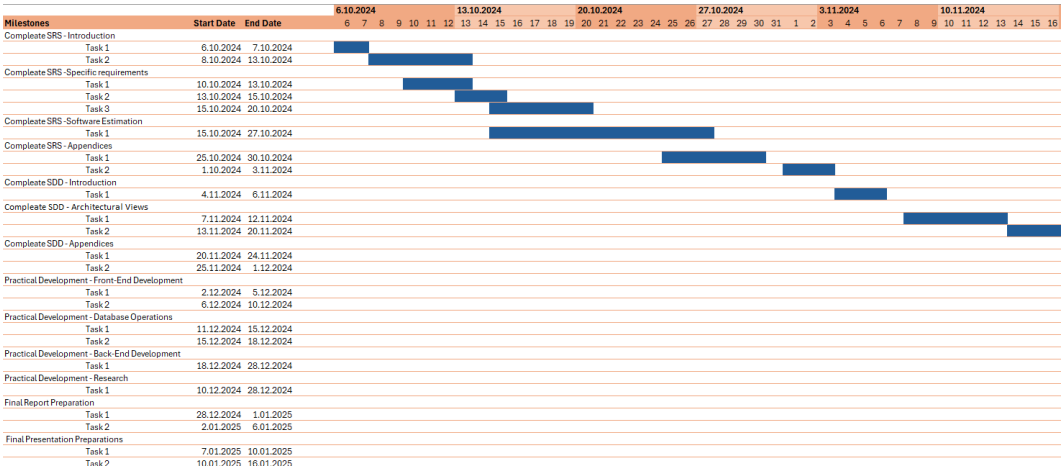

Figure 56: Gantt Chart-general

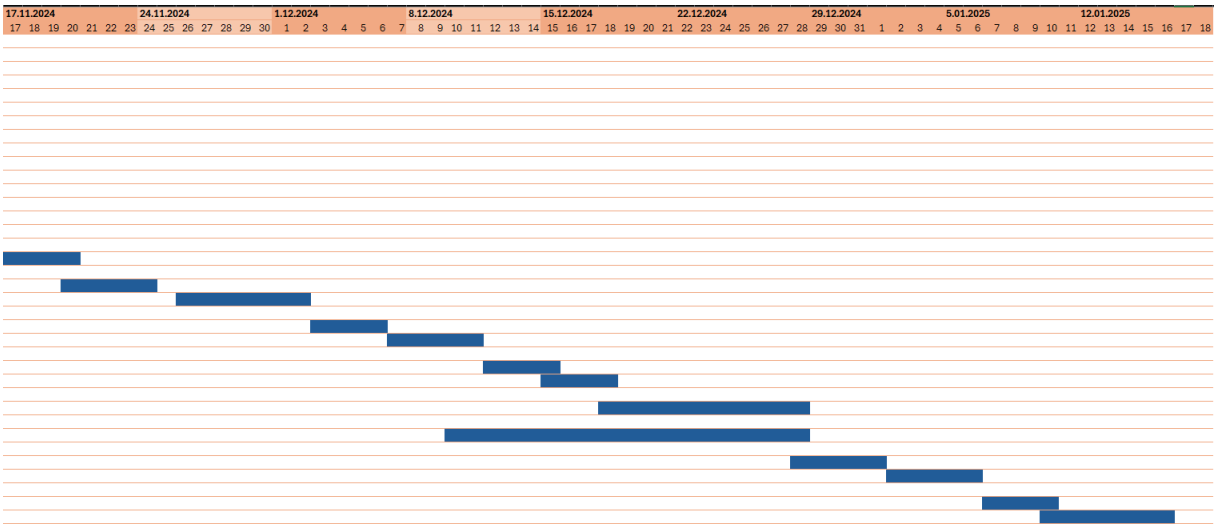

Figure 57: Gantt Chart-genera

5.2 Sprint 2 Overview

5.2.1 Sprint Backlog

In this sprint, the following tasks were completed along with the time allocated for each:

- Identifying and scheduling tasks - 8 hours: This task had a medium priority and focused on identifying tasks, assigning them to members and foundational work.
- Activity Diagrams- 10 hours: A high-priority task that required a significant amount of effort, focusing on creating activity diagrams.
- Class Diagrams- 6 hours: A medium-priority task aimed at designing class contents and relationships.
- Sequence Diagrams- 11 hours: A high-priority task that required a significant amount of effort, focusing on creating sequence diagrams.
- Data Flow Diagram- 18 hours: A high-priority task that required a significant amount of effort, focusing on creating data flow diagrams.
- Deployment Diagram- 6 hours: A medium-priority task focused on component and system deployment.
- SDD Report Writing- 8 hours: A medium-priority task
- Correcting SRS Report- 4 Hours: A low-priority task involving minor corrections and adjustments based on the feedback our team has recieved.

The total effort spent on this sprint amounted to 71 hours, as shown in the work estimation chart, which tracks story points against the baseline for each week.

PBI ID	PBI (Product Backlog Item)	Priority	Hours
1	Identifying and Scheduling Tasks	Medium	8
2	Activity Diagrams	High	10
3	Class Diagram	Medium	6
4	Sequence Diagram	High	11
5	Data Flow Diagrams	High	18
6	Deployment Diagram	Medium	6
7	SDD Report Writing	Medium	8
8	Correcting SRS	Low	4
			Total = 71

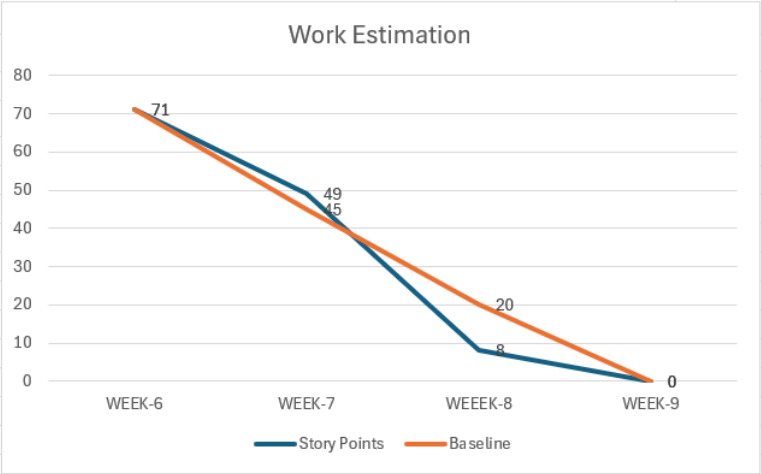

Figure 58: Sprint Backlog Estimation

5.2.2 Sprint Burndown Chart

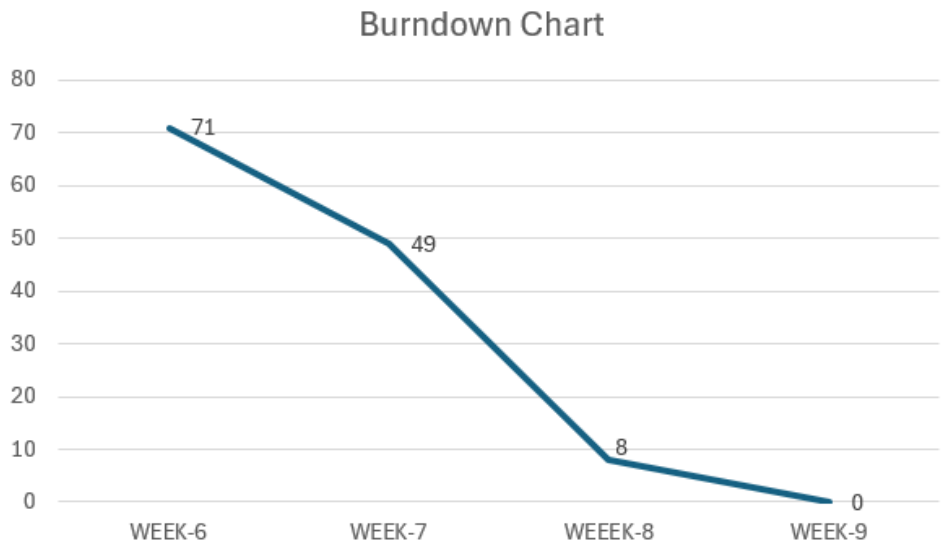

Figure 59: Burndown Chart

5.2.3 Sprint Review

In this sprint, we successfully completed the Software Design Description (SDD) report, achieving our primary goal of finalizing the design documentation for the project. This milestone sets the stage for the next phase, Software Prototype, where we will delve into the implementation and building of the system. Completing the SDD allowed us to gather and organize all relevant diagrams, define the architectural views, and establish a clear understanding of the project's structural design and software requirements.

During this sprint, we also identified some areas that could benefit from additional detail or clarification. In this phase our team also managed to work in a more coordinated way as a team, avoiding repeating our mistakes from the previous Software Requirements Specification and producing a better report.

5.2.4 Spring Retrospective

During this sprint, we encountered challenges in effectively managing our tasks, which led to delays in task allocation and completing development preparations. As a result, additional effort will be required in the next phase to ensure the timely implementation of the software prototype.

In the next phase, we plan to focus on database design, web interface development, mail service integration, and backend system implementation. Tasks will be distributed more effectively, aligning responsibilities with team members' skills and interests. By improving our workflow and collaboration, we aim to complete the next phase efficiently and meet the project objectives.