
Traineeship Application

Sprint Report

ΟΜΑΔΑ 5386-5324-5213

ΔΑΝΑΗ ΧΑΝΛΑΡΙΔΟΥ- 5386

**ΒΑΣΙΛΕΙΟΣ ΠΑΠΑΚΥΡΙΑΚΟΥ-
5324**

ΕΥΑΓΓΕΛΙΑ ΔΙΠΛΑ- 5213

VERSIONS HISTORY

Date	Version	Description	Author
10/03/2025	0.1	Initial requirements analysis and architecture design	5386-5324-5213
20/03/2025	0.2	Implementation of user registration and login (US1–US3)	5386-5324-5213
10/04/2025	1.0	Profile and traineeship position management (US4, US5, US7, US10)	5386-5324-5213
01/05/2025	2.0	Logbook functionality and traineeship viewing (US6, US8, US9, US13, US14)	5386-5324-5213
25/05/2025	3.0	Committee operations and final evaluation process (US16–US21)	5386-5324-5213

1 Introduction

This document provides information concerning the **1st to 4th sprint** of the *Traineeship Management App* of the project.

1.1 Purpose

The purpose of this document is to present the deliverables and development progress of the *Traineeship Management App* during the first four sprints of implementation. The application is intended to assist the traineeship committee of the university in managing internship opportunities offered by companies and assigned to students. The application supports distinct roles (students, professors, companies, committee members) and provides personalized features for each user group.

1.2 Document Structure

The rest of this document is structured as follows. Section 2 describes our Scrum team and specifies the this Sprint's backlog. Section 3 specifies the main design concepts for this release of the project.

2 Scrum team and Sprint Backlog

2.1 Scrum team

Product Owner	5386-5324-5213
Scrum Master	5386-5324-5213
Development Team	5386-5324-5213

2.2 Sprints

<List below the sprints that you performed and the user stories that have been realized in each Sprint>

Sprint No	Begin Date	End Date	Number of weeks	User stories
1	10/03/2025	30/03/2025	3	US1, US2, US3 – User Registration/Login/Logout
2	31/03/2025	20/04/2025	3	US4, US5, US7, US10 – Profiles & Announcements
3	21/04/2025	11/05/2025	3	US6, US8, US9, US11, US13, US14 – Logbooks & View
4	12/05/2025	25/05/2025	3	US16–US21 – Committee Operations & Evaluations

3 Use Cases

<Specify the concrete Use Cases that describe the interaction of the user with the applications, as derived from the abstract user stories. Give a **UML Use Case diagram** and the **detailed use case descriptions**.>

3.1 CreateAccount

Use case ID	1
Actors	User (Student / Company / Professor / Committee member)
Pre conditions	The user has not yet created an account.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the user navigates to the registration page. 2. The user fills out the required registration form. 3. The system verifies the input and creates a new account. 4. The user is redirected to the login page.
Alternative flow 1	If any required field is missing or invalid, the system prompts an error and requests correction.
Alternative flow 2	
Post conditions	A new user account is registered in the system.

3.2 CreateStudentProfile

Use case ID	2
Actors	Student
Pre conditions	The student is logged in.
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the student accesses the "My Profile" section.2. The system presents a form to input full name, university ID, interests, skills, and preferred location.3. The student submits the form.4. The profile is stored in the database.
Alternative flow 1	If any field is invalid or left blank, the form is not submitted and an error is shown.
Alternative flow 2	
Post conditions	The student's profile is created or updated.

3.3 PostTraineeshipPosition

Use case ID	3
Actors	Company
Pre conditions	The company is logged in and has an active profile.
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the company accesses the "New Position" section.2. The system provides a form for start/end date, description, skills, and topics.3. The company submits the form.4. The system adds the new traineeship position to the list of available positions.
Alternative flow 1	If required fields are missing, the system does not save the position and prompts the company
Alternative flow 2	

Post conditions	A new traineeship position is posted and visible to committee/students.
------------------------	---

3.4 AssignTraineeshipToStudent

Use case ID	4
Actors	Traineeship Committee
Pre conditions	A student has applied, and there are available positions.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the committee accesses the list of applicants. 2. The committee selects a student and requests matching positions. 3. The system provides suggestions based on interests/location/skills. 4. The committee assigns a position to the student.
Alternative flow 1	If no matching position is found, the system shows an empty result or suggestions to improve student profile.
Alternative flow 2	
Post conditions	The selected position is assigned to the student and marked as unavailable.

3.5 ApplyForTraineeship

Use case ID	5
Actors	Student
Pre conditions	The student is logged in and has completed their profile.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the student navigates to the list of available traineeships. 2. The student selects a traineeship and submits an application. 3. The system stores the application and notifies the committee.
Alternative flow 1	If the student profile is incomplete, the application is blocked.

Alternative flow 2	
Post conditions	An application is submitted and visible to the committee.

3.6 FillInLogbook

Use case ID	6
Actors	Student
Pre conditions	The student is assigned to an active traineeship.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the student accesses the logbook section. 2. The student enters daily or weekly activity reports. 3. The system stores and timestamps the entries.
Alternative flow 1	If the user tries to enter a log without being assigned, an error is shown.
Alternative flow 2	
Post conditions	Logbook entries are stored and can be accessed by supervisors.

3.7 DeleteTraineeshipPosition

Use case ID	7
Actors	Company
Pre conditions	The company is logged in and has at least one available traineeship.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the company navigates to the list of open positions. 2. The company selects one and chooses "Delete". 3. The system removes the position if it has not been assigned.

Alternative flow 1	If the position is already assigned, deletion is not allowed.
Alternative flow 2	
Post conditions	The position is removed from the list of available traineeships.

3.8 EvaluateTrainee (by Company)

Use case ID	8
Actors	Company
Pre conditions	The student has an in-progress traineeship assigned by this company.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the company selects a position and clicks "Evaluate". 2. A form is shown with rating criteria (motivation, effectiveness, efficiency). 3. The company submits the evaluation.
Alternative flow 1	If any required rating is missing, submission is rejected.
Alternative flow 2	
Post conditions	An evaluation is saved and visible to the committee.

3.9 EvaluateTrainee (by Professor)

Use case ID	9
Actors	Professor
Pre conditions	The professor is assigned as a supervisor.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the professor selects an assigned traineeship. 2. The professor fills out evaluation form for the student and company.

	3. The evaluation is submitted.
Alternative flow 1	If fields are incomplete, the form is not submitted.
Alternative flow 2	
Post conditions	The evaluation is stored and considered in final grading.

3.10 AssignSupervisor

Use case ID	10
Actors	Traineeship Committee
Pre conditions	A student has been assigned to a position.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the committee selects the traineeship. 2. The system suggests professors based on matching criteria or load. 3. The committee confirms the assignment.
Alternative flow 1	If no professor is available, the system prompts to try again later.
Alternative flow 2	
Post conditions	A professor is assigned as supervisor for the traineeship.

3.11 FinalizeEvaluationAndCompleteTraineeship

Use case ID	11
Actors	Traineeship Committee
Pre conditions	All evaluations (company & professor) have been submitted.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the committee opens the traineeship record.

	2. The system shows all evaluations. 3. The committee selects Pass/Fail and finalizes the process.
Alternative flow 1	If some evaluations are missing, finalization is blocked.
Alternative flow 2	
Post conditions	The traineeship is marked as complete with outcome stored.

3.12 Retrieve Applicants For Evaluation

Use case ID	12
Actors	Committee Member
Pre conditions	The user is a logged-in committee member.
Main flow of events	<p>1. The use case starts when the committee navigates to the "Student Applications" section.</p> <p>2. The system displays a list of students who have applied for a traineeship.</p> <p>3. The committee can view each student's profile and submitted applications.</p>
Alternative flow 1	If no applications exist, a message is shown.
Alternative flow 2	
Post conditions	The committee has full visibility of all current applicants.

3.13 View Candidate Positions

Use case ID	13
Actors	Committee Member

Pre conditions	The student has applied for a traineeship and completed their profile.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the committee selects a student from the application list. 2. The system suggests available positions based on the student's location, interests, and skills. 3. The committee assigns a selected position to the student.
Alternative flow 1	If no suitable positions are found, the system displays suggestions to improve the student profile.
Alternative flow 2	
Post conditions	The student is assigned to a traineeship position, and it is marked as unavailable.

3.14 ListUnsupervisedPositions

Use case ID	14
Actors	Committee Member
Pre conditions	There are traineeship positions without a supervisor assigned.
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the committee navigates to the "Unsupervised Positions" section. 2. The system displays a list of such positions. 3. The committee selects a professor and assigns them as supervisor.
Alternative flow 1	If no professors are available, the system prompts to try again later.
Alternative flow 2	
Post conditions	The selected professor is assigned to the traineeship.

3.15 FinalizeTraineeship

Use case ID	15
Actors	Committee Member
Pre conditions	All evaluations from company and professor have been submitted.
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the committee opens the trainee's record.2. The system displays submitted evaluations.3. The committee chooses "Pass" or "Fail" and finalizes the traineeship.
Alternative flow 1	If some evaluations are missing, finalization is not allowed.
Alternative flow 2	
Post conditions	The traineeship is marked as complete with the final result recorded.

3.16 DashboardAccess

Use case ID	16
Actors	Committee Member
Pre conditions	The committee member is logged in.
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the user navigates to the "Dashboard" section.2. The system displays summary data: active applications, assigned positions, pending evaluations, etc.
Alternative flow 1	If no data exists, the system displays an empty list or default message.

Alternative flow 2	
Post conditions	The committee member gains an overview of all active system operations.

4 Design

4.1 Architecture

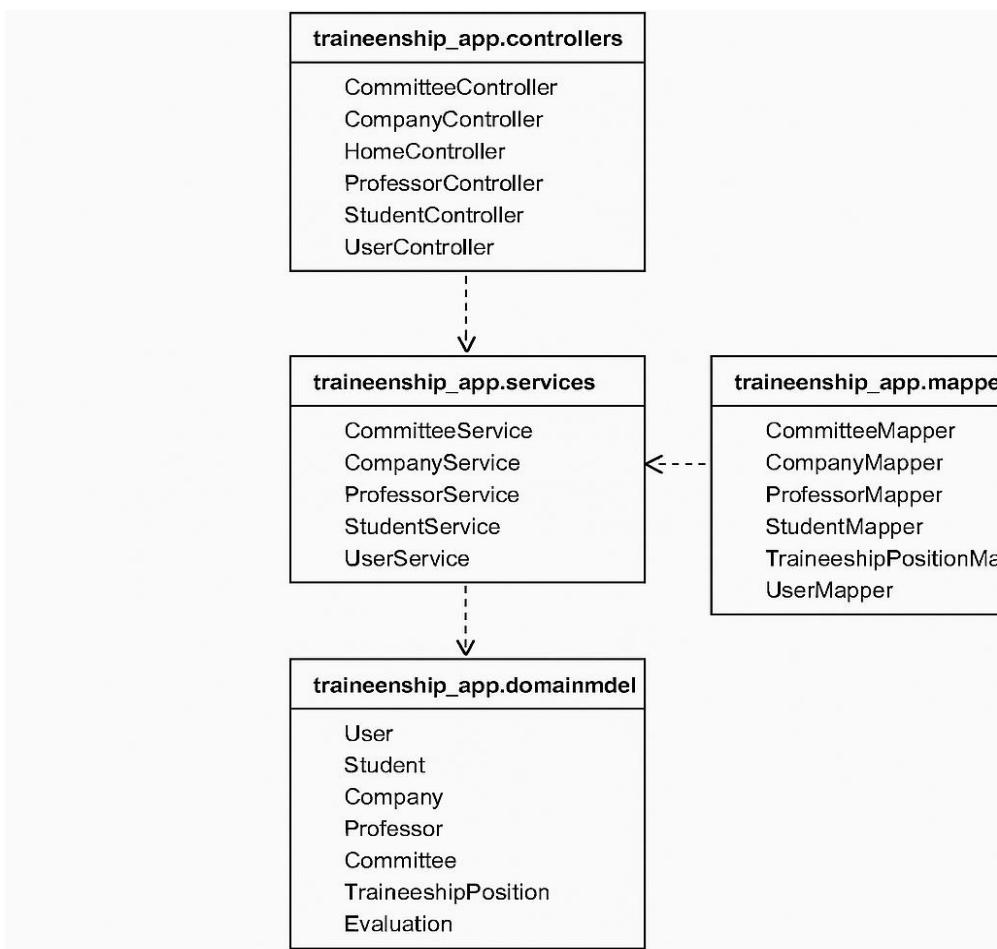


FIGURE 1: PACKAGE DIAGRAM

4.2 Design

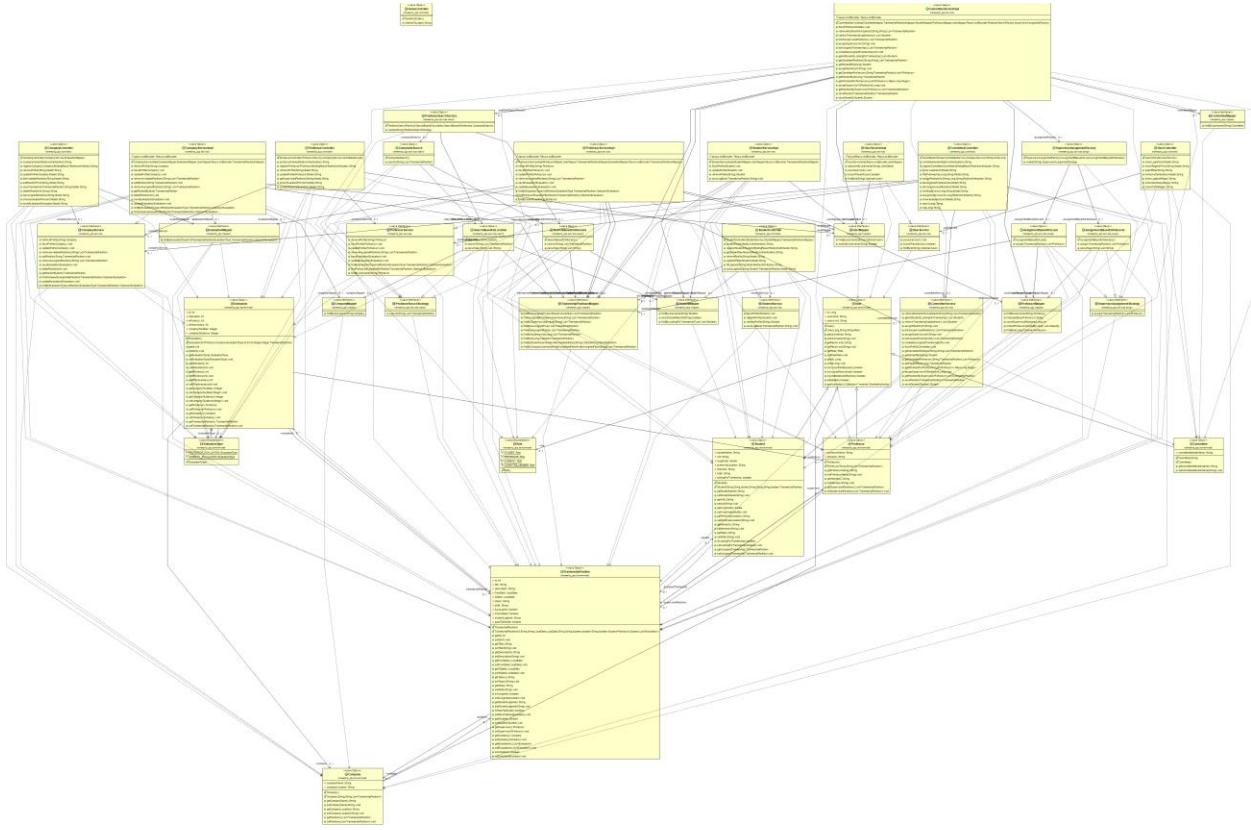


FIGURE 2: CLASS DIAGRAM (GO TO LAST PAGE FOR LARGER IMAGE)

Class Name: User (Domain Model)	
Responsibilities: <ul style="list-style-type: none">▪ Manage shared user attributes (name, email, password, role)▪ Provide basic accessor methods (getters/setters)	Collaborations: <ul style="list-style-type: none">▪ Student, Company, Professor, Committee

Class Name: Student (Domain Model)

Responsibilities: <ul style="list-style-type: none"> ▪ Extend User with student-specific information (e.g., interests, university ID) ▪ Fill in logbook, submit applications, receive evaluations 	Collaborations: <ul style="list-style-type: none"> ▪ TraineeshipPosition, Evaluation, Professor
--	---

Class Name: Company (Domain Model)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Manage company data (name, location) ▪ Post and delete traineeship positions ▪ Evaluate assigned students 	<ul style="list-style-type: none"> ▪ TraineeshipPosition, Evaluation

Class Name: Professor (Domain Model)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Supervise traineeships ▪ Evaluate both students and companies 	<ul style="list-style-type: none"> ▪ Student, Evaluation

Class Name: Committee (Domain Model)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Assign students to positions ▪ Assign supervising professors ▪ Finalize and complete traineeships 	<ul style="list-style-type: none"> ▪ Student, Company, TraineeshipPosition, Professor

Class Name: TraineeshipPosition (Domain Model)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Represent an available internship position ▪ Manage description, required skills, and duration 	<ul style="list-style-type: none"> ▪ Company, Student, Evaluation

Class Name: Evaluation (Domain Model)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Store evaluations submitted by professors and companies ▪ Contain evaluation criteria (e.g., motivation, effectiveness) 	<ul style="list-style-type: none"> ▪ Company, Professor, TraineeshipPosition, Student

Class Name: Role (Domain Model)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Define user types: STUDENT, COMPANY, PROFESSOR, COMMITTEE 	<ul style="list-style-type: none"> ▪ User

Class Name: StudentController (Controllers)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Handle student-related actions from the frontend ▪ Submit applications, record logbook entries, display traineeship details 	<ul style="list-style-type: none"> ▪ StudentService, TraineeshipService

Class Name: CompanyController (Controllers)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Handle company actions (posting, deleting positions, evaluations) ▪ Communicate with the company's services 	<ul style="list-style-type: none"> ▪ CompanyService, EvaluationService, TraineeshipService

Class Name: ProfessorController (Controllers)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Display supervised students ▪ Submit evaluations 	<ul style="list-style-type: none"> ▪ ProfessorService, EvaluationService

Class Name: CommitteeController (Controllers)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Manage the assignment process ▪ Final evaluation and completion of traineeships 	<ul style="list-style-type: none"> ▪ CommitteeService,StudentService, ProfessorService

Class Name: UserController (Controllers)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Handle user registration/login ▪ Retrieve and save user data 	<ul style="list-style-type: none"> ▪ UserService

Class Name: StudentService (Services)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Student business logic ▪ Submit logbook entries, fetch available positions 	<ul style="list-style-type: none"> ▪ StudentRepository, TraineeshipPositionRepository

Class Name: CompanyService (Services)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Company business logic ▪ Post/delete traineeship positions, record evaluations 	<ul style="list-style-type: none"> ▪ CompanyRepository, EvaluationRepository
Class Name: ProfessorService (Services)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Assign supervision and handle evaluations 	<ul style="list-style-type: none"> ▪ ProfessorRepository, StudentRepository

Class Name: CommitteeService (Services)	
Responsibilities:	Collaborations:
<ul style="list-style-type: none"> ▪ Select and assign traineeship positions 	<ul style="list-style-type: none"> ▪ StudentService,ProfessorService, TraineeshipService

<ul style="list-style-type: none"> ▪ Oversee the entire traineeship process 	
--	--

Class Name: UserService (Services)	
Responsibilities: <ul style="list-style-type: none"> ▪ User management (CRUD), authentication 	Collaborations: <ul style="list-style-type: none"> ▪ UserRepository

Class Name: StudentMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none"> ▪ Database access for Student entity 	Collaborations: <ul style="list-style-type: none"> ▪ Student

Class Name: CompanyMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none"> ▪ Database access for Company entity 	Collaborations: <ul style="list-style-type: none"> ▪ Company

Class Name: ProfessorMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none"> ▪ Database access for Professor entity 	Collaborations: <ul style="list-style-type: none"> ▪ Professor

Class Name: CommitteeMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none"> ▪ Database access for Committee entity 	Collaborations: <ul style="list-style-type: none"> ▪ Committee

Class Name: TraineeshipPositionMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none"> ▪ Database access for TraineeshipPosition entity 	Collaborations: <ul style="list-style-type: none"> ▪ TraineeshipPosition

Class Name: EvaluationMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none">▪ Database access for Evaluation entity	Collaborations: <ul style="list-style-type: none">▪ Evaluation

Class Name: UserMapper (Repository/Mappers)	
Responsibilities: <ul style="list-style-type: none">▪ Database access for User entity	Collaborations: <ul style="list-style-type: none">▪ User

