Traineeship Application

Sprint Report

Kotopoulos Vasilis, **5264**
Spanou Maria, **5351**
Gkovaris Christos-Grigorios, **5203**

VERSIONS HISTORY

Date	Version	Description	Author
29/3/2025 – 4/4/2025	V1.0	US1-15	The whole team
5/4/2025 – 15/4/2025	V2.0	US16-21	The whole team
16/4/2025 – 25/4/2025	V3.0	NF1,NF2, HTML Login, Logout, Student	Spanou Maria
26/4/2025 – 25/5/2025	V4.0	HTML Company, Professor, Committee, Report	The whole team

1 Introduction

This document provides information concerning the **final** sprint of the project.

1.1 Purpose

This document aims to report the activities and progress made during the final Sprint of the **Traineeship Management Application**. It summarizes the Scrum team structure, realized user stories, use cases, and design decisions.

1.2 Document Structure

The rest of this document is structured as follows. Section 2 describes out 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	Kotopoulos Vasilis (Manages backlog and prioritizes features)
Scrum Master	Gkovaris Christos-Grigorios (Ensures Scrum process is followed, facilitates meetings)
Development Team	Spanou Maria, Gkovaris Christos-Grigorios, Gkovaris Christos-Grigorios (Implements features, fixes bugs, writes tests)

2.2 Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories
Sprint 1	**29/3/2025**	**4/4/2025**	**1.5**	**US1 - US15**
Sprint 2	**5/4/2025**	**15/4/2025**	**1.5**	**US16 - US21**

Sprint 3	**24/5/2025**	**25/5/2025**	**0**	**UML Diagrams, CRC Cards**
Sprint 4	**25/5/2025**	**25/5/2025**	**0**	**Unit Tests, Fixes, Documentation**

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 <Use Case 1>

Use case ID	UC01
Actors	User
Pre conditions	The user is not registered in the system
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The user opens the registration form 2. The user fills in the required information (name, email, password, role) 3. The user submits the form 4. The system validates the input and creates a new account 5. A confirmation message is displayed
Alternative flow 1	Invalid input (e.g., missing fields) → error message shown
Alternative flow 2	Email already in use → error message shown
Post conditions	A new user account is created and stored in the system

3.2 <Use Case 2>

Use case ID	UC02
Actors	User

Pre conditions	The user has an existing account
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The user opens the login page 2. The user enters valid credentials 3. The system verifies the credentials 4. The user is redirected to the dashboard
Alternative flow 1	Wrong credentials → error message shown
Post conditions	The user is logged into the system

3.3 <Use Case 3>

Use case ID	UC03
Actors	User
Pre conditions	The user is logged in
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The user clicks the logout button 2. The session is terminated 3. The user is redirected to the home page
Post conditions	The user is logged out

3.4 <Use Case 4>

Use case ID	UC04
Actors	Student
Pre conditions	The student is logged in
Main flow of	[Main flow of events that describes the interaction between the user and the

events	application] 1. The student navigates to the profile creation page 2. The student fills in full name, university ID, interests, skills, preferred location 3. The student saves the profile
Alternative flow 1	Missing or invalid data → error message
Post conditions	The student's profile is stored in the system

3.5 <Use Case 5>

Use case ID	UC05
Actors	Student
Pre conditions	The student has a completed profile and is logged in
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The student browses available positions 2. The student selects a traineeship 3. The student submits an application
Alternative flow 1	Application already exists → warning message
Post conditions	An application is registered and visible to the committee

3.6 <Use Case 6>

Use case ID	UC06
Actors	Student
Pre conditions	The student has been assigned a traineeship
Main flow of events	[Main flow of events that describes the interaction between the user and the application]

	1. The student navigates to the logbook page 2. The student writes the log entry 3. The entry is saved
Alternative flow 1	Invalid format → error message
Post conditions	A new log entry is stored in the database

3.7 <Use Case 7>

Use case ID	UC07
Actors	Company
Pre conditions	The company is registered and logged in
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The company accesses the profile creation page 2. Company enters name and location 3. Company saves the profile
Post conditions	The company profile is stored in the system

3.8 <Use Case 8>

Use case ID	UC08
Actors	Company
Pre conditions	The company has created at least one position
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The company accesses the “My Positions” page 2. The system displays a list of available positions
Post conditions	The company views the list of advertised positions

3.9 <Use Case 9>

Use case ID	UC09
Actors	Company
Pre conditions	Some positions have been assigned to students
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The company accesses the “Assigned Positions” page 2. The system shows a list of assigned positions with student info
Post conditions	The company sees all assigned traineeships

3.10 <Use Case 10>

Use case ID	UC10
Actors	Company
Pre conditions	Company profile exists
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The company fills out form with title, description, dates, skills, topics 2. The position is submitted and published
Alternative flow 1	Required fields missing → error message
Post conditions	A new traineeship becomes available for students

3.11 <Use Case 11>

Use case ID	UC11
Actors	Company
Pre conditions	The traineeship is not yet assigned
Main flow of	[Main flow of events that describes the interaction between the user and the

events	application] 1. The company selects a position 2. Clicks “Delete” 3. The position is removed
Alternative flow 1	Assigned positions cannot be deleted → warning
Post conditions	The position is removed from the list

3.12 <Use Case 12>

Use case ID	UC12
Actors	Company
Pre conditions	The company has a student assigned
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. Company selects a traineeship 2. Company fills in evaluation (motivation, efficiency, effectiveness) 3. Submits the form
Post conditions	The evaluation is stored and visible to the committee

3.13 <Use Case 13>

Use case ID	UC13
Actors	Professor
Pre conditions	The professor is registered and logged in
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The professor navigates to the profile page 2. Fills in full name and interests 3. Saves the profile

Post conditions	The professor's profile is saved in the system

3.14 <Use Case 14>

Use case ID	UC14
Actors	Professor
Pre conditions	The professor is assigned to at least one traineeship
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The professor opens the "My Supervised Positions" page 2. The system shows the list of traineeships the professor supervises
Post conditions	The professor can view supervised traineeship positions

3.15 <Use Case 15>

Use case ID	UC15
Actors	Professor
Pre conditions	The professor supervises the traineeship
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The professor opens an evaluation form for a supervised student 2. Fills in motivation, effectiveness, efficiency 3. Evaluates company guidance/facilities 4. Submits the evaluation
Post conditions	The evaluation is stored and visible to the committee

3.16 <Use Case 16>

Use case ID	UC16
Actors	Traineeship Committee Member

Pre conditions	Students have submitted applications
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The committee member accesses the applications panel 2. Views list of students who applied
Post conditions	The member can browse student applications

3.17 <Use Case 17>

Use case ID	UC17
Actors	Traineeship Committee Member
Pre conditions	A student has applied and has a profile
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The committee selects a student 2. Searches for positions by interest or location 3. Filters positions based on required vs student skills
Alternative flow 1	No match found → empty result
Post conditions	The committee sees a tailored list of matching positions

3.18 <Use Case 18>

Use case ID	UC18
Actors	Traineeship Committee Member
Pre conditions	The student is eligible and a matching position exists
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The committee selects a student and a matching position

	2. Confirms the assignment 3. The position becomes unavailable
Post conditions	The student is linked to the position; position is marked as assigned

3.19 <Use Case 19>

Use case ID	UC19
Actors	Traineeship Committee Member
Pre conditions	The traineeship is assigned and no supervisor exists yet
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The committee chooses an assignment strategy (by interest or load) 2. Selects or accepts suggested professor 3. Assigns professor to the traineeship
Post conditions	The professor is linked to the traineeship position

3.20 <Use Case 20>

Use case ID	UC20
Actors	Traineeship Committee Member
Pre conditions	There are active traineeships in progress
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The member accesses the list of in-progress traineeships 2. The system displays current status and assignments
Post conditions	The member can monitor ongoing positions

3.21 <Use Case 21>

Use case ID	UC21

Actors	Traineeship Committee Member
Pre conditions	A traineeship has completed and evaluations exist
Main flow of events	[Main flow of events that describes the interaction between the user and the application] 1. The committee views evaluations from professor and company 2. Reviews ratings and comments 3. Assigns a final pass/fail decision
Post conditions	The traineeship is marked as complete with a final result

4 Design

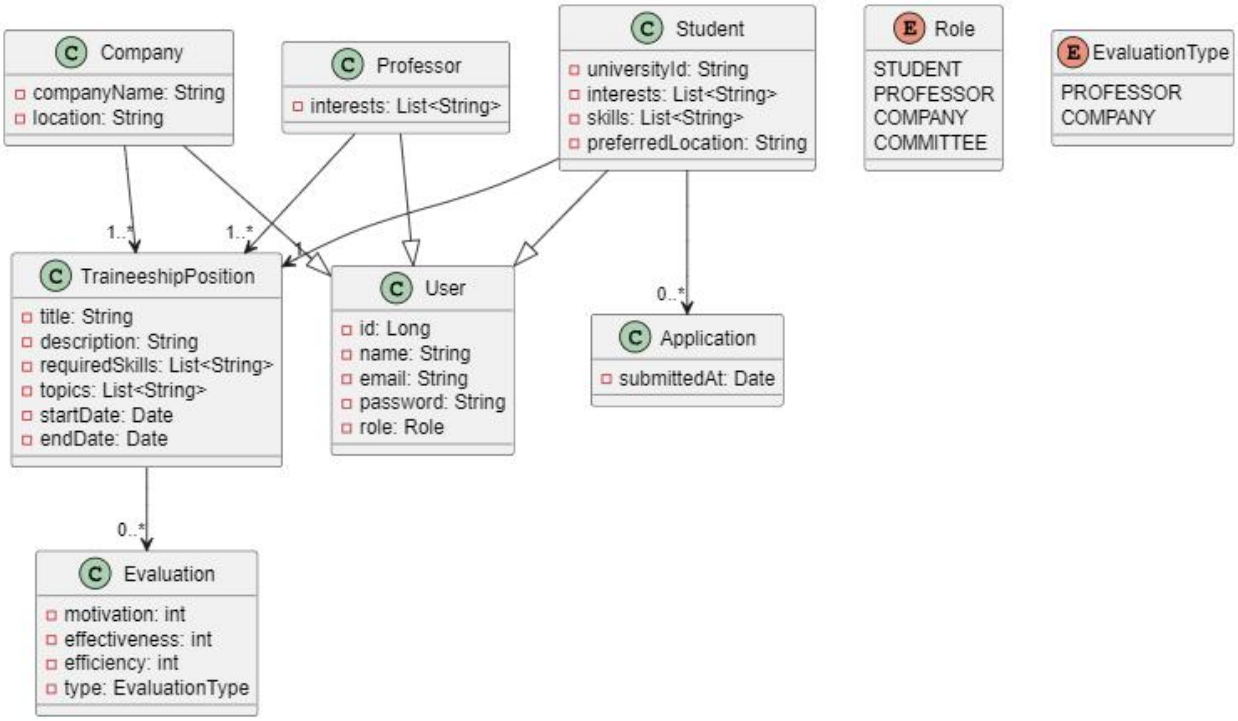
4.1 Architecture

The architecture of the **traineeship management system** follows a typical layered approach, separating concerns into different packages such as:

- **controller**: handles HTTP requests and forwards actions to services.
- **service**: contains business logic and orchestrates operations between controllers and repositories.
- **repository**: provides CRUD access to the database through JPA.
- **model**: contains the domain classes (Student, Company, TraineeshipPosition, etc.)
- **config**: holds configuration components such as security and professor assignment logic.

4.2 Design

The diagram below represents the **main entity classes** of the **traineeship management system**. The central abstraction is the **User class**, from which **Student**, **Company**, and **Professor** inherit, following a role-based hierarchy.



<Document the classes that are included in this release in terms of CRC cards according to the template that is given below.>

Class Name: User	
Responsibilities: ▪ Stores authentication credentials (username, password) ▪ Defines user role (Student, Company, Professor, Committee) ▪ Serves as superclass for all user types	**Collaborations:** ▪ Role (enum) ▪ Student, Company, Professor (inherit)

Class Name: Student	
Responsibilities: ▪ Stores student profile data (interests, skills, location) ▪ Applies for traineeship positions ▪ Submits logbook entries	**Collaborations:** ▪ Application ▪ TraineeshipPosition ▪ Evaluation ▪ Inherits from User

Class Name: Professor	
Responsibilities: ▪ Supervises assigned traineeships ▪ Submits evaluations for students and companies ▪ Manages professor profile (interests)	**Collaborations:** ▪ TraineeshipPosition ▪ Evaluation ▪ Inherits from User

Class Name: TraineeshipPosition	
Responsibilities: ▪ Stores internship offer data (description, dates, required skills) ▪ Links company, professor, and student ▪ Holds evaluation records	**Collaborations:** ▪ Company ▪ Professor ▪ Student ▪ Evaluation

Class Name: Application	
Responsibilities: ▪ Represents a student's application to a position ▪ Stores submission timestamp ▪ Used by the committee for matching and assignment	**Collaborations:** ▪ Student ▪ TraineeshipPosition

Class Name: Evaluation	
Responsibilities: ▪ Stores numerical evaluations from professor or company ▪ Identifies source of evaluation (EvaluationType) ▪ Supports final decision of committee	**Collaborations:** ▪ Professor, Company (as evaluators) ▪ TraineeshipPosition ▪ EvaluationType (enum)