Traineeship Application

Sprint Report

Vasileios Pistikopoulos 5336,

Georgia-Panagiota Kloukina 4701,

Eutuxia Markopoulou 5281

**Versions History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24-3-25 | 1.0 | Initialization of the Spring application  (pom.XML, application.properties) | 4701,5336 |
| 12-4-25 | 1.1 | Models / Database Entities Initialization (Data JPA) | 5336,5281 |
| 22-4-25 | 1.2 | Implementation of Authentication (Spring Security) | 5281 |
| 3-5-25 | 1.3 | Bug fixes | 4701 |
| 14-5-25 | 1.4 | DTOs, initial frontend | 5336,5281,4701 |
| 20-5-25 | 1.5 | Frontend improvements and Testing | 5336,5281,4701 |
| 23-5-25 | 2.0 | Final improvements | 5336,5281,4701 |

# Introduction

## Purpose

This application was built to make the process of managing student internships easier and more organized for everyone involved. It brings together students, companies, professors, and committee members on a single platform where each user can interact based on their role.

For students, the app offers a place to create a profile, browse available internships, apply for positions, and keep track of their progress through logbook entries. Companies can post internship openings, review applications, and later evaluate the interns they hosted. Professors can supervise specific students, approve or reject their internship choices, and monitor their development. Finally, committee members oversee the whole process, making sure everything runs smoothly and intervening when needed.

The goal of the project was to build a complete system that supports this workflow in a user-friendly and efficient way, combining backend logic with a functional and clean user interface.

## 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.

# Scrum team and Sprint Backlog

## Scrum team

|  |  |
| --- | --- |
| **Product Owner** | Mr. Zarras Apostolos |
| **Scrum Master** | Vasileios Pistikopoulos (5336) |
| **Development Team** | Georgia Kloukina Panagiota (4701), Vasileios Pistikopoulos (5336), Eutaxia Markopoulou (5281) |

## 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** | **15-3-2025** | **30-5-2025** | **2** | **Student** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 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**.>

**Use Cases for General Registration and Login**

## UC1 – Create Account

|  |  |
| --- | --- |
| **Use case ID** | UC1 |
| **Actors** | User |
| **Pre conditions** | The user does not already have an account. |
| **Main flow of events** | 1. The use case starts when the user navigates to the registration page. 2. The user fills in the registration form with required data (name, email, password, role). 3. The user submits the form. 4. The system validates the input and creates a new account. 5. A confirmation or success message is shown to the user. |
| **Alternative flow 1** | If the email is already in use, the system displays an error message and prompts the user to use a different email. |
| **Alternative flow 2** | If required fields are missing, the system highlights them and prevents submission. |
| **Post conditions** | The user account is stored in the database and the user can now log in. |

## UC2 – Login

|  |  |
| --- | --- |
| **Use case ID** | UC2 |
| **Actors** | User |
| **Pre conditions** | The user has already created an account and the account is active. |
| **Main flow of events** | 1. The use case starts when the user navigates to the login page. 2. The user enters their username/email and password. 3. The system checks the credentials against stored records. 4. If credentials are valid, the user is logged in and redirected to the dashboard. |
| **Alternative flow 1** | If credentials are invalid, an error message is shown and the user remains on the login page. |
| **Alternative flow 2** | If the account is inactive, the user is informed and prompted to contact support. |
| **Post conditions** | A session is created for the user and access to the application is granted. |

## UC3 – Logout

|  |  |
| --- | --- |
| **Use case ID** | UC3 |
| **Actors** | User |
| **Pre conditions** | The user is logged into the application. |
| **Main flow of events** | The use case starts when the user clicks the 'Logout' button.  The system invalidates the user's session.  The user is redirected to the homepage or login screen. |
| **Alternative flow 1** | If the session is already expired, the user is simply redirected to the login screen. |
| **Alternative flow 2** | - |
| **Post conditions** | The user's session is terminated and the application is in a neutral state. |

## **UC4 – Create Student Profile**

|  |  |
| --- | --- |
| **Use case ID** | UC4 |
| **Actors** | Student |
| **Pre conditions** | The student is logged in. |
| **Main flow of events** | 1. The use case starts when the student accesses the profile creation page. 2. The student fills in personal details (name, university ID, interests, skills, preferred location). 3. The student submits the profile form. 4. The system validates and stores the data. |
| **Alternative flow 1** | If required fields are missing, the system prompts the user to complete them. |
| **Alternative flow 2** | – |
| **Post conditions** | The student profile is saved and can be used for matching to traineeships. |

## UC5 – Apply for Traineeship

|  |  |
| --- | --- |
| **Use case ID** | UC5 |
| **Actors** | Student |
| **Pre conditions** | The student has a completed profile and is logged in. |
| **Main flow of events** | 1. The use case starts when the student navigates to the traineeship application section. 2. The student selects a traineeship position. 3. The student submits an application request. 4. The system records the application and notifies the committee. |
| **Alternative flow 1** | If the student has already applied for a position, the system prevents duplicate applications. |
| **Alternative flow 2** | – |
| **Post conditions** | The student's application is stored and ready for committee review. |

## UC6 – Fill Logbook

|  |  |
| --- | --- |
| **Use case ID** | UC6 |
| **Actors** | Student |
| **Pre conditions** | The student is assigned to an active traineeship. |
| **Main flow of events** | 1. The use case starts when the student accesses the logbook interface. 2. The student enters the daily or weekly activities. 3. The student submits the log entry. 4. The system stores the log in the student's record. |
| **Alternative flow 1** | If the log entry is incomplete, the system prompts for the missing data. |
| **Alternative flow 2** | - |
| **Post conditions** | The student's progress is recorded in the system logbook. |

## UC7 – Create Company Profile

|  |  |
| --- | --- |
| **Use case ID** | UC7 |
| **Actors** | Company |
| **Pre conditions** | The company user is logged in. |
| **Main flow of events** | 1. The use case starts when the company user navigates to the profile setup. 2. The user fills in the company name and location. 3. The form is submitted and the system stores the profile. |
| **Alternative flow 1** | If the fields are empty, the system requests completion. |
| **Alternative flow 2** | - |
| **Post conditions** | The company profile is now registered and visible to the committee. |

## UC8 – View Open Positions

|  |  |
| --- | --- |
| **Use case ID** | UC8 |
| **Actors** | Company |
| **Pre conditions** | The company has posted traineeship positions. |
| **Main flow of events** | 1. The use case starts when the company views the 'Open Positions' page. 2.  The system displays all available, unassigned traineeships posted by the company. |
| **Alternative flow 1** | - |
| **Alternative flow 2** | - |
| **Post conditions** | The company can monitor the status of open positions. |

## UC9 – View Assigned Positions

|  |  |
| --- | --- |
| **Use case ID** | UC9 |
| **Actors** | Company |
| **Pre conditions** | The company has assigned traineeship positions. |
| **Main flow of events** | 1. The use case starts when the company navigates to 'Assigned Positions'. 2. The system displays a list of positions currently assigned to students. |
| **Alternative flow 1** | - |
| **Alternative flow 2** | - |
| **Post conditions** | The company can track current internships and student assignments. |

## UC10 – Create Traineeship Position

|  |  |
| --- | --- |
| **Use case ID** | UC10 |
| **Actors** | Company |
| **Pre conditions** | The company profile is completed and the user is logged in. |
| **Main flow of events** | 1. The use case starts when the company selects 'Create New Position'. 2. The user fills in start/end dates, description, required skills, and topics. 3. The form is submitted and validated. 4. The position becomes available for viewing by the committee and students. |
| **Alternative flow 1** | If the form is incomplete, the system prompts for missing information. |
| **Alternative flow 2** | - |
| **Post conditions** | The traineeship position is published and open for applications. |

## UC11 – Delete Traineeship Position

|  |  |
| --- | --- |
| **Use case ID** | UC11 |
| **Actors** | Company |
| **Pre conditions** | The position is still open and not assigned. |
| **Main flow of events** | 1. The use case starts when the company selects a position to delete. 2. The system prompts for confirmation. 3. Upon confirmation, the position is removed from the listings. |
| **Alternative flow 1** | If the position is already assigned, deletion is blocked and an error is shown. |
| **Alternative flow 2** | - |
| **Post conditions** | The position is no longer visible to students or committee. |

## UC12 – Evaluate Trainee (Company)

|  |  |
| --- | --- |
| **Use case ID** | UC12 |
| **Actors** | Company |
| **Pre conditions** | A student is actively completing a traineeship at the company. |
| **Main flow of events** | 1. The company accesses the evaluation form. 2. The user enters ratings for motivation, effectiveness, and efficiency (scale 1–5). 3. The form is submitted and saved. |
| **Alternative flow 1** | If ratings are not entered, the system prevents submission. [Alternative flow of events IF ANY that takes place in case of an exception] |
| **Alternative flow 2** | - |
| **Post conditions** | The evaluation is stored and becomes accessible to the committee. |

## UC13 – Create Professor Profile

|  |  |
| --- | --- |
| **Use case ID** | UC13 |
| **Actors** | Professor |
| **Pre conditions** | The professor is logged in. |
| **Main flow of events** | 1. The use case starts when the professor navigates to profile setup. 2. The professor fills in name and interests. 3. The system validates and saves the profile. |
| **Alternative flow 1** | If fields are missing, the system requests completion. |
| **Alternative flow 2** | - |
| **Post conditions** | The professor's profile is saved and can be used for supervision assignments. |

## UC14 – View Supervised Positions

|  |  |
| --- | --- |
| **Use case ID** | UC14 |
| **Actors** | Professor |
| **Pre conditions** | The professor is assigned to one or more traineeship positions. |
| **Main flow of events** | 1. The professor accesses the 'Supervised Positions' section. 2. The system displays a list of all traineeships assigned to the professor. |
| **Alternative flow 1** | - |
| **Alternative flow 2** | - |
| **Post conditions** | The professor can review and monitor the assigned traineeships. |

## UC15 – Evaluate Traineeship (Professor)

|  |  |
| --- | --- |
| **Use case ID** | UC15 |
| **Actors** | Professor |
| **Pre conditions** | The professor is supervising a traineeship. |
| **Main flow of events** | 1. The professor opens the evaluation form for a supervised traineeship. 2. The professor fills in ratings for student (motivation, effectiveness, efficiency). 3. The professor also evaluates the company (facilities, support). 4. The form is submitted and saved. |
| **Alternative flow 1** | If fields are missing, submission is blocked. |
| **Alternative flow 2** | - |
| **Post conditions** | The evaluation is saved and available to the committee. |

## UC16 – View Student Applications

|  |  |
| --- | --- |
| **Use case ID** | UC16 |
| **Actors** | Committee Member |
| **Pre conditions** | There are student applications submitted in the system. |
| **Main flow of events** | 1. The committee member accesses the list of student applicants. 2. The system displays all pending traineeship applications. |
| **Alternative flow 1** | - |
| **Alternative flow 2** | - |
| **Post conditions** | Committee members can review all active applications. |

## UC17 – Search Matching Positions

|  |  |
| --- | --- |
| **Use case ID** | UC17 |
| **Actors** | Committee Member |
| **Pre conditions** | There are available positions and student applications. |
| **Main flow of events** | 1. The committee member selects a student from the applicant list. 2. The member sets filters (interests, location, skills). 3. The system displays matching traineeship positions. |
| **Alternative flow 1** | If no matches are found, a message is shown and filters can be adjusted. |
| **Alternative flow 2** | - |
| **Post conditions** | A list of relevant positions is available for assignment. |

## UC18 – Assign Position to Student

|  |  |
| --- | --- |
| **Use case ID** | UC18 |
| **Actors** | Committee Member |
| **Pre conditions** | A student has applied and there is an available traineeship position. |
| **Main flow of events** | 1. The committee member selects a student. 2. The member chooses a matching traineeship position. 3. The system assigns the position to the student. |
| **Alternative flow 1** | If the position is already assigned, an error is shown. |
| **Alternative flow 2** | - |
| **Post conditions** | The student is now officially assigned to a traineeship. |

## UC19 – Assign Supervisor

|  |  |
| --- | --- |
| **Use case ID** | UC19 |
| **Actors** | Committee Member |
| **Pre conditions** | There is an in-progress traineeship without a professor assigned. |
| **Main flow of events** | 1. The committee member accesses the list of in-progress traineeships. 2. Selects a traineeship. 3. Searches for a professor based on interests or workload. 4. Assigns the professor to the traineeship. |
| **Alternative flow 1** | If no matching professor is found, a message is displayed. |
| **Alternative flow 2** | - |
| **Post conditions** | The professor is assigned as the supervisor. |

## UC20 – View Ongoing Traineeships

|  |  |
| --- | --- |
| **Use case ID** | UC20 |
| **Actors** | Committee Member |
| **Pre conditions** | There are active traineeships in the system. |
| **Main flow of events** | 1. The committee member opens the 'Ongoing Traineeships' section. 2. The system displays all currently active assignments. |
| **Alternative flow 1** | - |
| **Alternative flow 2** | - |
| **Post conditions** | Committee can monitor all ongoing traineeships. |

## UC21 – Final Evaluation Decision

|  |  |
| --- | --- |
| **Use case ID** | UC21 |
| **Actors** | Committee Member |
| **Pre conditions** | Evaluations for a traineeship have been submitted. |
| **Main flow of events** | 1. The committee member selects a traineeship from the active list. 2. Reviews all evaluations (from professor and company). 3. Chooses to mark the traineeship as 'Pass' or 'Fail'.   System updates the traineeship status. |
| **Alternative flow 1** | If evaluations are missing, the system blocks finalization. |
| **Alternative flow 2** | - |
| **Post conditions** | The traineeship is completed and status is finalized. |

# Design

## Architecture

<Specify the overall architecture for this release in terms of a **UML package diagram**.>

## Design

<Specify the detailed design for this release in terms of **UML class diagrams**.>

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

|  |  |
| --- | --- |
| **Class Name: ....** | |
| **Responsibilities:**   * ... * .... * ... | **Collaborations:**   * ... * .... * .... |

|  |  |
| --- | --- |
| **Class Name: ....** | |
| **Responsibilities:**   * ... * .... * ... | **Collaborations:**   * ... * .... * .... |

...................................................

...................