

---

Traineeship Management Application

Product Backlog Specification

---

---

## VERSIONS HISTORY

---

Date	Version	Description	Author
2/18/25	<1.0>	1st version of the requirements definition document	A. Zarras

## 1 Introduction

---

The objective of this project is to develop an application that allows the traineeship committee of the University to manage and monitor open and assigned traineeship positions. Specifically, the application shall allow companies to announce open traineeship positions. The students will be able to look for available traineeship positions. The traineeship committee shall assign positions to students via different alternative criteria. The traineeship committee will further allocate professors as supervisors to the assigned traineeship positions. Professors and companies will be responsible for the final evaluations of the students' traineeships.

The rest of this document is structured as follows. In Section 2 we focus on the development process that will be followed and other scoring and organizational issues. Sections 3 and 4 provide the Product Backlog, i.e., the "raw" functional and non-functional requirements that should be further analyzed to drive the design, implementation and testing of the application.

## 2 Development process and organization issues

---

To realize the project, we shall rely on a Scrum approach. Each team shall organize a number of sprints during which the team shall implement **user stories** from the project backlog and their **tests**. The deadline for the project is: **26/5/2025**.

### 2.1 Deliverables

---

**Definition of "done" story:** A user story is done if it is **implemented correctly** and validated with one or more appropriate **tests**.

At the end of the project the Scrum team shall **deliver (via turnin)**

- **The project implementation.**
- **A Sprint report**, according to the given **Sprint report template (SprintReport-v0.doc)**, describing **the sprints that they performed and the "done" user stories that have been developed during each sprint**. The report should also comprise the specification of **detailed use cases, derived from the given user stories**, the **detailed design** of the application and **CRC cards** that document the responsibilities and the collaborations between the different part of the application.
- Turn in the **project** and the other deliverables using **turnin deliverables@myy803 <your-project>.zip**, where your-project is a zip file of your Eclipse project.

### 2.2 Scoring

---

1. Working implementations of the **user stories** is **60%** of the total score.
2. **Acceptance, integration and unit tests** are **15%** of the total score.

3. **Design quality**, usage of recommended **patterns** and **best practices** to satisfy the extensibility and maintainability requirements is **15%** of the total score.
4. The quality of **reporting** is **10%** of the total score.

### 3 Functional Requirements / User Stories

---

#### 3.1 General user stories

---

The following user stories concern general registration and login actions.

US ID	User story
<b>US1</b>	As a user, I want to be able to create a new account, so that I have access to the functionalities of the traineeship management application.
<b>US2</b>	As a user, I want to be able to login my account to start using the application.
<b>US3</b>	As a user, I want to be able to log out from my account to terminate my interaction with the traineeship management application.

#### 3.2 User stories for the students

---

The user stories below allow a student to personalize the traineeship allocation process.

US ID	User story
<b>US4</b>	As a student, I want to be able to create a profile that includes my full name, my university ID number, a list of interests, a list of skills, and a preferred location where I would like to pursue my traineeship.
<b>US5</b>	As a student I want to apply for a traineeship to the traineeship committee.
<b>US6</b>	As a student I want to be able to fill in my traineeship logbook to report what I do in the traineeship that has been assigned to me.

#### 3.3 User stories for companies

---

The following user stories allow a company to manage a list of traineeship positions, open or assigned to students.

US ID	User story
<b>US7</b>	As a company, we want to be able to create a profile that includes the company name and location.
<b>US8</b>	As a company we want to have access to the list of available traineeship positions that I have advertised.
<b>US9</b>	As a company we want to have access to the list of traineeships positions that have been assigned to students.

<b>US10</b>	As a company we want to be able to announce an available traineeship position. The announcement shall include start and end dates, a short description of the work to be done, a list of required skills and a list of topics of interest that relate to the traineeship position.
<b>US11</b>	As a company we want to be able to delete a traineeship position that is no longer available.
<b>US12</b>	As a company we want to be able to fill in an evaluation for a traineeship that is in progress. The evaluation should rate the student motivation, effectiveness and efficiency on a scale from 1 (bad) to 5 (perfect).

### 3.4 User stories for professors

---

The following user stories allow a professor to manage a list of assigned traineeship positions that he/she supervises.

<b>US ID</b>	<b>User story</b>
<b>US13</b>	As a professor I want to be able to create a profile that includes my full name and a list of interests.
<b>US14</b>	As a professor I want to have access to the list of traineeship positions that I supervise.
<b>US15</b>	As a professor I want to be able to fill in an evaluation for a traineeship that I supervise. The evaluation should rate the student motivation, effectiveness and efficiency on a scale from 1 (bad) to 5 (perfect). The evaluation should also evaluate the hosting company in terms of the facilities provided and the guidance of the trainee student on a scale from 1 (bad) to 5 (perfect).

### 3.5 User stories for the traineeship committee

---

The following user stories allow the traineeship committee to assign and monitor traineeships to students.

<b>US ID</b>	<b>User story</b>
<b>US16</b>	As a traineeship committee member, I want to have access to the list of students who applied for a traineeship position.
<b>US17</b>	As a traineeship committee member, I want to be able to select a student who applied for a traineeship position and look for a list of available traineeship positions for the student. The search process should be done based on the student's interests or the student's preferred location, or both. The list should include only available positions with required skills that match the student's skills.
<b>US18</b>	As a traineeship committee member, I want to be able to assign a traineeship position to a student.
<b>US19</b>	As traineeship committee, I want to be able to assign a supervising professor to an in-progress traineeship position. The selection of the professor should be done based on the professor's interests or the professor's load (i.e. number of supervised traineeships).
<b>US20</b>	As a traineeship committee member, I want to have access to the list of traineeships that are

	in progress.
<b>US21</b>	As a traineeship committee member, I want to be able select a traineeship that is in progress, monitor the evaluations of the position and complete the whole process with a pass or fail mark depending on the evaluations.

## 4 Non-Functional Requirements

---

**[NF1] Maintainability:** In software engineering, maintainability is the degree of effectiveness and efficiency with which a product or system can be modified by the maintainers. In the case of this project, we specifically focus on the following concerns:

- **[NF1.1]** A first important concern is to employ an architecture that promotes low coupling and high cohesion. To this end, you can employ **Fowler's enterprise application architecture patterns** [2] that allow to clearly separate the different parts (views, controllers, domain model, database) of the application and facilitate the mapping of the domain model to the underlying database schema.
- **[NF1.2]** In the long term, we want to be able to easily **extend** and **configure** the application with **different traineeship search strategies, without having to change the code of the application.** The initial set of **strategies** that will be supported by the application should include the following strategies.
  - The first strategy, calculates a list of available traineeship positions for a given student which match the student's interests. The matching of the student's interests with the position's topics is done based on Jaccard similarity metric. For a set of interests, I and a set of topics T  $J(I, T) = \frac{|\text{intersection}(I, T)|}{|\text{union}(I, T)|}$ . The similarity should be greater than a given threshold.
  - The second strategy calculates a list of available traineeship positions for a given student which match the student's preferred location with the location where the company that offers the position is situated.
  - The third strategy combines the first two, it calculates a list of available traineeship positions for a given student which match the student's interests and preferred location.
- **[NF1.3]** In the long term we want to be able to easily **extend** and **configure** the application with **different supervisor allocation strategies.** The initial set of **strategies** that will be supported by the application should include the following strategies.
  - According to the first strategy the interests of the supervisor should match the position's topics. The matching of the professor's interests with the position's topics is done based on Jaccard similarity metric. For a set of interests I and a set of topics T  $J(I, T) = \frac{|\text{intersection}(I, T)|}{|\text{union}(I, T)|}$ . The similarity should be greater than a given threshold.

- According to the second strategy the allocated supervisor is the professor with the minimum load (i.e., number of supervised traineeships).

**HINT** To satisfy the [NF1.2] and [NF1.3] requirements the application should be designed according to well-known **principles** and exploit **best practices** like the **GoF design patterns**. **Specifically, you can employ the GoF strategy pattern and possibly the GoF template method pattern for the implementation of alternative/interchangeable strategies.**

**[NF2] Usability:** In software engineering usability concerns the ease of use and learnability. In the context of this project the application should provide a simple and user-intuitive interface. The application should also provide help, in the form of user guidelines concerning its main functionalities of the application.

## 5 IDE, Java and External API Requirements/Constraints/Recommendations

---

The application should be implemented in **Java**. You can use the **Spring framework**. Spring and lately **Spring Boot** are very popular technologies that facilitate the development of **Web-based applications**. As a base database management system, you can use **MySQL**. To write automated tests for the application you can use **JUnit and Mockito**. The preferred IDE is **Eclipse**.

## 6 References

---

[1] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

[2] Martin Fowler. *Catalog of Patterns of Enterprise Application Architecture*. Addison-Wesley. <https://martinfowler.com/eaCatalog/>