Diploma Projects Management App

Product Backlog Specification

VERSIONS HISTORY

Date	Version	Description	Author
2/8/23	<1.0>	1st version of the requirements definition document	A. Zarras

1   Introduction

The objective of this project is to develop a Web application that allows students to browse available diploma thesis projects from various professors and apply for the diploma thesis projects that interest them. The application further allows professors to assign diploma thesis projects to students, supervise the assigned theses projects and assess the outcomes.

The rest of this document is structured as follows. In Section 2 we focus on the development process that shall 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: 18/5/2023.

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 shall 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 **50%** of the total score.

2. **Acceptance, integration and unit tests** is **20%** of the total score.

3. **Design qualit**y, usage of recommended **patterns** and **best practices** to satisfy the extensibility and maintainability requirements is **20%** of the total score.

4. Quality of **reporting** is **10%** of the total score.

3 Functional Requirements / User Stories

3.1 General user stories

ID	AS A <User Type>	I WANT <An Action>	SO THAT <A Benefit/Value>
U1	User	To create a new account to the application with my role (STUDENT or PROFESSOR), username and password.	I can use the application.
U2	User	To login to the application with my user name and password.	I can use the provided functionalities.

3.2 Student user stories

ID	AS A <User Type>	I WANT <An Action>	SO THAT <A Benefit/Value>
S1	Student	To set personal profile information (full name, year of studies, current average grade, number of remaining courses for graduation).	The professors can evaluate of my application to available diploma thesis subjects.
S2	Student	To have access to a list of available diploma thesis subjects that are offered by the professors.	I apply to the ones that look interesting.
S3	Student	To see a more detailed description of a diploma thesis subject (title, objectives, supervisor).	I choose if I will actually apply for the thesis.
S4	Student	Apply for a diploma thesis subject.	I let the professor know that I am willing to take over the diploma thesis subject.

3.3 Professor user stories

ID	AS A <User Type>	I WANT <An Action>	SO THAT <A Benefit/Value>
P1	Professor	To set personal profile information (full name and specialty).	The students can find out more about me before applying to

			available diploma thesis subjects.
P2	Professor	To have access to the list of the **available** diploma thesis subjects that I offer.	I can manage related information.
P3	Professor	Add a new diploma thesis subject (title, objectives) to my list.	I keep my list up to date.
P4	Professor	Delete a diploma thesis subject from my list.	I keep my list up to date.
P5	Professor	See the list of applications from the students who want to take over a diploma thesis subject.	I assign the diploma thesis subject to a particular student.
P6	Professor	To assign a diploma thesis subject to one of the students who applied for the project via a strategy selected from the following list: ▪ random choice, ▪ student with the best average courses grade, ▪ student with the fewest remaining courses for graduation, ▪ student with average courses grade greater than a given threshold Th1 and number of remaining courses for graduation less than a given threshold Th2.	The student can start working on the project.
P7	Professor	To have access to the list of assigned diploma thesis projects that I supervise.	I manage related information.
P8	Professor	Set the grades for the implementation, the report and the presentation of a diploma thesis subject that I supervise.	I evaluate the work done by the student.
P9	Professor	Calculate the overall grade of a diploma thesis that I supervise based on the following weighted average formula: total grade = 0.7 * implementation grade + 0.15 * report grade + 0.15 * presentation grade	I evaluate the work done by the student.

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:

- Be able to easily **extend** the application with **new strategies** for the assignment of diploma thesis subjects to students.

- Keep the application **independent** from the **database management system** that is going to be used for the data.

- Be able to easily **change** the **database management system** for another without significant modifications to the application logic.

- Be able to change or **add different views** for the interaction with the **provided services** (e.g. add a restful api layer for using the application programmatically).

To achieve these maintainability concerns the application should be designed according to the well known **principles** and exploit **best practices** like the **GoF design patterns** (Strategy, Template Method, Adapter, etc. [1]) and **Fowler's enterprise application architecture patterns** [2].

[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.** The application should be compatible at least with **Java 8**. You can further 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**. Moreover, a popular library for descriptive statistics is **Apache Math 3**. 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/eaaCatalog/