Laboratory Assignment AND Assessment Requirements Specification

Version 1.0

8 March 2020

Developed by:

     Hudema Dana

     Ilie Oana Andreea

     933,

Version History

Version	Description of Change	Author	Date
V01	Initial/Modification of document	Hudema Dana Ilie Oana Andreea	1 March 2020
V02	Completion of document	Hudema Dana	8 March 2020

		Ilie Oana Andreea	

Contents

1. Introduction

The application is written in Java and is designated for teachers to be able to assign themes to students and keep track of them.

1.1. Purpose

The application allows the user to easily manage a list of students, assign themes to them, mark grades for each theme, extend deadlines and manage delays.

1.2. Scope

The scope of the document is to give information about the system: regarding the users, functionalities, purpose, usability, data management and user scenarios.

1.3. Definitions, Acronyms, and Abbreviations

MAP – Metode Avansate de Programare

FR – Feature Reeuirement

1.4. Document Overview

The document is organized in chapters and subchapters describing the general purpose and scope of the document, the product description and requirements such as functional and user interface requirements.

2. Product/Service Description

The application allows the user to add students and themes. Also, he/she can assign a mark for a student on a theme. The user can modify at any time the student or delete it.

2.1. Product Context

The product is independent and self-contained.

2.2. User Characteristics

Users that will be using this product are university teachers teaching MAP subject.

3. Requirements

When the program starts, the input data is read from the following text files:

- Students.txt file, which contains information about idStudent (student's number), name, group, email, name of the professor
- Assignment.txt file, which contains the following information: laboratory number (unique identifier), brief description of the requirement, deadline – the week of the semester in which the assignment should be delivered (1. 14), the week in which the theme was received (1..14).

The teacher should be able to see all students and themes, add a student or a theme, delete, find or edit any information about a student.

Also, the teacher can assign a grade from 1 to 10 for each assignment. Each week of delay will be penalized by 2.5 points.

An assignment can be delivered at most 2 weeks after its deadline, otherwise it will be marked with 1.

The file Catalog.txt will store information about the grades. For each grade, it will contain the id of the grade, the name of the student, the lab id and the value of the mark given.

3.1. Functional Requirements

List the functional requirements (FR) of the system.

Section/ Requirement ID	Requirement Definition
FR1.0	Implement CRUD operations for the Student entity
FR1.1	Add student
FR1.2	Edit student
FR1.3	Delete student
FR1.4	Display all students
FR1.5	Display specific student
FR2.0	Manage laboratory themes and subjects.
FR2.1	Add a new laboratory theme
FR2.2	Edit laboratory theme
FR2.3	Delete laboratory theme
FR2.4	Display all laboratory themes
FR2.5	Extend the deadline for an existing subject
FR3.0	Manage grading
FR3.1	Add a grade to a particular laboratory theme to a particular student
FR3.2	Delete a grade to a particular laboratory theme to a particular student
FR3.3	Display all grades
FR3.4	Find a grade from a particular laboratory theme to a particular student
FR4.0	Search students based in different criteria
FR5.0	Generate reports
FR5.1	Send weekly reports about grading through email
FR5.2	Report of laboratory grade for each student
FR5.3	Report of hardest theme
FR5.4	Report of students who can enter the exam
FR5.5	Report of students who have delivered all the themes on time
FR6.0	Generate notifications

Section/ Requirement ID	Requirement Definition
FR6.1	Notify students by email when adding a new laboratory theme or modifying the delivery date of an existing subject

3.2. User Interface Requirements

The user should be presented a menu where each option is describing one of the functional requirements. After choosing an option the program should ask the user to enter the needed information.

3.3. Usability

- The user documentation and help should be complete
- The help should be context sensitive and explain how to achieve common tasks
- The system should be easy to learn.

3.4. Data Management

The data should be stored in text files.

4. User Scenarios/Use Cases

The application allows the user to:

- Add a student
- Edit student
- Delete student
- Find student
- Show all students
- Add a laboratory theme
- Delete a laboratory theme
- Edit a laboratory theme
- Show all themes
- Extend a deadline
- Assign a grade to a student on a theme
- Delete grade of a student on assigned theme
- Edit grade
- Find grade
- Show all grades
- Find students
- Generate reports about students
- Notify students when adding a new laboratory theme or modifying the delivery date of an existing subject

Please refer to Analysis and Design Document.