

## **Laboratory Assignment AND Assessment Requirements Specification**

**Version 1.0**

**8 March 2020**

Developed by:

Chereji Iulia-Maria

Campean Catalin-Alexandru

932,

Version History

<b>Version</b>	<b>Description of Change</b>	<b>Author</b>	<b>Date</b>
V01	Improvement of document	Both students	19 March 2023
V02	Completion of document	Both students	19 March 2023

## Contents

Laboratory Assignment AND Assessment Requirements Specification .....	1
Version 1.0 .....	1
8 March 8, 2020 .....	1
1. Introduction .....	3
1.1. Purpose .....	3
1.2. Scope .....	3
1.3. Definitions, Acronyms, and Abbreviations .....	3
1.4. Document Overview .....	3
2. Product/Service Description .....	3
2.1. Product Context .....	3
2.2. User Characteristics .....	3
3. Requirements .....	3
3.1. Functional Requirements .....	4
3.2. User Interface Requirements .....	4
3.3. Usability .....	4
3.4. Data Management .....	5
4. User Scenarios/Use Cases .....	5

## 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

.....

### 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 the students:
  - **idStudent** (student's number) – string
  - **name** – string
  - **group** – int
  - **email** – string
  - example (“1”, “Andrei Cupes”, 932, “[andrei.cupes@gmail.com](mailto:andrei.cupes@gmail.com)”)

- Assignment.txt file, which contains information about the assignments:
  - **laboratory number** (unique identifier) - string,
  - **brief description** of the requirement - string,
  - **deadline** – the week of the semester in which the assignment should be delivered (1..14) - int,
  - **received week**, the week in which the theme was received (1..14) – int
  - example (“1”, “Tema nr 1”, 2, 1)

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.

- Catalog.txt file will store information about the grades:
  - **id of the grade** - string,
  - **name of the student** - string,
  - **lab id** - string,
  - **grade value** – double
  - example (“1”, “Andei Cupes”, “1”, 10)

### 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
FR2.0	Manage laboratory themes and subjects.
FR2.1	Extend the deadline for an existing subject
FR2.2	Add a new laboratory theme
FR2.3	Notify students by email when adding a new laboratory theme or modifying the delivery date of an existing subject
FR2.4	Add a grade to a particular laboratory theme to a particular student
FR3.0	Filter students based on different criteria
FR4.0	Generate reports

### 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 / update / delete a student
- Add / update / delete a lab theme
- Assign a grade to a student on a theme
- Delete grade of a student on a theme
- Extend a deadline
- Find student
- Find grade
- Find lab theme
- Show all students
- Show all themes
- Show all grades
- Filter students (based on some criteria, selected from a predefined list)
  - Example: filter by group, grades, late assignments
- Notify students when adding a new laboratory theme or modifying the delivery date of an existing subject
- Generate reports (saved in Report.txt file)

Please refer to Analysis and Design Document.