Laboratory Assignment AND Assessment Requirements Specification

Version 1.0

2 March 2024

Developed by:

George Râpeanu, Pricop Laurențiu

936,

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Description of Change** | **Author** | **Date** |
| V01 | Initial/Modification of document |  | 1 March 2020 |
| V02 | Completion of document | George Râpeanu, Pricop Laurențiu | 2 March 2024 |

Contents

[Laboratory Assignment AND Assessment Requirements Specification 1](#_Toc1859473)

[Version 1.0 1](#_Toc1859474)

[March 2, 2024 1](#_Toc1859475)

[1. Introduction 3](#_Toc1859476)

[1.1. Purpose 3](#_Toc1859477)

[1.2. Scope 3](#_Toc1859478)

[1.3. Definitions, Acronyms, and Abbreviations 3](#_Toc1859479)

[1.4. Document Overview 3](#_Toc1859480)

[2. Product/Service Description 3](#_Toc1859481)

[2.1. Product Context 3](#_Toc1859482)

[2.2. User Characteristics 3](#_Toc1859483)

[3. Requirements 3](#_Toc1859484)

[3.1. Functional Requirements 4](#_Toc1859485)

[3.2. User Interface Requirements 4](#_Toc1859486)

[3.3. Usability 4](#_Toc1859487)

[3.4. Data Management 4](#_Toc1859488)

[4. User Scenarios/Use Cases 4](#_Toc1859489)

# 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.

## 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.

## Scope

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

## Definitions, Acronyms, and Abbreviations

……………………………………………………………………………………

## Document Overview

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

# 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.

## Product Context

The product is independent and self-contained.

## User Characteristics

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

# 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.

## Functional Requirements

List the functional requirements (FR) of the system.

|  |  |
| --- | --- |
| Section/ Requirement ID | Requirement Definition |
| FR1.0 | CRUD operations for the Student entity |
| FR2.0 | Manage laboratory homework |
| FR2.1 | Extend the deadline for an existing homework |
| FR2.2 | Add a new laboratory homework |
| FR2.3 | Notify students by email when adding a new laboratory homework or modifying the delivery date of an existing homework |
| FR2.4 | Add a grade to a particular laboratory homework to a particular student. Delays will be calculated automatically |
| FR3.0 | Filter students based on different criteria |
| FR4.0 | Generate reports   * Laboratory grade for each student * The hardest homework * Students who can enter the exam * Students who have delivered all their assignments on time |

## 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.

## 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.

## Data Management

The data should be stored in 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 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.

# User Scenarios/Use Cases

The application allows the user to:

* Add a student
* Add a lab assignment
* Assign a grade to a student on an assignment
* Update student
* Delete student
* Find student
* Show all students
* Show all themes
* Filter student
* Notify students when adding a new laboratory assignment or modifying the delivery date of an existing assignment
* Extend a deadline
* Generate reports
  + Laboratory grade for each student
  + The hardest homework
  + Students who can enter the exam
  + Students who have delivered all their assignments on time

Please refer to Analysis and Design Document.