

Laboratory Assignment AND Assessment Requirements Specification

Version 1.0

8 March 8, 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 Ilie Andreea	Dana Oana	8 March 2020
-----	------------------------	---------------------------	--------------	--------------

Contents

Laboratory Assignment AND Assessment Requirements Specification	1
Version 1.0	1
15 March, 2019	1
1 Functional Requirements	3
2 Actors	4
3 Use cases – diagram	4
3.1 Use case number 1 (Description of the use case)	4
4 Analysis.....	8
4.1 Entities.....	8
4.2 Relations between entities	8
4.3 Attributes	9
4.4 System behavior	9
4.4.1 Use case 1-2-3	9
4.5 System events	4
5 Design.....	9

Analysis and design Document

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

2 Actors

- Teachers for MAP subject

3 Use cases – diagram

3.1 Use case number 1 (Description of the use case)

Actors: Teacher

Description: Add a **new** student

Precondition: User enters **valid data** about a student

Postcondition: student added to the repository.

User action	System response
1 Completes the necessary fields for adding	
	2 Checks if everything is all right. Adds the student if it is ok or displays an error message otherwise. Then it returns to menu

Exceptions: When the fields aren't filled, id already exists, one of the fields has an invalid data type.

3.2 Use case number 2 (Description of the use case)

Actors: Teacher

Description: Edit a student

Precondition: User enters **a valid id** of the student he wants to modify, then enters **valid data to all the fields** of the student

Postcondition: Data about the student is updated

User action	System response
1 Completes the necessary fields for updating	
	2 Checks if student exists and all the fields are valid . If so, it updates it, otherwise it displays an error. Then it returns to menu

Exceptions: Student with that id doesn't exist, one of the fields has an invalid data type, **one of the fields has not been filled**

3.3 Use case number 3 (Description of the use case)

Actors: Teacher

Description: Delete a student

Precondition: User gives a **valid id (existing one)**

Postcondition: Student is deleted from its repository

User action	System response
1 Inputs the id	
	2 Checks if student exists. If so, it deletes it, otherwise it displays an error. Then it returns to menu

Exceptions: Student with that id doesn't exist, **id field is not filled**

3.4 Use case number 4 (Description of the use case)

Actors: Teacher

Description: Print all students

Precondition: -

Postcondition: **All students are shown to the screen**

User action	System response
1	
	2 Prints all students. Then it returns to menu

3.5 Use case number 5 (Description of the use case)

Actors: Teacher

Description: Find information about a specific student

Precondition: User gives a valid student id (existing one)

Postcondition: Data about the student is shown to the screen

User action	System response
1 Gives student id	
	2 Check if student exists. If it does, it prints the data, otherwise it displays an error. Then it return to the menu.

Exception: **Student with id doesn't exist**

3.6 Use case number 6 (Description of the use case)

Actors: Teacher

Description: Add a lab theme

Precondition: User gives valid information about theme

Postcondition: Theme is added to the repository

User action	System response
1 Completes the necessary fields for adding	
	2 Checks if everything is all right. Adds the theme if it is ok or displays an error message otherwise. Then it returns to menu

Exceptions: When the fields aren't filled, id already exists, one of the fields has an invalid data type.

3.7 Use case number 7 (Description of the use case)

Actors: Teacher

Description: Edit laboratory theme

Precondition: User enters a valid id of the theme he wants to modify, then enters valid data to all the fields of the theme

Postcondition: Data about the theme is updated

User action	System response
1 Completes the necessary fields for updating	
	2 Check if theme exists and if all necessary fields are valid . If it does, it updates with new set of data, otherwise it displays an error. Then it return to the menu.

Exception: Theme with id doesn't exist, fields of invalid type

3.8 Use case number 8 (Description of the use case)

Actors: Teacher

Description: Delete laboratory theme

Precondition: User gives a valid theme id (existing one)

Postcondition: Data about the theme is deleted

User action	System response
1 Gives theme id	
	2 Check if theme exists. If it does, it is deleted, otherwise it displays an error. Then it return to the menu.

Exception: Theme with id doesn't exist

Use case number 9 (Description of the use case)

Actors: Teacher

Description: Print all laboratory themes

Precondition: -

Postcondition: All themes are shown on the screen

User action	System response
1	
	2 Prints all assignments, then it returns to menu

Use case number 10 (Description of the use case)

Actors: Teacher

Description: Add a grade to a specific student on a specific laboratory theme.

Precondition: User gives a valid student (student without grade), theme and grade

Postcondition: Grade is added to the given student

User action	System response
1 Completes info about student, theme and grade	
	2 If the student and the given theme exist and the grade is valid, it adds the grade to the student. Otherwise, it displays an error. Then it returns to menu

Exceptions: When a student or theme doesn't exist, or student already has a grade, or one of the fields is not filled, or grade invalid type.

3.9 Use case number 11 (Description of the use case)

Actors: Teacher

Description: Delete a grade to a specific student on a specific laboratory theme.

Precondition: User gives a valid student (student with grade), theme

Postcondition: Grade is deleted from the given student

User action	System response
1 Completes info about student and theme	
	2 If the student and the given theme exist and a grade is assigned, it deletes the grade of the student. Otherwise, it displays an error. Then it returns to menu

Exceptions: When a student or theme doesn't exist, or student doesn't has a grade, or one of the fields is not filled, or grade invalid type.

3.10 Use case number 12(Description of the use case)

Actors: Teacher

Description: Print all grades

Precondition: -

Postcondition: All grades are shown to the screen

User action	System response
1	
	2 Prints all grades. Then it returns to menu

Use case number 13 (Description of the use case)

Actors: Teacher

Description: Filter the students, assignments, themes and grades based on different criteria

Precondition: -

Postcondition: -

User action	System response
1 Completes the necessary fields	
	2 Returns the result set of the selected filter. Then returns to menu

Use case number 14 (Description of the use case)

Actors: Teacher

Description: Generate reports

Precondition: Select report type

Postcondition: Information about report is shown to the screen

User action	System response
1 Choose one of the given reports types	
	2 Shows the corresponding report

4 Analysis

4.1 Entities

- Student
- Laboratory assignments
- Grades

4.2 Relations between entities

A student can have more assignments and an assignment can be assigned to more students.

A grade can be given for a specific student on an assignment.

4.3 Attributes

- Student
 - id: String
 - name: String
 - group: Int
 - email: String
 - teacher: String
- Grade
 - id: Map<String, Int>
 - st: Student
 - assign: Assignment
 - value: Float
 - date: Int
- Assignment
 - id: Int
 - description: String
 - deadline: Int
 - delivery_week: Int

4.4 System behavior

4.4.1 Use case 1-2-3

The system will probably act as a subsystem to a larger environment, in order to speed up a certain process in the company's workflow.

5 Design

5.1 Class diagram

