Laboratory Assignment AND Assessment Requirements Specification

Version 1.0

15 March, 2020

Developed by:

Name Burian Raul, Cernov Alexandru

Group 931

Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description of Change | Author | Date |
| V01 | Initial/Modification of …. | Burian Raul,Cernov Alexandru | 02.03.2020 |
| V02 | Project Analysis completion | Burian Raul,Cernov Alexandru | 15.03.2020 |
|  |  |  |  |

Contents

[Laboratory Assignment AND Assessment Requirements Specification 1](#_Toc1859878)

[Version 1.0 1](#_Toc1859879)

[15 March, 2019 1](#_Toc1859880)

[1 Functional Requirements 3](#_Toc1859881)

[2 Actors 3](#_Toc1859882)

[3 Use cases – diagram 3](#_Toc1859883)

[3.1 Use case number 1 (Description of the use case) 3](#_Toc1859884)

[4 Analysis 3](#_Toc1859885)

[4.1 Entities 3](#_Toc1859886)

[4.2 Relations between entities 3](#_Toc1859887)

[4.3 Attributes 3](#_Toc1859888)

[4.4 System behavior 4](#_Toc1859889)

[4.4.1 Use case 1-2-3 4](#_Toc1859890)

[4.5 System events 4](#_Toc1859891)

[5 Design 4](#_Toc1859892)

**Analysis and design Document**

# Functional Requirements

List the functional requirements (FR) of the system.

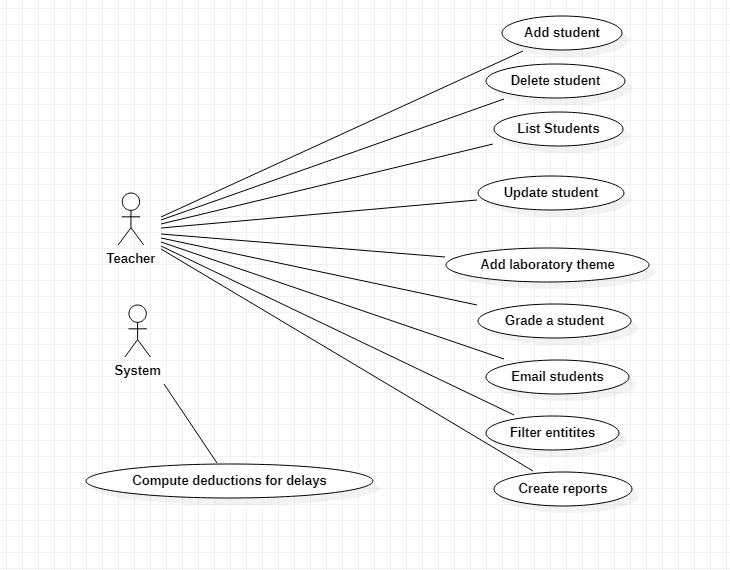
| Section/ Requirement ID | Requirement Definition |
| --- | --- |
| FR 1 | CRUD operations for student |
| FR 2 | Add a laboratory theme |
| FR 3 | Update deadline for existing laboratory theme |
| FR 4 | Students notified by email when a new theme is added or the deadline is updated |
| FR 5 | Grade students for the laboratories |
| FR 6 | Information about a student grade is saved in a text file “StudentName.txt” with the contents as in the requirements |
| FR 7 | Email the text files with the student grades’ information to students weekly |
| FR 8 | Filter students/themes/grades based on different “criteria” |
| FR 9 | Create reports as said in the requirements document |

# Actors

The teacher is the one who will use the application, him being the only actor

Some may consider the system as an actor with the function to compute automatically the deduction to delays to a laboratory grade.

# Use cases – diagram



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

Actors: Teacher

Description: List Students

Precondition: - An XML file with the students is present as input, else an error is thrown

Postcondition: - Students are listed on the console

|  |  |
| --- | --- |
| User action | System response |
| Start the application |  |
|  | The list of commands is shown on screen |
| User presses “11”, the command to list students |  |
|  | The list of students is printed on the console |

Exceptions: If the input file is not present or not correctly formatted

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

Actors: Teacher

Description: Add student

Precondition: - none

Postcondition: - A new student is added to the list of students and the corresponding students file

|  |  |
| --- | --- |
| User action | System response |
| Start the application |  |
|  | The list of commands is shown on screen |
| User presses “21”, the command to add a stuent |  |
|  | The system prompts the user for a student ID |
| User enters student ID |  |
|  | The system prompts the user for student’s name |
| User enters student’s name |  |
|  | The system prompts the user for student’s group |
| User enters student’s group |  |
|  | The systems shows a message that the student was added |

Exceptions: If entered student does not respect the validation conditions

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

Actors: Teacher

Description: Remove student

Precondition: - A valid student ID is given

Postcondition: - The student is removed from memory and students file

|  |  |
| --- | --- |
| User action | System response |
| Start the application |  |
|  | The list of commands is shown on screen |
| User presses “31”, the command to remove a stuent |  |
|  | The system prompts the user for a student ID |
| User enters student ID |  |
|  | The system shows a message that the student was deleted |

Exceptions: If entered student ID does not exist

# Analysis

## Entities

Grade, student, laboratory theme

## Relations between entities

A student is graded at many laboratory themes with exactly one grade at each one.

## Attributes

Grade: id, grade, deadline, feedback

Student: id, name, group

Assignment: id, description, deadline, startline

## System behavior

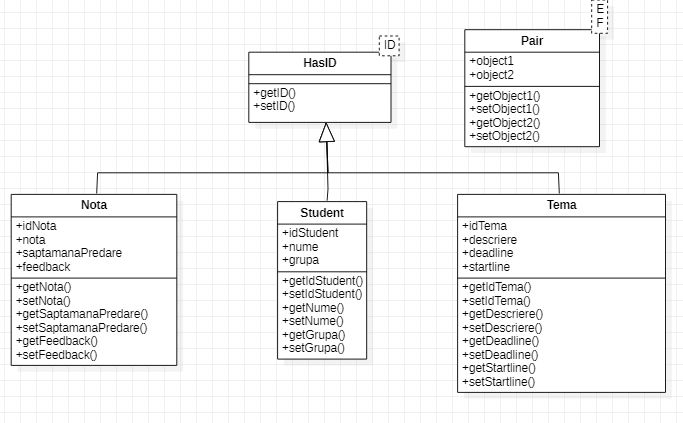
The system is designed to be used as a standalone software to aid teachers in an educational institution to better handle the grading and communicating the students about their grade.

## System events

Events are triggered by command line input and the output is presented back to the user in the console after each command is executed. One command can be executed at a time in sequential mode.

# Design

* 1. **Class diagram (DOMAIN)**



* 1. **Sequence diagrams (for each use case)**

# 

* 1. **GRASP**

GRASP= General Responsibility Assignment Software Patterns

General guidelines for assigning responsibilities to classes and objects

GRASP patterns : use of a service as an abstraction layer, polymorphism, templates, low coupling