Assigment 1

Analysis and Design Document

Student:Giurgiu Diana-Ioana

**Group:30233**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

The application is designed to manage the students, their grades, personal information at Technical University of Cluj-Napoca in JAVA.

# Functional Requirements

*The functional requirements are divided between the 2 existing users: students and teachers/admin.*

*The students can:*

* *update/view/add their personal information*
* *create/update/delete/view their student profile*
* *process class enrolment*

*The teachers or admins can:*

* *create/read/update/delete the students’ information*
* *generate reports for a particular period containing the activities performed by a student*

# Non-functional Requirements

*Some of the non-functional requirements refer to not allowing students to enter invalid data the moment they try to perform an action.*

*Also, using an MVC in order to protect the integrity of the data stored in the application and the possible extensibility of the application.*

2. Use-Case Model

*[Create the use-case diagrams and provide one use-case description (according to the format below).*

*Use-Case description format:*

*Use case: <use case goal>*

*Level: <one of: summary level, user-goal level, sub-function>*

*Primary actor: <a role name for the actor who initiates the use case>*

*Main success scenario: <the steps of the main success scenario from trigger to goal delivery>*

*Extensions: <alternate scenarios of success or failure>*

*]*

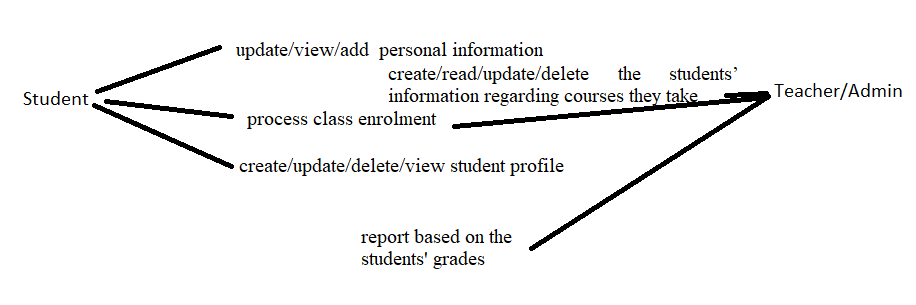
*Use case: Interface interaction between the 2 users*

*Level: user and admin*

*Primary actor: Students and Admins/Teachers*

*Main success scenario: Successful into acquiring the data or do the modification they wish to do*

*Extensions: Detailing each action with the subclasses they use in order to fulfill their role.*



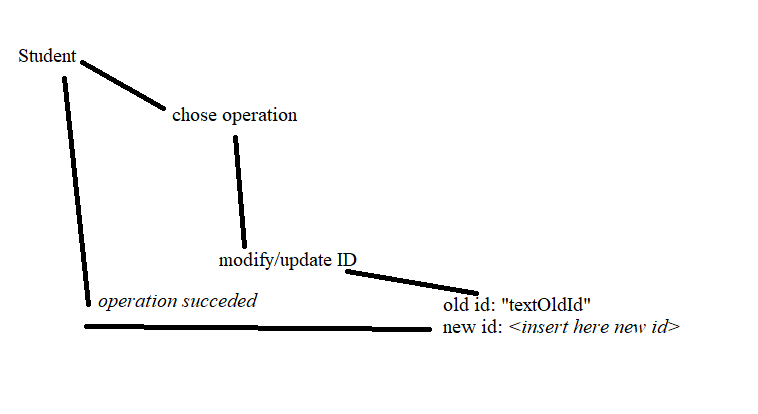
*Use case: student modifies ID*

*Level: student level*

*Primary actor: student*

*Main success scenario: successfully modifies the ID*

*Extensions: wrong input in ID retry.*



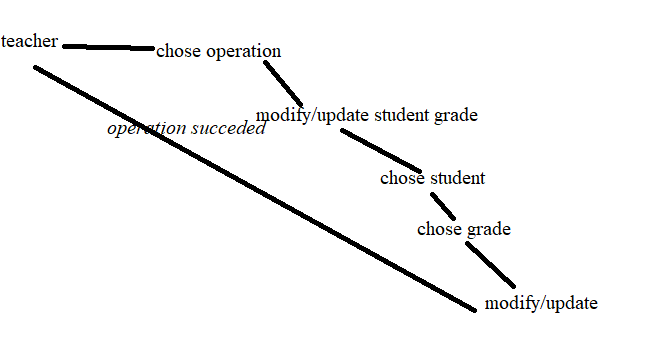
*Use case: Teacher applies Update on student*

*Level: teacher level*

*Primary actor: student*

*Main success scenario: successfully modifies the student aprt*

*Extensions: ---*

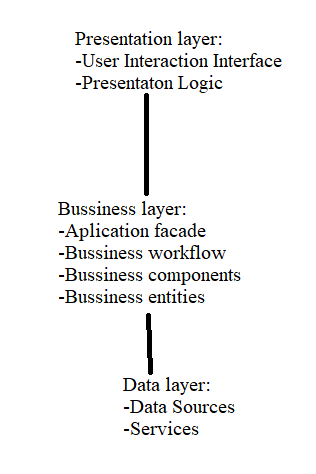


3. System Architectural Design

**3.1 Architectural Pattern Description**

*The layers pattern was used, which is grouping functionalities on layers   organized into horizontal layers, each layer performing a specific role within the application . The top level layers depend on lower level layers.*

**3.2 Diagrams**

**

*The application will be divided in packages, each package will be significant of a layer.*

*There will be 3 layer(Presentation, Business, Data).*

4. UML Sequence Diagrams

*[Create a sequence diagram for a relevant scenario.]*

Interface

Server

Student

Validate log in

Log in

Chose operation

Grab old id

Modifi personal ID

Requesting input new id

Requesting write new id

Checking that the new ID is valid

Input new id

ID is valid

Success message

5. Class Design

**5.1 Design Patterns Description**

*The Observer pattern could be useful in order to notify the students in case something changes in their profile or their grades are updated.*

**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

6. Data Model

*[Present the data models used in the system’s implementation.]*

Teacher/Admin

-name

-TeacherID

-CRUD on student info

-reports for students

Student

-name

-IDCARD

-ID

-address

-age

7. System Testing

*The testing will be done with a JUNIT testing unit which will create new students/ modify some students’ information and will also use the admins rights to update grades or student info.*

8. Bibliography