Student: Iacob Gabriel

**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

A Java application for the management of students in the CS Department at TUCN.

# Functional Requirements

The application should have two types of users (student and teacher/administrator user) which have to provide a username and a password in order to use the application.

The regular user can perform the following operations:

* Add/update/view client information (name, identity card number, personal numerical code, address, etc.).
* Create/update/delete/view student profile (account information: identification number, group, enrolments, grades).
* Process class enrolment (enroll, exams, grades).

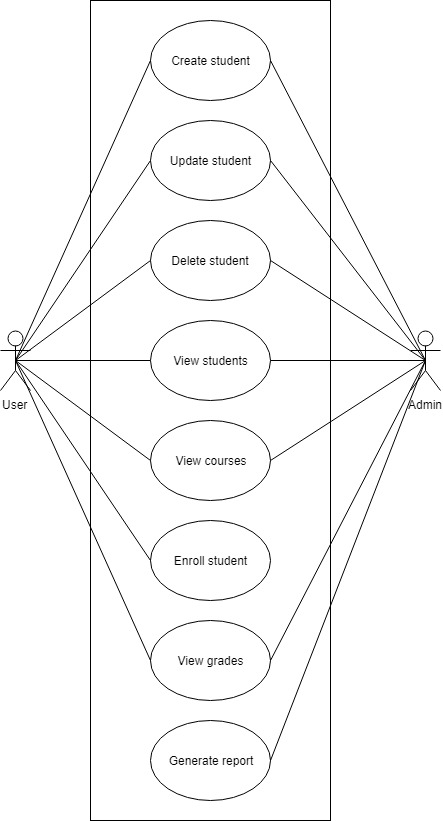
The administrator user can perform the following operations:

* CRUD on students information.
* Generate reports for a particular period containing the activities performed by a student.

# Non-functional Requirements

* The data will be stored in a relational database (Postgres / MySql / MsSql).
* Layers architectural pattern will be used for the application.
* Data Access Layer will be created using SQL statements in the most suitable way for the application.
* All the inputs of the application will be validated against invalid data before submitting the data and saving it in the database.

2. Use-Case Model



**Use case:** View Grades

**Level:** summary level

**Primary actor:** User

**Main success scenario:** User logs in. User goes into the “View Grades” section of the interface. User inputs the ID of the student. System displays all grades for the courses the student was enrolled in.

**Extensions:**

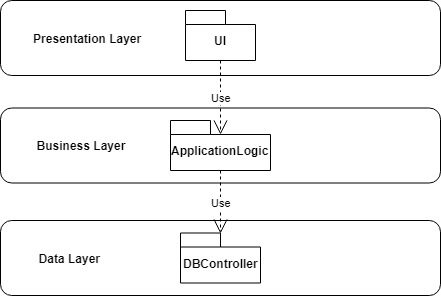
* User can input a course ID. System will display all grades for all students who were enrolled in that course.
* User can input both a course ID and a student ID. System will display the grade of that student for that course.
* User inputs invalid student or course ID. System will display no results.

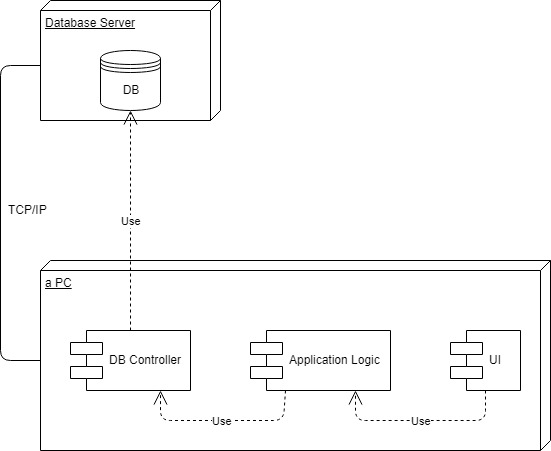
3. System Architectural Design

**3.1 Architectural Pattern Description**

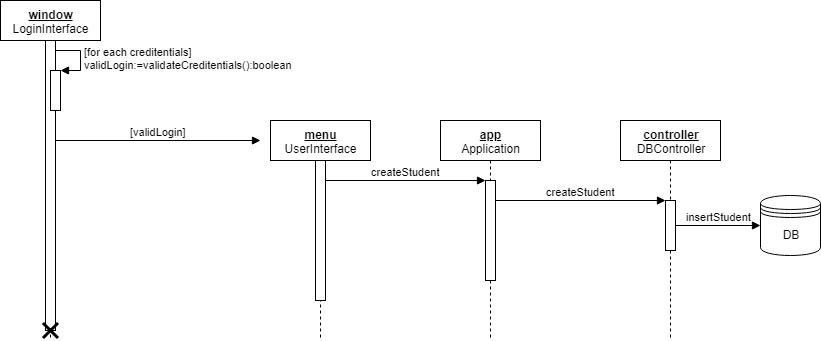
Layers design pattern – The idea of structuring software components into logical groupings called layers. Having separate layers with distinct roles and functionalities helps maximizing the maintainability of the code, provides a clear delineation where certain design decisions must be made and optimize the way the application works when deployed.

**3.2 Diagrams**





4. UML Sequence Diagrams

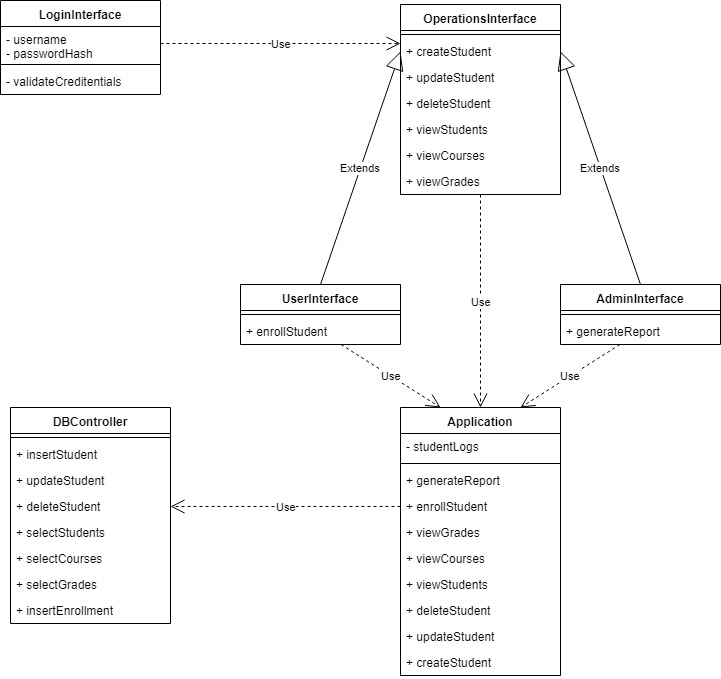


5. Class Design

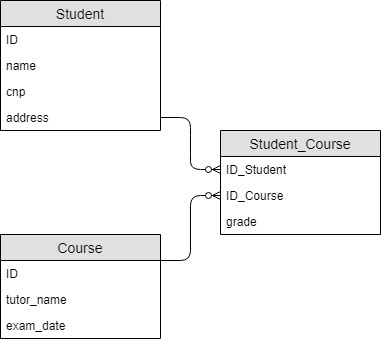
**5.1 Design Patterns Description**

-

**5.2 UML Class Diagram**



6. Data Model



7. System Testing

-

8. Bibliography

* **Layered Application Guidelines** <https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658109%28v%3dpandp.10%29>
* **Diagrams** <https://github.com/utcn-cs-software-design-tudorv-2019/LabInfo/blob/master/resources/UML%20-%20embarcadero.pdf>