Assignment 1

Student: Radu Petrisel

**Group: 30432**

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 used for the management of students in the Computer Science Department of Technical University of Cluj Napoca. The application has two users (student and teacher/administrator) which must provide a username and a password in order to use the application.

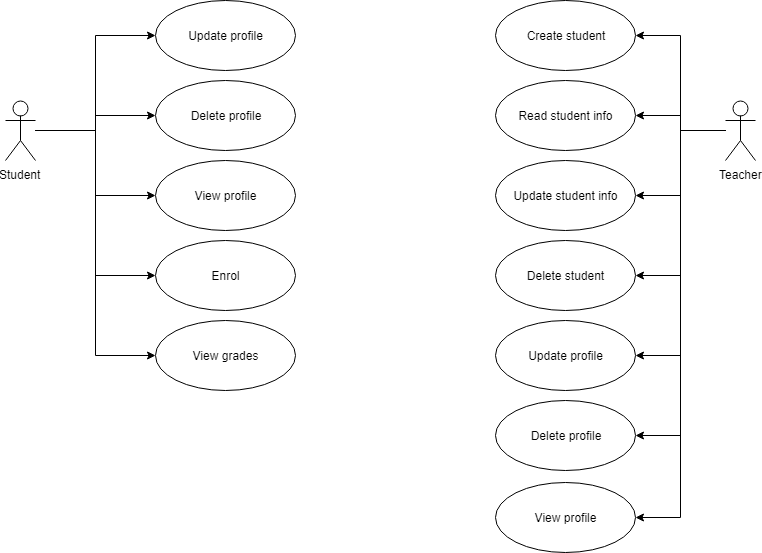
# Functional Requirements

* authentication
* create/read/update/delete students’ information (for teacher/administrator)
* generate reports (teacher/admin)
* add/update/view info (all users)
* create/update/delete/view profile (all users)
* process class enrolment (all users)

# Non-functional Requirements

* reliability

2. Use-Case Model



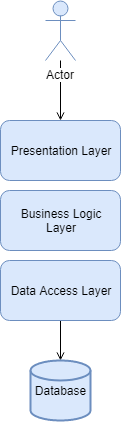
**Use case:** create student

**Level:** user-goal level

**Primary actor:** teacher

**Main success scenario:** insert student info 🡪 system validate info 🡪 student created

**Extensions:** failure 🡪 invalid data or student already exists

**3. System Architectural Design

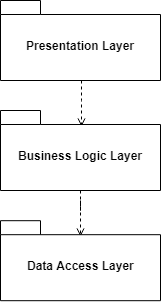
**3.1 Architectural Pattern Description**

This system uses layered architecture pattern.

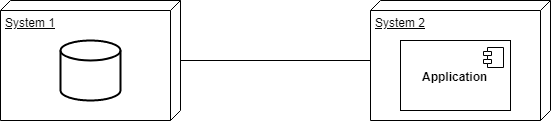
The layered architecture is the most common architectural pattern nowadays. The pattern consists of logically diving the application in layers, each with its own part to play. This application has 3 layers: data access (DAL), business logic (BLL) and presentation layer (PL).

**3.2 Diagrams**

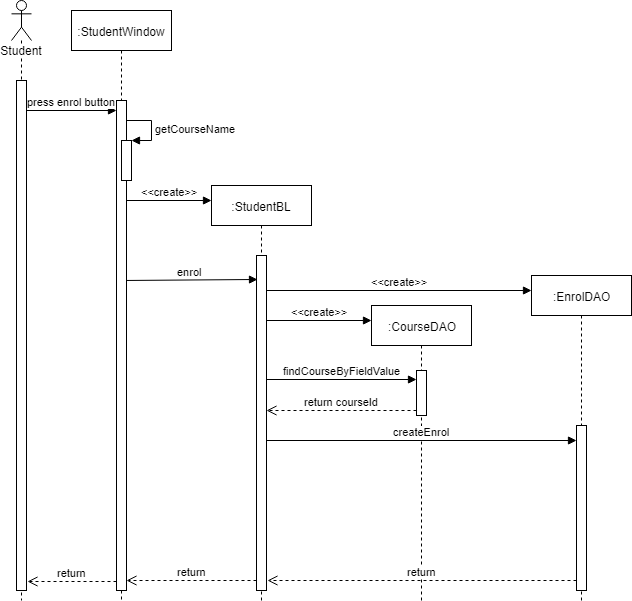
**Package Diagram**

****

**Deployment Diagram**



4. UML Sequence Diagrams

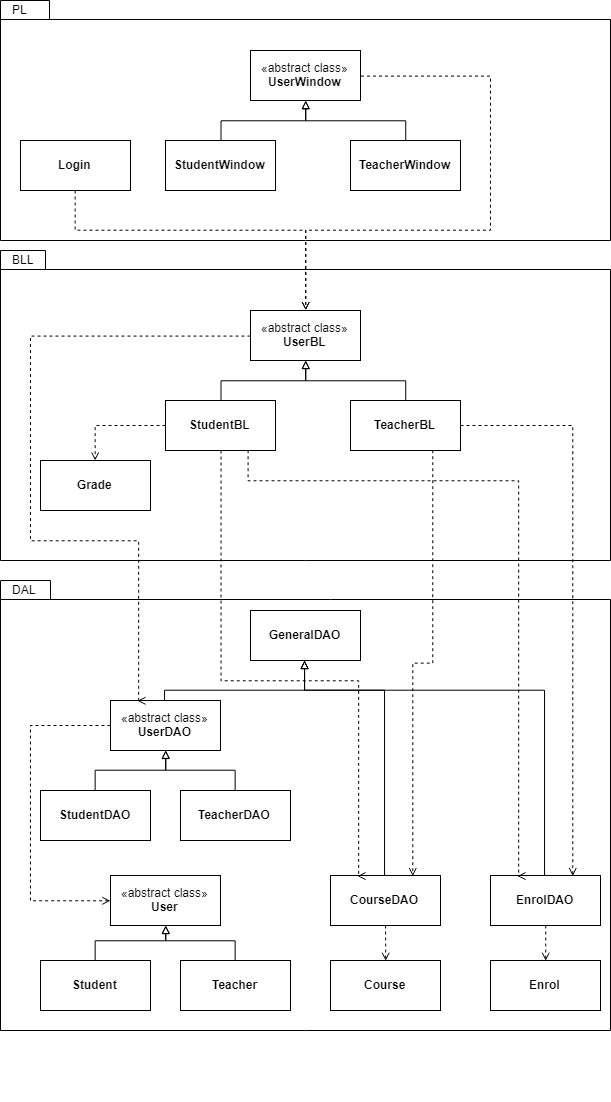


5. Class Design

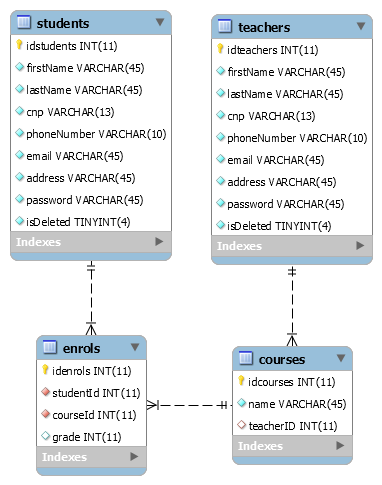
**5.1 Design Patterns Description**

* Singleton – database connection has only one instance
* Design by contract – database access methods validate data before writing it to the database

**5.2 UML Class Diagram**

**

6. Data Model



7. System Testing

8. Bibliography

[MySQL jdbc guide](https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-connect-drivermanager.html)

[JavaFX Docs](https://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm)