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.

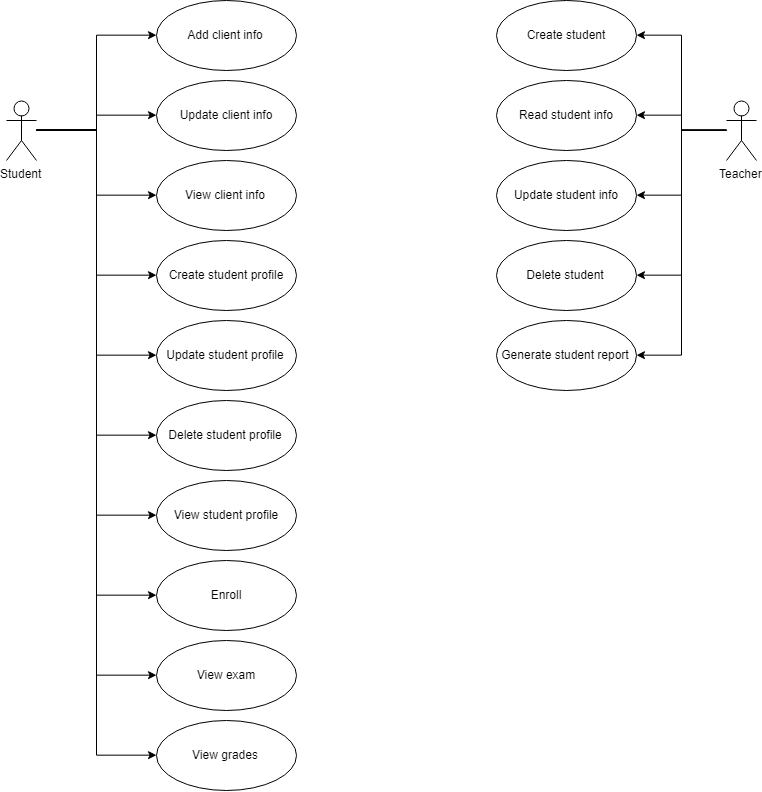
# Functional Requirements

* authentication
* CRUD on 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**

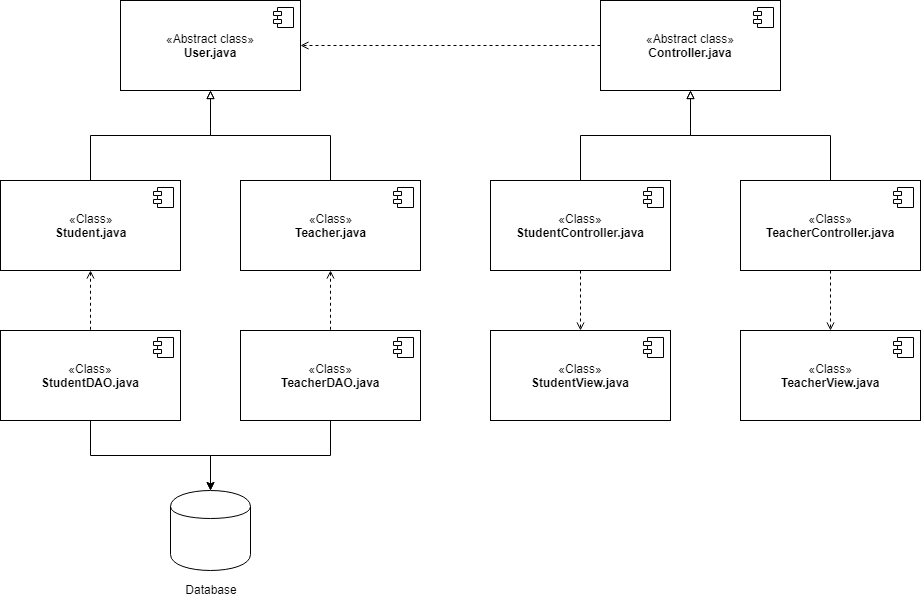
This project uses two architectural patterns: layered architecture and Model-View-Controller (MVC).

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).

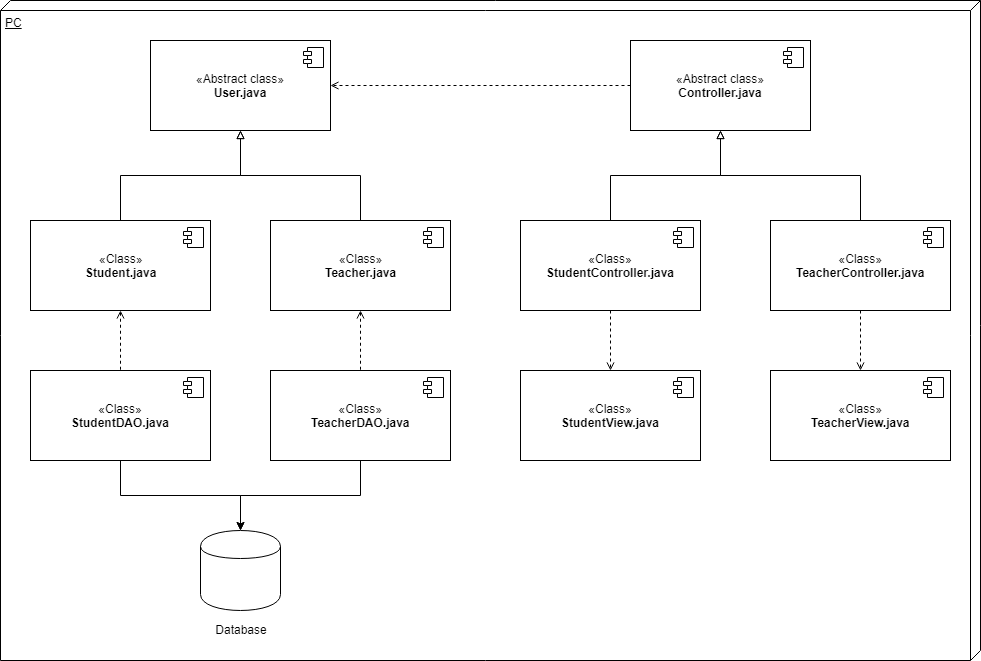
MVC is a commonly used architectural pattern for user interfaces. The main idea of MVC is that data is ‘stored’ in the Model (M) and the controller (C) connects the data with the View (V) (what the user sees).

**3.2 Diagrams**

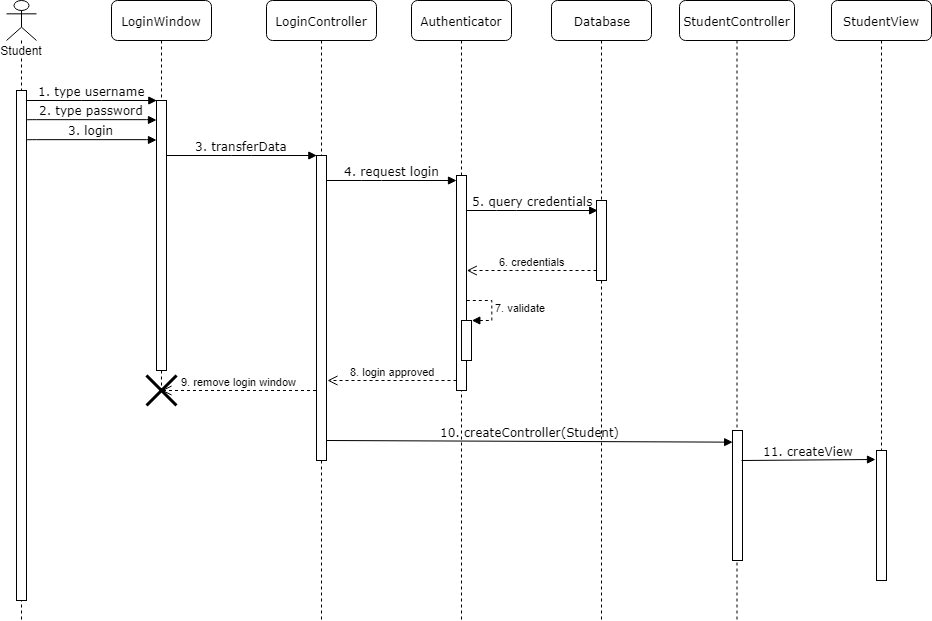
**Component Diagram**

**

****Package Diagram**

****Deployment Diagram**

4. UML Sequence Diagrams

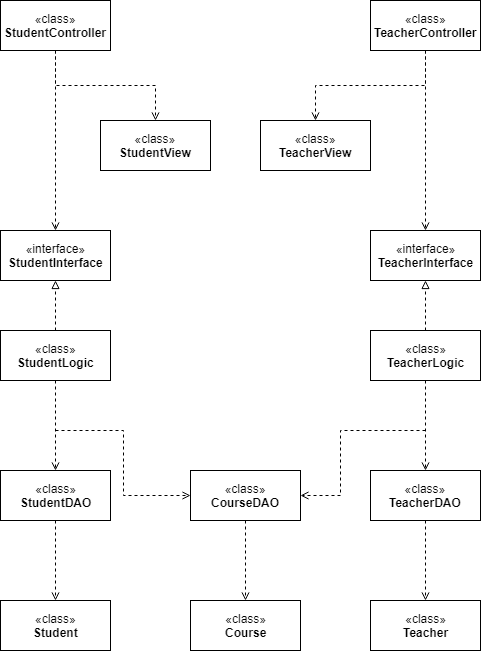


5. Class Design

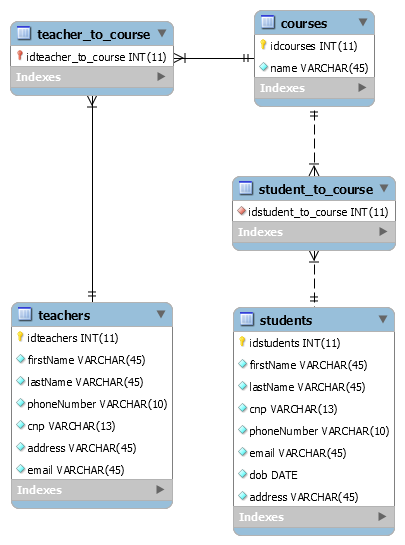
**5.1 Design Patterns Description**

* Observer design pattern – student will receive email notification when a new grade is added on one of his exams

**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)