Students Management

Analysis and Design Document

Student: Danila Vlad-Mihai

**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 4

4. UML Sequence Diagrams 5

5. Class Design 6

6. Data Model 7

7. System Testing 7

8. Bibliography 7

1. Requirements Analysis

# Assignment Specification

This application aims to develop a management system for TUCN CS Department students using Layers architectural pattern.

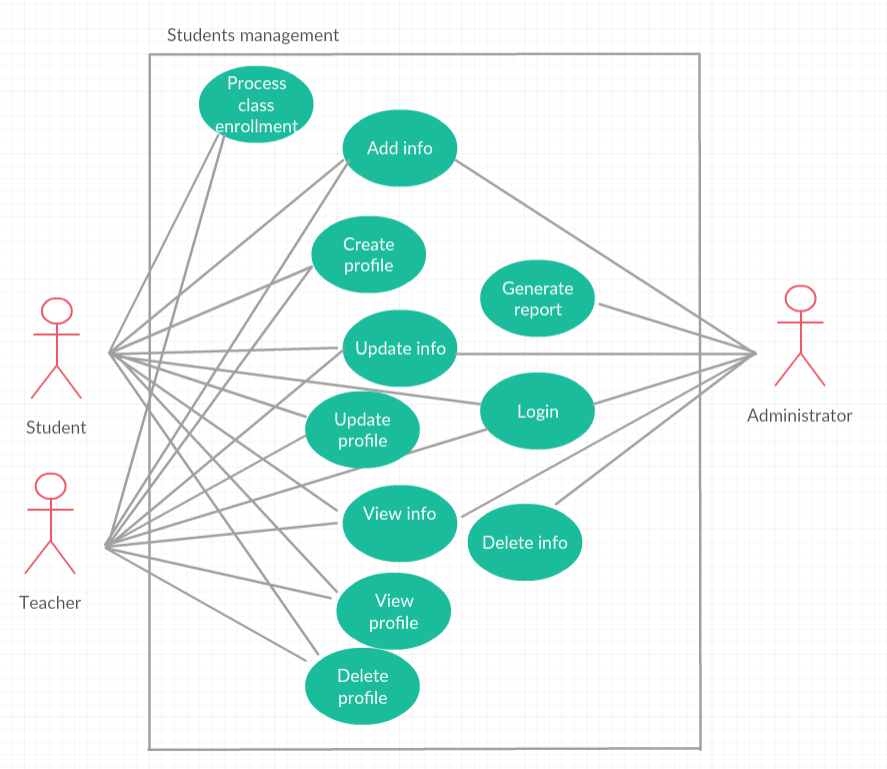
# Functional Requirements

* System shall block users from access to use the application if they are not logged in
* Add/Update/View client information
* Create/Read/Update/Delete student profile
* Process class enrolment
* Generate reports
* Validate input before saving it in the database

# Non-functional Requirements

* The system shall be organized using Layers architectural pattern
* Data Access Layer shall be created using SQL statements
* Data will be stored in a relational database

2. Use-Case Model



**Use case**: Student login using username and password

**Level**: User-goal level

**Primary** **actor**: Student

**Main** **success** **scenario**:

1. Student opens the application
2. Student enters the username and password
3. Student receives a message confirming the authentication
4. Student login successfully completed

**Extensions:**

1. Student opens the application
2. Student enters the username and password
3. Student receives an error message
4. Student enters the username and password again
5. Student receives a message confirming the authentication
6. Student login successfully completed

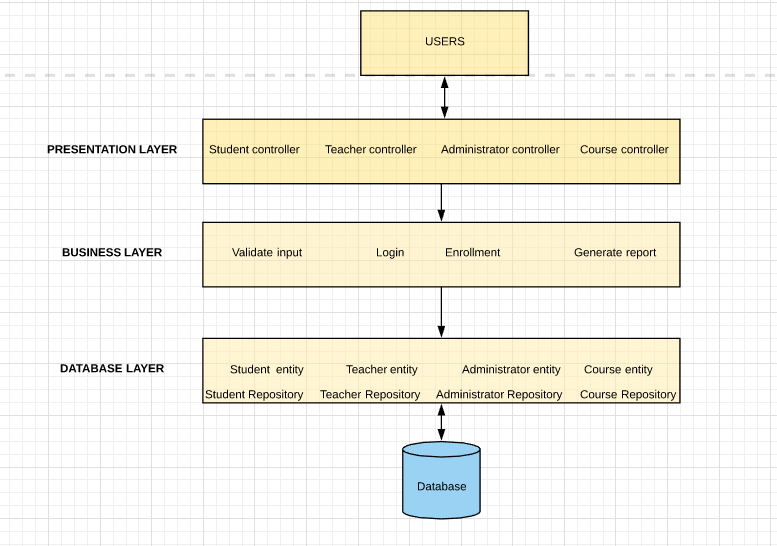
3. System Architectural Design

**3.1 Architectural Pattern Description**

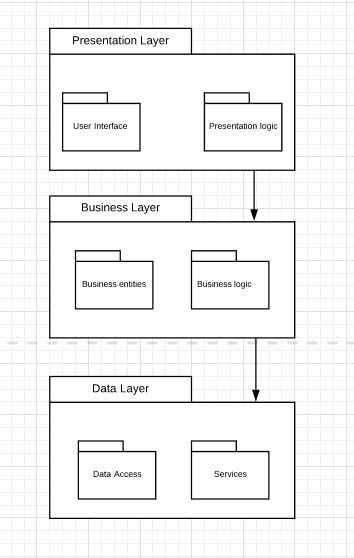
The system will be built using Layers architectural pattern. Logically similar functionalities will be grouped from the technical point of view by dividing the entire program logic into three layers: Presentation Layer, Business Layer and Database Layer. Our goal is to minimize the amount of overlapping functionalities across the entire application.

**3.2 Diagrams**

System’s conceptual architecture

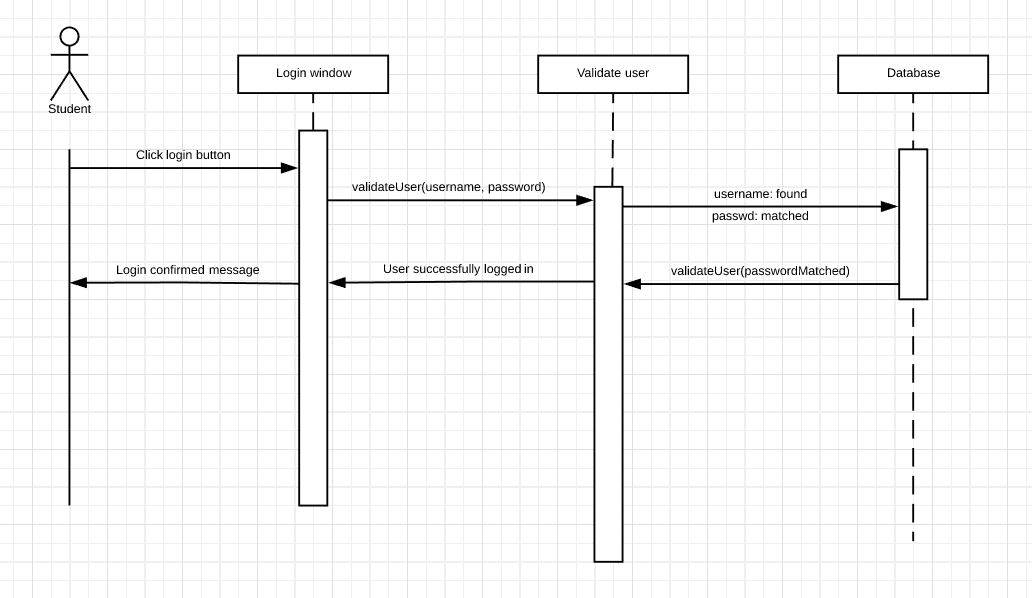


Package diagram

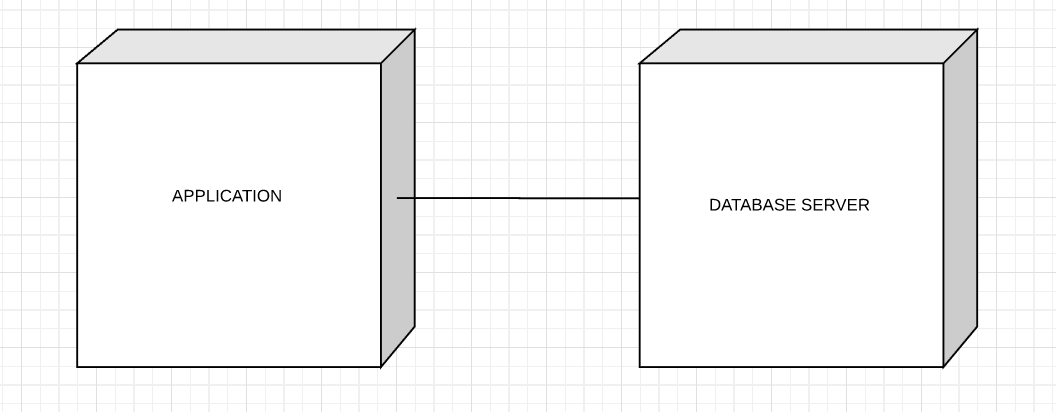


1. UML Sequence Diagrams

Scenario: Student login



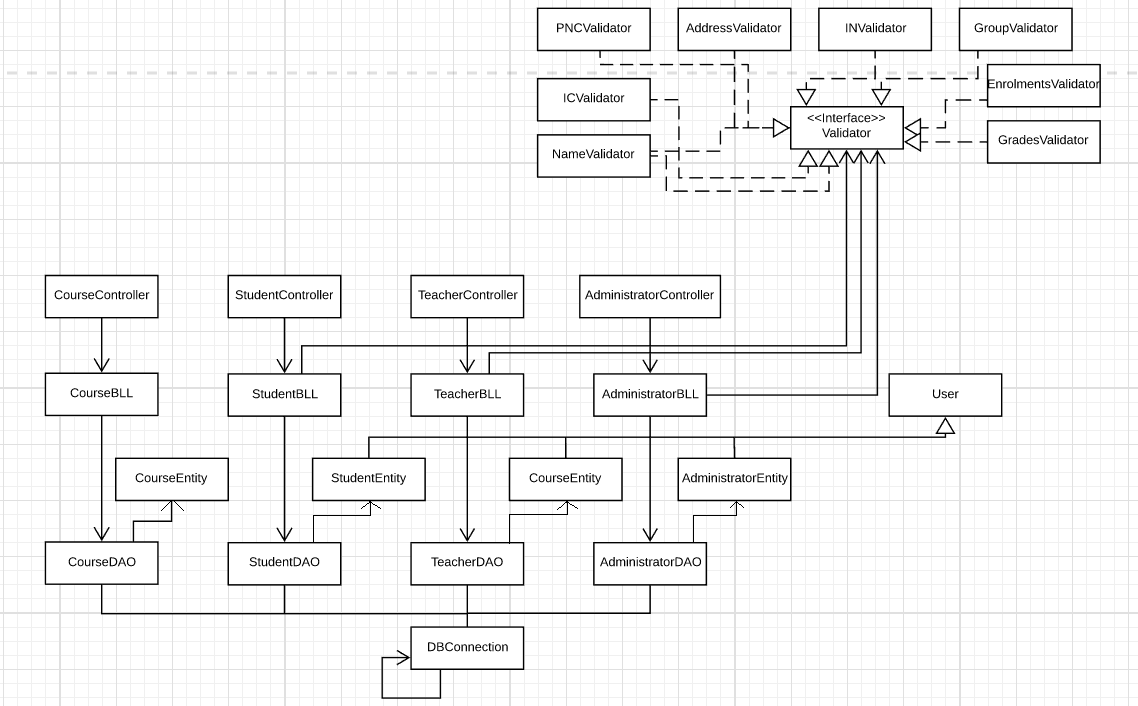
Deployment diagram



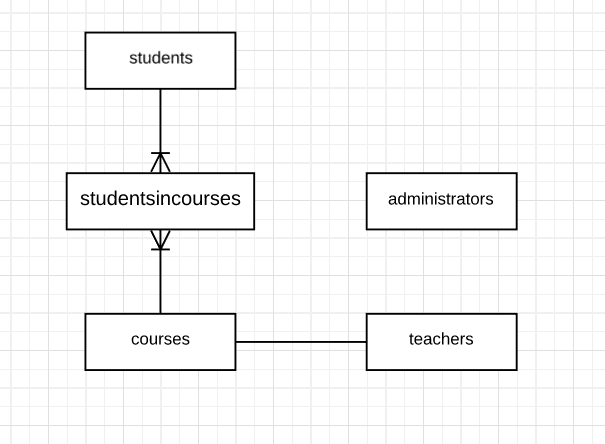
5. Class Design

* 1. **Design Patterns Description**

Singleton design pattern – one single database connection

* 1. **UML Class Diagram**

6. Data Model



7. System Testing

* Unit Testing – write test to verify that a relatively small piece of code is doing what it is intended to do.

8. Bibliography

[stackoverflow.com](https://stackoverflow.com)

[lucidchart.com](https://www.lucidchart.com)

<http://users.utcluj.ro/~dinso/PS2018/>