Assignment 2

Student: Barabas Hunor

**Group: 30432**

Table of Contents

Assignment 1 1

1

Student: Barabas Hunor 1

Table of Contents 2

1. Requirements Analysis 3

1.1Assignment Specification 3

1.2Functional Requirements 3

1.3Non-functional Requirements 3

2. Use-Case Model 3

Use case: Generate Report 3

Level: sub-function 3

Primary actor: teacher (as Administrator) 3

Main success scenario: 3

Extensions: 3

3. System Architectural Design 4

4. UML Sequence Diagrams 9

5. Class Design 9

6.Data Model 10

7.System Testing 11

8. Bibliography 11

1. Requirements Analysis

# Assignment Specification

This assignment involves the design and implementation of a student management system for the CS Department at TUCN.

# Functional Requirements

For Users, to be able to:

* Add/Update/View Client Information
* Create/View/Update/Delete Student Information
* Log in
* Process class enrollment

For Administrators ( Teachers), to be able to:

* Generate Reports
* Perform CRUD operations

# Non-functional Requirements

Possible non-functional requirements could be:

* Security – To prevent malicious students from modifying data that they should not be able to modify
* Reliability – To have a system that the university can use and rely on.
* Availability – To have a system with an uptime as close as possible to 100% ( looking at you, sinu…)
* Usability – To have an application that is simple to use for students and teachers alike.

2. Use-Case Model

*Use case: Generate Report*

*Level: sub-function*

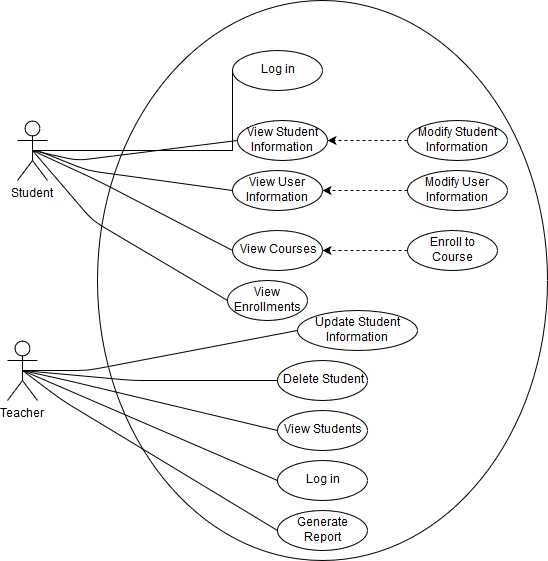
*Primary actor: teacher (as Administrator)*

*Main success scenario:*

1. Teacher logs in to the Administrator panel
2. Selects a student from the list
3. Specifies time interval
4. Presses Generate Report
5. After a while the report will be generated as a separate file.

*Extensions:*

If there is a problem in the connection with the database, the operation might fail. Otherwise, even if there is no activity a report will be generated ( it will be blank, but it will be generated nonetheless).



3. System Architectural Design

**3.1 Architectural Pattern Description**

The utilized pattern will be the Layered Architectural Pattern. As the name suggests, our application will be structured into layers, each of them having a specific role. Higher level layers do not need to know how the layers below them perform their operations, making it easy to extend the system if need be.

**Presentation Layer**

Responsible for user interface logic. This layer deals with the core functionality of the system, and the communication between the UI and the database.

**Business Layer**

Responsible for business logic. Deals with the communication between the user interface and the database.

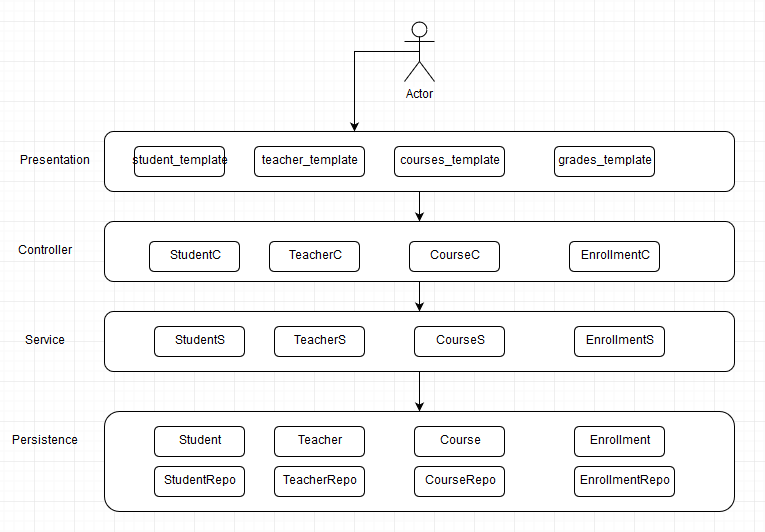
**Data Access Layer**

Provides access to the data.

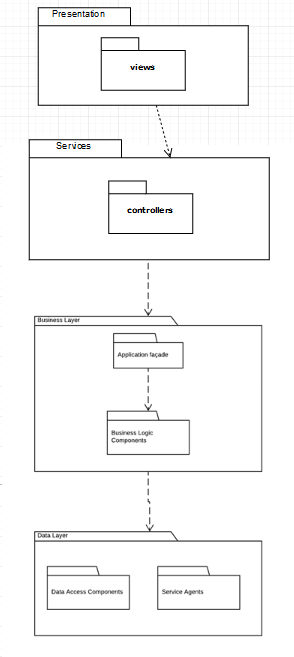
**MVC**

In addition, an MVC architecture will be used, thus refactoring the whole Presentation Layer. We will have a separate package for views and one for controllers, essentially creating a whole new layer, the Services Layer. This will help us separate the presentation of the UI from the application control.

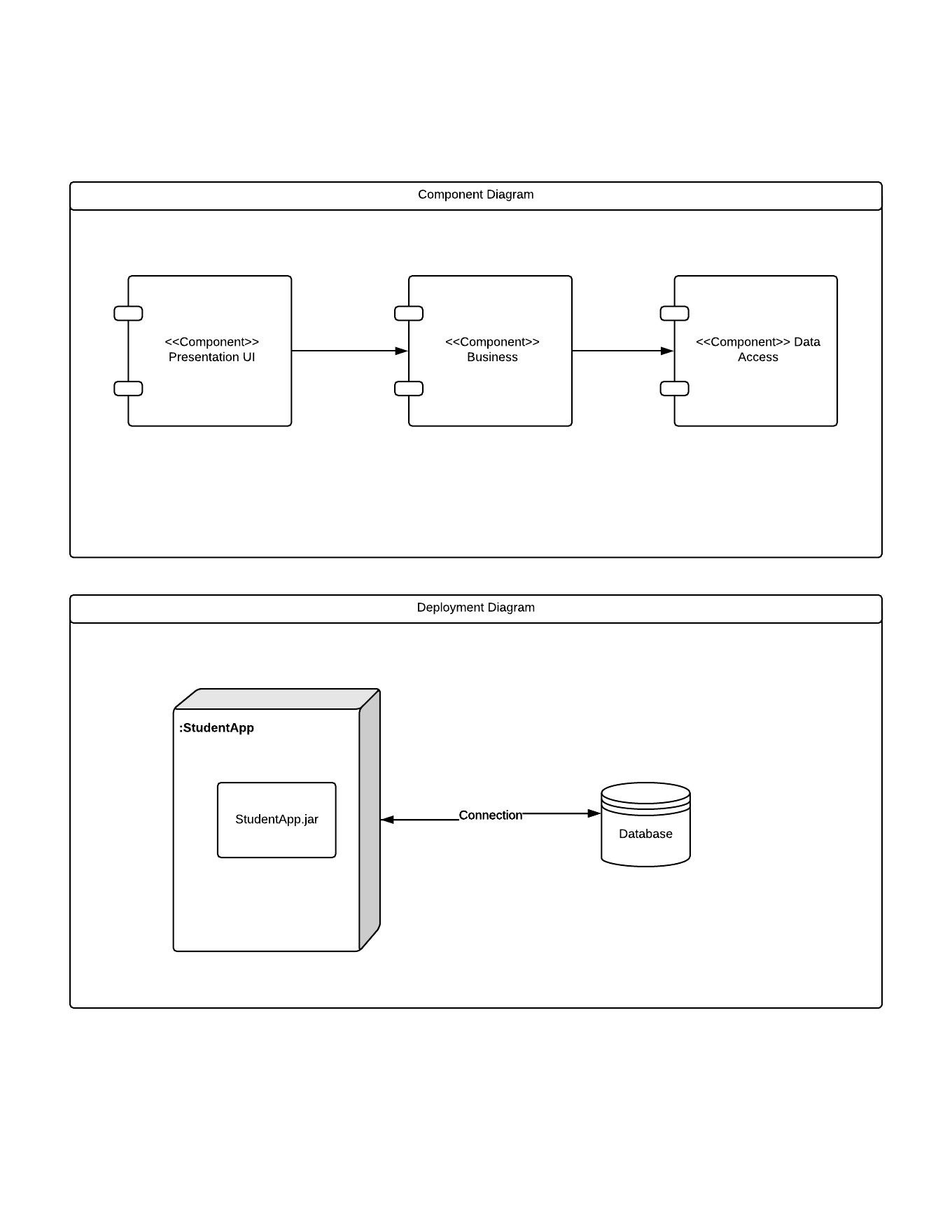
**3.2 Diagrams**



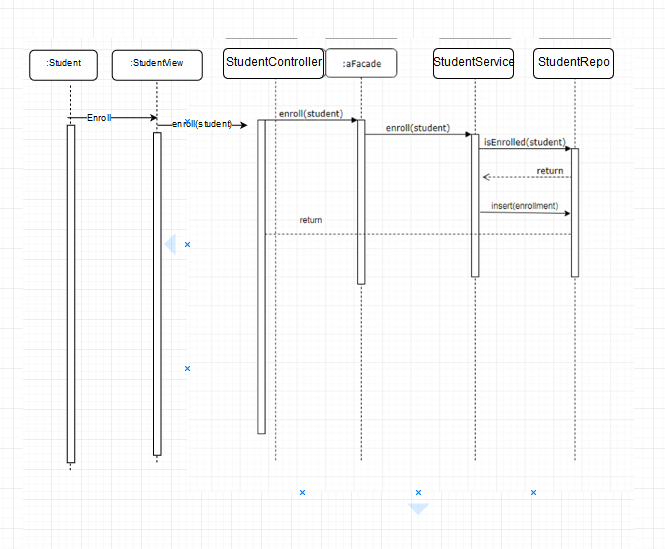
*Conceptual Architecture of the Application*



*Package Diagram*



4. UML Sequence Diagrams

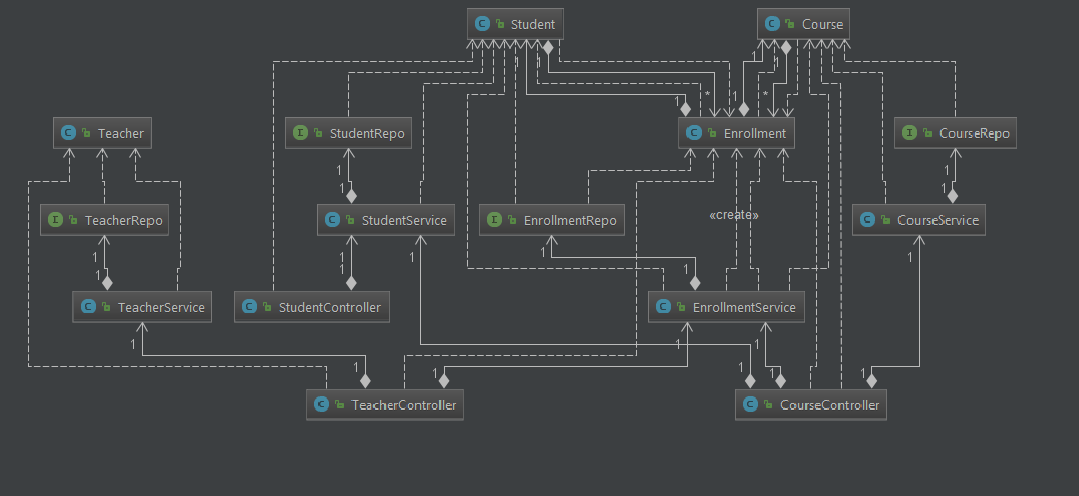


5. Class Design

**5.1 Design Patterns Description**

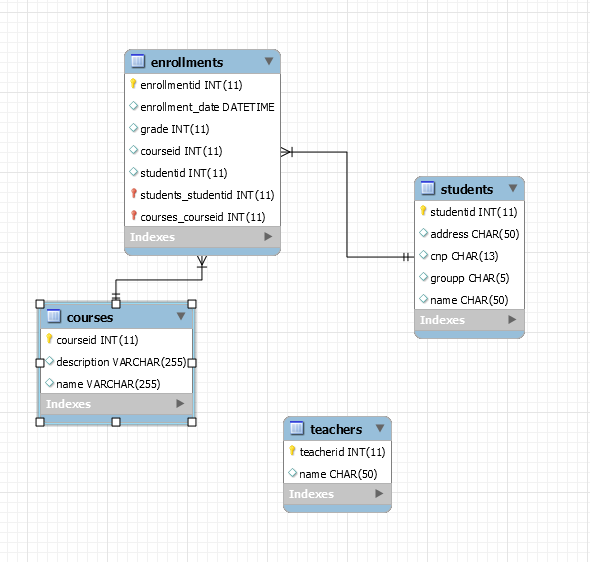
The business logic layer contains a Façade Design Pattern, which provides a unified, higher level interface for the business logic components

* 1. **UML Class Diagram**



1. Data Model

The data model consists of is slightly less complex thanks to the features of the Spring Framework ( and the requirements). The users and logins tables are omitted and we are left with only 4 essential tables: enrollments, students, courses and teachers.



7.System Testing

The testing strategy used is unit testing. The used tool is JUnit*.*

8. Bibliography

Microsoft Application Architecture Guide - <https://msdn.microsoft.com/en-us/library/ff650706.aspx>

* Layered Application Diagram
* Various pieces of knowledge

Tool used for diagram creation: - Lucidchart - https://www.lucidchart.com