Assignment1

Student: Cristea Andrei

**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

Design and implement a Java application for the management of students in the CS Department at UTCN. 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.

# Functional Requirements

The regular user can perform the following operation:

* Add/update/view client information.
* Create/update/delete/view student profile.
* Process class enrolment.

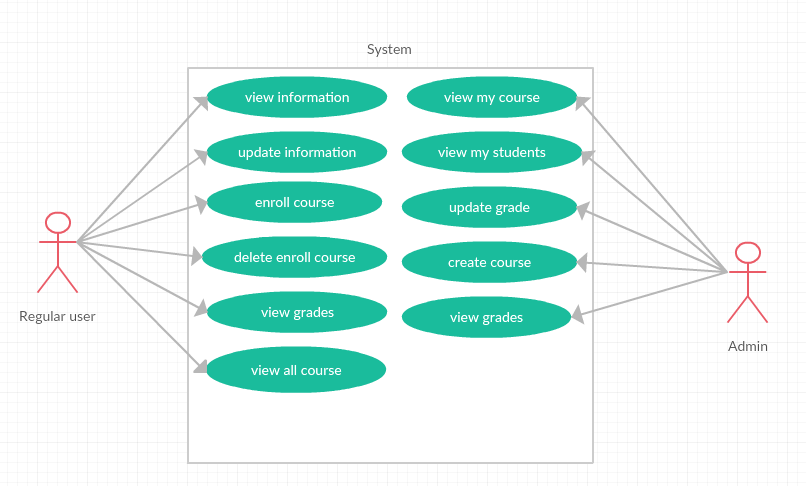
The administrator user can perform the following operation:

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

# Non-functional Requirements

* The app must be easy to use
* Have a friendly interface
* To ensure data security

2. Use-Case Model

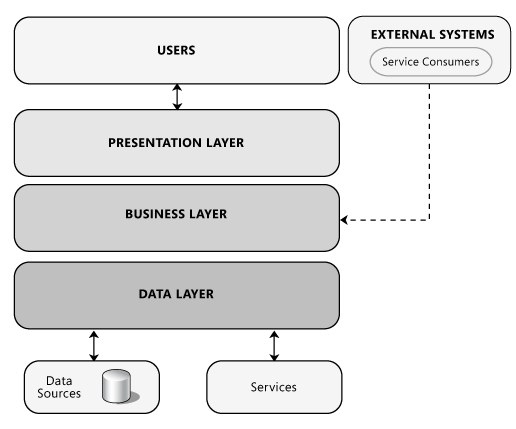
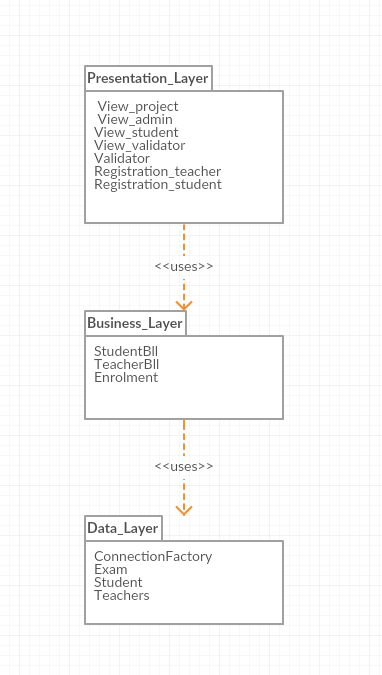
**

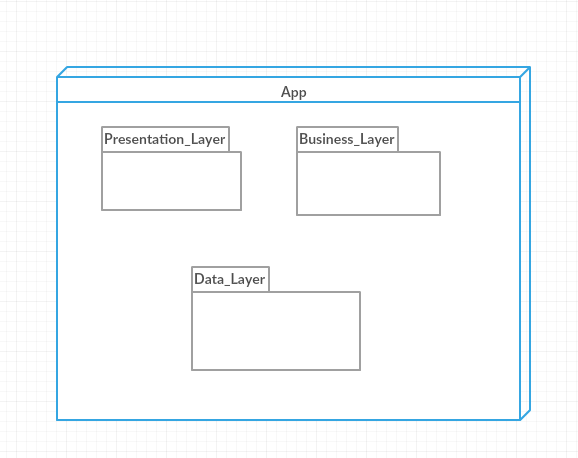
3. System Architectural Design

**3.1 Architectural Pattern Description**

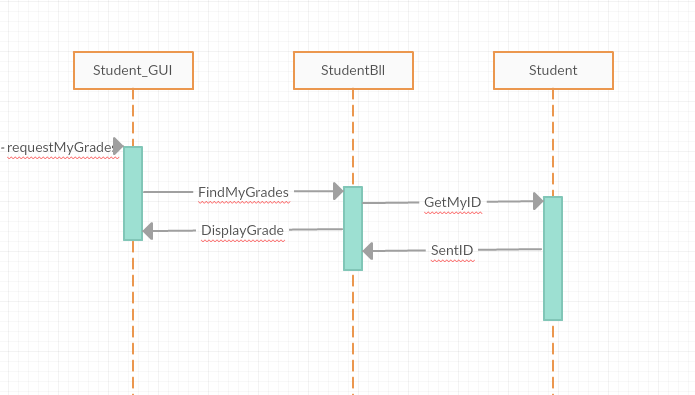
For this application I will use Layered Design. The logic of this architecture is to separate the features of the components, making it easier to create a design that supports reusability of components. Each logical layer contains a number of discrete component types grouped into sub layers, with each sub layer performing a specific type of task.

**3.2 Diagrams**

****

****

4. UML Sequence Diagrams

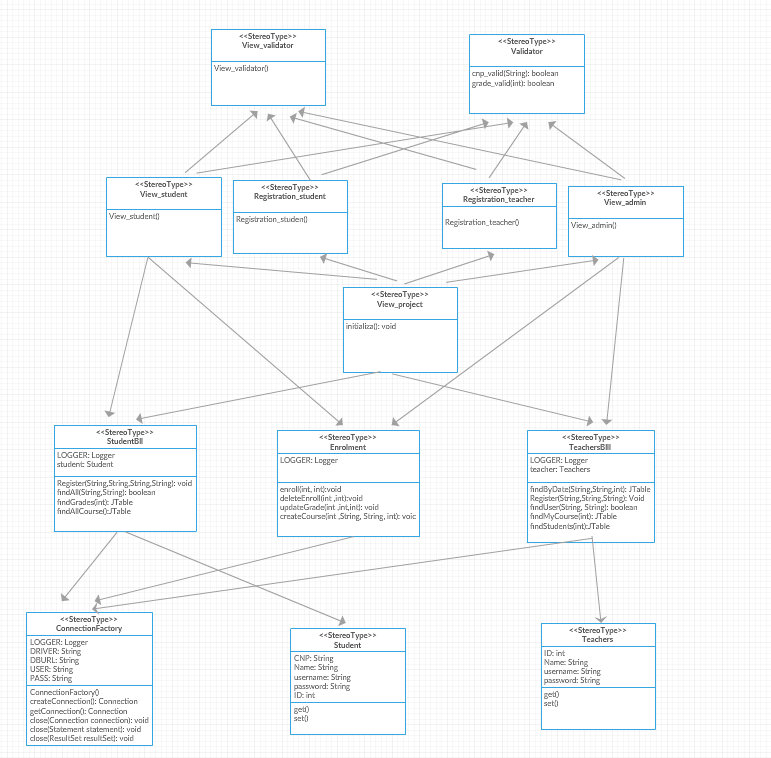
**

5. Class Design

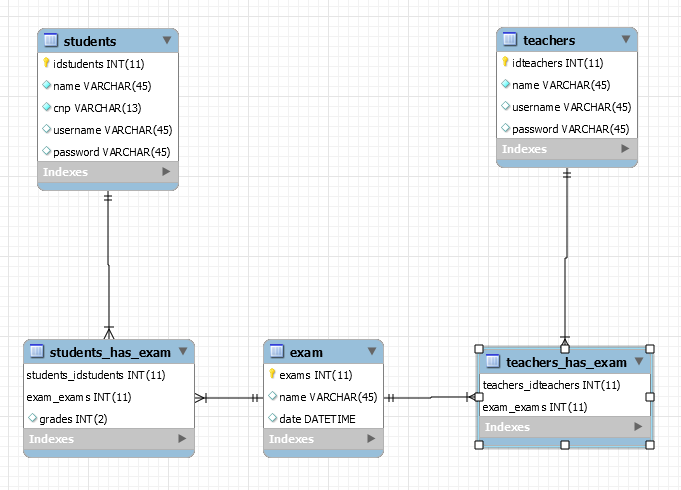
**5.1 Design Patterns Description**

**Factory method pattern** is a creational pattern that uses factory methods to deal with the problem of creating objects without having specify the exact class of the object that will be created. This is done by creating objects by calling a factory method ither specified in an interface and implemented by child classes.

**5.2UMLClassDiagram**



6. Data Model



For this app I will use a database to store the information about users. This model will contain three entity(student, admin, courses) that will be associated with a certain logic.

7. System Testing

Integration testing:- Testing performed to expose defects in the interfaces and in the

interactions between integrated components or systems.

Data Flow:- Data Flow testing is one of the testing strategies, which focuses on the data variables and their values. The basic idea behind this form of testing, is to reveal the coding errors and mistakes, which may results in to improper implementation and usage of the data variables or data values in the programming code i.e. data anomalies, such as:

* All the data variables, present in the programming code have been initialized or not,
* Data variables which are put into use, have been, priorly initialized or not,
* If the initialized data variables, has been used, at least once, in the programming code.

8. Bibliography

* <https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff650706(v=pandp.10)>
* <http://www.professionalqa.com/data-flow-testing>
* <https://www.tutorialspoint.com/uml/>
* <https://dev.to/chrisvasqm/introduction-to-unit-testing-with-java-2544>