Assignment A1

Analysis and Design Document

Student: Chendris Calin 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 ( user and admin ) which have to provide a username and a password in order to use the application.

# Functional Requirements

The regular user can perform the following operations:

- Add/update/view client information (name, identity card number, personal numerical code etc.)

- Create/update/delete/view student profile (account information: identification number, group enrolment, grades).

- Process class enrolment (enroll, exams, grades).

The administrator can perform the following operations:

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

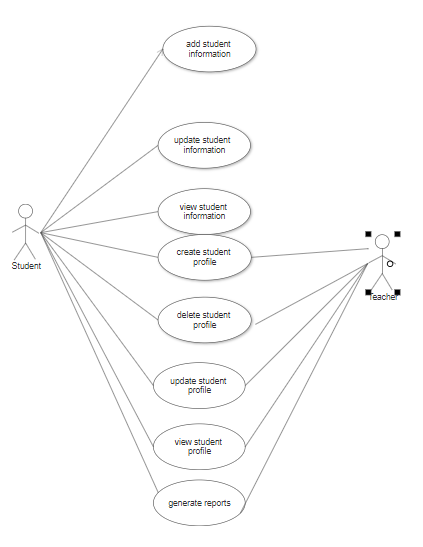
The data will be stored in a relational database (Postgres / MySql / MsSql). Use the Layers architectural pattern to organize your application. Create the Data Access Layer using SQL statements in the way you find most suitable for the application.

All the inputs of the application will be validated against data before submitting the data and saving it.

# Non-functional Requirements

Availability, Confidentiality, Efficiency,.

2. Use-Case Model



Use case: Create User

Level: user-goal level

Primary actor: Student

Main success scenario:

1. The student enters first time.

2. The student push the create student profile button.

3. The student introduce a username and a password

4. An account is created and it takes the student to the log-in window where he/she introduces the username and password that he/she created.

5.The student pushes the log-in button and the application allows the log-in if the details are correct.

Extensions: 5.The student pushes the log-in button but the application doesn’t allow then log-in because he/she entered an incorrect username/password.

3. System Architectural Design

**3.1 Architectural Pattern Description**

The most common architectural pattern is the Layered architecture.

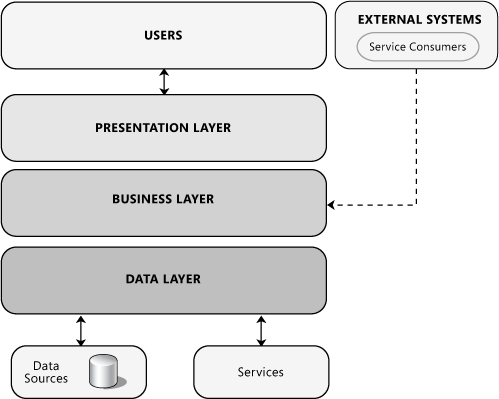
In a layered architecture, the layers can be used in a strict way, where a layer only knows the layer directly beneath it, or in a more flexible approach where a layer can access any layer beneath it.

Components of the Layered architecture:

* **Presentation layer**. This layer contains the user oriented functionality responsible for managing user interaction with the system, and generally consists of components that provide a common bridge into the core business logic encapsulated in the business layer.
* **Business layer**. This layer implements the core functionality of the system, and encapsulates the relevant business logic
* **Data layer**. This layer provides access to data hosted within the boundaries of the system, and data exposed by other networked systems; perhaps accessed through services.

**3.2 Diagrams**

**System’s conceptual architecture**

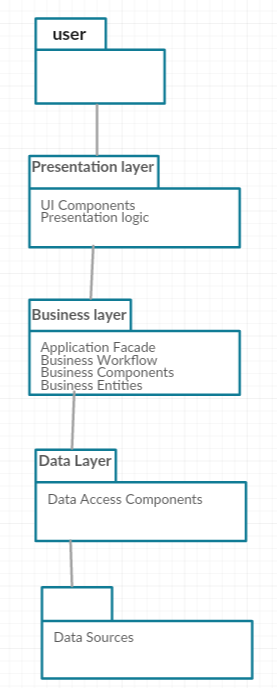
****

Presentation Layer includes the UI of the application

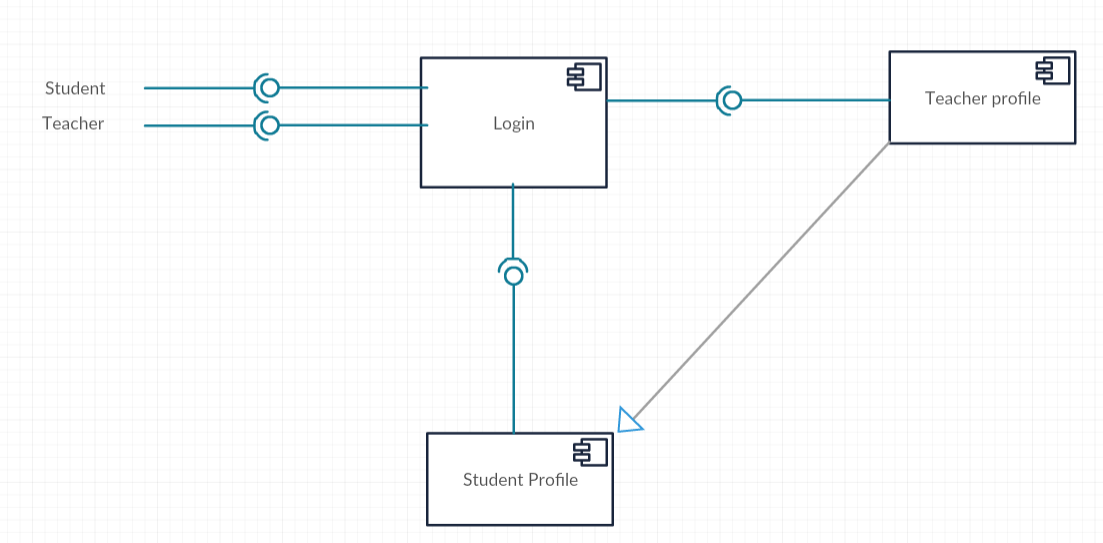
Business Layer : the logic of the application

Data Layer: classes which communicate with the database.

Package Diagram



Component diagram



4. UML Sequence Diagrams

5. Class Design

**5.1 Design Patterns Description**

*[Describe briefly the used design patterns.]*

**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

6. Data Model

*[Present the data models used in the system’s implementation.]*

7. System Testing

*[Present the used testing strategies (unit testing, integration testing, validation testing) and testing methods (data-flow, partitioning, boundary analysis, etc.).]*

8. Bibliography

<https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ee658109%28v%3dpandp.10%29>

<https://herbertograca.com/2017/08/03/layered-architecture/>