Assignment1

Student: Álvaro Suárez del Cueto

**Group: Erasmus**

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

# Application for the management of students in the University of Cluj Napoca.

# Functional Requirements

The application has two types of user’s student and teacher/administrator  
user, which have to provide a username and a password in order to use the application and will perform different operations on the system:

* Regular user can perform the following operations:  
  - Add/update/view client information (name, identity card number, personal numerical code, address, etc.).  
  - Create/update/delete/view student profile (account information: identification number, group, enrolments and grades).  
  - Process class enrolment (enroll, exams, grades).
* The administrator user can perform the following operations.  
  - CRUD on student’s information.  
  - Generate reports for a particular period containing the activities performed by a student

# Non-functional Requirements

# 1.3.1. Availability.

* Source of stimulus: teacher.
* Stimulus: Access the web service through the login webpage.
* Environment: runtime
* Artifact: Management System.
* Response: the system ask for a pair username/password which will uniquely identify the user.
* Response measure: 200 OK is the response if user exist, 404 not found if not

## 1.3.2. Performance.

* Source of stimulus: teacher.
* Stimulus: submit grades.
* Environment: runtime
* Artifact: Management System.
* Response: the system receives the marks written by the teacher.
* Response measure: The update of the information should be less than 2 seconds.

## 1.3.3. Security.

* Source of stimulus: Administrator.
* Stimulus: Access the web service through the login webpage using an administrator account.
* Environment: runtime
* Artifact: Management System.
* Response: the system ask for a pair username/password which will uniquely identify the administrator.
* Response measure: if the user is an administrator he/she will be redirected to its corresponding webpage, if a regular user is instead, trying to access an administrator account an error message will be shown.

## 1.3.4. Usability

* Source of stimulus: teacher.
* Stimulus: submit grades.
* Environment: runtime
* Artifact: Management System.
* Response: the system receives the marks written by the teacher.
* Response measure: webpage design and navigability should be easy for non-informatics familiar people.

2. Use-Case Model

* Use case diagram:

***Use case: Log In.***

***Level: user-goal level.***

***Primary actor: Teacher.***

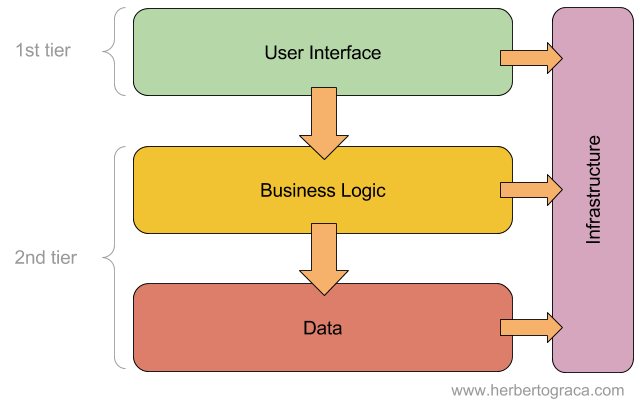
***Main success scenario: User introduces Username/Password and he enters the system.***

***Extensions: in case of success he will face the chat webpage, in case of failure he will see a 404 user not found message.***

3. System Architectural Design

**3.1 Architectural Pattern Description**

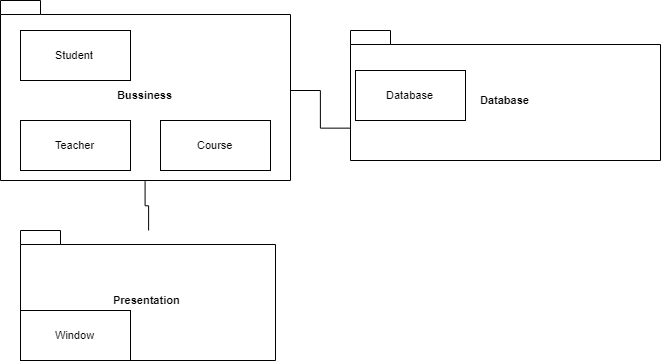
The system will follow the Layered architecture:



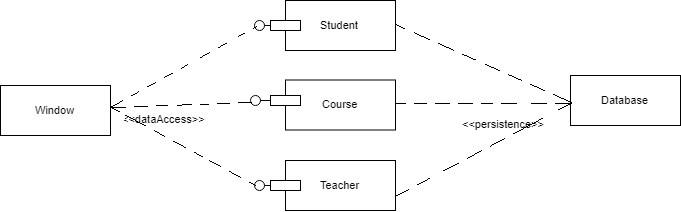
The user interface will be the Java Swing window, the business logic the classes and their correspondent behavior which will be exposed afterwards and the data layer which will be de database class with all the database operations on it.

**3.2 Diagrams**

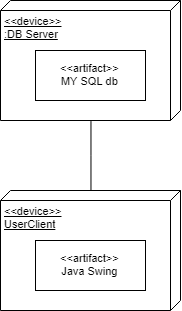
* Package:

**

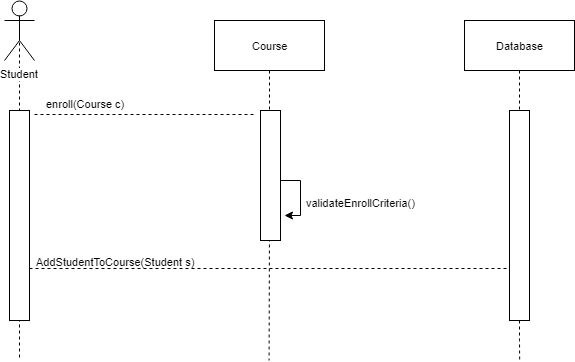
* Component:



* Deployment:



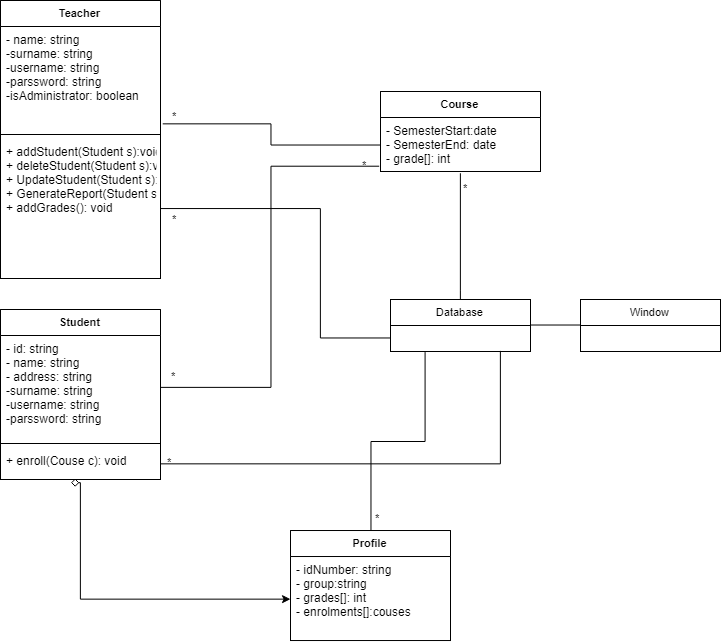
4. UML Sequence Diagrams



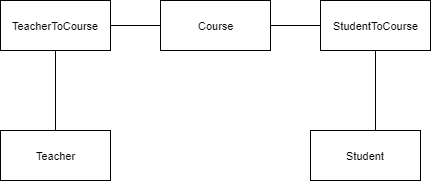
5. Class Design

**5.1 Design Patterns Description**

**5.2 UML Class Diagram**

**

6. Data Model

**

7. System Testing

The system will be tested using unit testing and using the dataflow method.