ASSIGNMENT A1
Students Management Application

Analysis and Design Document

Student: Biris Alexandra
Group: 30432

Table of Contents

1. Requirements Analysis

1.1 Assignment Specification

Java application for the management of students in the CS Department at Technical University of Cluj-Napoca.

1.2 Functional Requirements

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.

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

The administrator user can perform the following operations:

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

1.3 Non-functional Requirements

a) Availability: the system needs to be available 90% for the user with less activity during summer time
b) Accuracy: the system should accurately provide real time information
c) Performance: the system should respond to the user in less than several seconds from the time of the request submittal
d) Security: all system data must be backed up every 24 hours and this can be achieved by developing a second database
e) Usability: the system will have a GUI, it will be user friendly and it will not require a special training

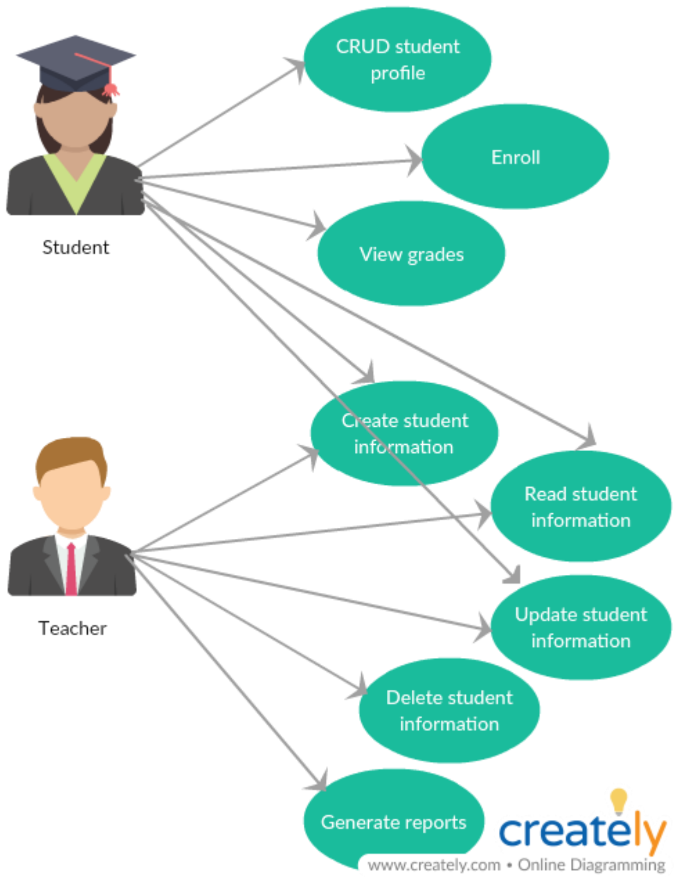
2. Use-Case Model

Use case: View grades
Level: User-goal level
Primary actor: Student
Main success scenario:

-the student logs in her/his account using a username and a password
-the view grades button is pressed
-a list with all the exam grades is displayed on the screen

Extensions:

-problems at the log in, such as student forgetting the username or the password
-error encountered when accessing the database


Student
Teacher
CRUD student profile
Enroll
View grades
Create student information
Read student information
Update student information
Delete student information
Generate reports
creately
www.creately.com • Online Diagramming

3. System Architectural Design

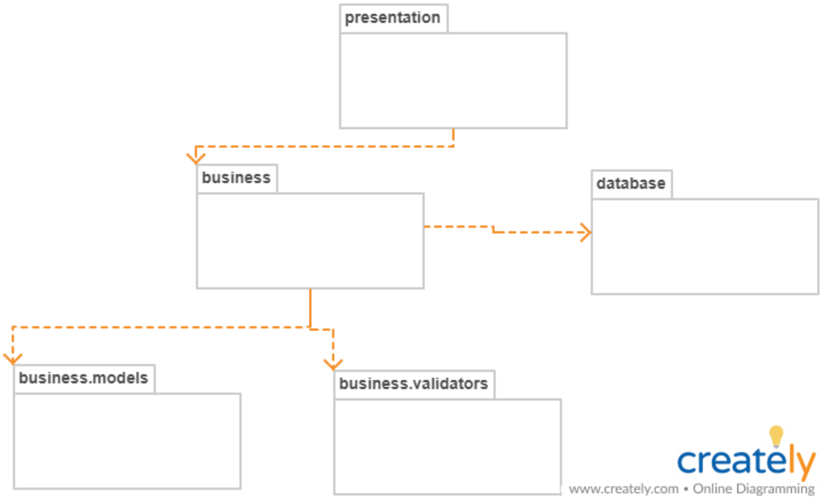
3.1 Architectural Pattern Description

The architectural pattern used is Layers. Components within this pattern are organized into horizontal layers, each layer performing a specific role within the application. Although it does not specify the number and types of layers that must exist in the pattern, most layered architectures consist of three standard layers: presentation, business and database.

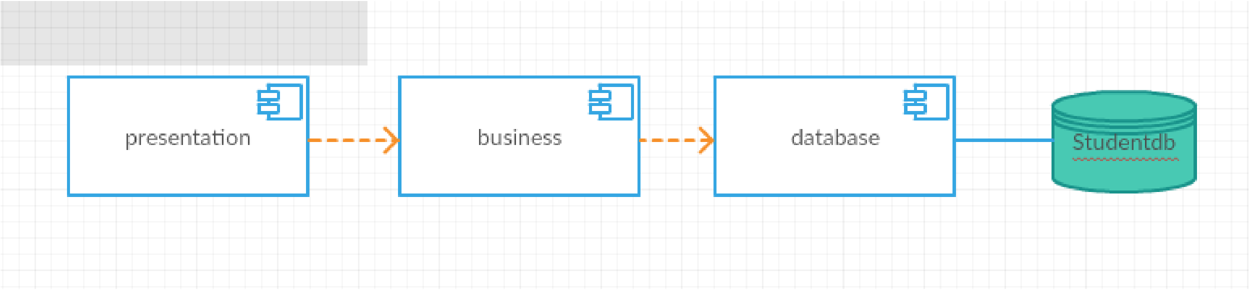
1. Presentation layer: responsible for handling all user interface and browser communication logic
2. Business layer: responsible for executing specific business rules associated with the request
3. Database layer: responsible for executing SQL statements to retrieve the corresponding data and pass it back up in the business layer.

3.2 Diagrams

Package diagram:

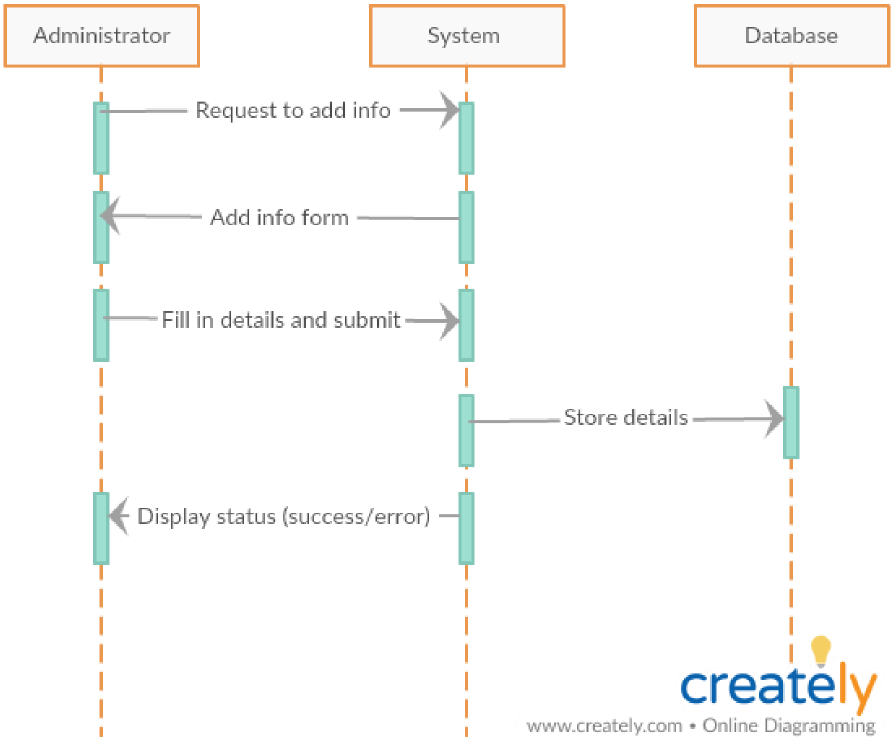


Component diagram:



4. UML Sequence Diagrams

The sequence diagram for creating a student information:

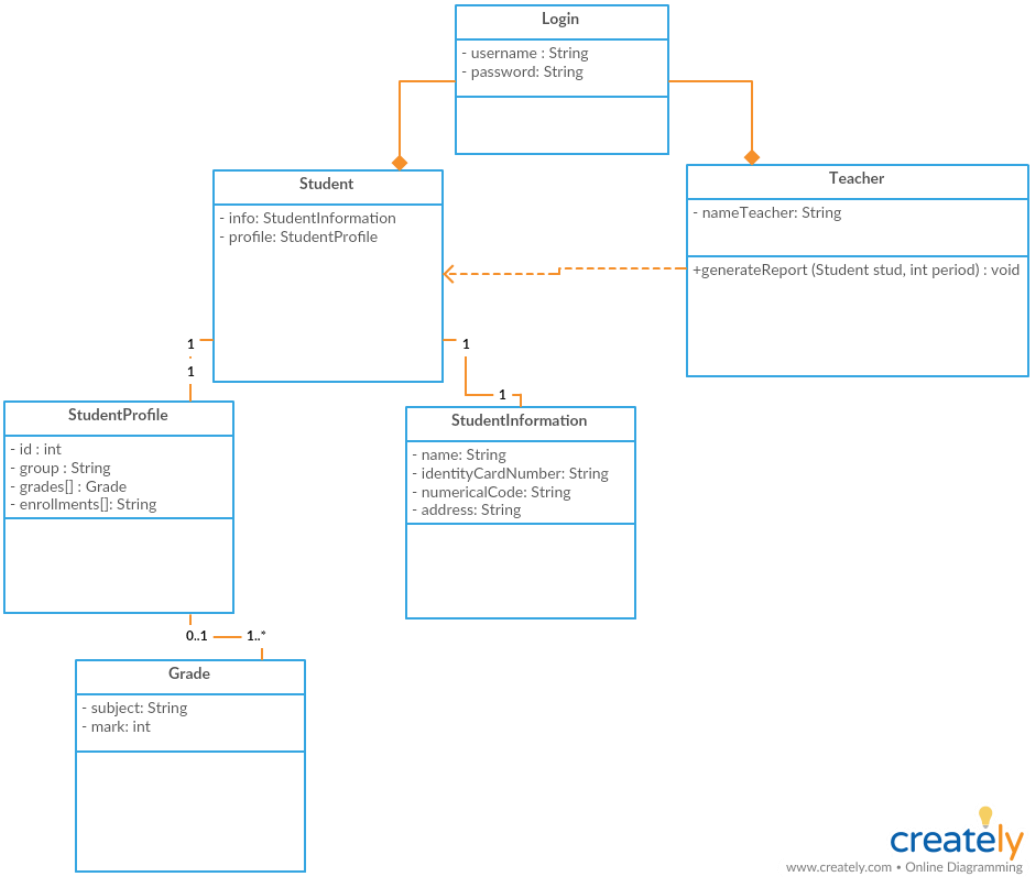

Administrator
System
Database
Request to add info
Add info form
Fill in details and submit
Store details
Display status (success/error)
creately
www.creately.com • Online Diagramming

5. Class Design

5.1 Design Patterns Description

Singleton pattern involves a single class which is responsible to create an object while making sure that only single object gets created. I used this pattern in the creation of the connection to the database, to ensure the fact that a single connection is established and to maintain it.

5.2 UML Class Diagram



6. Data Model

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

7. System Testing

The testing technique that will be used is unit testing, which involves breaking my program into pieces and subjecting each piece to a series of tests. Usually we write the test cases ourselves, but some can be automatically generated.

8. Bibliography

https://www.safaribooksonline.com/library/view/software-architecture-patterns/9781491971437/ch01.html
https://msdn.microsoft.com/en-us/library/ee658109.aspx