

Class Management System

PROJECT JAVA2 LA2 INFORMATIQUE

REALIZED BY:

* Jaziri Taha
* Mannai Azza

Higher Institute of Computer Science and Mathematics of Monastir 2018-2019

PLAN

Chapter 1: Project Presentation………………………………………………………….. 3

Introduction…………………………………………………………………………..... 4

1.Specification and Analysis of Functional Needs……………………… 5

2.Specification and Analysis of Non-Functional Needs..................... 9

Conclusion .......................................................................................................... 7

Chapter 2: Analysis and needs Conception.........................…………………….. 8

1. Identification of actors............................................................................... 9
2. Use case diagram......................................................................................... 12
3. Sequence diagram....................................................................................... 22
4. Class diagram................................................................................................ 26

Conclusion.......................................................................................................... 27

Chapter 3: Realization and solution.........................……………………..................28

Introduction.......................................................................................................... 29

1.Working Environment................................................................................... 29

2. Interfaces..........................................................................................................30

General Conclusion ............................................................................................... 45

Chapter 1: Project Presentation

**Introduction**

Seen that we are always having trouble with accessing our grades and results we choose to work on the first step of an application that is still as its primitive form to help solve that problem.

The main objective of this project is to put in place a new system of class management in which entry, sharing and deliberation are very easy to access, our application mainly consists on establishing a completed work of schooling: saving lists of students as well as their overall averages. This chapter’s focus point is to identify the functional and non-functional needs as well as the management rules necessary for this system’s development.

# Specification and Analysis of Functional Needs:

Users Management:

It has to assure:

* Add
* Delete

Users

* Update
* Consult

Students Management:

It has to assure:

* Add
* Delete

Students

* Update
* Consult

Subjects Management:

It has to assure:

* Add
* Delete

Subjects

* Update
* Consult

Grades Management:

- Save the grades of each student.

- Change the grades of each student

- Delete the grades of each student

- Consult the grades of each student

-Calculate the averages of the subjects of each student.

2.Specification and Analysis of Non-Functional Needs:

Ergonomics requirements:

* A user-friendly interface, readable and easy to use.
* Short Response Time
* Use lists containing correct values ​​to avoid errors.

Security:

* The system must be secure with the obligation for each user to enter a password and a login.
* Authentication: It is provided by an authentication module; it manages access between users and the application by a login and password.

**Conclusion:**

In this chapter we analysed our system (its implementation, its architecture) and we identified its functional and non-functional needs. In the next chapter I will discuss design methodologies, while presenting a use case design, sequence diagrams, as well as the class diagram.

Chapter 2: Analysis and needs Conception



**Introduction:**

We will present in this chapter the design part of the project. We

Will also build a complete view in the form of use case diagrams

, sequence and class diagram using UML as method and StarUML as design software.

1. Identification of actors :

The actors in our Application Are:

1. Employee
2. Student
3. Professor

Student:

* + - * + Consult grades.
        + Consult timeline.
        + Access account

Professor:

* + - * + Consult grades.
        + Consult timeline.
        + Access account
        + Change grade in case of error

=

Employee:

Manage Students:

* Add Student
* Delete Student
* Modify Student
* Consult Student

Manage Subjects:

* Add Subject
* Delete Subject
* Modify Sujbect
* Consult Subject

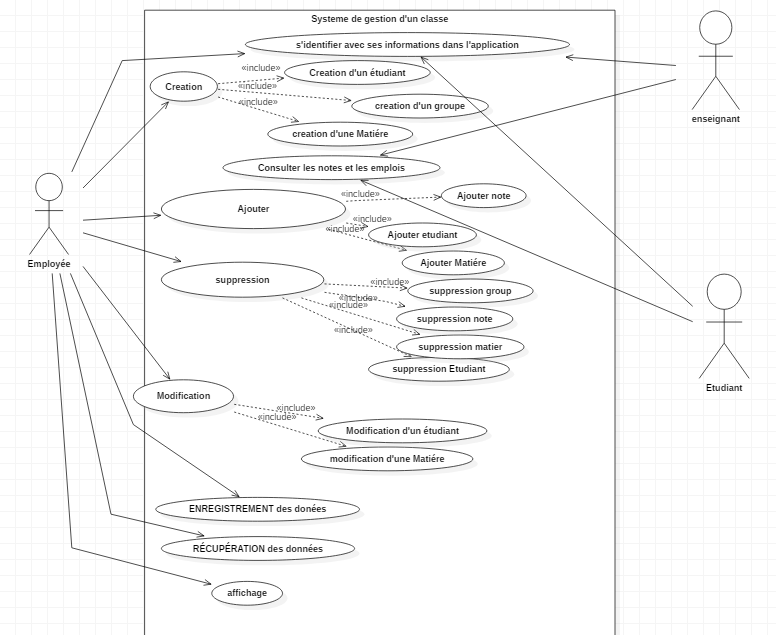
Manage Grades:

* Add grade
* Delete grade
* Modify grade
* Consult grade

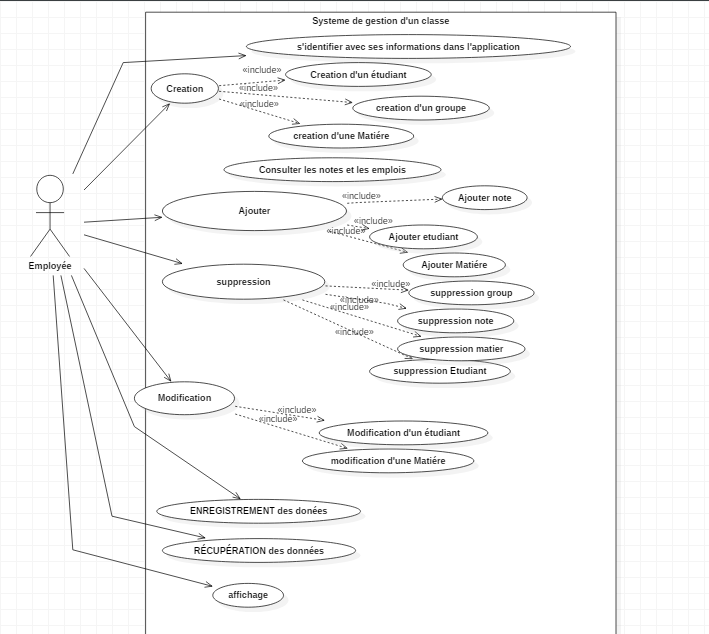
2.Use case Diagram:

Use cases describe the Performance of the system from the user’s perspective in the form of actions and reactions.

Global use case Diagram:



Employee use case Diagram:



Add Student:

|  |  |
| --- | --- |
| Actor | Employee |
| Pre-condition | - The system Employee authenticates.  - Account Employee is consulted |
| Post-Condition | The addition of a new Student is successful and saved in the database. |
| Description of the main scenario | -The Employee enters the information corresponding to the new Student (including his grades) and confirms his request for addition.  -The system adds the new user to the database and sends a message indicating the success of the operation |

Delete Student:

|  |  |
| --- | --- |
| Actor | Employee |
| Pre-condition | - The system Employee authenticates.  - Account Employee is consulted |
| Post-Condition | The removal of an already existing Student is successful and saved in the database. |
| Description of the main scenario | -The Employee enters the information corresponding to the already existing Student and confirms his request for removal.  -The system removes the already existing student from the database and sends a message indicating the success of the operation. |

Modify Student

|  |  |
| --- | --- |
| Actor | Employee |
| Pre-condition | - The system Employee authenticates.  - Account Employee is consulted |
| Post-Condition | The modification of the already existing Student is successful and saved in the database. |
| Description of the main scenario | -The Employee enters the information corresponding to the already existing Student (including his grades) and confirms his request for modification  -The system modifies the student in the database and sends a message indicating the success of the operation |

Add Subject:

|  |  |
| --- | --- |
| Actor | Employee |
| Pre-condition | - The system Employee authenticates.  - Account Employee is consulted |
| Post-Condition | The addition of a new Subject is successful and saved in the database. |
| Description of the main scenario | -The Employee enters the information corresponding to the new Subject and confirms his request for addition.  -The system adds the new Subject to the database and sends a message indicating the success of the operation |

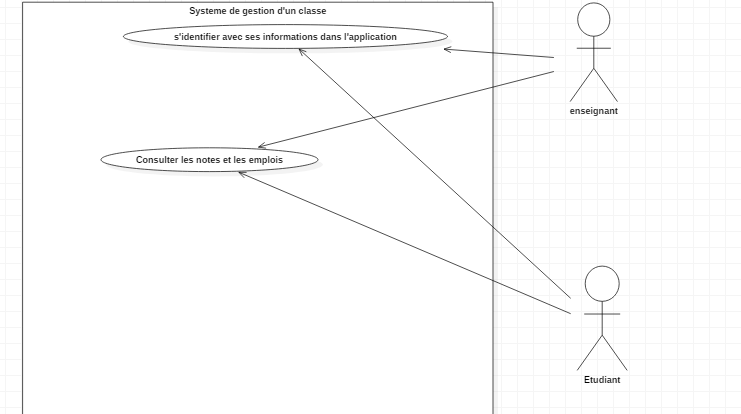
Delete Subject:

|  |  |
| --- | --- |
| Actor | Employee |
| Pre-condition | - The system Employee authenticates.  - Account Employee is consulted |
| Post-Condition | The removal of an already existing Subject is successful and saved in the database. |
| Description of the main scenario | -The Employee enters the information corresponding to the already existing Subject and confirms his request for removal.  -The system removes the already existing subject from the database and sends a message indicating the success of the operation. |

Modify Subject:

|  |  |
| --- | --- |
| Actor | Employee |
| Pre-condition | - The system Employee authenticates.  - Account Employee is consulted |
| Post-Condition | The modification of the already existing Subject is successful and saved in the database. |
| Description of the main scenario | -The Employee enters the information corresponding to the already existing Subject (including its name , coefficient and code ) and confirms his request for modification  -The system modifies the student in the database and sends a message indicating the success of the operation |

Student and Professor use case Diagram:

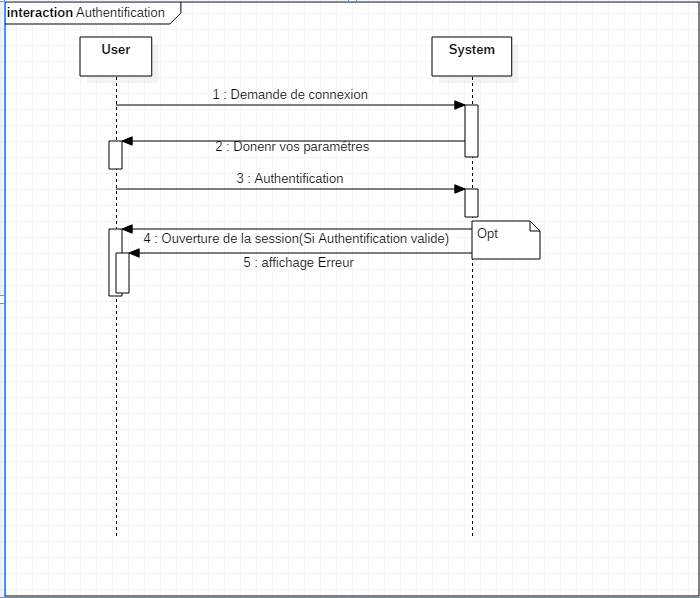


Login

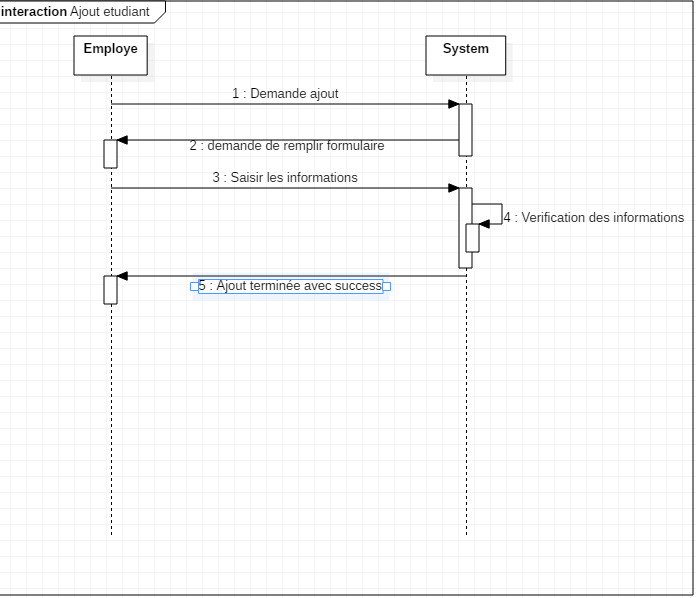
|  |  |
| --- | --- |
| Actor | Professor/Student |
| Pre-condition | - The system Professor/Student possesses a Password and an ID |
| Post-Condition | The Login is successful and the application can be consulted |
| Description of the main scenario | -The Professor/Student enters the information needed (ID, Code) and confirms his request for login  -The system checks the Professor/Student’ s information’s in the database and sends a message indicating the success of the operation  In case of new Professor/Student an account can be created |

3.Sequence Diagram:

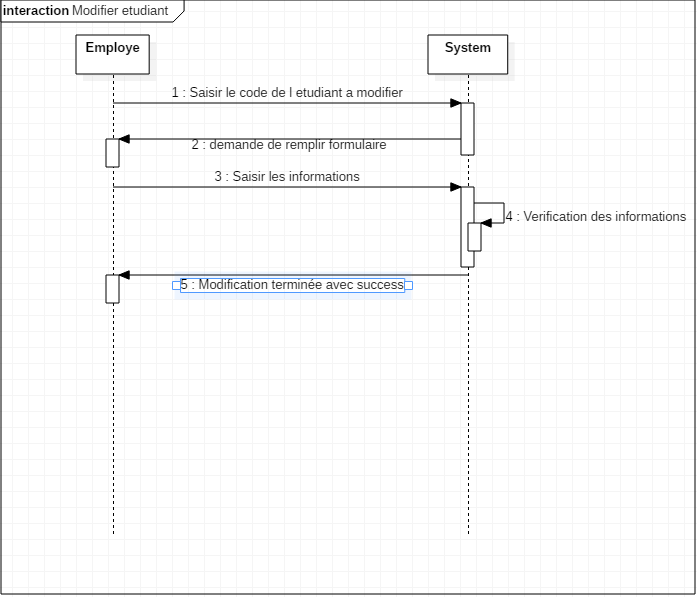
Authentication:



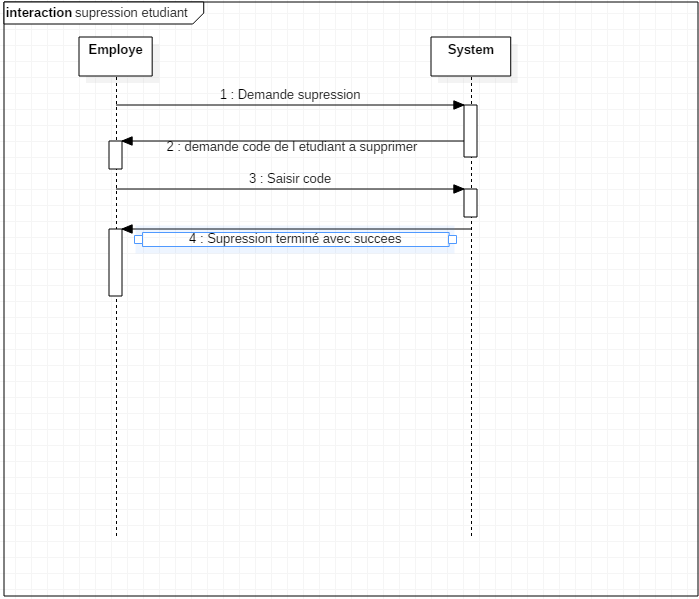
Add:



Modify:

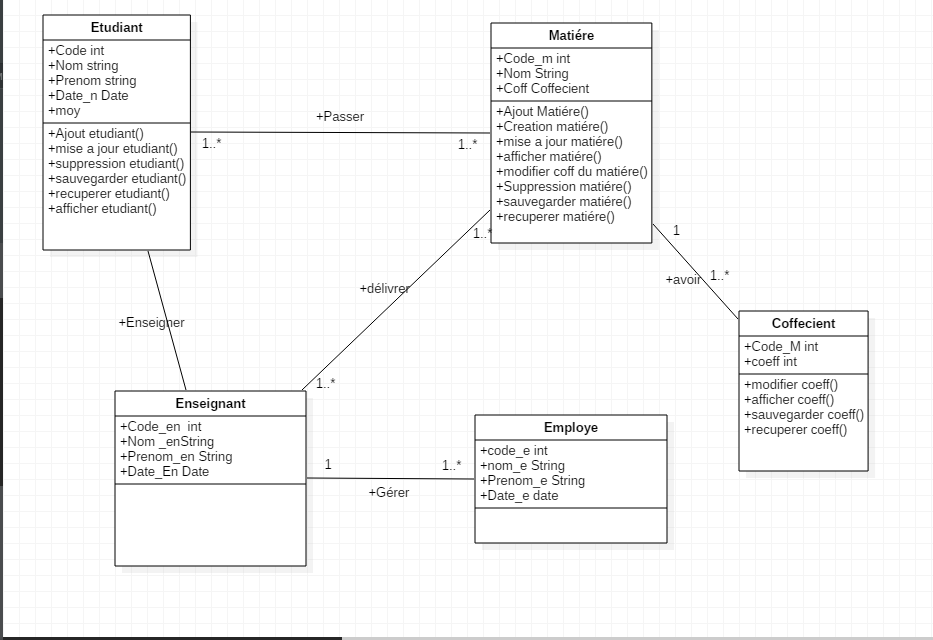


Delete:



4.Class Diagram:

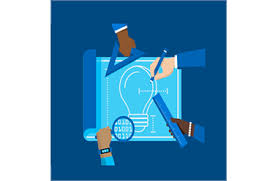
Now we will explain the classes developed and the associated packages during the application development cycle



**Conclusion**

In this chapter we have presented the different diagrams defined by UML which made it possible to understand the needs of the system to be developed as well as the different interactions between the objects participating in its functioning, which will facilitate the implementation and coding phase.

Chapter 2: Realization and solution



**Introduction:**

This chapter is the last part of the report. It aims to expose the work done. In which we will present the material environment of the project as well as the development tools used, we will then show the different screenshots illustrating the features of the application will present in this chapter the design part of the project.

1.Working Environment

Languages used:

* Java Language
* SQL: A Database Management System

API Java used:

1-Java.awt. \*

2-Javax.Swing. \*

3-Java.sql. \*

4-java.util.\*

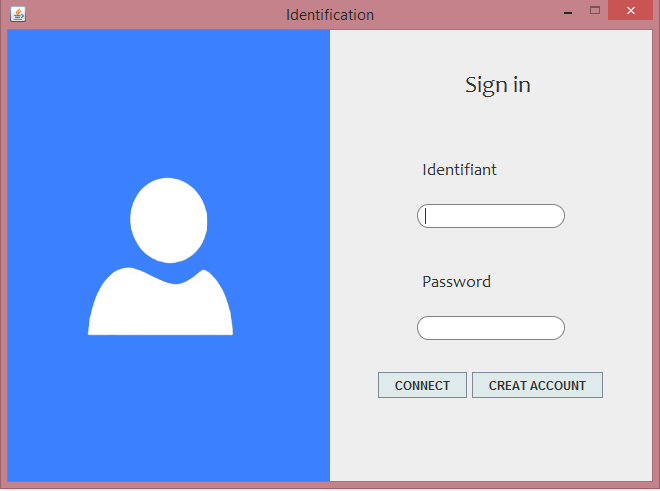
5- com.mysql.jdbc.connection

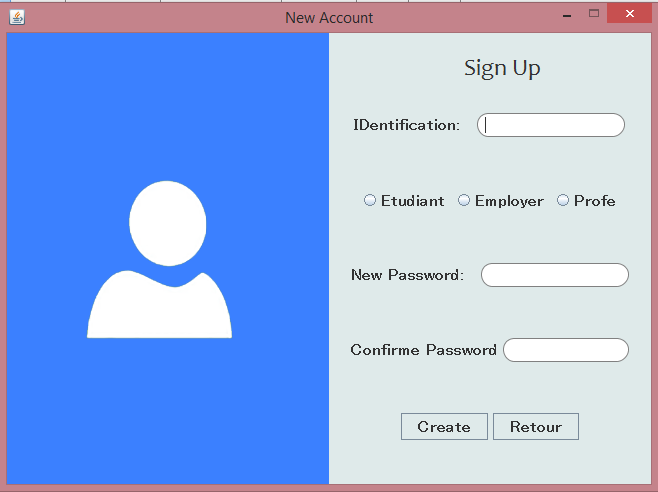
JDK used:

* JDK 1.8.0.201

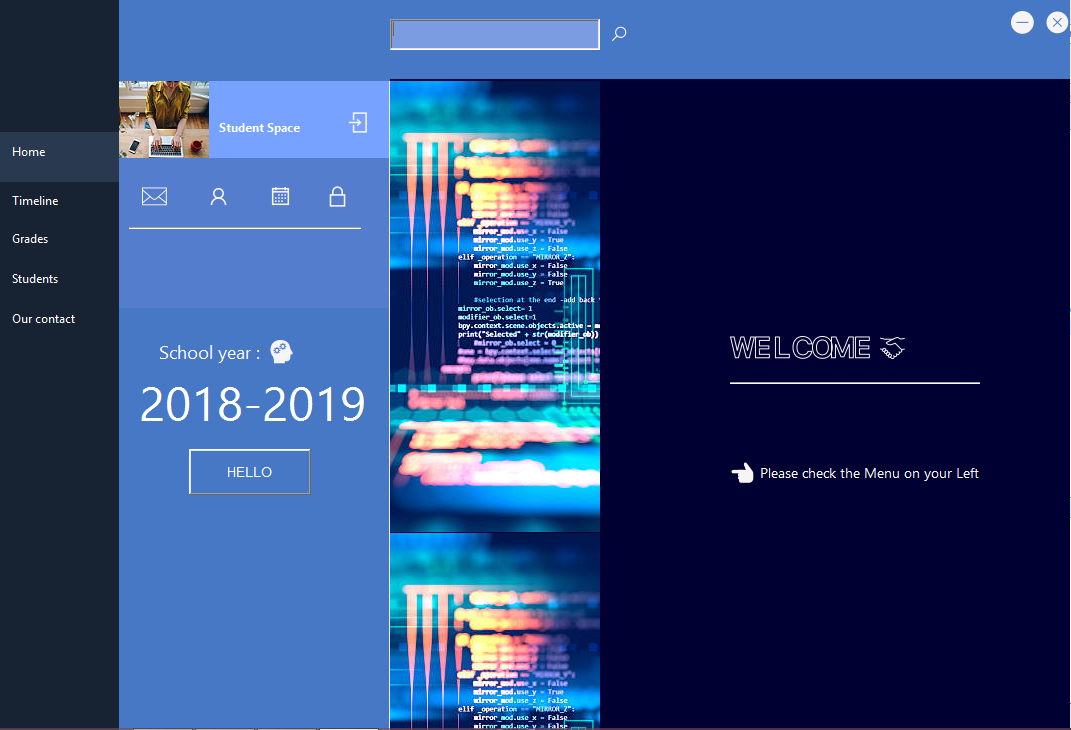
2.Interfaces:

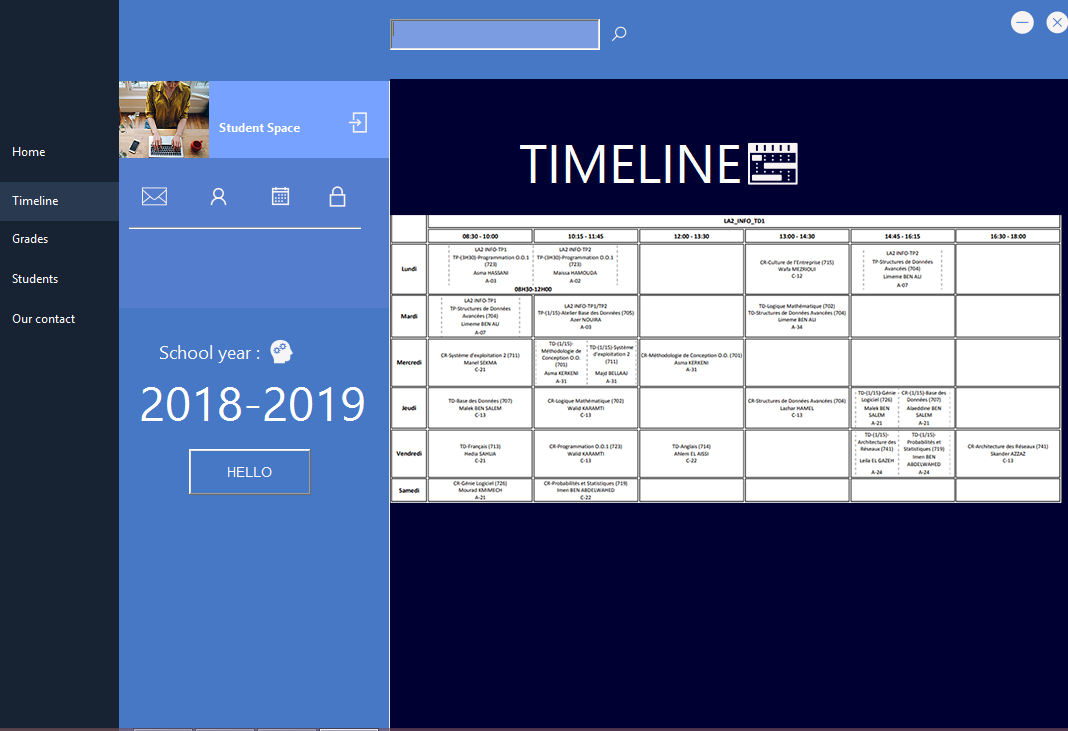
Authentication space:

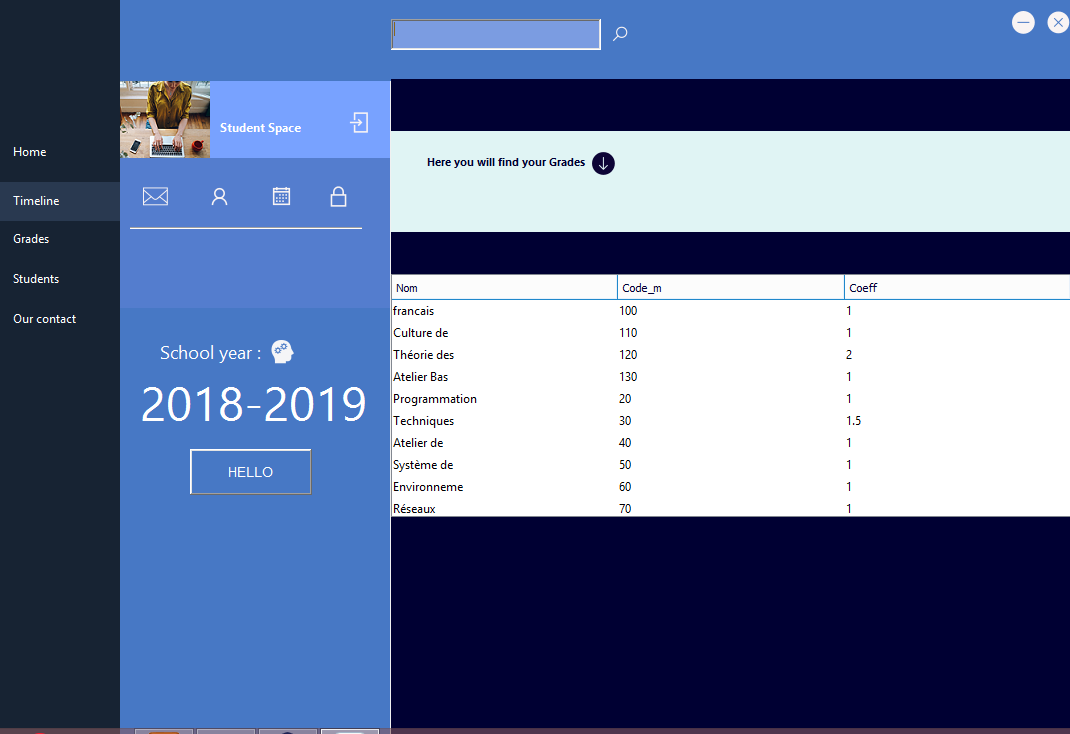


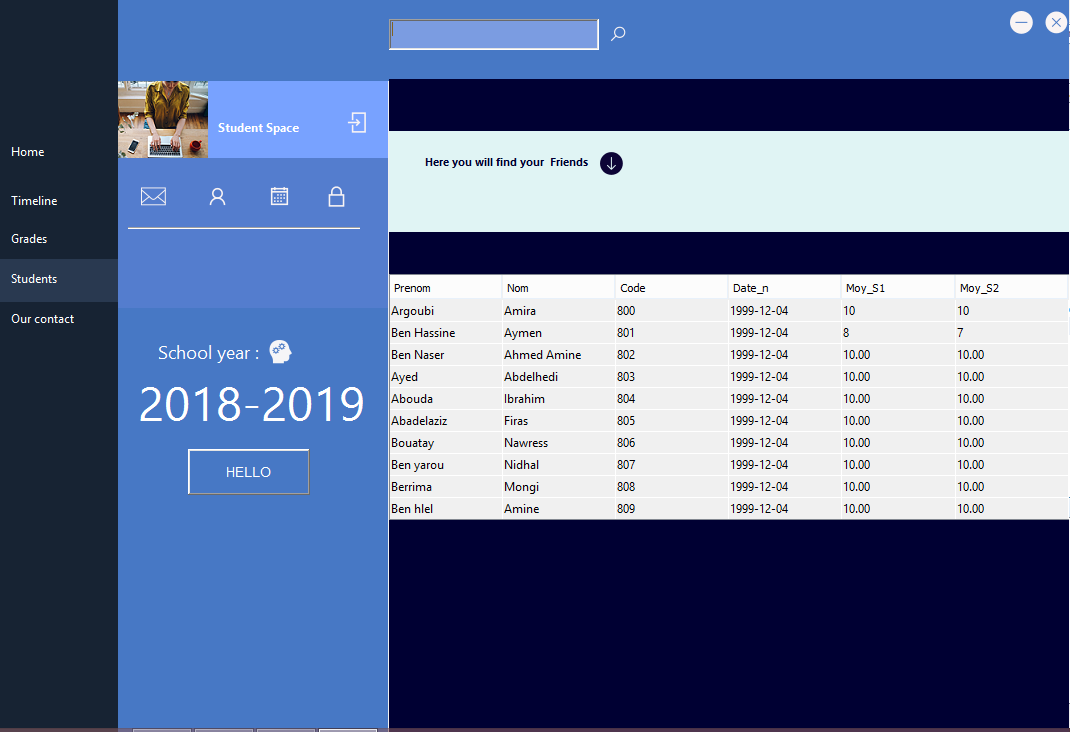


Student Space:

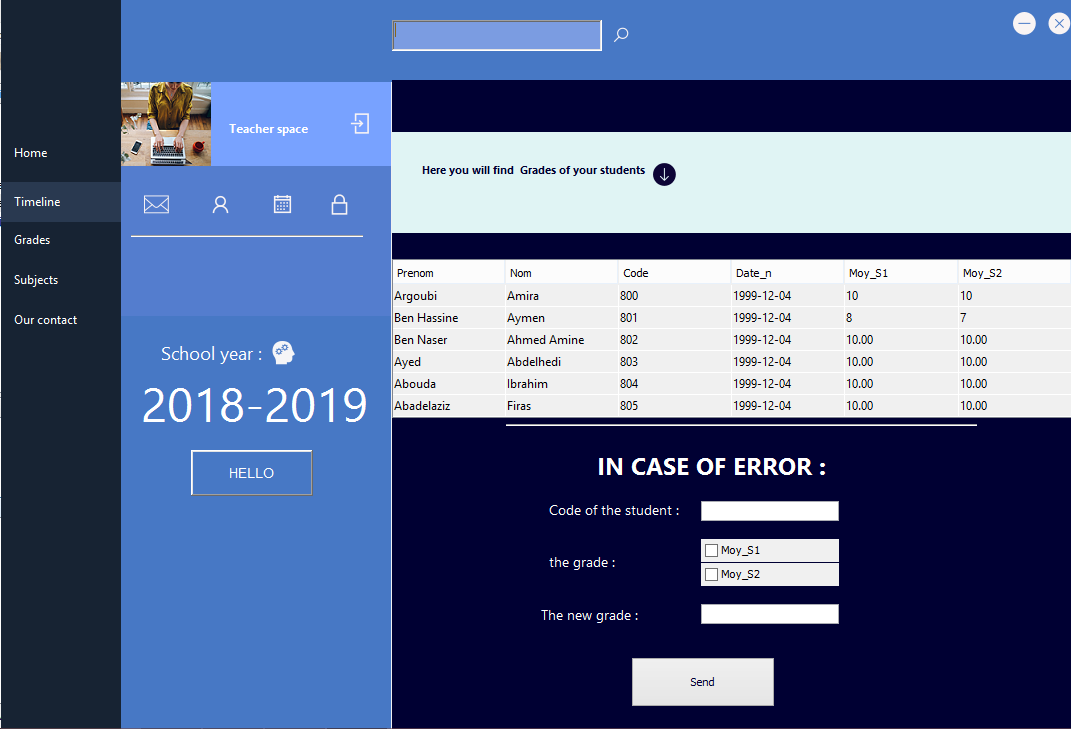


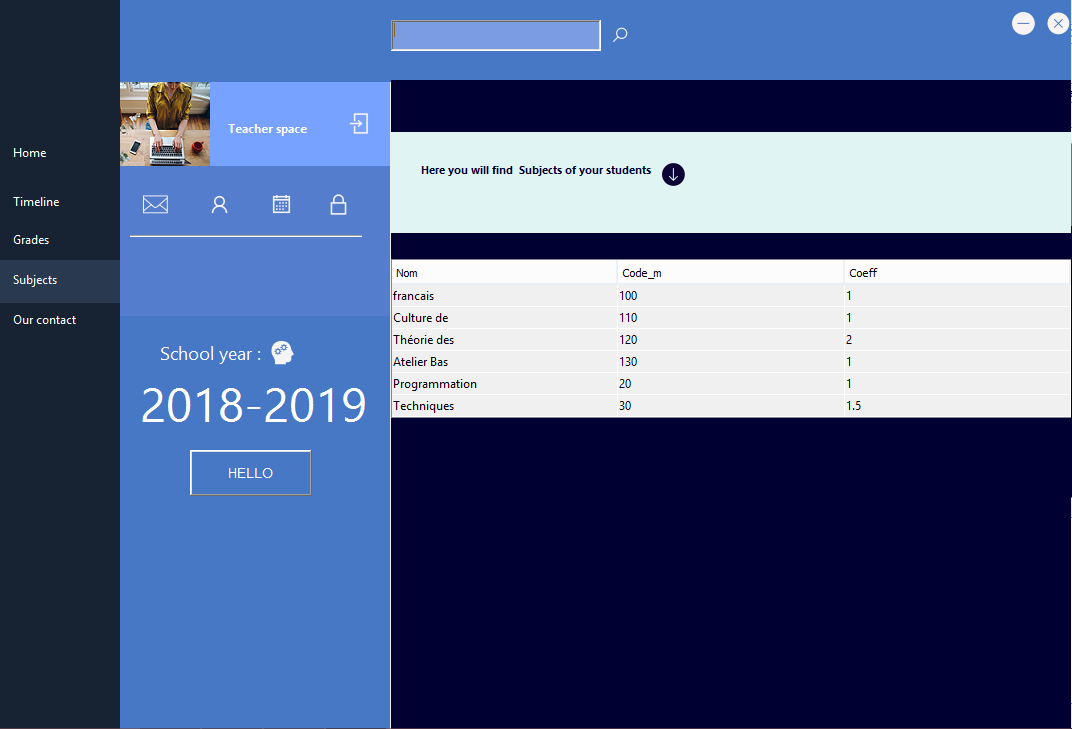






Professor Space:





Employee Space:

