Architecture Design

for

DegreeOverview

Version 1.0 approved

Prepared by:

Lei XIA (1830026125)

Yiping GAO (1830026026)

Zhenyu REN (1830026095)

Zhengyang SHEN (1730026092)

Jasmine

2021/4/21

Table of Contents

Table of Contents ii

Revision History ii

1. Overview 1

1.1 Project description 1

1.2 References 1

1.3 Design purpose 1

2. Overall description 1

2.1 Use case diagram and class diagram 1

2.2 Design model 3

2.3 System architecture 3

3. System architecture 3

3.1 LoginSubsystem 3

3.1.1 Description 4

3.1.2 Database 4

3.2 VisSubsystem 5

3.2.1 Description 5

3.2.2 Database 5

3.3 PerforSubsystem 6

3.3.1 Description 6

3.3.2 Database 6

3.4 LecSubsystem 7

3.4.1 Description 7

3.4.2 Database 8

4. Assessment 8

4.1 Stability 8

4.2 Reusability 8

4.3 Scalability 8

5. Alternative design (optional) 9

6. More considerations 9

7. Appendix 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Initial version | 2021/4/21 |  | 1.0 |
|  |  |  |  |

# Overview

## Project description

The **DegreeOverview** project is to help university student and teacher better understand the relationship between courses. The course designer could do some operation on the course. The students could visualize their outcomes easily.

## References

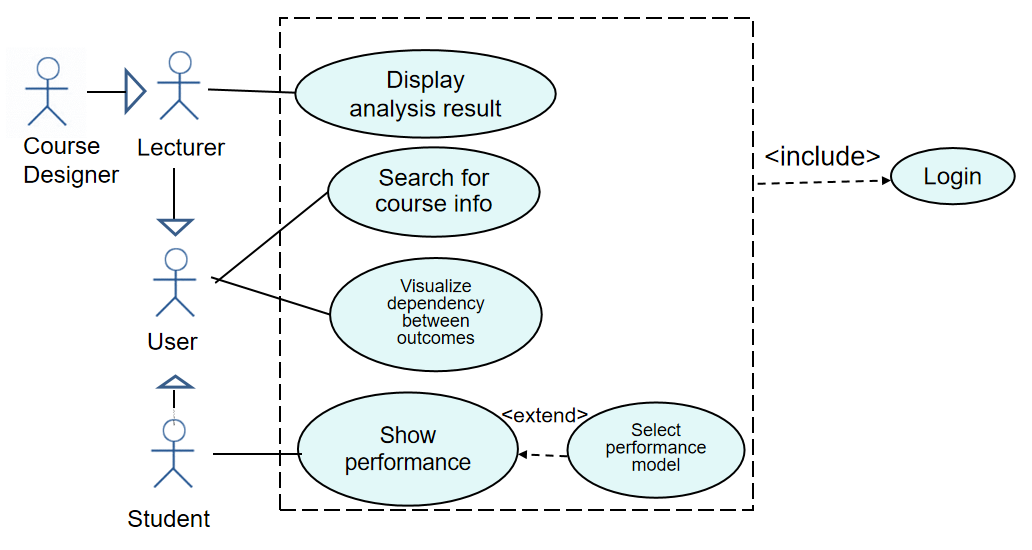
* Lei XIA, Yiping GAO, Zhenyu REN, Zhengyang SHEN. (April 19, 2021). SRS\_Jasmine1.4.docx.
* SDWIII Project V4.docx.

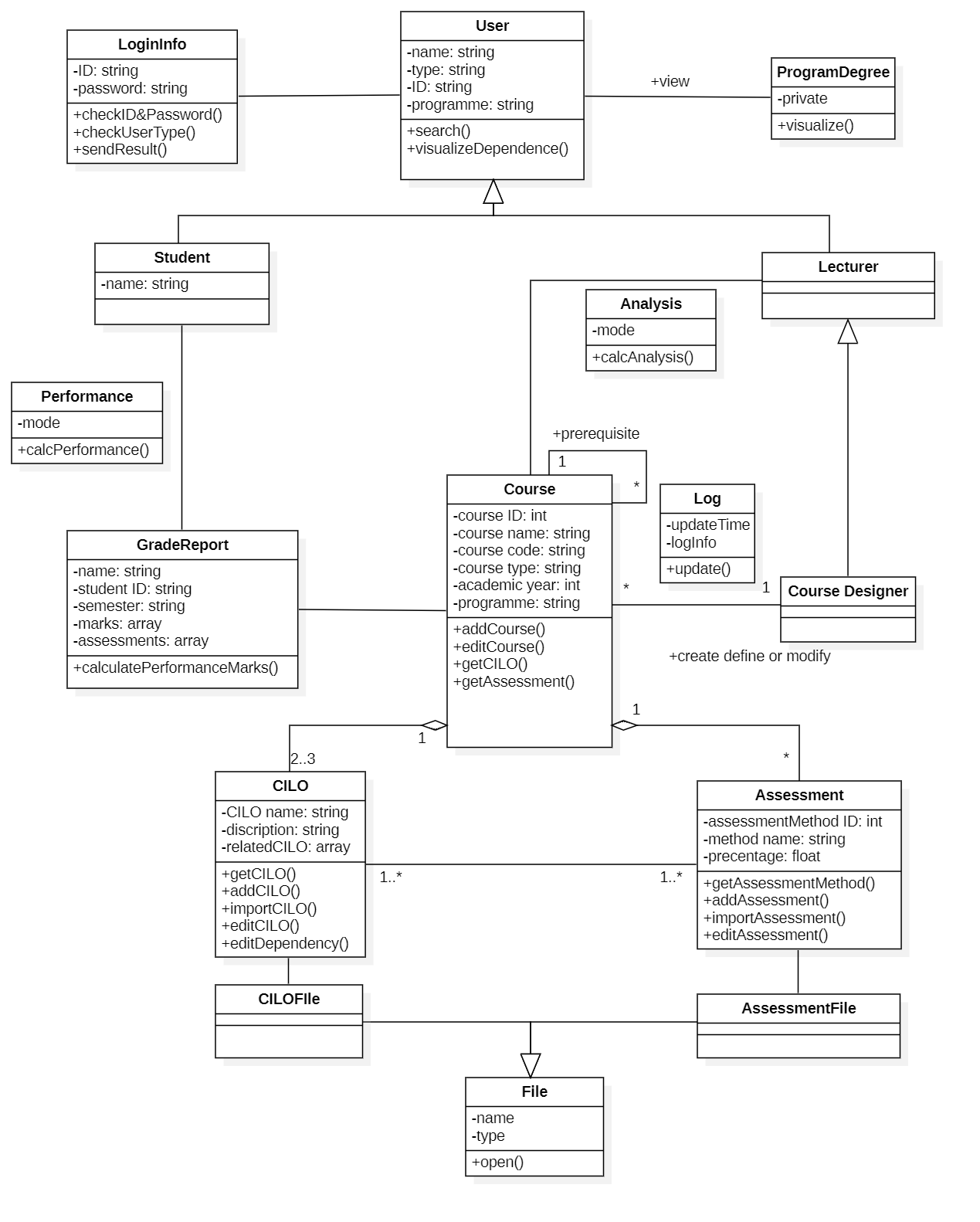
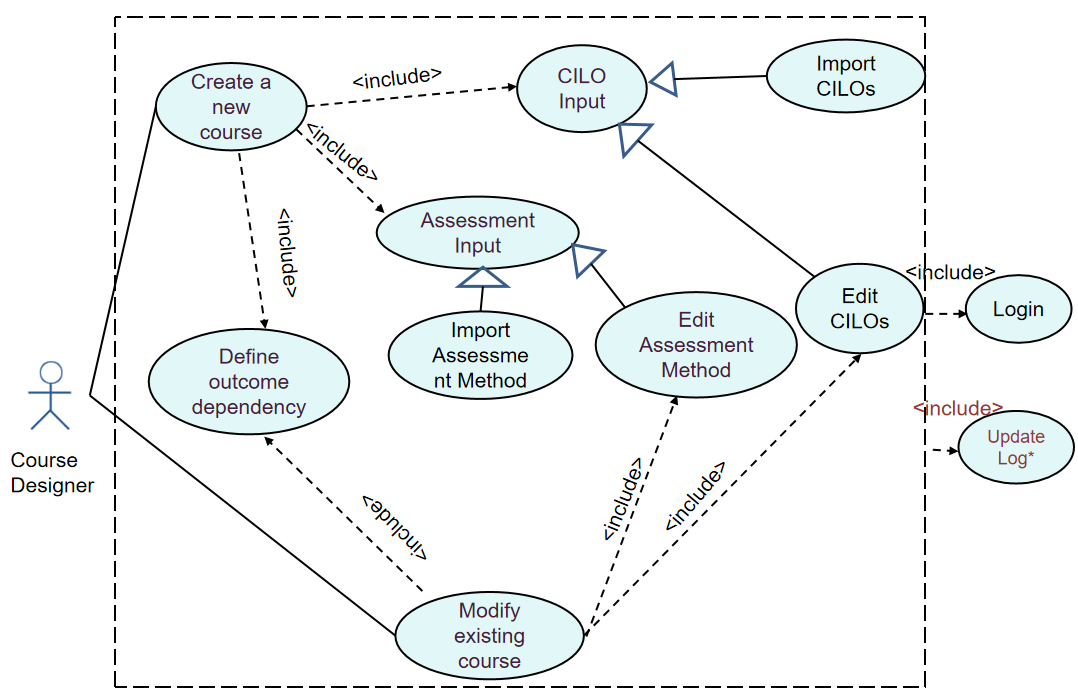
## Design purpose

* This design architecture looks clearly that help engineer programming the program easily.
* Based on our architecture design, the program could be developed by several group and each group could program the project parallelly.

# Overall description

## Use case diagram and class diagram

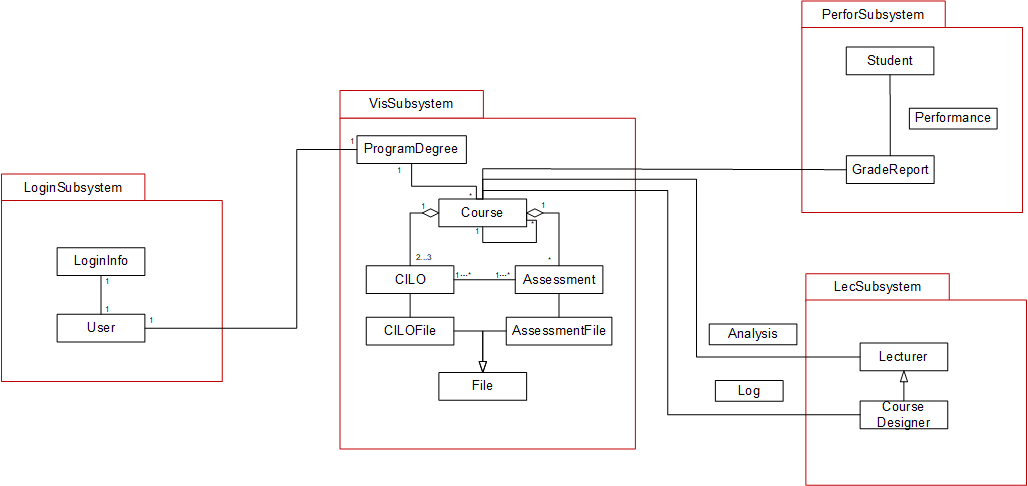
**Case diagram:**

****Class Diagram:**

## Design model

* The system Uses MVT (Model-View-Template) three-tier architecture.
* MVT has low coupling, high reusability, low life cycle cost, fast deployment, high maintainability and favorable software engineering management.
* The Three Tier model can clearly overview the relationship between the interface, application and storage.

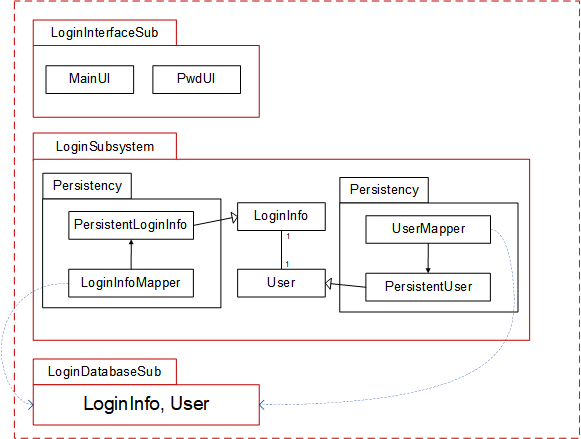
## System architecture

Following parts are the overall design of our system in architecture design diagram:

# System architecture

## LoginSubsystem

Login subsystem is designed for conduct login operation for all users.



### Description

This subsystem mainly includes the user class and login information class. The interface in this subsystem includes login interface and main page for each user. The database in this subsystem includes the table for login information and user information.

### Database

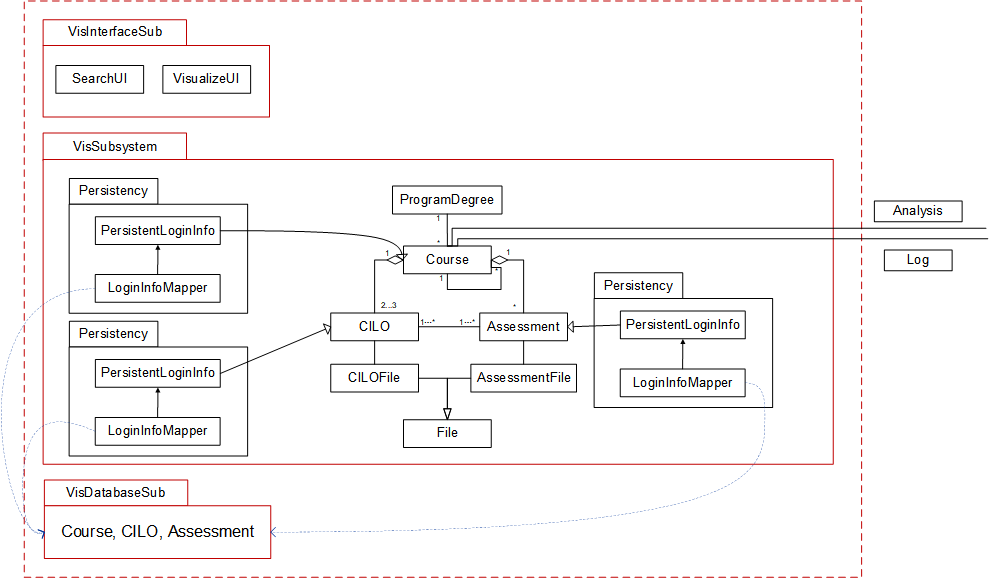
|  |  |  |
| --- | --- | --- |
| LoginInfo | | |
| UserID | Password | Type |
|  |  |  |
|  |  |  |

UserID is the primary key in this table, and there are also two attributes password and type. Type is to identify the user type, there are student, courseDesigner and non-courseDesigner in our system.

|  |  |  |
| --- | --- | --- |
| User | | |
| UserID | UserName | Program |
|  |  |  |
|  |  |  |

UserID is the primary key in this table, and there are also two attributes username and program. Programs identify the program that the user belongs to.

## VisSubsystem

This subsystem is designed for visualize the relative information of courses. It includes contents and details for course information including CILOs and assessments.

### Description

This subsystem mainly includes the course class, CILO class and assessment class. The interface in this subsystem includes searchUI and VisualizeUI. The database in this subsystem includes the table for course, CILO and assessment.

### Database

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Course | | | | |
| CourseID | CourseName | Code | Type | Program |
|  |  |  |  |  |
|  |  |  |  |  |

CourseID is the primary key in this table, and there are also four attributes: course name code type and program. Code is the assigned code that to arrange all the course in our system. Type is to identify the course type, like FE, ME, GC, etc. Program is used to identify the subject that a course belongs to.

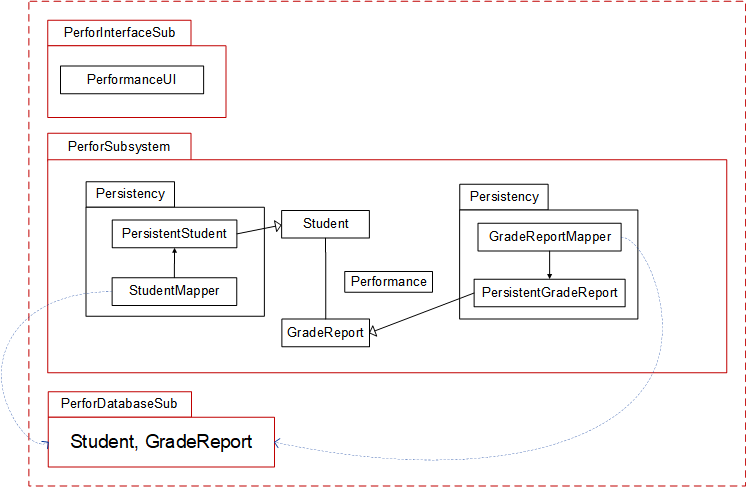
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CILO | | | | |
| CILOID | CILOName | Description | preCILO | CourseID |
|  |  |  |  |  |
|  |  |  |  |  |

CILOID is the primary key and CourseID is the foreign key in this table. There are also four attributes: CILO name, description, pre CILO and course. Pre CILO indicate that which CILO or CILOs are necessary before learning this CILO.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assessment | | | | |
| AssessmentID | MethodName | Percentage | CILOIDs | CourseID |
|  |  |  |  |  |
|  |  |  |  |  |

AssessmentID is the primary key and CourseID is the foreign key in this table. There are also four attributes: method name, percentage, CILOIDs and coursed. Method is the value method such like quiz and midterm. Percentage is the weight that this method takes for final result. CILOIDs is the associated CILO of current assessment.

## PerforSubsystem

This subsystem is used for showing the performance of the student based on the grade report.

### Description

This subsystem mainly includes the student class and GradeReport class. The interface in this subsystem includes PerformanceUI. The database in this subsystem includes the table for student and GradeReport.

### Database

|  |  |  |  |
| --- | --- | --- | --- |
| Student | | | |
| StudentID | StudentName | StudentYear | Program |
|  |  |  |  |
|  |  |  |  |

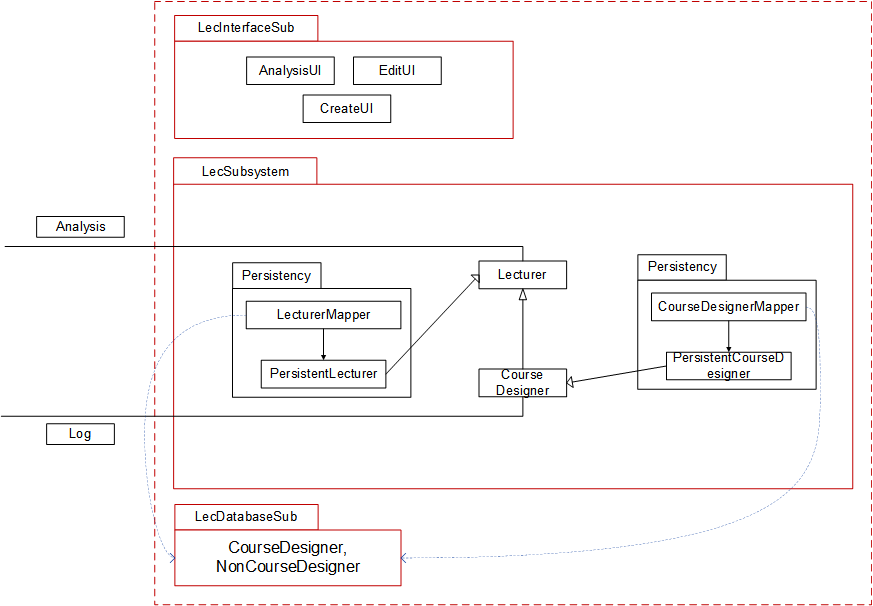
StudentID is the primary key in this table. There are also three attributes: StudentName, StudentYear and program. StudentYear indicates which grade the student is currently in. Program indicate the subject of the student.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GradeReport | | | | | |
| StudentID | Name | Semester | Methods | Total | CourseID |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

StudentID combined with CourseID is the primary key in this table. There are also four attributes: Name, Semester, Methods and Total. Name is the student’s name and semester are the date of the grade report. Following attribute method is an array to store the assessment methods that include in the grade report. And Total is for the final mark of the grade.

## LecSubsystem

This subsystem is design for lecturer to process the courses and its information.



### Description

This subsystem mainly includes the lecturer class and CourseDesigner class. The interface in this subsystem includes Analysis UI, EditUI and CreateUI. The database in this subsystem includes the table for lecturer and CourseDesigner.

### Database

|  |  |  |
| --- | --- | --- |
| Lecturer | | |
| Lecturer ID | LecturerName | Program |
|  |  |  |
|  |  |  |

Lecturer is the primary key in this table. There are also two attributes: LecturerName and program. Program indicate the subject of the student.

|  |  |  |
| --- | --- | --- |
| CourseDesigner | | |
| CourseDesigner ID | CourseDesignerName | Program |
|  |  |  |
|  |  |  |

CourseDesigner is the primary key in this table. There are also two attributes: CourseDesigner Name and program. Program indicate the subject of the student.

# Assessment

## Stability

* Since we divide the whole project system into several subsystems. For example, the login subsystem, the visualize subsystem and others.
* Each system is independent of each other and does not affect each other.
* If one of the subsystems is broken, other system could continue work. So, the whole system is stability.

## Reusability

* The components of the system could use in other place without ang changes because the system has already integrated some universal components like login component.
* Other software engineer could use some components directly without any changes.

## Scalability

* The system is very scalable and can be extended by adding additional links to windows in the user interface for additional functionality.
* For example, at lecture homepage, except visualize the whole result about one course, the system could display specific one student’s grade report for lectures.

# Alternative design (optional)

N/A

# More considerations

To understand what the user interface looks like and how does the user case work, it’s better to refer to Jasmine\_UI1.3\_20210414.docx.

# Appendix

N/A