YanXuan NUID:1563047

WeiChen NUID:1562214

Sizhe Gu NUID:2198472

**INFO 5100**

**Fall 2021**

**Assignment No. 3**

**Deliverable**

1. Report outlining your proposed solution.
2. Sequence diagrams showing how to navigate the university object model to deliver performance metrics needed for performance and feedback.
3. A class diagram showing the changes to the university model to support the new capabilities. This diagram must include the additional methods and attributes required to deliver the results.
4. System UI design

In this report, we designed a performance measurement solution to enable universities to measure the quality of the education they deliver to their students. It can show how an educational system in terms of faculty and courses contribute to the growth of their graduates over a 5-year period. We also figured out a way to track the jobs and promotions graduates get over time and assign rankings accordingly. In addition, we designed a way to track the connection of courses and their relevance to graduates' growth.

**Class Introduction:**

There are 13 concrete classes totally:

(The structure of the classes will be showed like:

ClassName class:

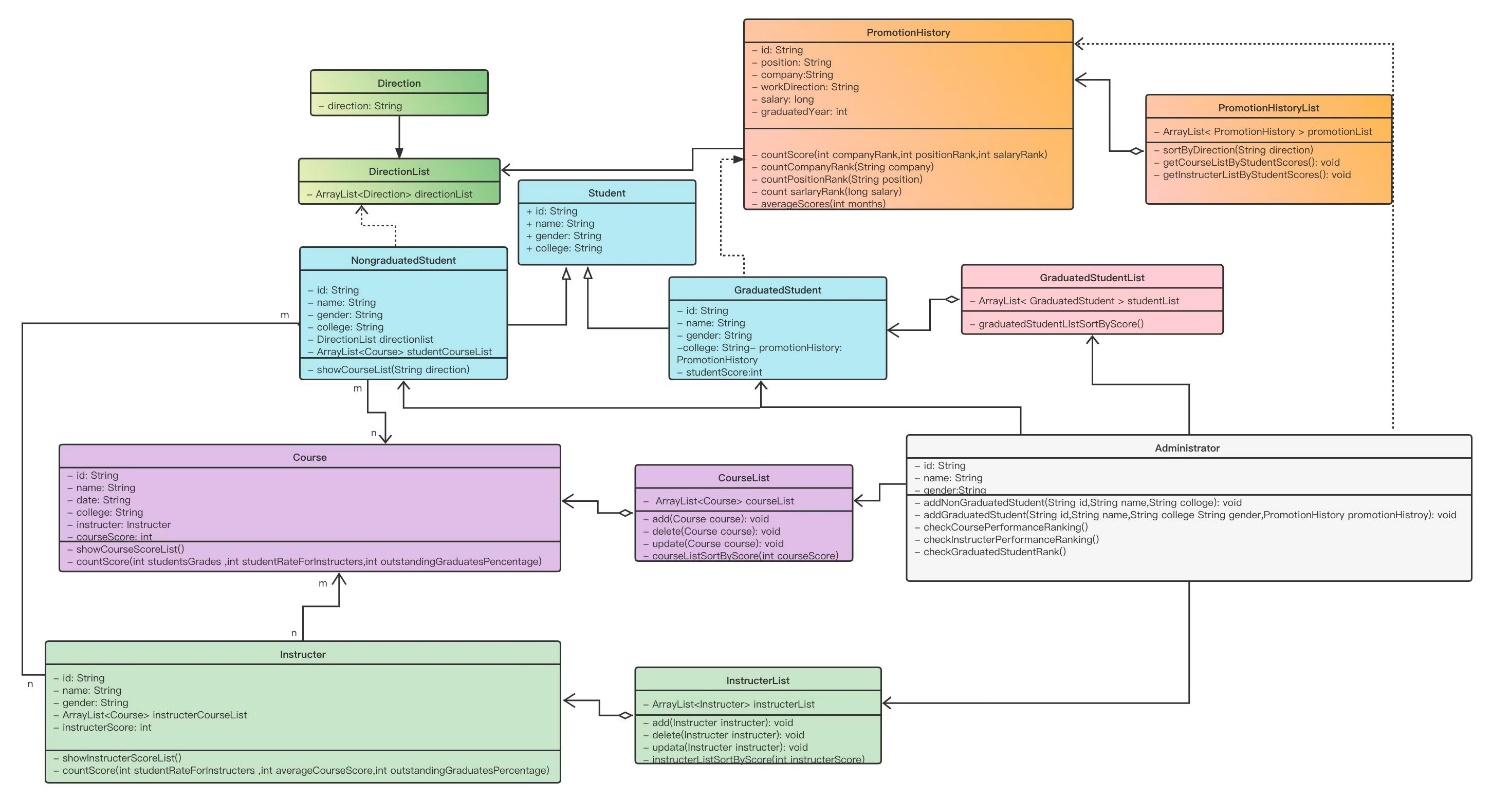
Property 1,

Property 2,

……)

1. Student class:
   1. String id;
   2. String name;
   3. String gender;
   4. String college
2. Nongraduated Student class extends Student class:
   1. String id;
   2. String name;
   3. String gender;
   4. String college;
   5. String direction;
   6. DirectionList directionList;
   7. ArrayList<Course> studentCourseList;
3. Graduated Student class extends Student class:
   1. String id;
   2. String name;
   3. String gender;
   4. String college;
   5. int graduateYear;
   6. PromotionHistory promotionHistory;
   7. int studentScore;
4. Graduated Student List class:
   1. ArrayList< GraduatedStudent > studentList;
5. Course Class:
   1. String id;
   2. String name;
   3. String date;
   4. String college;
   5. Instructor instructor;
   6. int courseScore;
6. Course List Class:
   1. ArrayList<Course> courseList;
7. Instructor Class:
   1. String id;
   2. String name;
   3. String gender;
   4. ArrayList<Course> instructorCourseList;
   5. int instructorScore;
8. Instructor List Class:
   1. ArrayList< Instructor> instructorList;
9. Administrator Class:
   1. String id;
   2. String name;
   3. String gender;
10. Promotion History Class:
    1. String id;
    2. String position;
    3. String company;
    4. String workDirection;
    5. long salary;
    6. int graduatedYears;
11. Promotion History List Class:
    1. ArrayList< PromotionHistory > promotionList;
12. Direction Class:
    1. String direction;
13. Direction List Class:
    1. ArrayList< Direction > directionList;

Their relationships are showed in the following class diagram:



**Design Ideas Introduction:**

In the class diagram showed before, our design ideas are as follows:

1. The administrators can be created, and they could login, create

non-graduated students and graduated students, as well as the courses and the teachers;

1. The graduated students can’t see their information and their

scores assigned by the rank system, so this part doesn’t need the UI design;

1. The non-graduated students can see their id, name and college

created by the administrators, and could choose their gender and their directions of future study. By choosing their future directions, there will show them a course list which is created by the algorithm’s design and they’d better choose;

1. We designed the performance of the university’s academic units

into two parts: the courses’ performance and the teachers’ performance.

For the courses’ performance, we use one of the courses’ properties, courseScore. It is calculated based on three conditions:



We choose the average of students’ rating’s sum for the course as the students’ grade for courses (full mark is 100), choose the average of students’ rating’s sum for teachers as the students’ rating for teachers (full mark is 100). The percentage of outstanding graduates is designed to firstly choose the graduated students who had the course, then choose the percent of students whose studentScore is more than 80 (includes 80, and the full mark of this part is 100). The full mark of the courseScore is 100.

For the teachers’ performance, we use one of the teachers’ property, teacherScore. It is also calculated based on three conditions:



We choose the average of students’ rating’s sum for the teacher as the students’ ratings for the teacher (full mark is 100). For courses' average grades, we choose the average of the sum of courseScore of all courses the teacher owes (full mark is 100). The percentage of outstanding graduates is designed to firstly choose the graduated students who had the teacher’s courses, then choose the percent of students whose studentScore is more than 80 (includes 80, and the full mark of this part is 100). The full mark of the teacherScore is 100.

Finally, we combine the two parts, the courses’ performance and the teachers’ performance as the university’s academic units. The system will show two lists, one is the courses and one is the teachers, ordered by their scores from high to low.

1. We design the graduated students’ studentScoret based on three

conditions:



The companies’ score is ranked by the code, some big companies’ score is marked 100, and other smaller companies’ score is marked 80 (full mark is 100). The positions’ score is also ranked by the code, boss position is marked 100, manager position is marked 90, and staff position is marked 80 (full mark is 100).



The salary’s score is also ranked by the code, as showed below (full mark is 100).

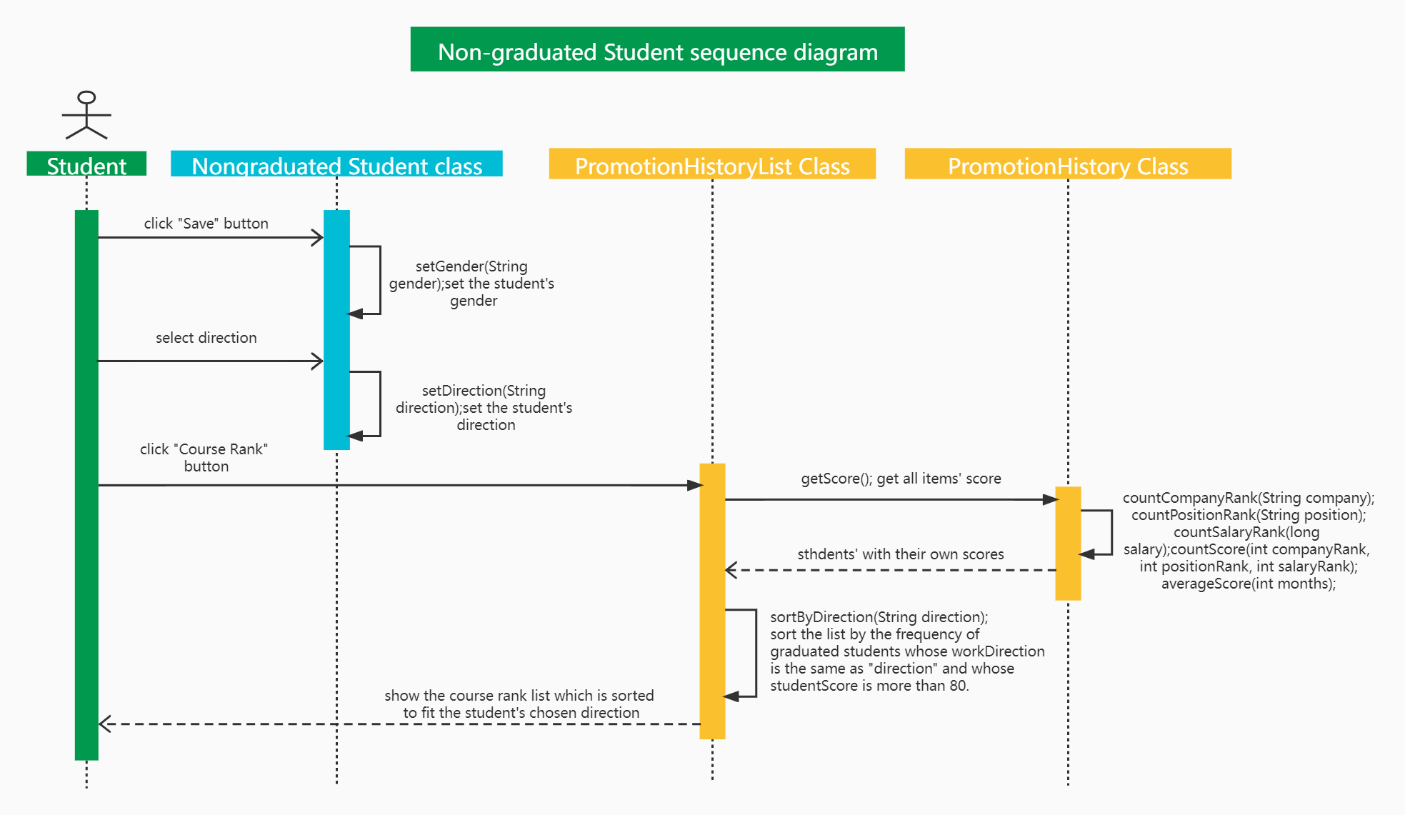


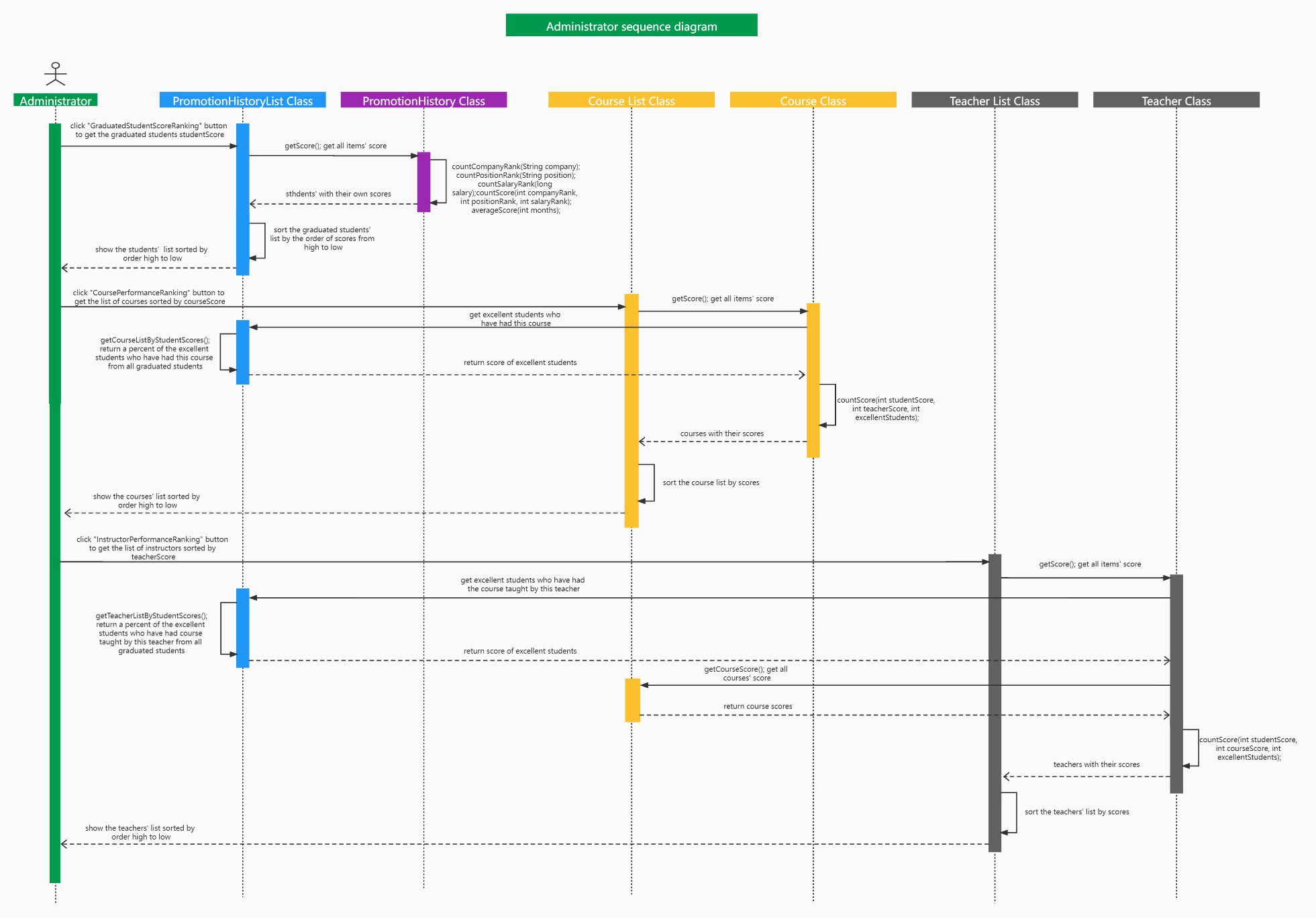
The full mark of the studentScore is 100. We choose the average score of 12 months as a one-year’s studentScore. The final studentScore then is the average studentScore of the student’s graduated years.

1. How to design the own ranking system for students to decide where they want to go for their studies?

We firstly choose the Promotion History List Class, then, according to the non-graduated student’s future study direction, we select in the promotionList for graduated students whose workDirection is the same as the non-graduated student’s chose and whose studentScore is more than 80. Then we will order the courses which the graduated students’ have had by frequency from high to low. Finally, we will show this selected course list to the non-graduated student.

The Sequence diagrams according the previous design ideas are the following:





**UI Design**

For the UI design, both administers and the non-graduated student can log in the system through name, ID and password. System separate the admin and student with name and id.

图形用户界面

描述已自动生成

For the non-graduated student, system shows the detail information and allow them to change gender and development directions.

图形用户界面, 应用程序

描述已自动生成

Students can see the course rank according to the development direction. Rank, name, course id, date, college and all instructors for the course are sorted in the table.

表格

描述已自动生成

For admin, system allows them to implement five functions.

图形用户界面, 应用程序

描述已自动生成

Course performance ranking is shown as follow. Course name, instructors and course score are listed in the table:

表格

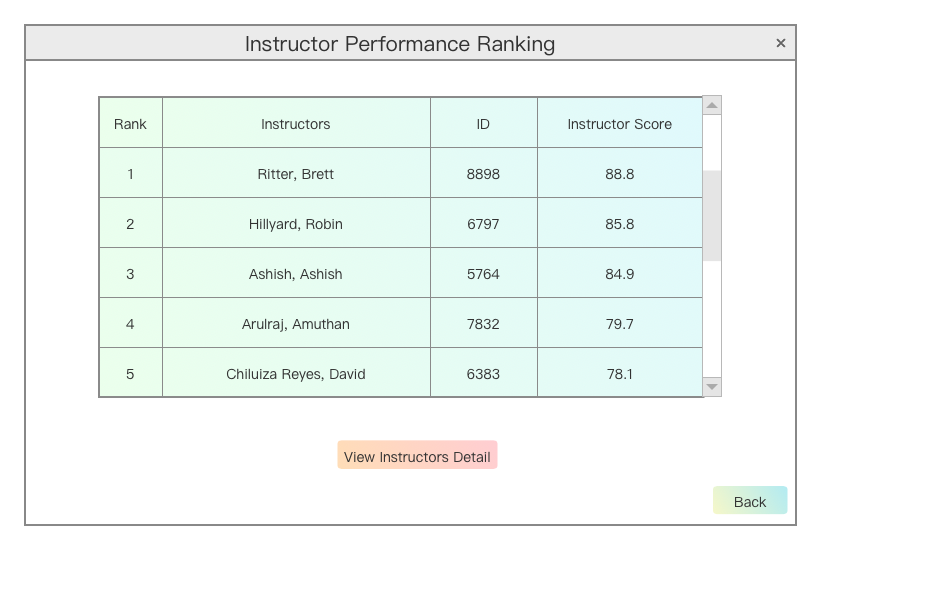
描述已自动生成

The course detail demonstrates the calculation of the course score through histogram, it shows the sum of the scores after scaling. Detail of each class and the total course score are also listed as follow:

图形用户界面, 图表

描述已自动生成

The instructors performance ranking is shown as follow. Instructors name, id, and instructors score are listed in the table:



The instructors detail demonstrates the calculation of the instructors score through histogram, it shows the sum of the scores after scaling. Detail of instructor such as courses and the total instructor score are also listed as follow:

图形用户界面, 图表, PowerPoint

描述已自动生成

Graduated performance ranking is shown as follow. Name, ID, position, college and graduated score are listed in the table:

表格

描述已自动生成

The instructors detail demonstrates the calculation of the instructors score through histogram, it shows the sum of the scores after scaling. Detail of graduated student such as positions and the graduated student score are also listed as follow:

图形用户界面, 应用程序

描述已自动生成

Admin can create graduated student in the system. The UI design is shown as follow:

图形用户界面

描述已自动生成

Admin can also create non-graduated student in the system. The UI design is shown as follow:

图形用户界面

描述已自动生成