

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;

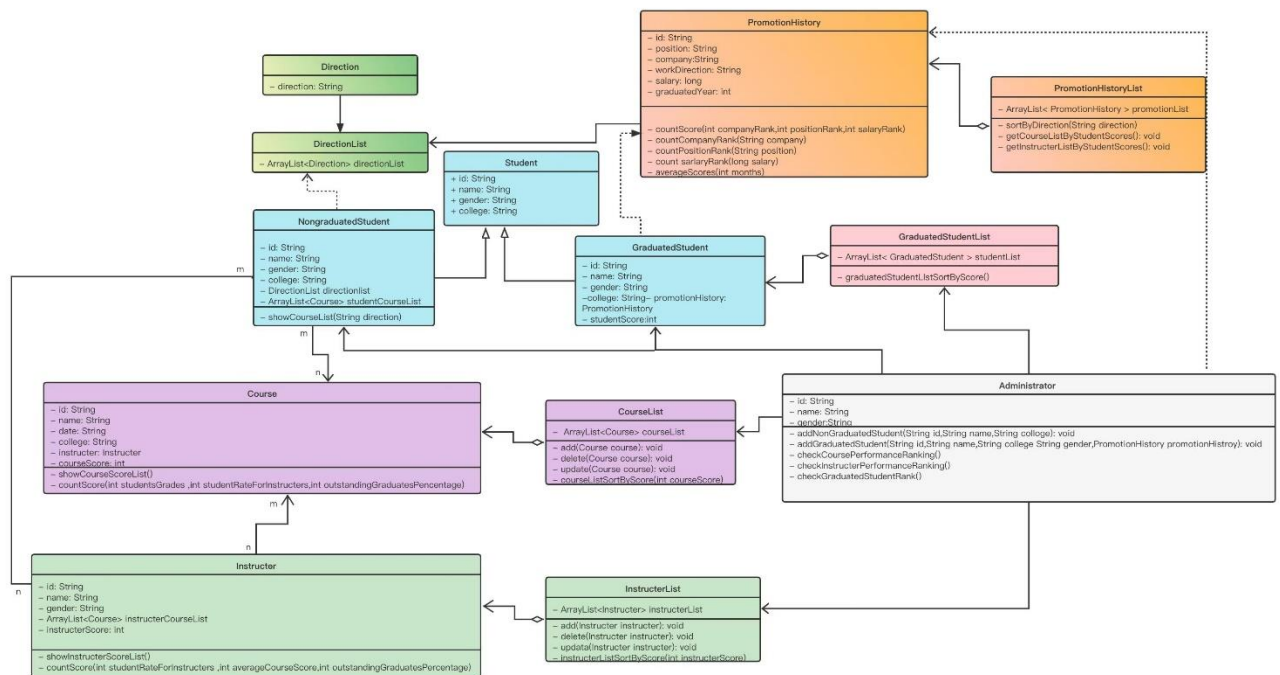
12.Direction Class:

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;
2. 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;
3. 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;
4. 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:

Students' grades for courses	30%
Students' ratings of teachers	30%
Percentage of outstanding graduates	40%

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:

Students' ratings of teachers	30%
Courses' average grades	30%
Percentage of outstanding graduates	40%

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.

5. We design the graduated students' studentScore based on three conditions:

company	10%
position	10%
salary	80%

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).

position	score
boss	100
manager	90
staff	80

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

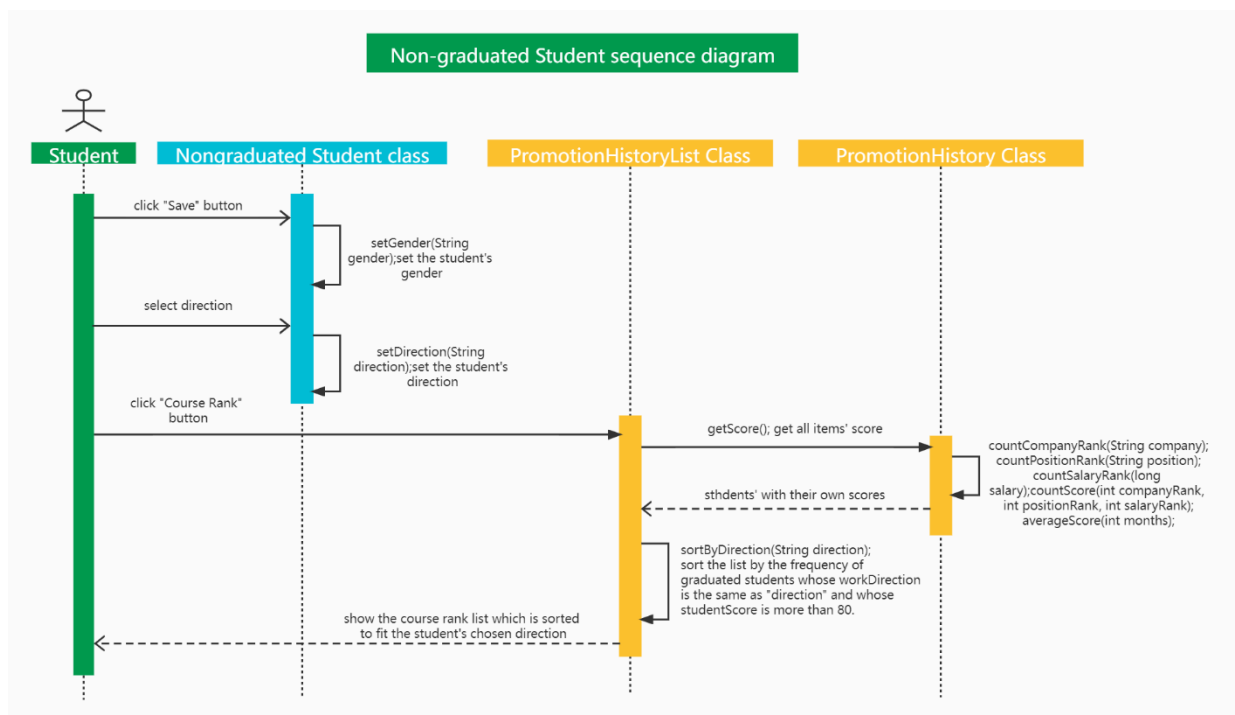
Salary(\$ / month)	score
>4000	100
>3000 && <=4000	90
>= 2000 && <= 3000	80
<2000	50

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.

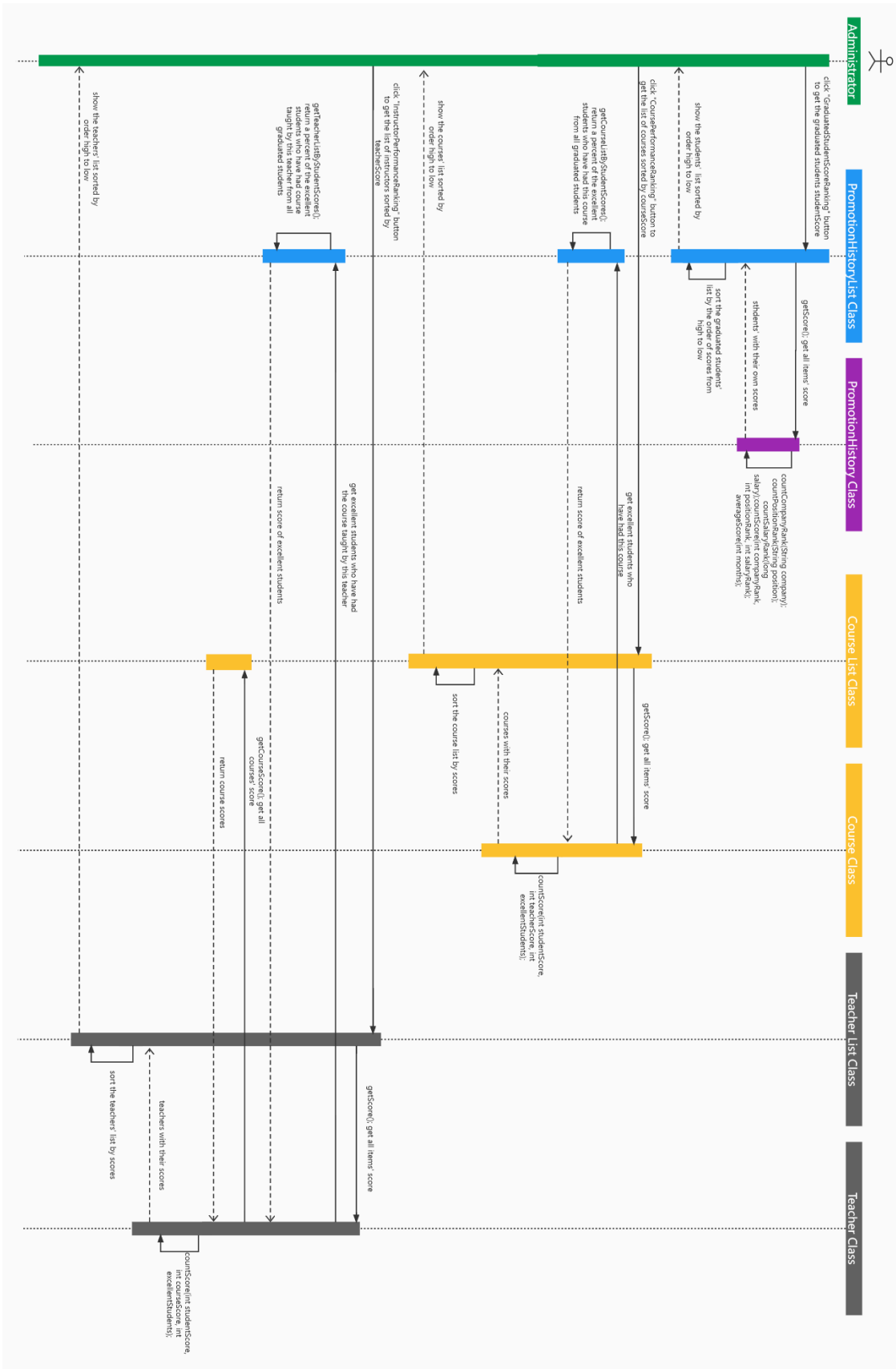
6. How to design the own ranking system for students to decide where they want to go for their studies?

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:

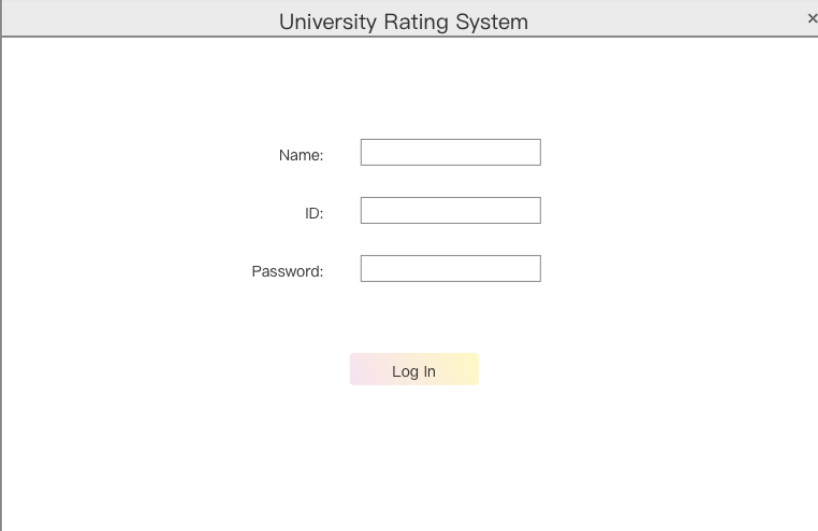


Administrator sequence diagram



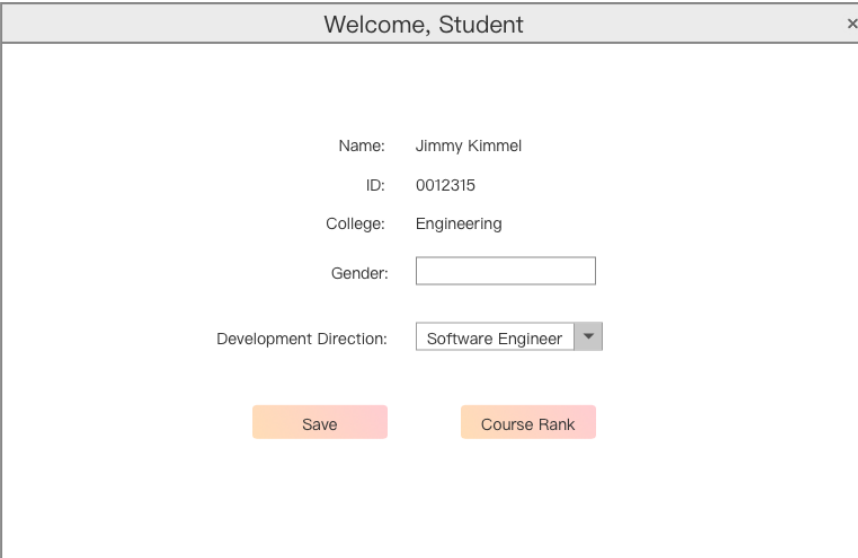
UI Design

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.



A login window titled "University Rating System" with a close button (x) in the top right corner. The window contains three input fields for "Name:", "ID:", and "Password:". Below these fields is a yellow "Log In" button.

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



A profile window titled "Welcome, Student" with a close button (x) in the top right corner. The window displays the following information: Name: Jimmy Kimmel, ID: 0012315, College: Engineering. Below this is a "Gender:" label followed by an empty input field. Further down is a "Development Direction:" label followed by a dropdown menu currently showing "Software Engineer". At the bottom are two buttons: "Save" and "Course Rank".

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.

Course Rank for Software Engineer					
Rank	Name	ID	Date	Instructors	College
1	Application Engineering and Development	5100	21 Fall Wed 03:15 PM – 07:00 PM	Arulraj, Amuthan Ritter, Brett	Engineering
2	Web Design and User Experience Engineering	6150	21 Fall Fri 11:15 AM – 01:55 PM	Hillyard, Robin Ashish, Ashish	Computer Science
3	Program Structure and Algorithms	6205	21 Spr Sun 10:15 AM – 01:15 PM	Ritter, Brett	Engineering
4	Agile Software Development	7245	21 Fall Sun 07:40 AM – 03:00 PM	Ashish, Ashish	Computer Science
5	Engineering of Big-Data Systems	7250	21 Spr Mon 10:15 AM – 01:15 PM	Hillyard, Robin	Engineering

Back

For admin, system allows them to implement five functions.

Welcome, Admin
<div>Course Performance Ranking</div> <div>Instructor Performance Ranking</div> <div>Graduated Student Score Ranking</div> <div>Add Graduated Student</div> <div>Add Non-Graduated Student</div>

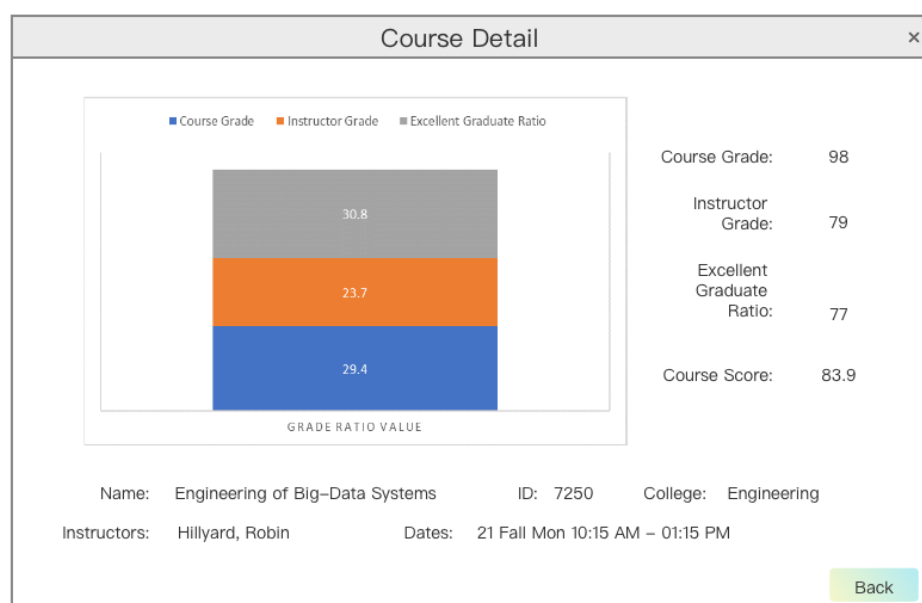
Course performance ranking is shown as follow. Course name, instructors and course score are listed in the table:

Course Performance Ranking			
Rank	Name	Instructors	Course Score
1	Engineering of Big-Data Systems	Hillyard, Robin	83.9
2	Web Design and User Experience Engineering	Hillyard, Robin Ashish, Ashish	81.5
3	Program Structure and Algorithms	Ritter, Brett	79.8
4	Agile Software Development	Ashish, Ashish	76.5
5	Application Engineering and Development	Arulraj, Amuthan Ritter, Brett	75.2

[View Course Detail](#)

[Back](#)

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:



The instructors performance ranking is shown as follow.

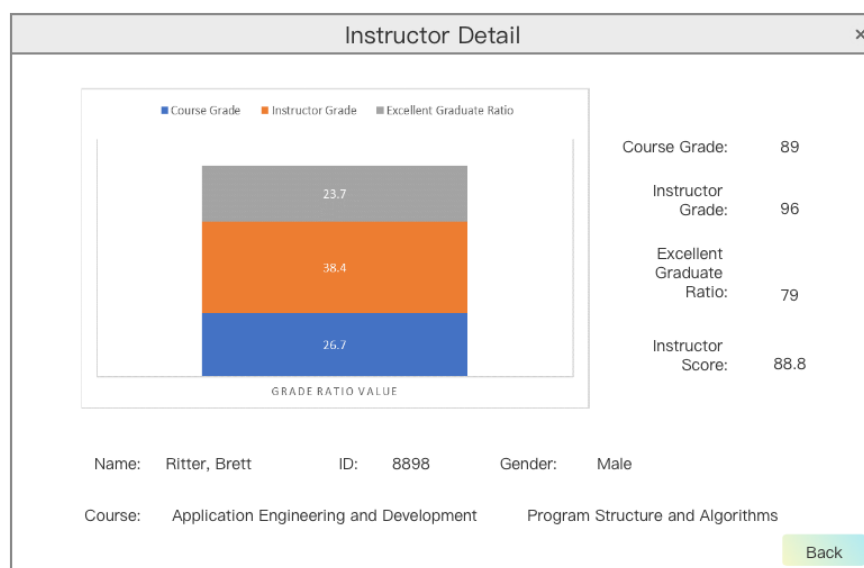
Instructors name, id, and instructors score are listed in the table:

Instructor Performance Ranking			
Rank	Instructors	ID	Instructor Score
1	Ritter, Brett	8898	88.8
2	Hillyard, Robin	6797	85.8
3	Ashish, Ashish	5764	84.9
4	Arulraj, Amuthan	7832	79.7
5	Chiluza Reyes, David	6383	78.1

[View Instructors Detail](#)

[Back](#)

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:



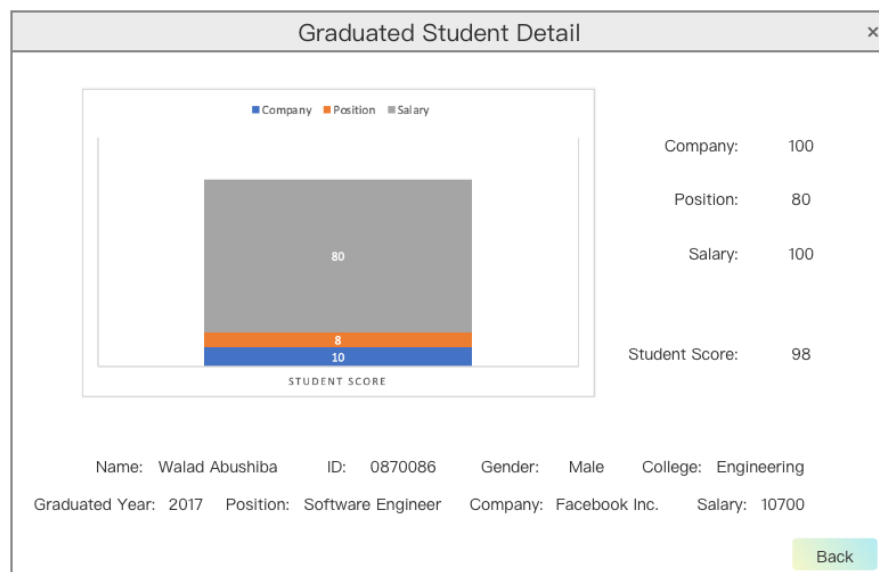
Graduated performance ranking is shown as follow. Name, ID, position, college and graduated score are listed in the table:

Rank	Name	ID	Position	College	Score
1	Walad, Abushiba	0870086	Software Engineer	Engineering	98
2	Black, Zhou	0589375	Data Scientist	Computer Science	97
3	Frank, Henze	1977375	Consultant	Engineering	95
4	Gordon, Callan	2984783	Architecture	Engineering	94
5	Nico, Decourt	0482745	Marketing	Business	92

View Graduated Detail

Back

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:



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

Add Graduated Student

Name:

ID:

College:

Gender:

Graduated Year:

Company:

Salary:

Position:

Software Engineer

Save

Back

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

Add Non-Graduated Student

Name:

ID:

College:

Save

Back