

ASSIGNMENT 3

UNIVERSITY MODEL

Team Members:

DIKSHA BHATIA-002175782

SIMRAN PATIL- 001563069

VIRENDRA SINGH RATHORE- 001584432

Problem Statement:

To utilize computer programming plan and programming strategies to work on the nature of instruction around the world and to foster an exhibition estimating metric to evaluate and work on the understudy's proficient development through the nature of schooling and feedback.

Objective:

To provide career-oriented students with quality education designed to meet the current industry needs and trends by improving real-time with the university's alumni's feedback by calculating the impact the courses have on their career. We have designed a system that analyzes statistical data obtained from student's (alumni) feedback and tracks students' growth in a professional setting and analyzes the courses that helped students in their careers.

Proposed Solution:

As indicated by our University Model, Student pursues a specific course which can be dictated by the faculty that is teaching that course.

Understudy is assessed by the teachers based on tasks, lab work and activities.

Faculty gives feedback and allots grades for the same.

In view of the course picked by the student, we can track the sort of job a student lands, and all the while track his development over a time of 5 years.

We have planned our own unique grading strategy to apportion grades which is a mix of grades doled out by the faculty and job performance evaluation.

Faculty Grading Strategy:

GPA	Grading
>3.8	A+
3.5-3.7	A
3.2-3.4	B+
2.8-3.1	B
<2	F

Job Performance Evaluation:

Securing a course relevant Job: 5 Points

Increment 1 Point for Appraisals over 5 years.

Our Unique Grading System:

Securing a relevant job with a GPA greater than 3	10
Securing a relevant job with a GPA greater than 2	8
Not securing a relevant job with a GPA greater than 2	6
Not securing a relevant job with a GPA less than 2	4

Overall Ranking System (Defined)

Overall Rank= College Grading System + Job Evaluation

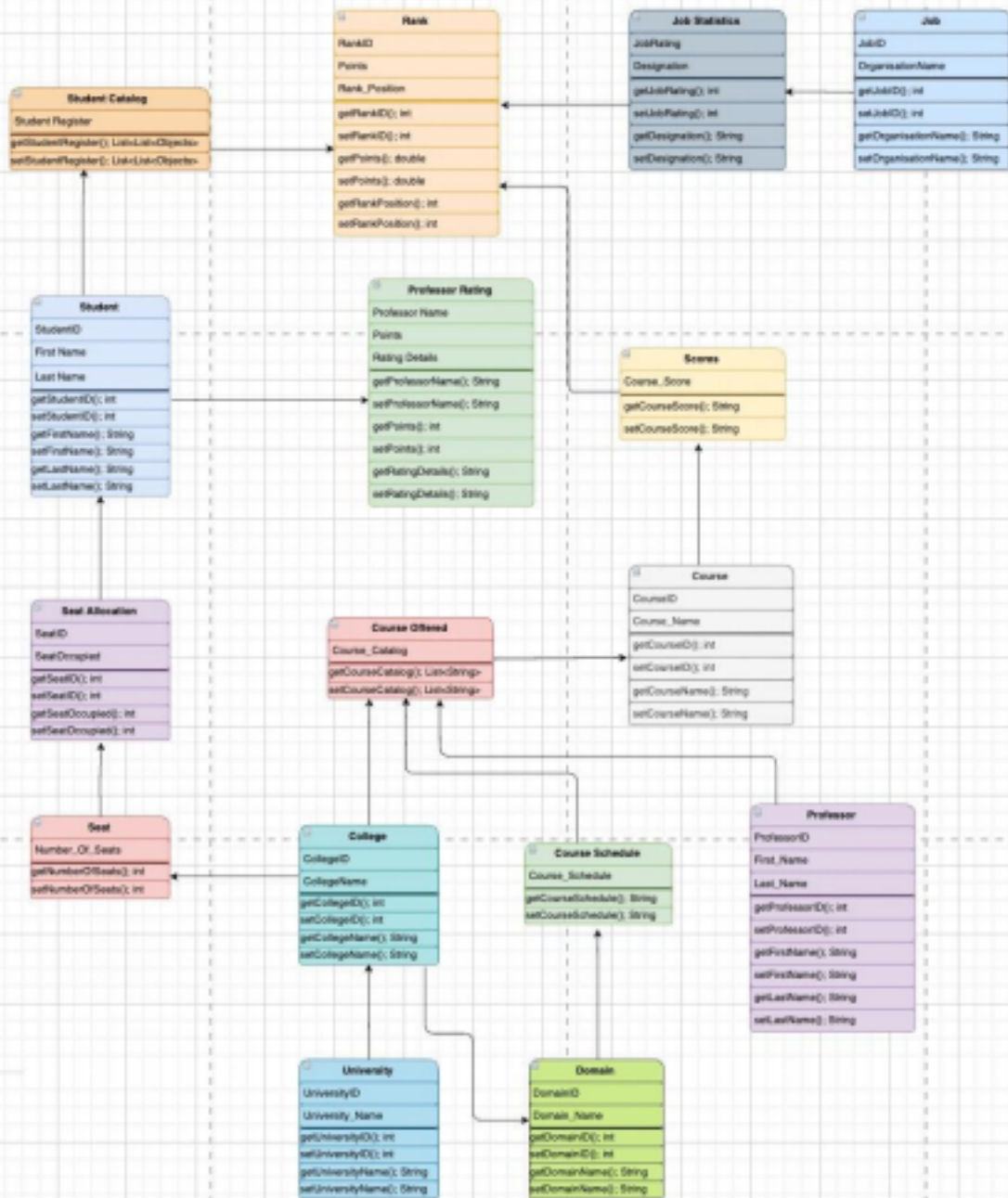
Suggestions to go for higher studies

The current system is highly biased towards recommending students to pursue higher studies in the field of research.

Considering today's fast paced IT industry, students who want to pursue better jobs in the market, for them studying in the field of research reduces the relevance of what they are studying. In order to cater to such students, the courses should be designed to make engineers ready for the IT market. In our university model, we would suggest those students who are currently stagnant in their jobs to go for higher studies in their field of interest.

University Model - Class Diagram:

University Model Class Diagram



Student Catalog:

A register claimed by the University authority.

A track of all student records which comprises student data, courses decided on, courses finished up, late fee payment penalties and so forth are contained in the Student Catalog.

Methods:

setStudentRegister() :List<List<Objects>>

getStudentRegister() :List<List<Objects>>

Student:

An individual admitted in the university program offered by a college.

Has a total track history about the courses undertaken for the degree.

Has a Combined Grade.

Methods:

getStudentID() :int

setStudentID() :int

getFirstName() :String

setFirstName() :String

getLastName() :String

setLastName() :String

Seat Allocation:

A particular seat in a course is allocated to a distinct student enrolled in the program and is represented by Seat Allocation.

Methods:

getSeatID() :int

setSeatID() :int

getSeatOccupied() :int

setSeatOccupied() :int

Seat:

All the available as well as vacant seats are depicted here.

Methods:

getNumberOfSeats() :int

setNumberOfSeats() :int

Domain:

A section of college where bifurcation of students is done based on their specialization. Domain will have specific courses.

Methods:

getDomainID() :int

setDomainID() :int

getDomainName() :String

setDomainName() :String

Professor:

An individual who is responsible for holding lectures and assessing students in the

University. Assigned with the responsibility of specific courses.

Methods:

```
getProfessorID() :int  
setProfessorID() :int  
getFirstName() :String  
setFirstName() :String  
getLastName() :String  
setLastName() :String
```

Job:

A paid position of regular employment for students.

A student can enroll into only one job at a time.

Methods:

```
getJobID() :int  
setJobID() :int  
getOrganisationName() :String  
setOrganisationName() :String
```

Job Statistics:

The rating given by the employer to the student dependent on his/her performance is portrayed here. Methods:

```
getJobRating() :int  
setJobRating() :int  
getDesignation() :String  
setDesignation() :String
```

College:

It oversees all students enrolled and all the different departments existing in the college.

Responsible for managing information about Professors and students.

Methods:

```
getCollegeID() :int  
setCollegeID() :int  
getCollegeName() :String  
setCollegeName() :String
```

University:

It oversees the academic specifications about various colleges.

Responsible for maintaining a catalog of all programs and courses offered by each college.

Takes care of business-related matters such as course financing, digital marketing etcetera. Administers the assets of every college.

Annual revenue generated from all colleges is looked after by the University.

Methods:

```
getUniversityID() :int  
setUniversityID() :int  
getUniversityName() :String
```

setUniversityName() :String

Course:

A series of lessons in which a student is enrolled for completion of his/her degree.

Each course has a professor assigned who is responsible for grading the course.

Methods:

getCourseID() :int

setCourseID() :int

getCourseName() :String

setCourseName() :String

Course Schedule:

It details the times of day and days of the week that each course will be offered. Looks for the course timings to keep away from time conflicts between two courses presented for a program.

Methods:

getCourseSchedule() :String

setCourseSchedule() :String

Course Offered:

Deals with a catalog of the courses presented by every department under a college for various programs. Administers interchange course offered list with different divisions under one college.

Methods:

getCourseCatalog() :List<String>

setCourseCatalog() :List<String>

Scores:

Portrays the rating assigned to every student depending on his/her performance in a specific course. An aggregate of all the tests organized such as assignments, lab work, midterms, projects, finals is calculated to provide the required score.

Methods:

getCourseScore() :String

setCourseScore() :String

Rank:

A cumulative aggregation of student list, scores obtained in the respective enrolled courses and job evaluation. A total point score for each student in the University is calculated and a rank is assigned to every student. A Student Ranking Directory for performance analysis and overall development of all colleges for different specializations is managed for cumulative years.

Methods:

getRankID() :int

setRankID() :int

getPoints() :Double

```
setPoints() :Double  
getRankPosition() :int  
setRankPosition() :int  
calculatePointsByStudentID(int studentID) :Double
```

The total point for a student is calculated based on the Job performance and the grades scored in their respective specializations in this method.

Professor Rating System:

A rating structure for the faculty by students based on the performance and effectiveness of the teaching strategies adopted by the Professor during the length of the course.

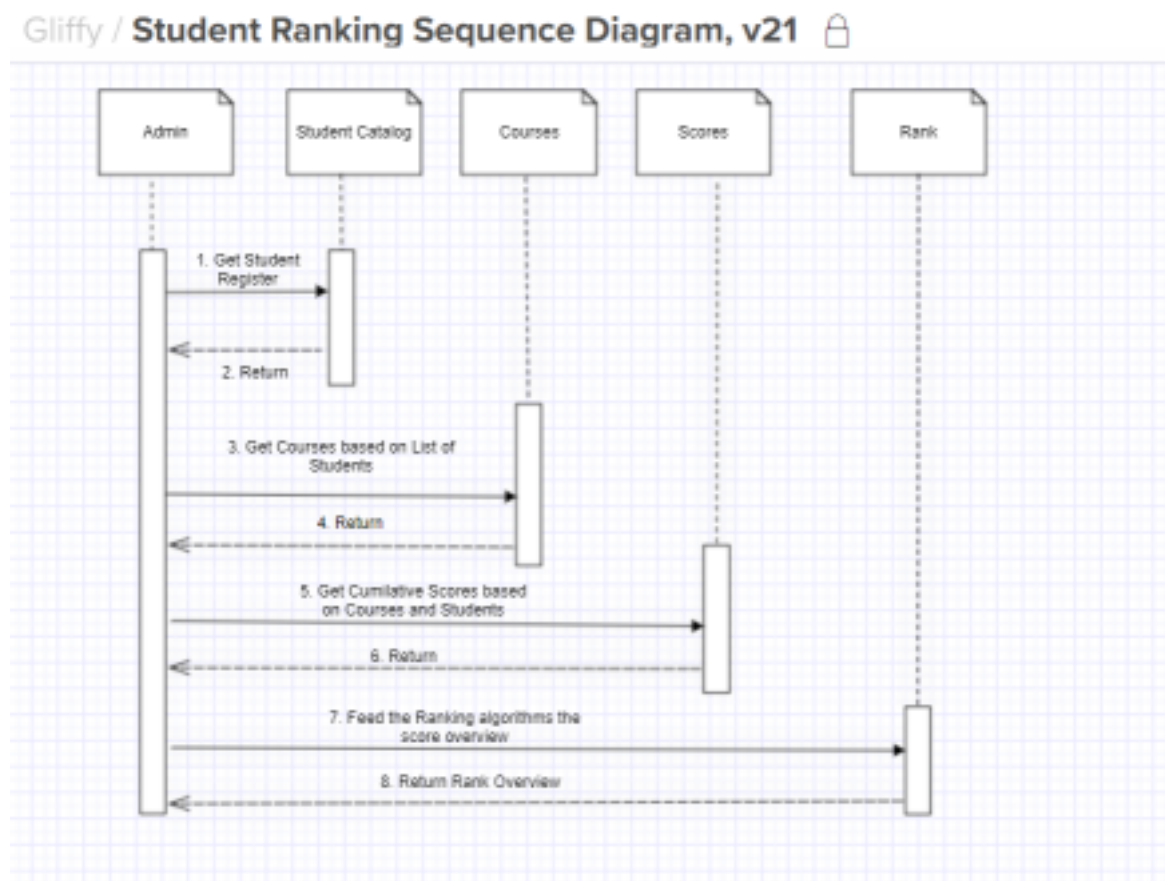
A general feedback score for every teacher is determined and contrasted with the earlier scholarly year. Performance statistics graphs for each professor is maintained for each term.

Methods:

```
getProfessorName() :String  
setProfessorName() :String  
getPoints() :int  
setPoints() :int  
getRatingDetails() :String  
setRatingDetails() :String
```

Sequence diagrams showing how to navigate the university object model to deliver performance metrics needed for performance and rating:

1. CGPA BASED STUDENT RANKING



CGPA Based Student Ranking System depicts how good students are faring on different courses provided by the university.

It gets information based on all the current students.

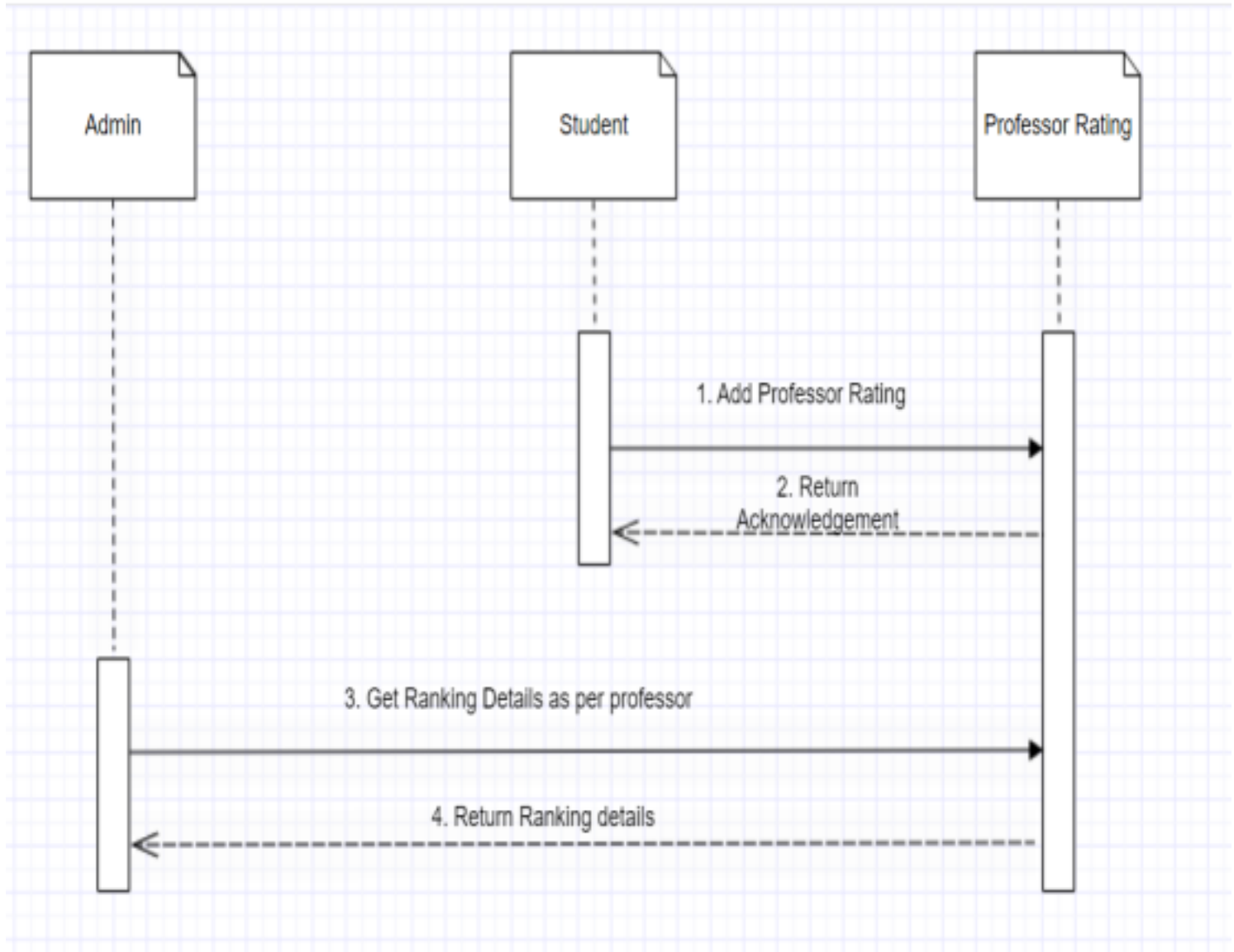
We will show the list of courses and the cumulative CGPA scored, and our a ranking algorithm would rank the students based on them.

2. FACULTY FEEDBACK

Students have been requested to rate the professors.

We depict the data in the form of a table showing how well the professor is rated.

Faculty Feedback, v20

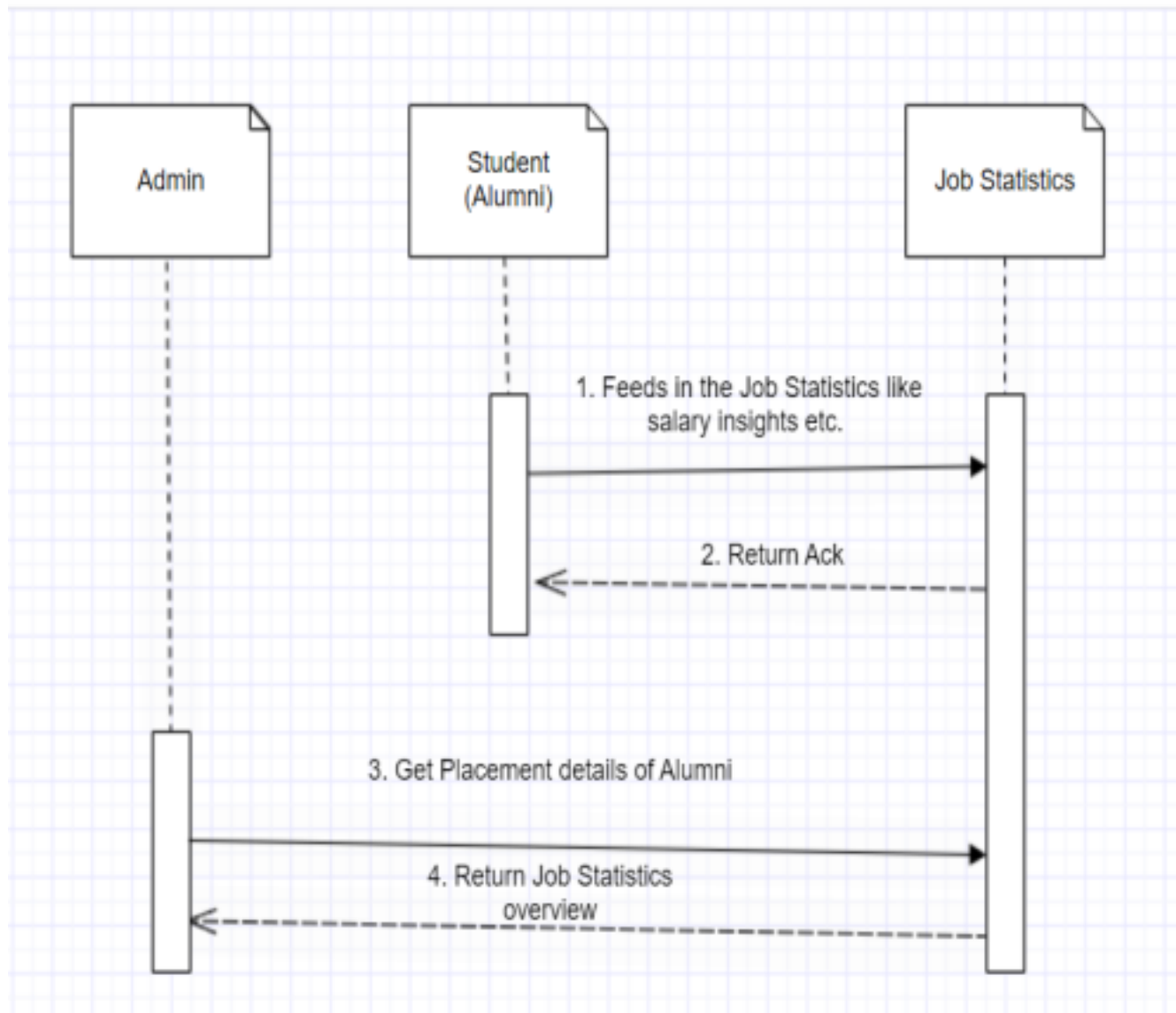


3. PLACEMENT STATISTICS

Students (alumni) are requested to provide salary insights, current job positions, track of promotions and the technical domain (Web Design, Software Engineer, Database architecture, etc.)

We have stored this information in Job Statistics class.

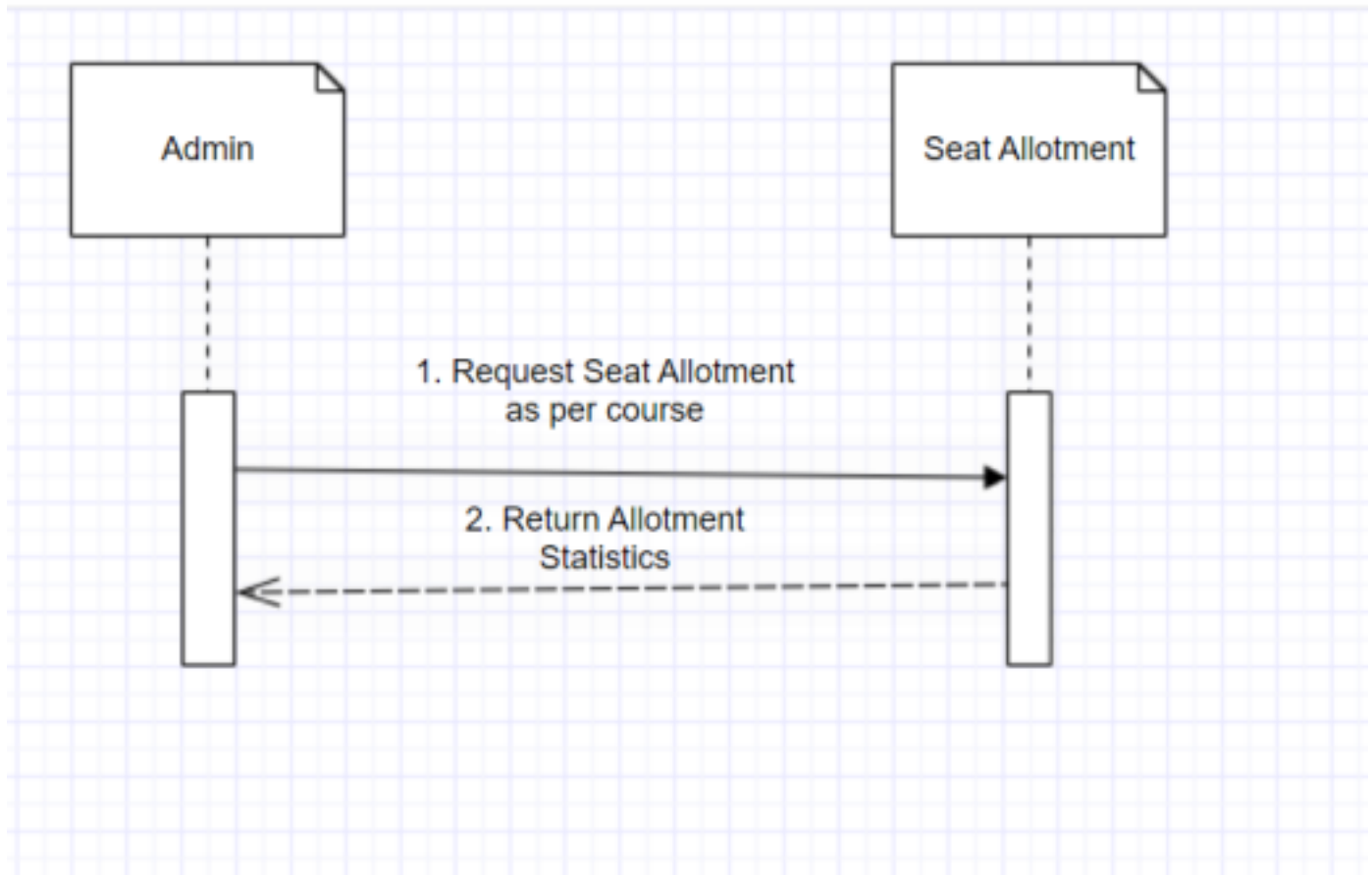
Placement Overview Ranking System, v17



4. ACCOUNTING INFORMATION

Based on student enrollment, we show the number of occupied seats versus the number of unoccupied seats. We refer to the Seat Allotment class for the same.

Accounting Seq Diagram, v5

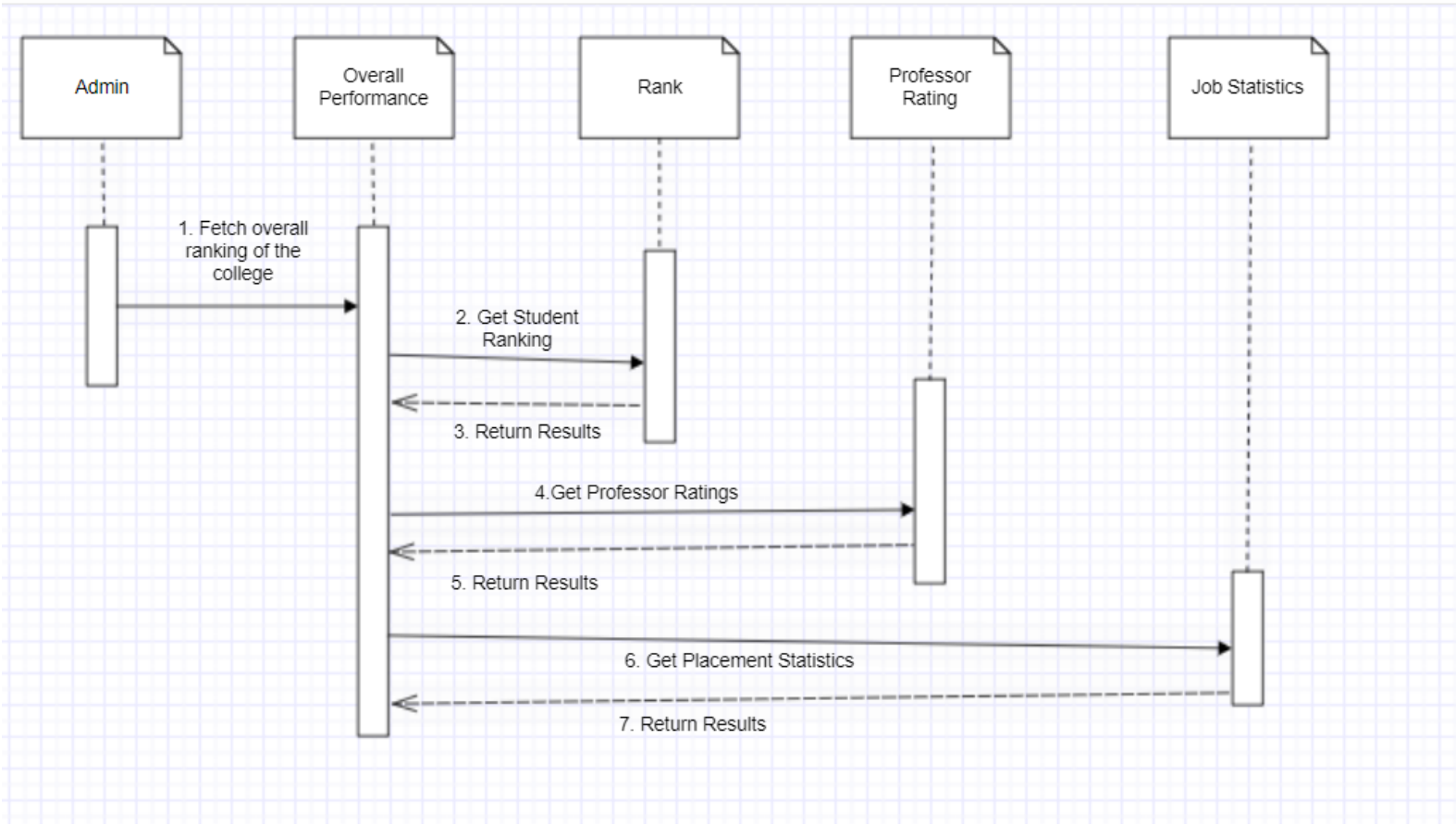


OVERALL RANKING SYSTEM

In the Overall Ranking System,
We take a summation of overviews across all domains.

- 1. We check how the students have performed in the courses to determine their understanding in multiple courses.
- 2. We gather feedback in the Professor Rating model. The evaluation would lead to how likely someone will recommend the professor’s class to his/her peers.
- 3. In the Accounting strategy, we also check seat allotment. This is also a signifier of how likely people are to take the professor’s class. If the number of unallocated students is high, then it indicates that students are less likely to take his class.
- 4. In the placement ranking, we consider how the college’s alumni are faring in the industry.
 - a. How likely are they to get jobs/ promotions?
 - b. Is the course relevant to the actual job domain they are working on?
 - c. Is the salary range good enough for the profile they are working for?

y / Overall Ranking, v11



Professor Grading Strategy:

We use the 1-5 star rating strategy to rate. Students would give a star rating on how likely they recommend the course taken by a professor to other students.

Rating	Meaning
5	Definitely

3-4	Most Likely
<=2	Not Recommended

Job Statistics Evaluation:

Achievement	Increment In The Points For Total Ranking
Getting a job relevant to course	5
Change in designation after 5 years	1
Working in different domain	-5

College Grading System (Defined):

Criteria	Points
Grabbed a relevant job with GPA > 3	10
Grabbed a relevant job with GPA > 2	7
Not landing a relevant job with GPA > 2	5
Not landing a relevant job with GPA < 2	3

Overall Ranking System (Defined)

Overall Rank= College Grading system + Job Performance

Example:

If Diksha gets a 3.9 GPA by her faculty and secures a job based on the courses that she took in her semesters and has received a promotion already in her 2nd year, then her Overall Rank would be as follows:

Faculty Grade: 3.9 (A+)

Job Rank: 8

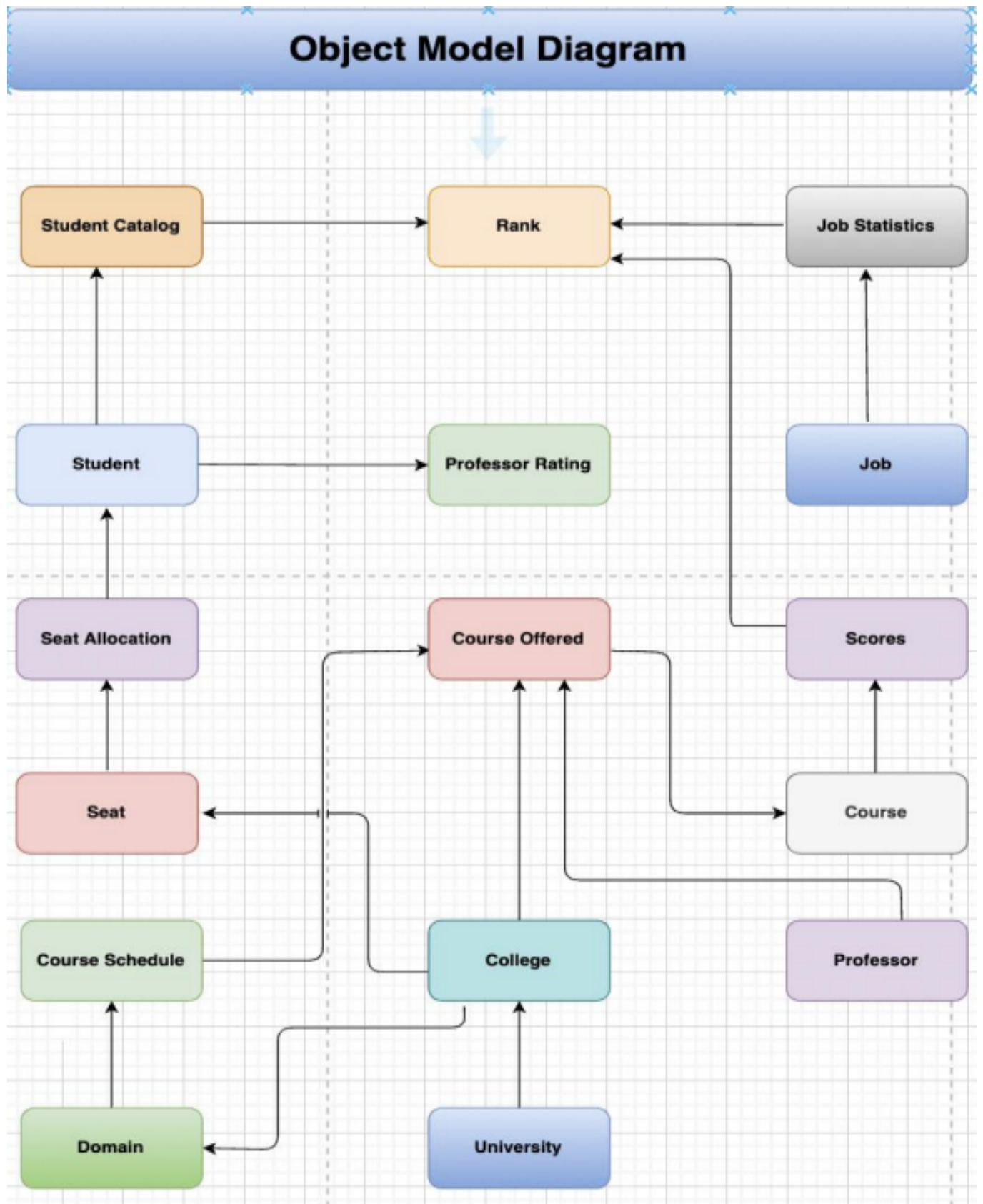
College Grading: 10

Overall Rank: 10 + 8 = 18

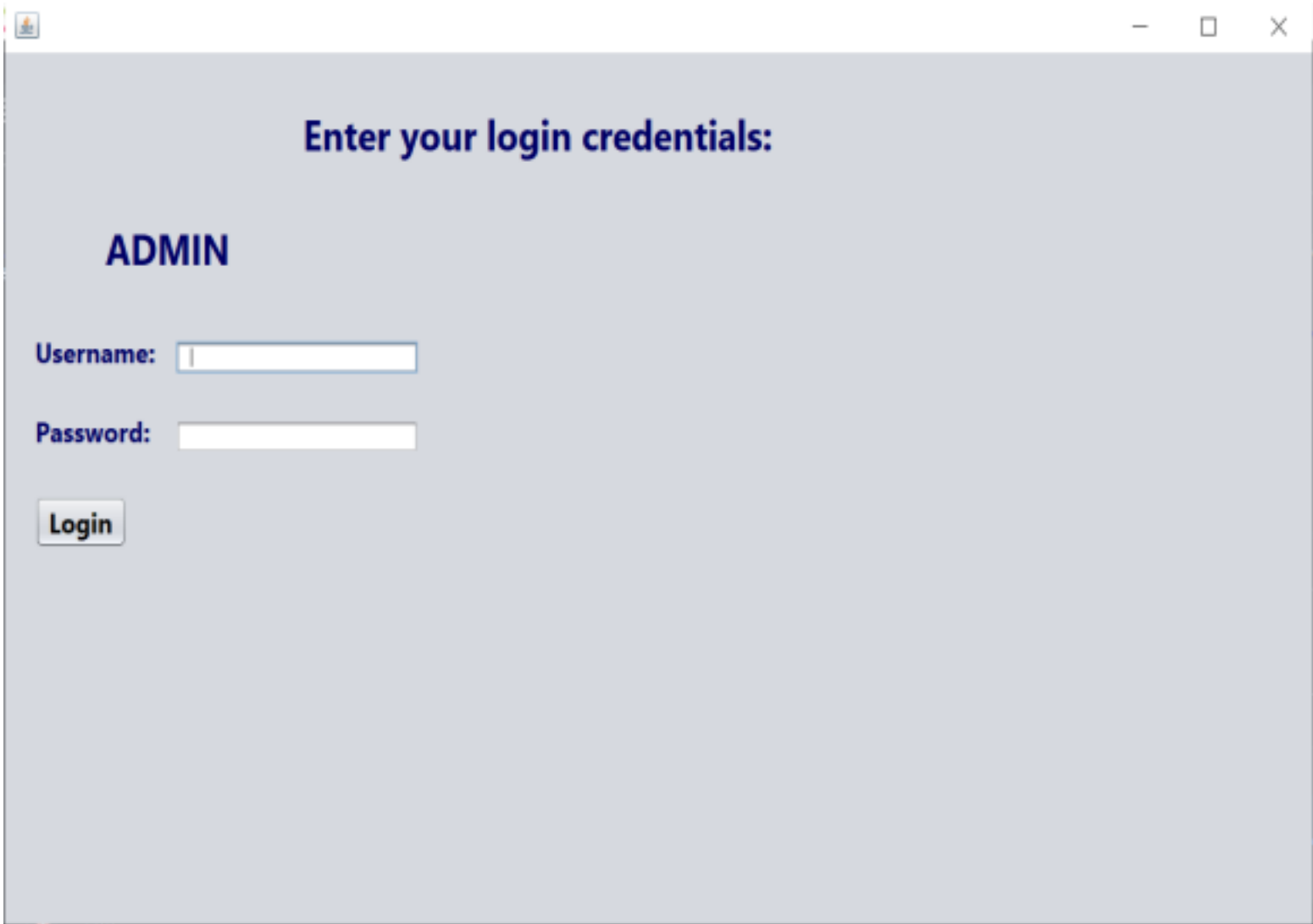
Suggestion for higher studies

Through the Job Statistics model, we would come to a conclusion that candidates are not moving up the career progression and feeling stagnant in their career. Considering the candidate has a good CGPA record during the undergrad, would be recommended to go for higher studies in the field of their interest.

An object model showing the changes to the university model to support the new capabilities:



Dashboard Login Screen



A screenshot of a web browser window displaying a login screen. The window has a standard title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The background of the page is a solid light gray. The text "Enter your login credentials:" is centered at the top in a bold, dark blue font. Below this, the word "ADMIN" is displayed in a bold, dark blue font. The login form consists of two rows: "Username:" followed by a white text input field with a blue border, and "Password:" followed by a white password input field with a blue border. Below the password field is a "Login" button with a gray background and a thin blue border.

Enter your login credentials:

ADMIN

Username:

Password:

Login

Performance Metrics Screen - Student Ranking

Performance Metrics for COE:

Current Students

Placement

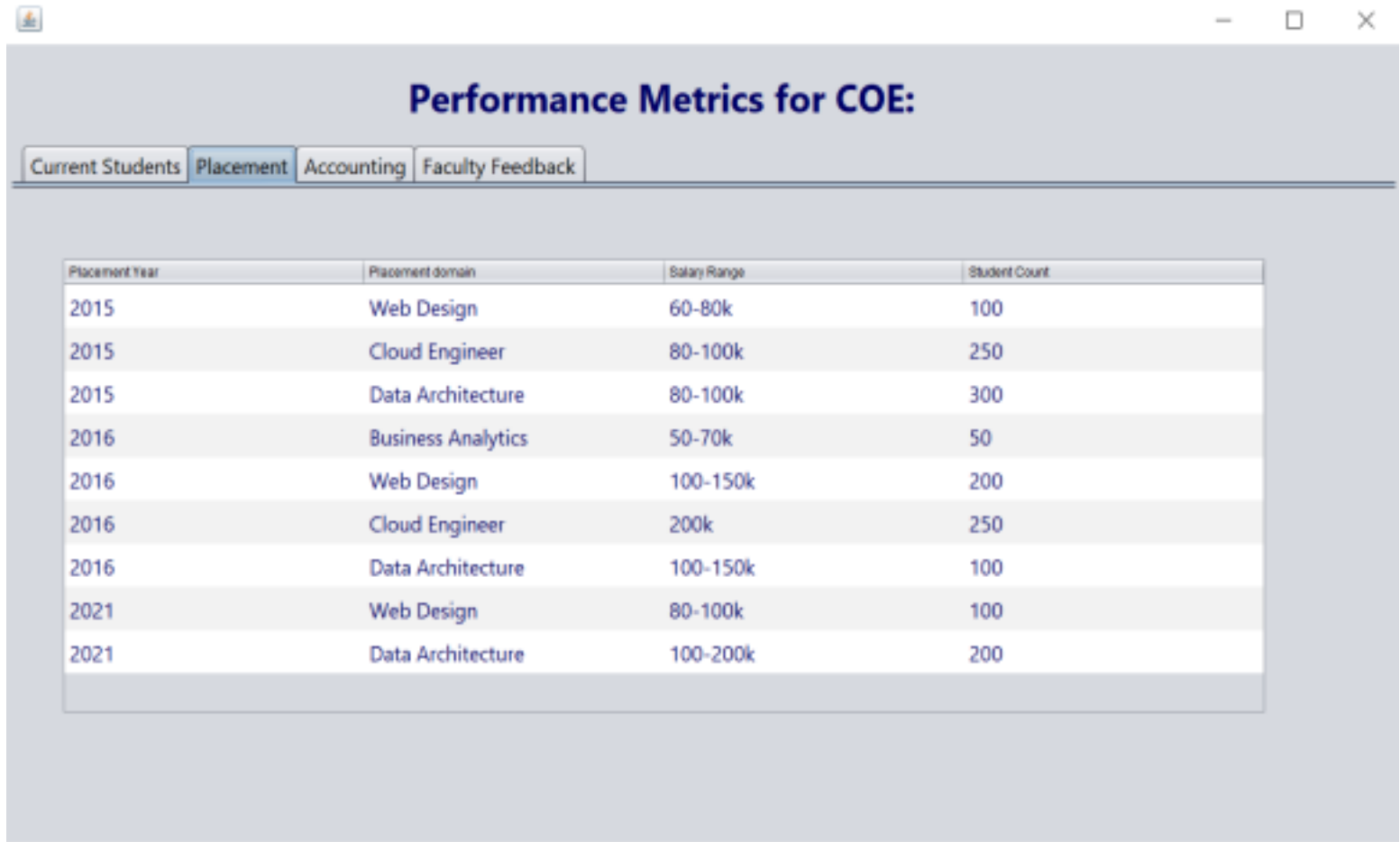
Accounting

Faculty Feedback

View Current Student Directory:

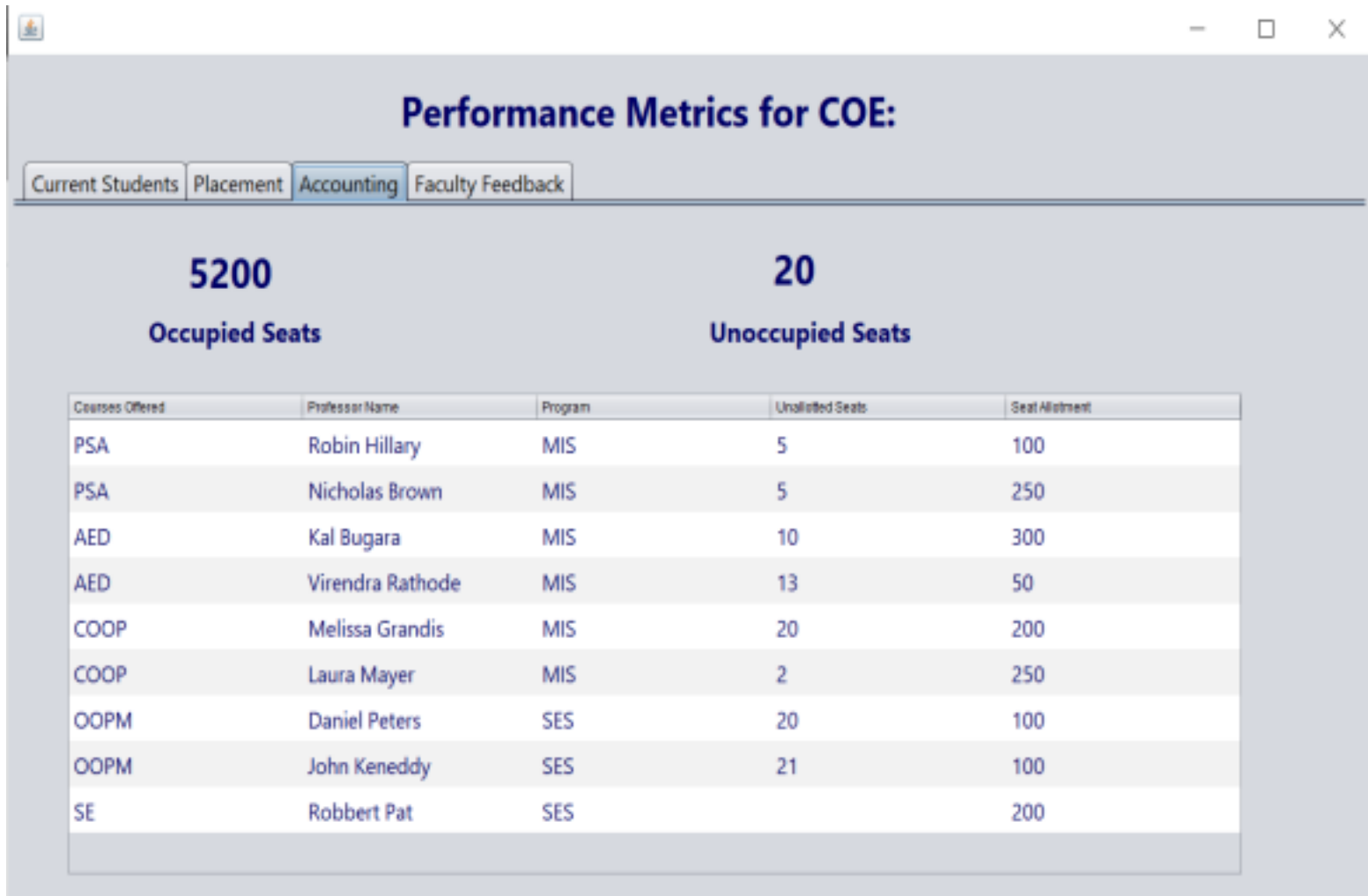
Student Name	List of Courses	CGPA	Rank
Brian M	Data Science, PSA	3.4	3
Andy R	COOP AED	3,6	2
Paul G	AED, PSA, COOP	3.8	1

Performance Metrics Screen – Placement Summary

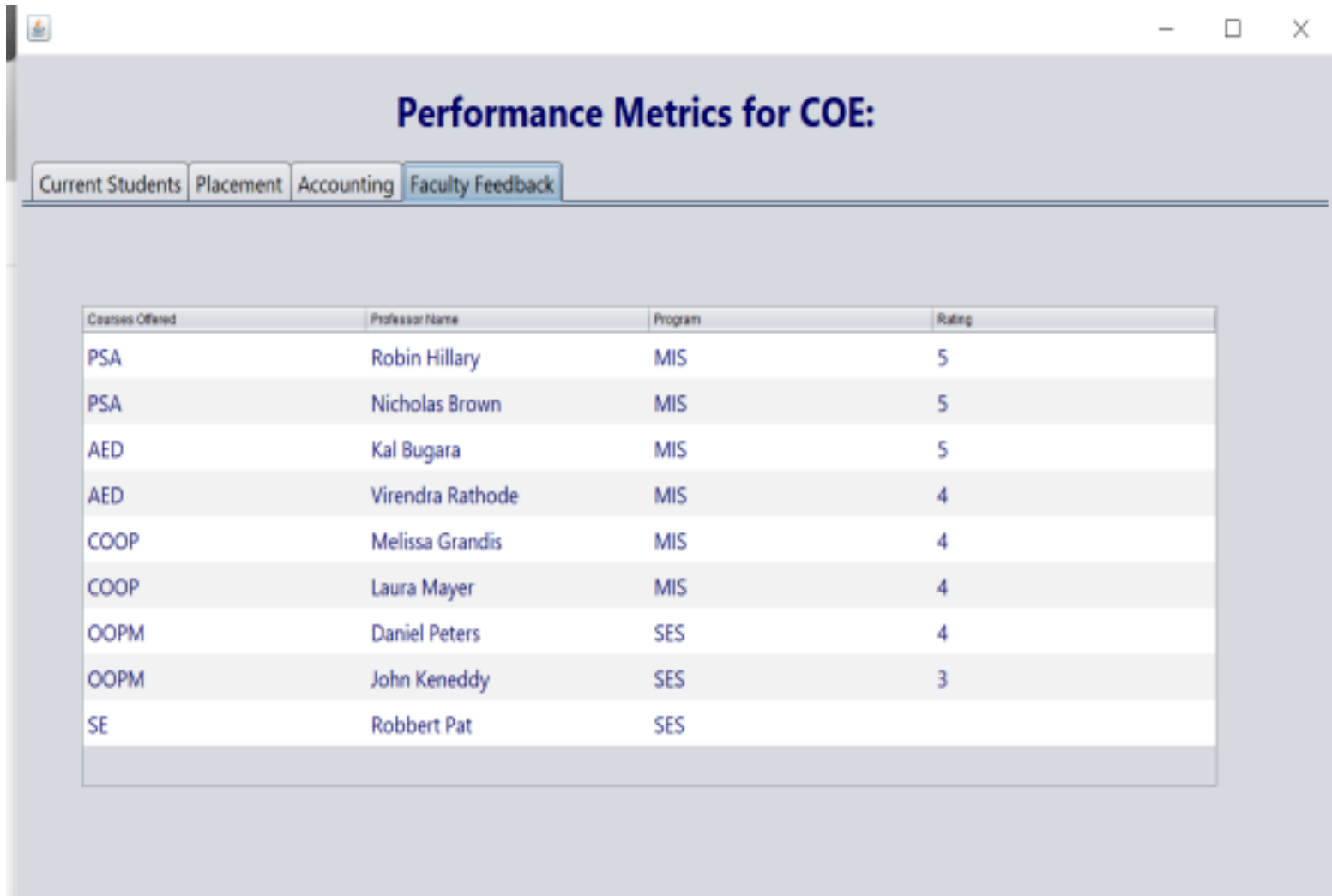


Current Students	Placement	Accounting	Faculty Feedback
Placement Year	Placement domain	Salary Range	Student Count
2015	Web Design	60-80k	100
2015	Cloud Engineer	80-100k	250
2015	Data Architecture	80-100k	300
2016	Business Analytics	50-70k	50
2016	Web Design	100-150k	200
2016	Cloud Engineer	200k	250
2016	Data Architecture	100-150k	100
2021	Web Design	80-100k	100
2021	Data Architecture	100-200k	200

Performance Metrics Screen – Accounting and Finance



Performance Metrics Screen – Faculty Rating



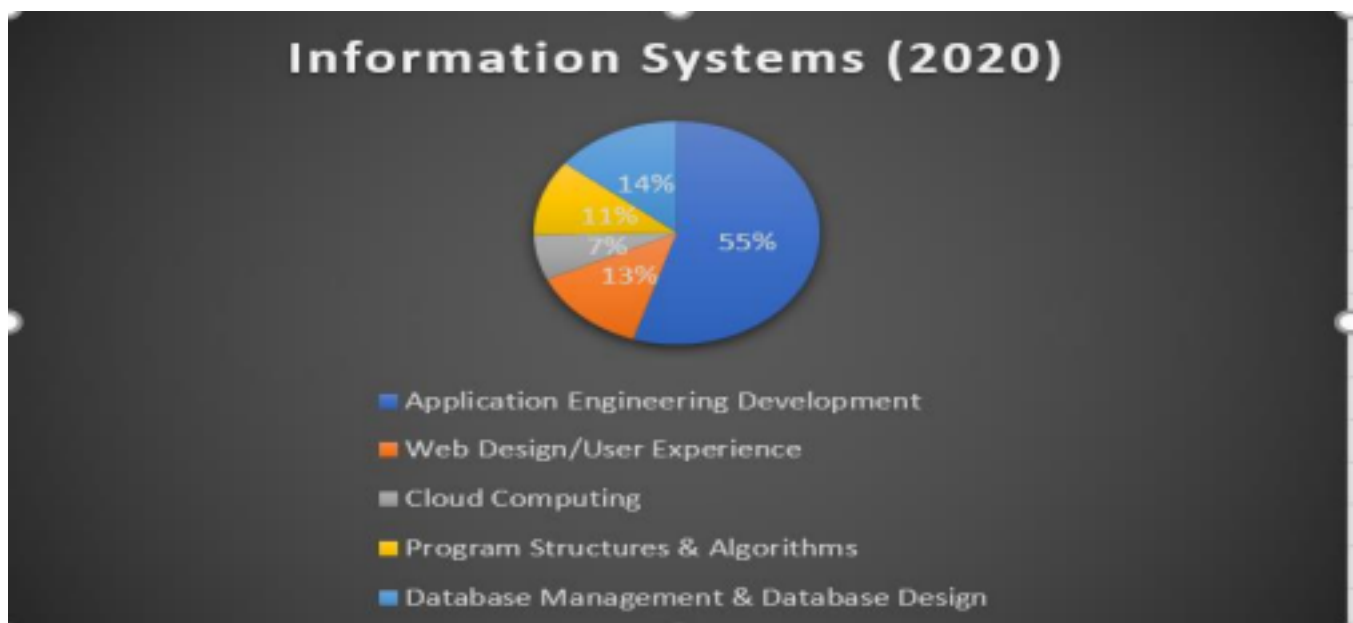
Current Students	Placement	Accounting	Faculty Feedback
Courses Offered	Professor Name	Program	Rating
PSA	Robin Hillary	MIS	5
PSA	Nicholas Brown	MIS	5
AED	Kal Bugara	MIS	5
AED	Virendra Rathode	MIS	4
COOP	Melissa Grandis	MIS	4
COOP	Laura Mayer	MIS	4
OOPM	Daniel Peters	SES	4
OOPM	John Keneddy	SES	3
SE	Robbert Pat	SES	

Administrator has access to view statistics in the form of graphs on a single button

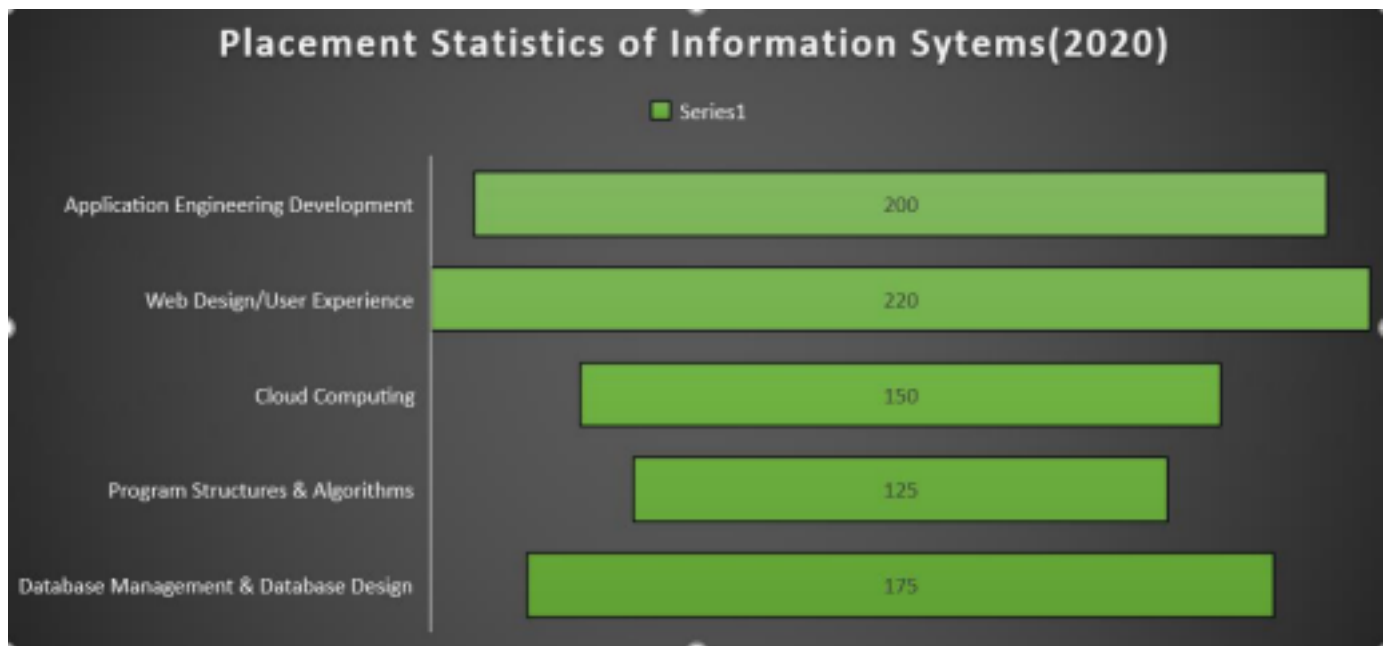
press. 1. Number of Students Enrolled in A Particular Course



2. Course Statistics of Information Systems (2020)



3. Placement Statistics of Information Systems (2020)



4. Placement Statistics of Previous 5 Years

