INFO 5100 - Application Engineering and Development

Assignment 3

UNIVERSITY PERFORMANCE EVALUATION SYSTEM

Ronak Patil
Prathamesh Nemade
Urvang Patel

Title: Designing a University Performance Evaluation System.

Purpose:

The purpose of this report is to understand the flow and approach of the education system and to generate a performance matrix of the students in the university to improve the university ranking.

Team Information:

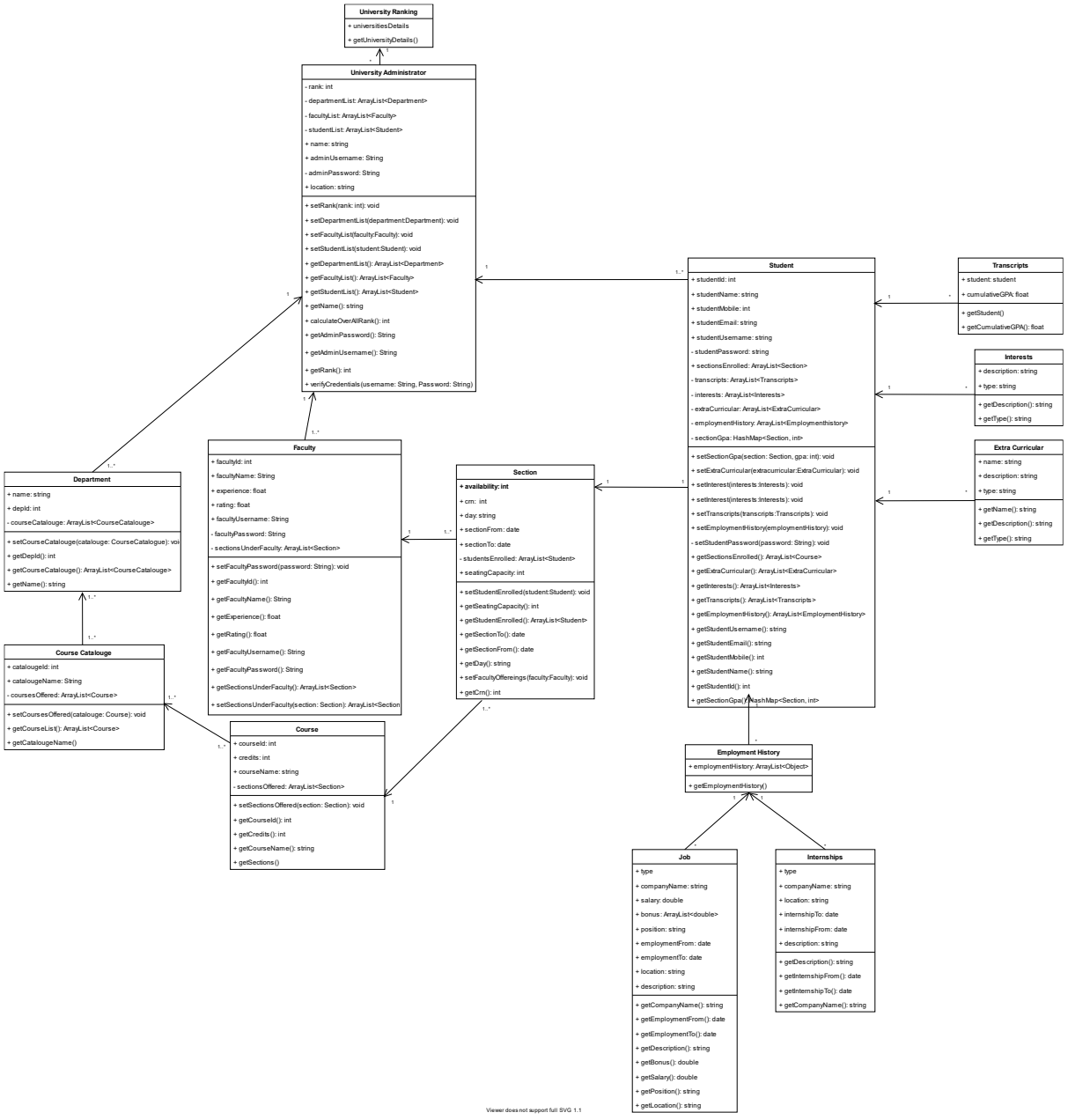
TEAM		
NAME	**E-MAIL**	**NUID**
Ronak Patil	patil.ron@northeastern.edu	001007035
Prathamesh Nemade	nemade.p@northeastern.edu	002139730
Urvang Patel	patel.ur@northeastern.edu	001568243

Problem Statement:

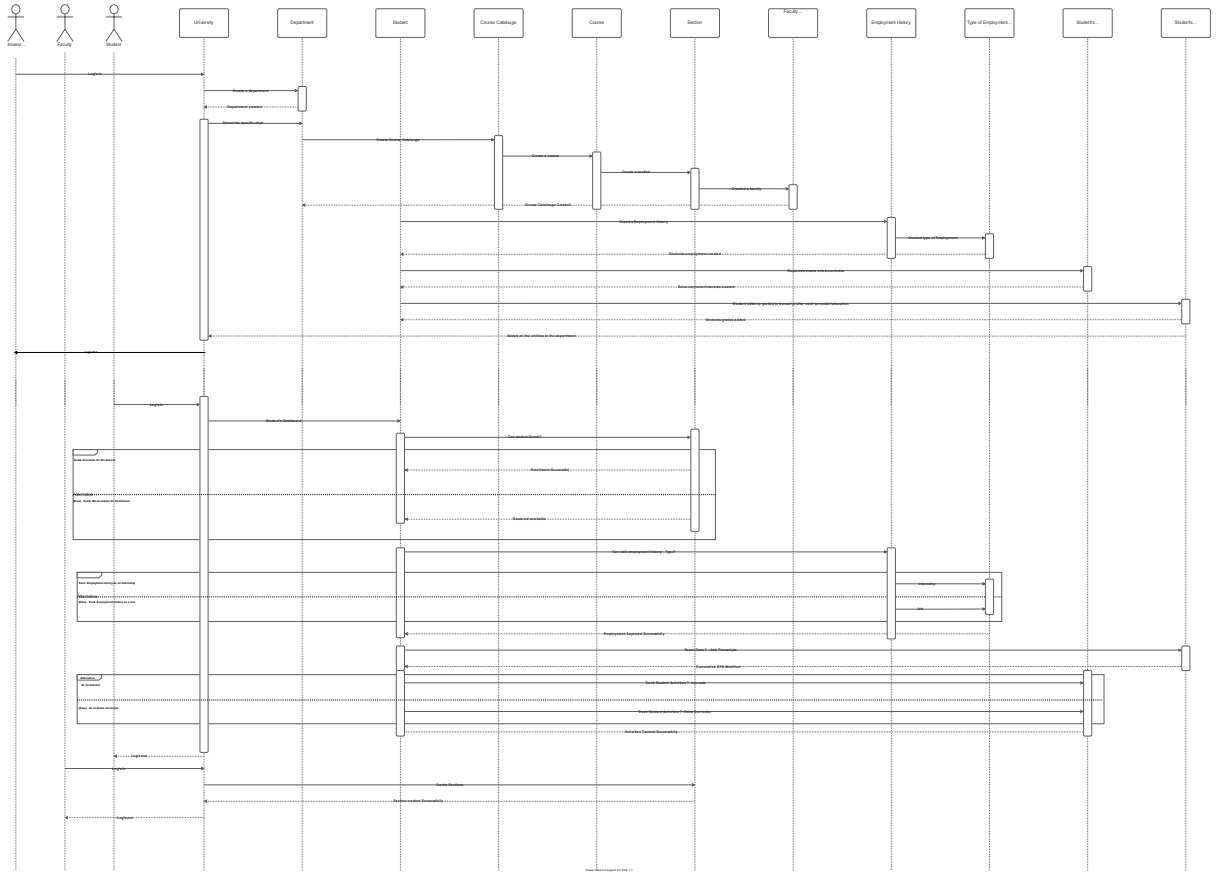
This model helps you to track down the students in the university and helps you out to find out which student has progressed over the period based on multiple factors. The major bodies of these models are **Admin**, **Faculty**, **Student**, **Employment History,** and **Student Activities**.

The current ranking system is biased towards **research,** but this system is designed by including all the major factors such as current salaries earned by the alumni, GPA, internships, the extracurricular activities are done by students all of this is considered for ranking up the university.

❖ UML Diagram:


University Ranking
+ universitiesDetails
+ getUniversityDetails()
University Administrator
- rank: int
- departmentList: ArrayList<Department>
- facultyList: ArrayList<Faculty>
- studentList: ArrayList<Student>
+ name: string
+ adminUsername: String
- adminPassword: String
+ location: string
+ setRank(rank: int): void
+ setDepartmentList(department:Department): void
+ setFacultyList(facultyFaculty): void
+ setStudentList(student:Student): void
+ getDepartmentList(): ArrayList<Department>
+ getFacultyList(): ArrayList<Faculty>
+ getStudentList(): ArrayList<Student>
+ getName(): string
+ calculateOverAllRank(): int
+ getAdminPassword(): String
+ getAdminUsername(): String
+ getRank(): int
+ verifyCredentials(username: String, Password: String)
Student
+ studentId: int
+ studentName: string
+ studentMobile: int
+ studentEmail: string
+ studentUsername: string
- studentPassword: string
+ sectionsEnrolled: ArrayList<Section>
- transcripts: ArrayList<Transcripts>
- interests: ArrayList<Interests>
- extraCurricular: ArrayList<ExtraCurricular>
- employmentHistory: ArrayList<Employmenthistory>
- sectionGpa: HashMap<Section, int>
+ setSectionGpa(section: Section, gpa: int): void
+ setExtraCurricular(extracurricular:ExtraCurricular): void
+ setInterest(interests:Interests): void
+ setInterest(interests:Interests): void
+ setTranscripts(transcripts:Transcripts): void
+ setEmploymentHistory(employmentHistory): void
- setStudentPassword(password: String): void
+ getSectionsEnrolled(): ArrayList<Course>
+ getExtraCurricular(): ArrayList<ExtraCurricular>
+ getInterests(): ArrayList<Interests>
+ getTranscripts(): ArrayList<Transcripts>
+ getEmploymentHistory(): ArrayList<EmploymentHistory>
+ getStudentUsername(): string
+ getStudentEmail(): string
+ getStudentMobile(): int
+ getStudentName(): string
+ getStudentId(): int
+ getSectionGpa(): HashMap<Section, int>
Transcripts
+ student: student
+ cumulativeGPA: float
+ getStudent()
+ getCumulativeGPA(): float
Interests
+ description: string
+ type: string
+ getDescription(): string
+ getType(): string
Extra Curricular
+ name: string
+ description: string
+ type: string
+ getName(): string
+ getDescription(): string
+ getType(): string
Faculty
+ facultyId: int
+ facultyName: String
+ experience: float
+ rating: float
+ facultyUsername: String
- facultyPassword: String
- sectionsUnderFaculty: ArrayList<Section>
+ setFacultyPassword(password: String): void
+ getFacultyId(): int
+ getFacultyName(): String
+ getExperience(): float
+ getRating(): float
+ getFacultyUsername(): String
+ getFacultyPassword(): String
+ getSectionsUnderFaculty(): ArrayList<Section>
+ setSectionsUnderFaculty(section: Section): ArrayList<Section
Section
+ availability: int
+ crn: int
+ day: string
+ sectionFrom: date
+ sectionTo: date
- studentsEnrolled: ArrayList<Student>
+ seatingCapacity: int
+ setStudentEnrolled(student:Student): void
+ getSeatingCapacity(): int
+ getStudentEnrolled(): ArrayList<Student>
+ getSectionTo(): date
+ getSectionFrom(): date
+ getDay(): string
+ setFacultyOfferings(faculty:Faculty): void
+ getCrn(): int
Department
+ name: string
+ deptId: int
- courseCatalouge: ArrayList<CourseCatalouge>
+ setCourseCatalouge(catalouge: CourseCatalogue): voi
+ getDeptId(): int
+ getCourseCatalouge(): ArrayList<CourseCatalouge>
+ getName(): string
Course Catalouge
+ catalougeId: int
+ catalougeName: String
- coursesOffered: ArrayList<Course>
+ setCoursesOffered(catalouge: Course): void
+ getCourseList(): ArrayList<Course>
+ getCatalougeName()
Course
+ courseId: int
+ credits: int
+ courseName: string
- sectionsOffered: ArrayList<Section>
+ setSectionsOffered(section: Section): void
+ getCourseId(): int
+ getCredits(): int
+ getCourseName(): string
+ getSections()
Employment History
+ employmentHistory: ArrayList<Object>
+ getEmploymentHistory()
Job
+ type
+ companyName: string
+ salary: double
+ bonus: ArrayList<double>
+ position: string
+ employmentFrom: date
+ employmentTo: date
+ location: string
+ description: string
+ getCompanyName(): string
+ getEmploymentFrom(): date
+ getEmploymentTo(): date
+ getDescription(): string
+ getBonus(): double
+ getSalary(): double
+ getPosition(): string
+ getLocation(): string
Internships
+ type
+ companyName: string
+ location: string
+ internshipTo: date
+ internshipFrom: date
+ description: string
+ getDescription(): string
+ getInternshipFrom(): date
+ getInternshipTo(): date
+ getCompanyName(): string
Viewer does not support full SVG 1.1

❖ **Sequence Diagram:**



Dashboards:

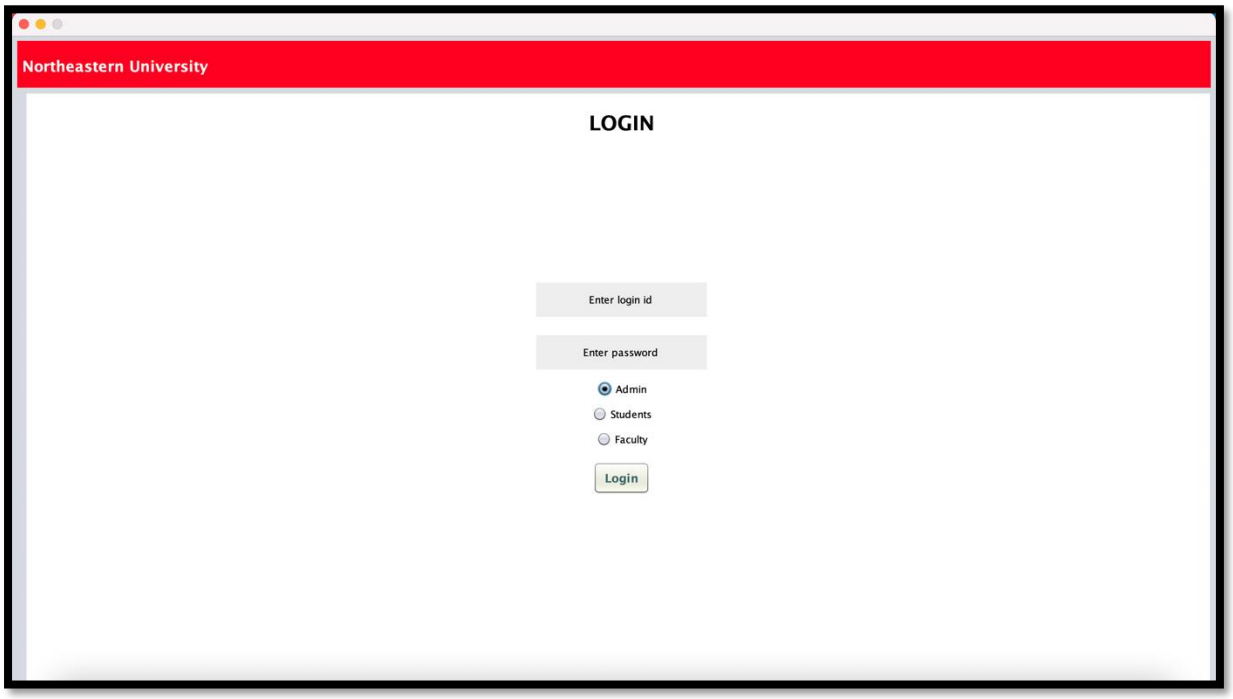
Mainly this model includes **three kinds** of dashboards or three types of logging portals. They are as follows:

1. University Administrator
2. Faculty
3. Students

Every dashboard denotes information related to the lower hierarchical bodies.

For example- The teacher or the faculty dashboard would denote the number of students enrolled in their specific subjects or respective subjects, whereas the student dashboard would represent the GPA in their ongoing semester, and this has been represented by special kinds of graphs.

- **Login Page:**



The login page allows the user to login into the appropriate role based on the credentials used. Mainly, there are three types of roles in this system. They are as follows:

- University Admin
- Student
- Faculty

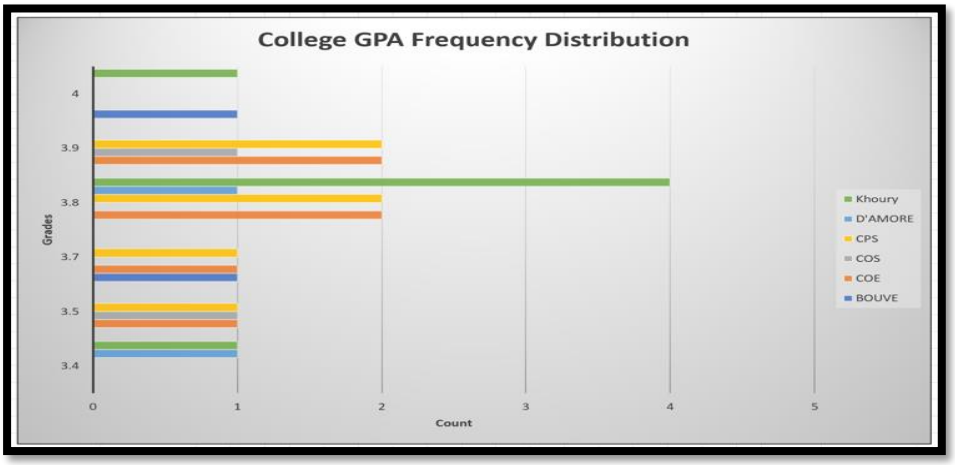
1. Following is the screenshot of the dashboard for the **University Administrator** login:

The above Admin's dashboard conveys details about the performance matrix of the overall university and the respective ranking with respect to other universities in the USA. The performance matrix is evaluated according to the average salary earned by the students in their respective companies down the line, based on the GPA they received during their masters, their participation in the other cultural and technical events in the university, and their interests in other activities.

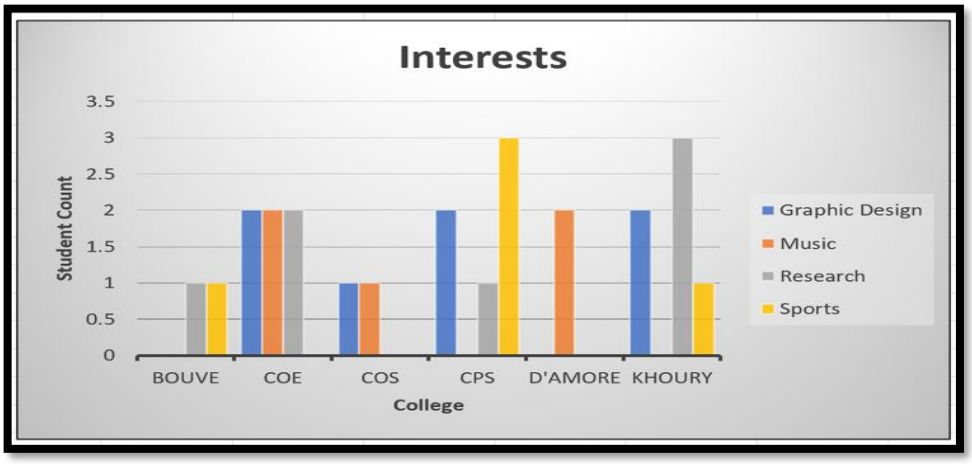
Calculation of **University Ranking** is based on the following factors:

a. Current Salary earned by the Students graduated from the respective university
b. GPA of graduated students
c. No. of students who received promotions
d. Net amount invested in Interests and Extra-Curricular activities.

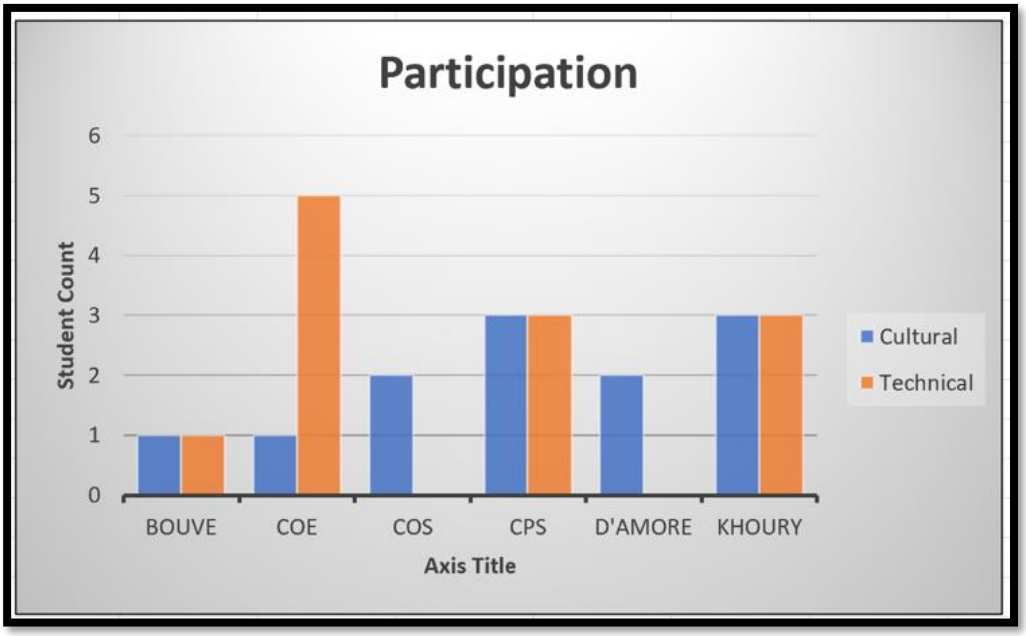
- The graph represents the average salary earned by the students from the respective colleges in their respective companies.
- The salary stands as the major factor to measure the growth of the students and the university.



- The graph represents the count of students according to their respective GPAs received during their masters.
- The GPA stands as the second major factor to check the growth of the students in the university.
- It would also help the recruiters to set the interview with the students performing well in their academics.

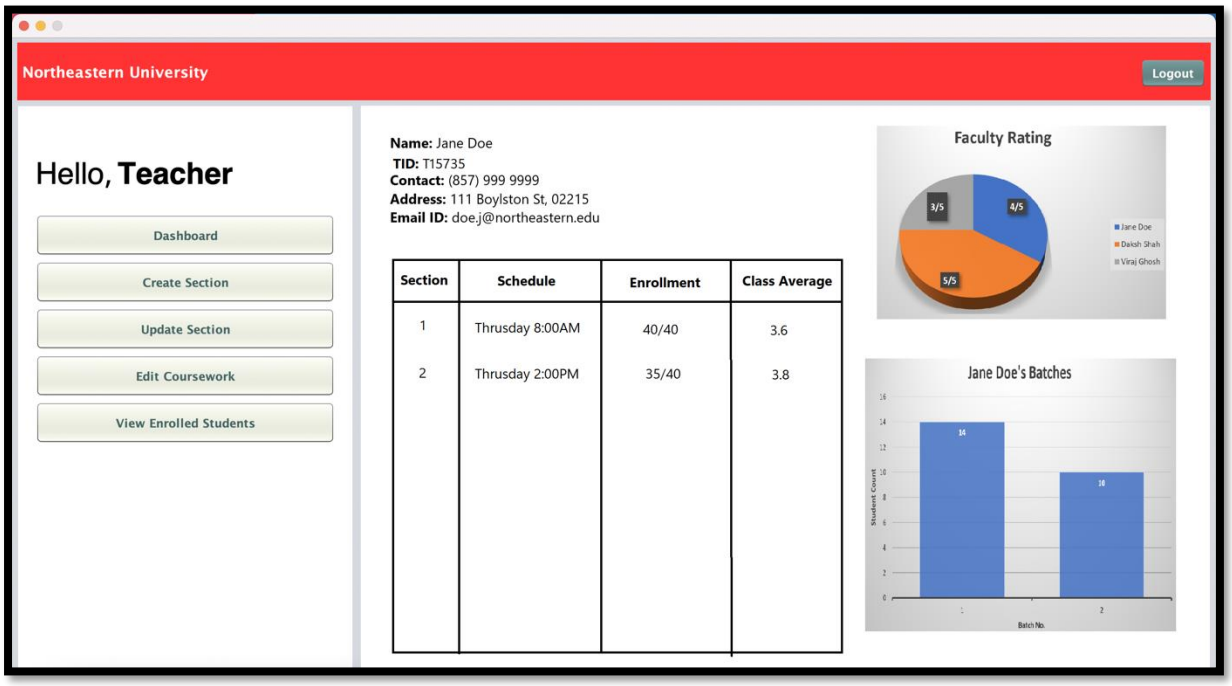


- Students' success rate depends on their interests.
- From the graph, we can see how the students from various departments have interests.



- Students' performance in their life not only depends on their studies but also on other activities/hobbies they pursue.
- We can see in the graphs how the students from different departments participated in various activities apart from their studies.

2. Following is the screenshot of the dashboard for the **faculty's** login:



The above faculty's dashboard conveys details about the logged-in user/teacher. Also, it shows up the details of the **sections** enrolled into by the logged-in faculty.
Moreover, the pie chart on the dashboard presents information/**ratings** about the rest of the faculty. Showing the **ranking** of the rest of the faculty helps the logged-in user to track down his/her **progress**. This can help the faculty to **improve**, **revamp** or even **add up** the required curriculum into the course/section.
The dashboard allows the teacher to see the enrolled students, create a section, and even edit the coursework.
Also, the bar graph on the bottom of the page represents the information of the enrolled students into the respective sections.

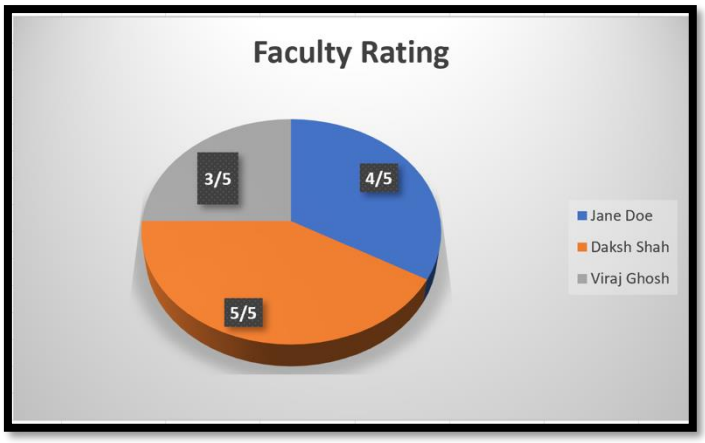
Pie chart – Rating of a faculty

Faculty rating is a critical factor contributing towards the success of a student.

The rating is calculated based on the following formula.

$\sum$(Student's rating of a teacher) $\equiv \sum$(All students enrolled in the specific class)

where, n is the total no. of students enrolled in the course.

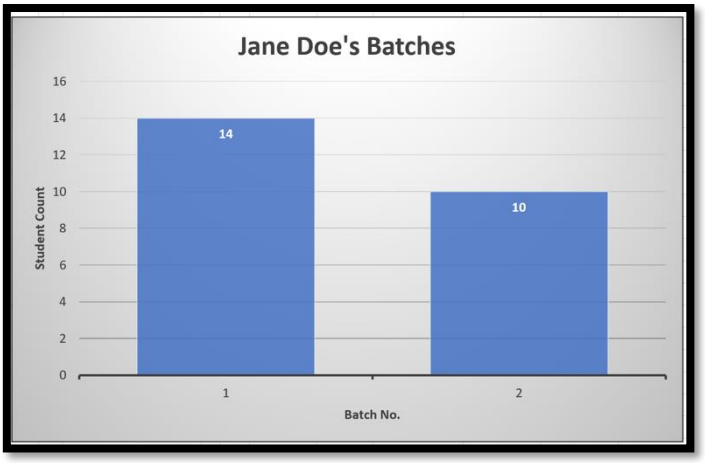


Bar Graph – Student count v/s Batch No.

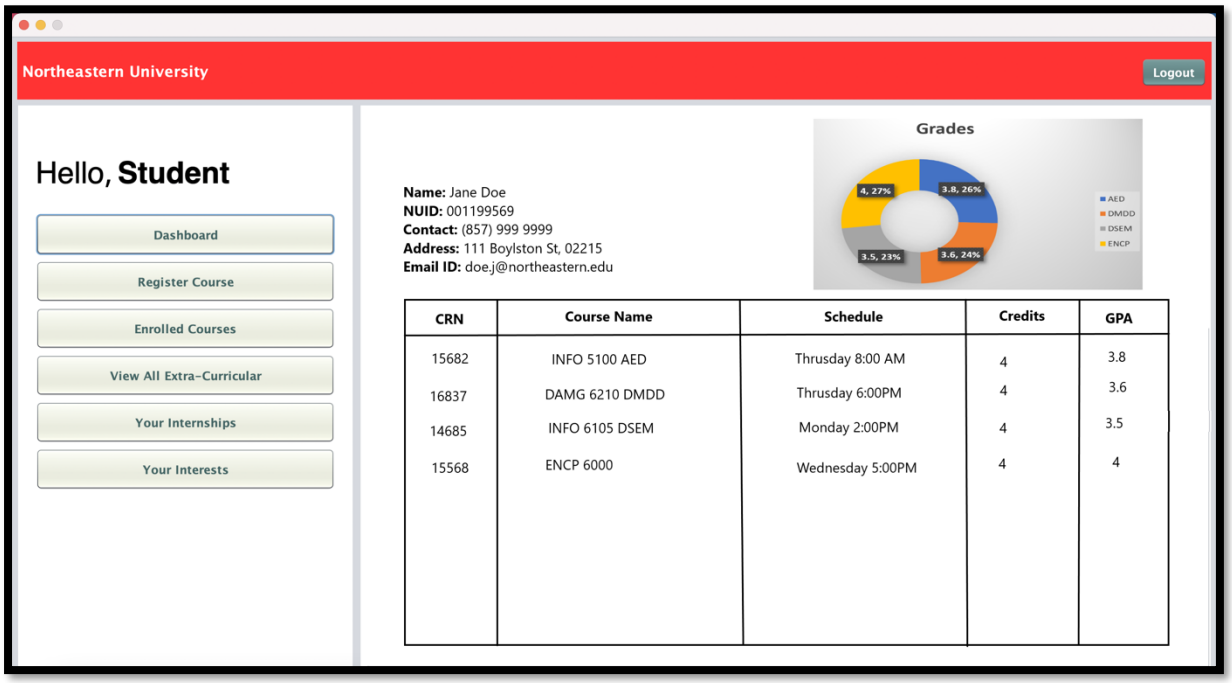
The batch size of a section is one of the major factors responsible for the productivity of the students. Less batch size can be responsible for a better reach of a student, and vice-versa.

Chances of Good Productivity $\propto$ 1 / High Batch Size

Variable Factor – Students Count;
Stagnant Factor – Batch Nos.



3. Following is the screenshot of the dashboard for the **student's** login:



The above student's dashboard conveys the information regarding the student's enrolled courses. It shows up the information of the logged-in user, also, always the students to register into a course, track down the interests and extra-curricular which the student is into.
Moreover, the students get a cumulative view of the progress of the ongoing semester.
The donut graph represents the information of the GPAs and the contributing percentages. This can help the student to track the progress in a single glance.

Donut Graph – Grades of the sections enrolled.

The grades help to track down the progress of the students. Also, it helps to figure out which are the important factors/sections that helped the students to progress over the period.

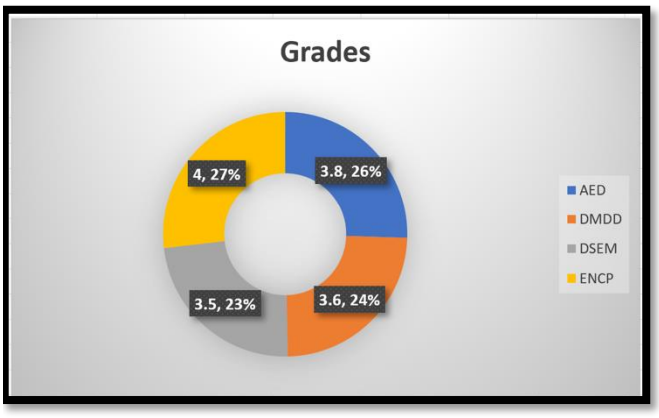
The grades are calculated based on the following formula.

$$\sum(\text{Student's overall grades}) = \sum_{I=n} (\text{All grades})$$

$$G\ (\text{rating, percentage contributing}) = \sum_{I=0}$$

$$(n, \sum(\text{Student's overall grades})) * 100)/n)$$

where, n is the total no. of sections the students has enrolled into.



Class Tables:

Following are the classes and their attributes used to develop the university model-

University Administrator:

This is an admin class for the university which manages the university and generates feedback reports to improve the university ranking.

Access	Attribute	Description	Data Type
Private	rank	Current rank of the university	Integer
Private	department list	A list of departments under the university	ArrayList<Department>
Private	facultyList	List of faculties employed by the university	ArrayList<Faculty>
Private	studentList	List of students enrolled in the university	ArrayList<Student>
Public	Name	University Name	String
Public	adminUserName	Admins user ID	String
Private	adminPassword	Password for admin login	String
Public	Location	University Address	String

Access	Method	Description	Return Type
Public	setRank(int)	Setter to set the university rank	Void
Public	setDepartmentList (Department)	Setter for departments	Void
Public	setFacultyList(Faculty)	Setter for faculty	Void
Public	setStudentList(Student)	Setter for students	Void
Public	getDepartmentList()	Getter for departments	ArrayList<Department>
Public	getFacultyList()	Getter for the faculty	ArrayList<Faculty>
Public	getStudentList()	Getter for students	ArrayList<Student>
Public	getName()	Getter for university name	String
Public	getLocation()	Getter for university address	String
Public	getAdminUserName()	Getter for admins user ID	String
Public	getRank()	Getter for university Rank	Int
Public	verifyCredentials (String, String)	Verify admin login	Void

Department:

This is a class department class, which creates an instance for all the departments in the university.

Access	Attribute	Description	Data Type
Public	name	The name of the department	String
Public	deptID	The ID of the department	Int
Private	courseCatalogue	List of different catalogues offered by the department	ArrayList<CourseCatalogue>

Access	Method	Description	Return Type
Public	setCourseCatalogue (CourseCatalogue)	Setter to set the course catalogue for the department	Void
Public	getCouseCatalogue()	Getter to fetch the course catalogues offered by the department	ArrayList <CourseCatalogue>
Public	getDeptID()	Getter for department ID	Int
Public	getName()	Getter for department name	String

Course Catalogue:

The class which stores the different course catalogues offered by the university and departments.

Access	Attribute	Description	Data Type
Public	catalougeID	The ID for the catalogue instance	Int
Public	catalogueName	Name of the catalogue	String
Private	coursesOffered	List of courses offered under this catalogue instance	ArrayList<Course>

Access	Method	Description	Return Type
Public	setCoursesOffered (Course)	Setter to set the courses in the catalogue	Void
Public	getCourseList()	Getter to get the list of courses offered	ArrayList<Course>
Public	getCatalogueName()	Getter to get the catalogue name	String

Course:

The class for courses offered in the university.

Access	Attribute	Description	Data Type
Public	courseID	The ID of the course	Int
Public	Credits	The course credits	Int
Public	courseName	The name of the course	String
Private	sectionOffered	List of sections offered under this course	ArrayList<Section>

Access	Method	Description	Return Type
Public	setSectionsOffered (Section)	Setter to set the sections of the course	Void
Public	getCourseID()	Getter to fetch the course number	Int
Public	getCredits()	Getter to fetch the course credits	Int
Public	getCourseName()	Getter to fetch the course name	String
Public	getSections()	Getter for the list of sections	ArrayList<Section>

Faculty:

This is the class which stores the faculties employed by the university.

Access	Attribute	Description	Data Type
Public	facultyID	The ID of the faculty	Int
Public	facultyName	Name of the faculty	String
Public	experience	Years of experience for the faculty	Float
Public	ranking	The faculty ranking	Float
Public	facultyUserName	The login ID for the faculty	String
Public	facultyPassword	The login password for the faculty	String

Access	Method	Description	Return Type
Public	setFacultyPassword(String)	Setter for the login password	Void
Public	getFacultyID()	Getter to fetch the faculty ID number	Int
Public	getFacultyName()	Getter to fetch the faculty name	String
Public	getExperience()	Getter to fetch the faculty experience	Float
Public	getRating()	Getter for the professor ranking	Float
Public	getFacultyUserName()	Getter to fetch the faculty login ID	String
Public	getFacultyPassword()	Getter to fetch the faculty password	String

Section:

This is a class which stores the details about the sections for courses.

Access	Attribute	Description	Data Type
Public	crn	The crn number for the course section	Int
Public	day	The day on which the class is scheduled	String
Public	sectionFrom	Start date for the section	Date
Public	sectionTo	End date for the section	Date
Private	facultyOffering	List of faculty offering the course section	ArrayList<Faculty>
Private	studentEnrolled	List of student enrolled in the section	ArrayList<Student>
Public	seatingCapacity	The maximum capacity of the class	int

Access	Method	Description	Return Type
Public	SetStudentEnrolled (Student)	Setter to set the enrolled students	Void
Public	getSeatingCapacity()	Getter to fetch the section capacity	Int
Public	getStudentEnrolled()	Getter to fetch the list of enrolled students	ArrayList<Student>
Public	getSectionTo()	Getter to fetch the section end date	Date
Public	getSectionFrom()	Getter to fetch the section Start Date	Date
Public	getDay()	Getter to fetch the day of the week the section is held	String
Public	getFacultyOfferings()	Getter to get the list of faculties offering the course sections	ArrayList<Faculty>

Public	setFacultyOfferings (Faculty)	Setter to set the list of faculties offering the course section	Void
Public	getCrn()	Getter to fetch the CRN of the section	Int

Student:

This is a class which stores the details about the students enrolled in the university.

Access	Attribute	Description	Data Type
Public	studentID	The unique ID for student	Int
Public	studentName	The name of the student	String
Public	studentMobile	The contact information of the student	Int
Public	studentEmail	Email ID of the student	String
Public	studentUsername	The login ID for the student	String
Private	studentPassword	The login password for the student	String
Public	coursesEnrolled	List of courses taken by the student	ArrayList<Course>
Private	transcripts	List of transcripts for the courses enrolled	ArrayList<Transcripts>
Private	interests	List of interests of the student	ArrayList<Interest>
Private	extraCurricular	List of extracurricular activities done by student	ArrayList <ExtraCurricular>
Private	employmentHistory	List of all the jobs	ArrayList <EmploymentHistory>
Public	gpa	The cumulative grade of the student	Float

Access	Method	Description	Return Type
Public	getCoursesEnrolled()	Getter to fetch the courses taken by student	ArrayList<Course>
Public	getExtraCurricular()	Getter to fetch the list of extra curricular activites done by student	ArrayList <ExtraCurricular>
Public	getInterests()	Getter to fetch the students interests	ArrayList<Interest>
Public	getTranscripts()	Getter to fetch the stusents transcripts	ArrayList<Transcript>
Public	setExtraCurricular (ExtraCurricular)	Set the extracurricular activities of the student	Void

Public	setInterests(Interest)	Set the student interests	Void
Public	setTranscripts(Transcript)	Set the student transcripts	Void
Public	setStudentPassword(String)	Set student login password	Void
Public	getEmploymentHistory()	Get the employment history of the student	ArrayList <EmploymentHistory>
Public	setEmploymentHistory (EmploymentHistory)	Set the employment history of the student	Void
Public	getStudentUsername()	Get the student login ID	String
Public	getStudentEmail()	Get the student email id	String
Public	getStudentMobile()	Get the student contact info	Int
Public	getStudentName	Get the student name	String
Public	getStudentID()	Get the student ID	Int
Public	getGpa()	Get the student GPA	float

Transcript:

The class which creates the transcripts for the students.

Access	Attribute	Description	Data Type
Public	student	The student object whose transcript is stored	Student
Public	cumulativeGpa	The cumulative gpa of the student	float

Access	Method	Description	Return Type
Public	getstudent()	Get the student object	Student
Public	getCumulativeGpa()	Get the student GPA	float

Interest:

The class which represents the interests for the student.

Access	Attribute	Description	Data Type
Public	description	The interest description	String
Public	type	The domain of the interest	String

Access	Method	Description	Return Type
Public	getDescription()	Get the description	String
Public	getType()	Get the type	String

Extracurricular:

The class represents the extracurricular activities done by the student.

Access	Attribute	Description	Data Type
Public	name	The name of the activity	String
Public	description	The description about the activity	String
Public	type	The domain of the activity	String

Access	Method	Description	Return Type
Public	getDescription()	Get the description	String
Public	getType()	Get the type	String
Public	getName()	Get the name	String

Employment History:

The class that stores the employment details about the students.

Access	Attribute	Description	Data Type
Public	employmentHistory	List of all the jobs	ArrayList <EmploymentHistory>

Access	Method	Description	Return Type
Public	getEmploymentHistory()	Get the employment history of the student	ArrayList <EmploymentHistory>

Job:

The class which stores the job details.

Access	Attribute	Description	Data Type
Public	Type	The domain of the job	String
Public	companyName	The name of the company	String
Public	Salary	The salary paid	Double
Public	Bonus	The bonus paid	ArrayList<double>
Public	Position	The job role	String
Public	employmentFrom	Start date of the job	Date
Public	employmentTo	End date of the job	Date
Public	Location	The location of the job	String
Public	Description	The job role description	String

Access	Method	Description	Return Type
Public	getcompanyName()	Get the name of the company	String
Public	getEmploymentFrom	Get the start date	Dare
Public	getEmploymentTo	Get the end date	Date
Public	getDescription	Get the job role description	String
Public	getBonus	Get the list of bonus received	ArrayList<double>
Public	getSalary	Get the salary	Double
Public	getPosition	Get the job designation	String
Public	getLocation	Get the company location	String

Internship:

The class stores the details about the internships done by the students.

Access	Attribute	Description	Data Type
Public	Type	The domain of the job	String
Public	companyName	The name of the company	String
Public	employmentFrom	Start date of the job	Date
Public	employmentTo	End date of the job	Date
Public	Description	The job role description	String

Access	Attribute	Description	Data Type
Public	getcompanyName()	Get the name of the company	String
Public	getEmploymentFrom()	Get the start date	Date
Public	getEmploymentTo()	Get the end date	Date
Public	getDescription()	Get the job role description	String

❖ **Conclusion**:

- In this presented solution, we have designed a system which can be used to check the measurement of the entities in the university.
- Based on the solution presented the major factors such as salaries earned by their current alumni, GPA, extra-curricular, interests and the no. of internships are the major key factors which are the responsible for the student's growth.
- Hence, this grading technique will help to improve the quality of education anywhere and hold people accountable for improving the quality of life through education, learning to learn, and feedback.