University Performance Measurement Model

Model Purpose:

The application's main purpose is for the University admins to monitor the university's performance, based on various factors. This model will take into consideration student performance, faculty performance, alumni performance, and other factors to maintain a score for each entity. Using LinkedIn data we can keep track of the details of our alumni. Overall we will maintain a score for each of the required entities and assign a rank based on their score which will be displayed to the University Admins.

Business Problems Addressed:

- Only college and university administrators have access to view the performance metric of their academic units.
- Admins can login to the application using username and password.
- The application measures the performance of a college based on the rankings acquired by the department, courses, faculty and students.
- college and university administrators to compare the performance of their academic units.
- Student details like name of the student, age, contact details, email address, address, department, college name, is stored for every student.
- University has access to college details and can add or modify the details.
- College details like college id, college name, number of departments, college ranking and tuition cost is stored in the college class.
- Department class stores information like id, name and the department ranking.
- Faculty class contains details of the faculty working in the college.These details include name, age, email, contact, qualification, designation, courses taught and the rank assigned to faculty.
- Alumni class stores all the details of the passed out students of the college.Details contain their name, address, year of graduation, their employment details, etc.
- Each department has a set of courses.These course details like course id, name, credits assigned and course ranking, are stored in Course class.

Business Rules:

- Only one registration per admin is permitted.
- Admin can login any number of times
- Admin can recover account in case he forgets the password
- A restaurant can receive multiple orders from the same or different customers.

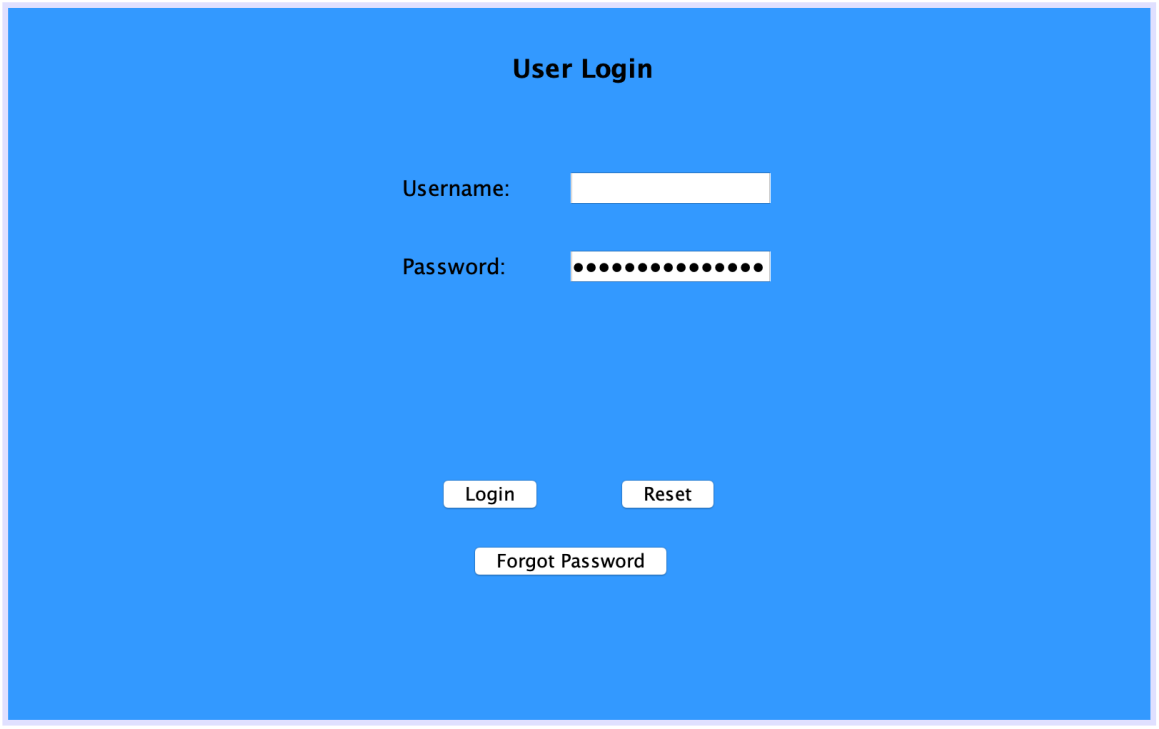
- Admin of one college can view performance data of another college as well for comparison.
- All the class level member variables would be defined as private.
- The class level member methods may be defined as public/private/protected as the case may be.

Design Requirements :

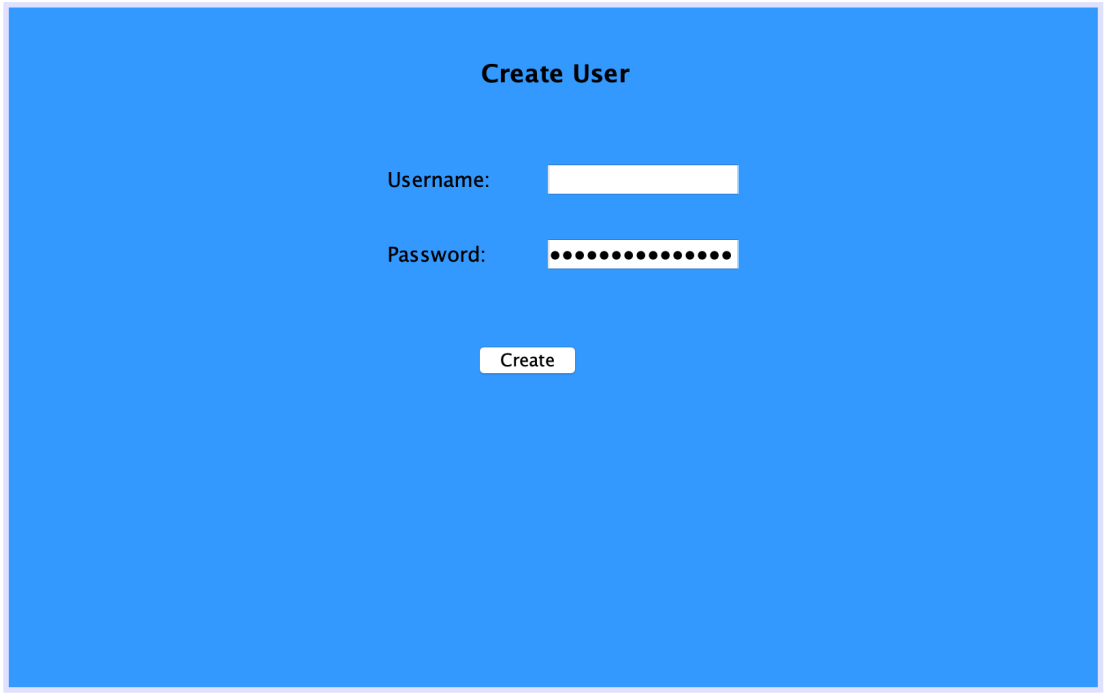
- Use UML class diagram.
- Each class represents a concept which encapsulates state (**attributes**) and behavior (**methods**).
- The name of the class appears in the first partition.
- Attributes,with their data types are shown in the second partition.
- Operations are shown in the third partition. They are services the class provides.
- “+” denotes public attributes or operations and “-“ denotes private attributes or operations.
- The lines joining classes shows association between them.

User Interface:

User Login Panel

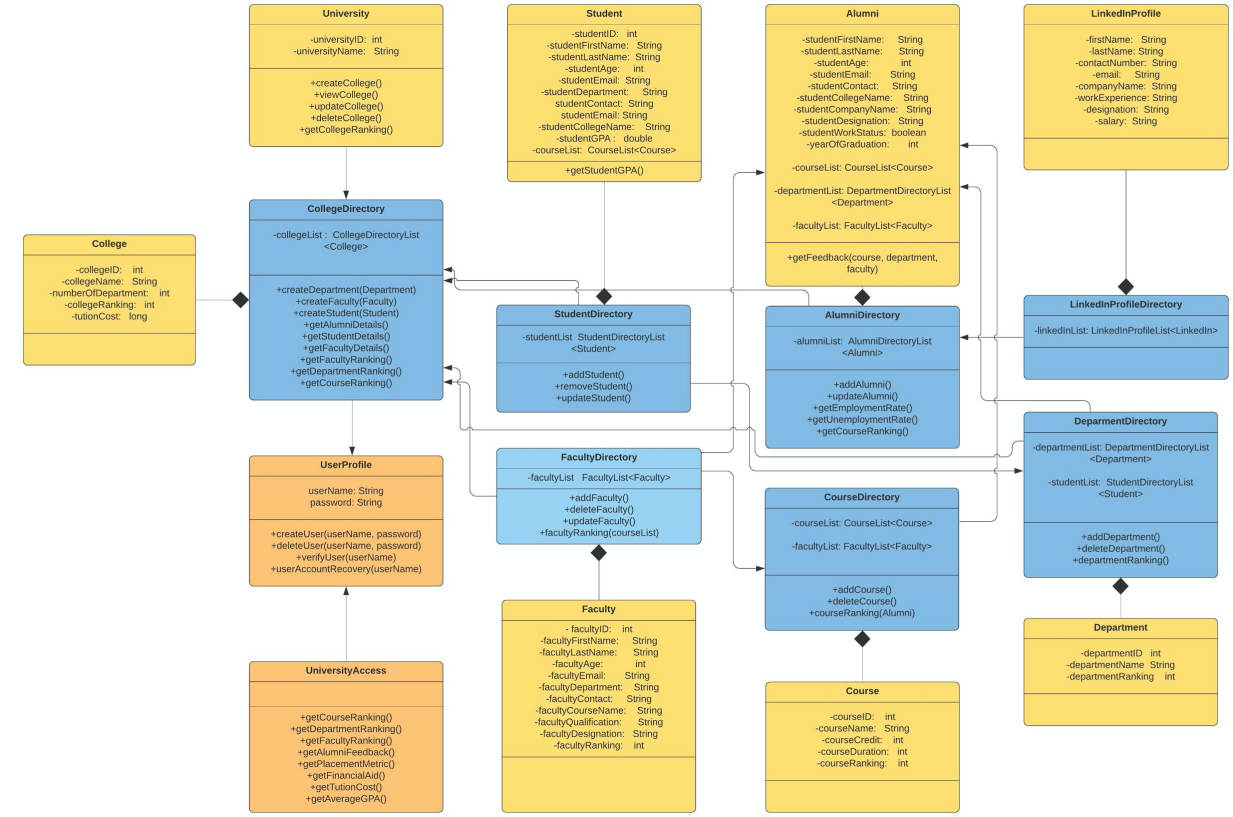


Create User Panel



Dashboard Panel

Object Model Diagram:



Class Details - Object Model:

Alumni:

- **getFeedback(course, department, faculty)**
    - This will get feedback about the different courses that the alumni had taken in their duration of the graduate school which has directly helped him to get a job.
    - The feedback also contains information about the current job status of the person whether they are employed or not.
    - If the person is employed, the user is asked to enter their job role, select the courses which were beneficial to them, department name, contact details, email id, and LinkedIn name to verify their details.
    - If the person is unemployed, the user is asked to enter their department, the courses they have taken, their LinkedIn profile, and their contact information. This information is used to calculate department ranking.

AlumniDirectory:

Functions:

- **addAlumni()**: This method is used to add specific alumni to the alumni directory.
- **updateAlumni()**: This method is used to update the contents of the alumni to the alumni directory.
- **getEmploymentRate():** This method is used to get the details of the working alumni based on their department and they will be allowed to fill the feedback only if they are working.
- **getUnemploymentRate():** This method shows how many alumni are unemployed from a particular department.
- **getCourseRanking():** This method fetches the ranking of a specific course from the feedback given by the alumni and stores the data in the alumniFeedback list.

University:

Functions:

- **createCollege():** This method is used to create the college under a university.
- **viewCollege():-** This method is used to view a college a the university.
- **updateCollege():-**This method is used to update the details of a college under a university.
- **deleteCollege():-**This method is used to delete a college from a university.
- **getcollegeRanking():-** This method is used to calculate the rank of the college based on the Department Ranking obtained from the getDepartmentRanking() class.

CollegeDirectory:

Functions:

- **createDepartment(Department):-** This method is used to create a department within the college and stores it in the Department directory.

- **createFaculty(Faculty):-** This method is used to create a Faculty for a college and stores it in the Faculty Directory.

- **createStudent(Student):-** This method is used to create a student and stores it in the Student directory.

- **getAlumniDetails(<Alumni Directory>**):- This method will get the alumni details from the alumni directory.

- **getStudentDetails(<StudentDirectory>):-** This method will get the student details in the student directory.

- **getFacultyDetails<FacultyDirectory>):-** This method will get the faculty details from the faculty Directory.

- **getDepartmentRanking()** : From the Alumni Employment and Unemployment rate the data is fetched for the alumni current status. If the Alumni are employed, they are directed to the feedback form and the mean of all the department ranks is calculated and ranked. It is also ranked based on the course rankings from each department. Each course is associated with a department and it sorted based on the highest-ranking and the department ranking is calculated.

- **getCourseRanking():-**The feedback of Alumni based on each course is stored in the alumniFeedbakArray list. From the array, the ranking is calculated based on the highest feedback rate that is given by the alumni.

- **getFacultyRanking()**: With the student id, we will be able to assign each course they have taken with the particular faculty and as each course is ranked, the faculty who has been assigned for that course will also be ranked.

Student Directory:

Functions:

- **addStudent():-** This method will add a student to the Student directory.

- **removeStudent():-** This method will remove a student from the Student directory.

- **updateStudent():**- This method is used to update the student information from the Student directory.

Student:

Functions:

- **getStudentGPA():-** This method is used to get the student GPA which will be used to calculate the course, Department & Faculty Rankings.

Faculty Directory:

Functions:

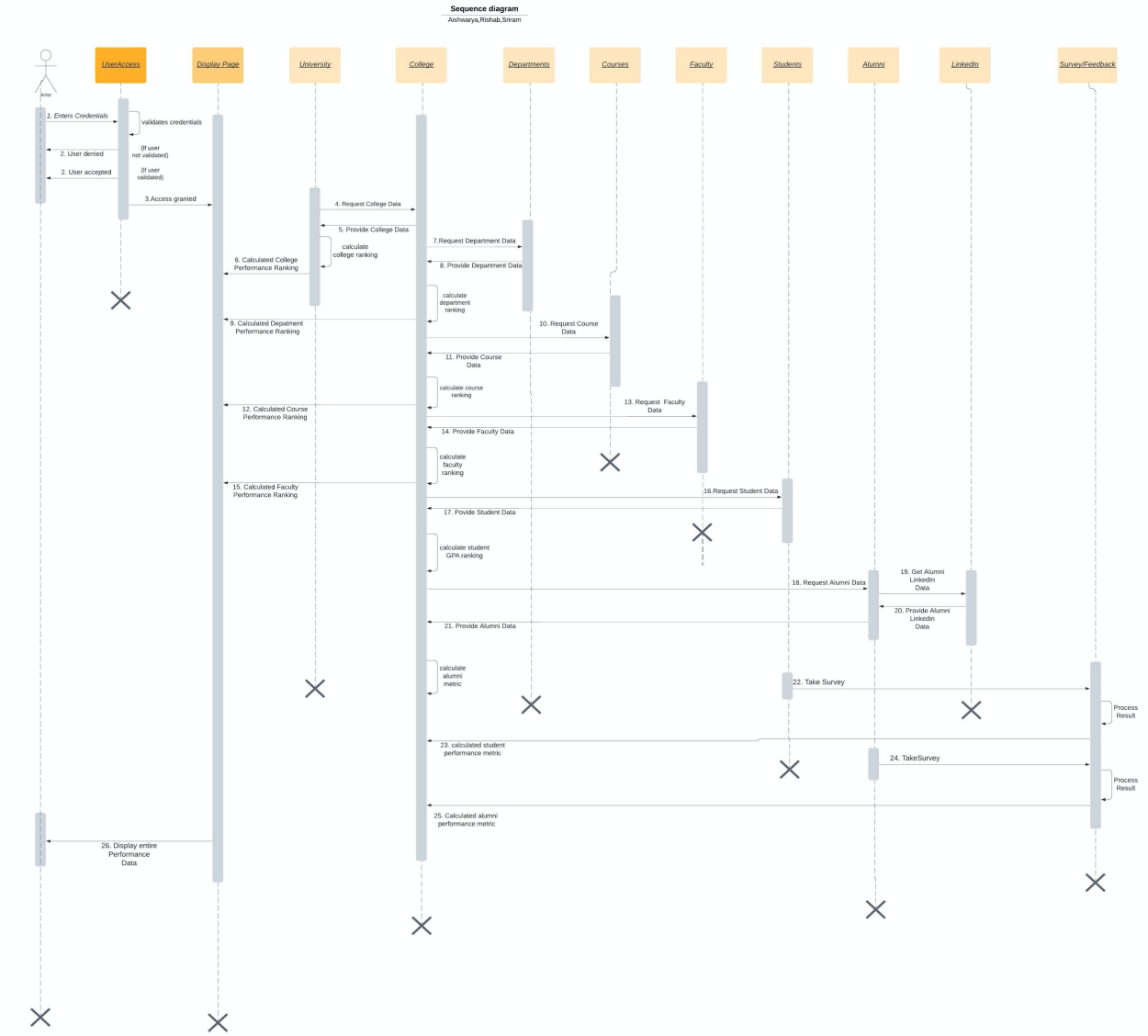
- **addFaculty():-** This method will add a faculty to the faculty directory.
- **deleteFaculty():**- This method will delete a faculty from the faculty directory.
- **updateFaculty():**- This method is used to update any details of a faculty from the faculty directory.
- **facultyRanking(courseList):-** This method is used to calculate the ranking of the faculty based on the defined criteria.

Course Directory:

Functions:

- **addCourse():-** This will add the course to the course directory.
- **deleteCourse():-** This will remove a course from the course directory.
- **courseRanking(Alumni) –** This will calculate the course ranking based on the defined criteria.

Sequence Diagram:



Conclusion:

The Object Model Diagram and Sequence Diagram for the university has been created and the class description of the Object model has been explained along with their respective functions.