UNIVERSITY COURSE RANKING SYSTEM

ABSTRACT

The purpose of this report is to create a course ranking system which helps universities to keep the track of their coursework that is being taught to the students in the university and its relevance to the student's growth even after graduating from the university. The course ranking system is based on the data received from the graduate students over the years i.e. from the time they joined the university till present. Course ranking system will help university administrators to keep track of students, courses and faculty that will eventually help them to make necessary changes to improve the quality of education.

PROBLEM STATEMENT

Most university ranking models base their data on research and accomplishments of the universities. These ranking systems are adequate for analyzing the facilities and innovations produced by the university, however, they do not study the overall success of the students who have graduated from the universities and how the courses offered by that university has contributed to their success.

PROJECT OBJECTIVE

The main objective is to develop a robust Student and Course Analysis Management System for the University.

Specific objectives:

- To create a dashboard which obtains information from graduated students about their current employment, designation, salary etc., and their rating of each course based on how gainful it was to them.
- To create an algorithm which assigns a score to each student based on the data collected from the dashboard and use those scores to rank the most beneficial courses and the most appreciated faculty of the university.
- To create an admin dashboard to View these statistics and rankings, and Update or Delete the least ranking courses. The dashboard will also allow the admin to create, view, update or delete a student, faculty or course in the system.

INTRODUCTION

We have designed a system which evaluates the vocational success of each student who have graduated over a period of 5 years, along with statistics about which courses bolstered their success, to rank the courses based on the overall contentment of the students who have studied it. Our system is an application that runs the cron and sends emails to graduate students with a link to a form that has different varieties of question ranging from personal information to feedback regarding the course ,current job etc. This email is sent at regular intervals to make sure that the response rate is good enough to make an ranking system out of it. The idea behind receiving data from the alumni is based on the assumption that it will be a 2-way exchange where the alumni, provides the entire details requested in the form and gets access to an alumni portal which will have all the alumni details along with the employer's details. The portal will be used to as a medium for employers to look out for alumni and vice versa.

Following are some of the questions asked in the form sent to the alumni.

1. What was your first salary and what is your current salary?

2.What is your current/past Companies and position?

3.How long have you been employed?

4.What is your job Location?

5.Have you had any promotions? If yes what role?

6.Rate your job Satisfaction from 1 to 10

7.Rate your Courses from 1 to 10 based on their relevance to your work?

8.Rate all your courses from 1 to 10 based on their course structure?

9.Rate the subject from 1 to 10 you feel helped you getting the job?

10. Please select one option if you think a change is needed in your relevant coursework based on the current industry standards?

The data received from the alumni helps in making a ranking algorithm which calculates the individual student score. On basis of the student score, course score can be calculated consequently leading to the course rank. Faculty ranking can be calculated too based on the data received and the student score. This details will help the University to decide which courses are useful and which are the ones that need upgrade.

SCOPE

The proposed system is intended to collect and analyze specific information of students such as personal details, course details and employment details etc.

This project when completed will provide a comprehensive study of the practical applicability of the courses offered by the university and the faculty who taught it.

The proposed system will have the following features:

- Login module: Login module will help in authentication of user accounts. Only Administrators and Students who have valid login credentials can access their respective dashboards.
- Administrative module: This module will enable an admin to perform all administrative duties and access all information.
- Student module: This module will collect information from the graduate student which will then be used to calculate their success score. The students can also view the course ranking for guidance in deciding their choice of courses.
- Algorithm: This algorithm will calculate a score for each of the students based on their employment, salary etc. The records will then be grouped by each course. The courses will then be sorted based on the aggregate score of its students.

ALGORITHM FOR DEFINED MODEL

Graduate Student Score:

set startingscore =0;

company_score = 10*(number of companies- rank of company)
promotion_score = 100* (number of promotions)
designation_score = 10 * (total designations- rank of designation)
patent_score = 100* number of patents
unemployment_score = 50* number of months unemployed
field_relevance_score = 50* relevanceFactor
job_satisfaction = 50 * satisfaction_scale(1-10)
salary_score = (current_salary - starting_salary) / (location_factor * number_of_years * multiplication_factor)

graduatescore = startingscore + company_score + salary_score + promotion_score + designation_score + patent_score + unemployment_score + field_relevance_score +job_satisfaction;

Course Ranking:

Course score = (sum of all graduate scores in the course) / (number of graduates in the course * ((course_Rating + relevant_CourseRating + helpful_Course / 3));

courseRanking[] = sortDescending(courseList)

Faculty Ranking:

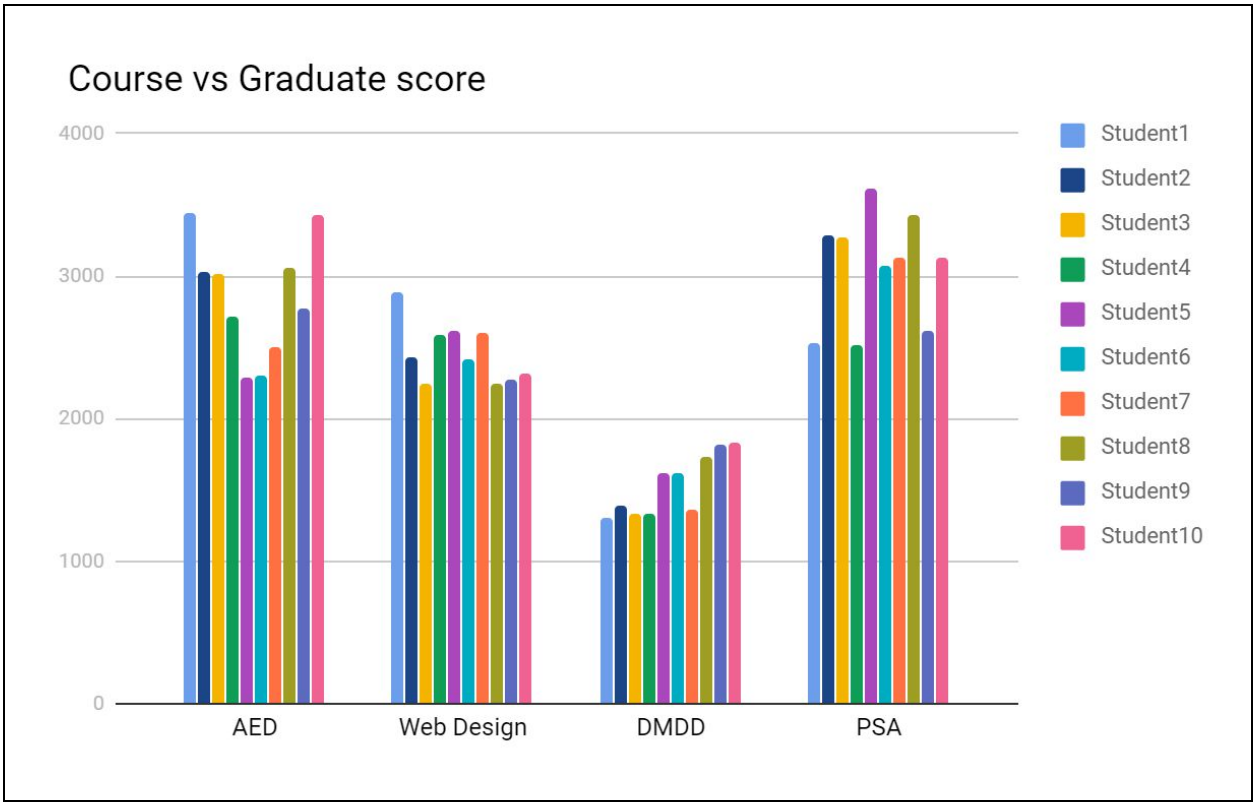
Faculty score = (sum of all graduate scores under the faculty) / (number of graduates under the faculty)

facultyRanking[] = sortDescending(facultyList)

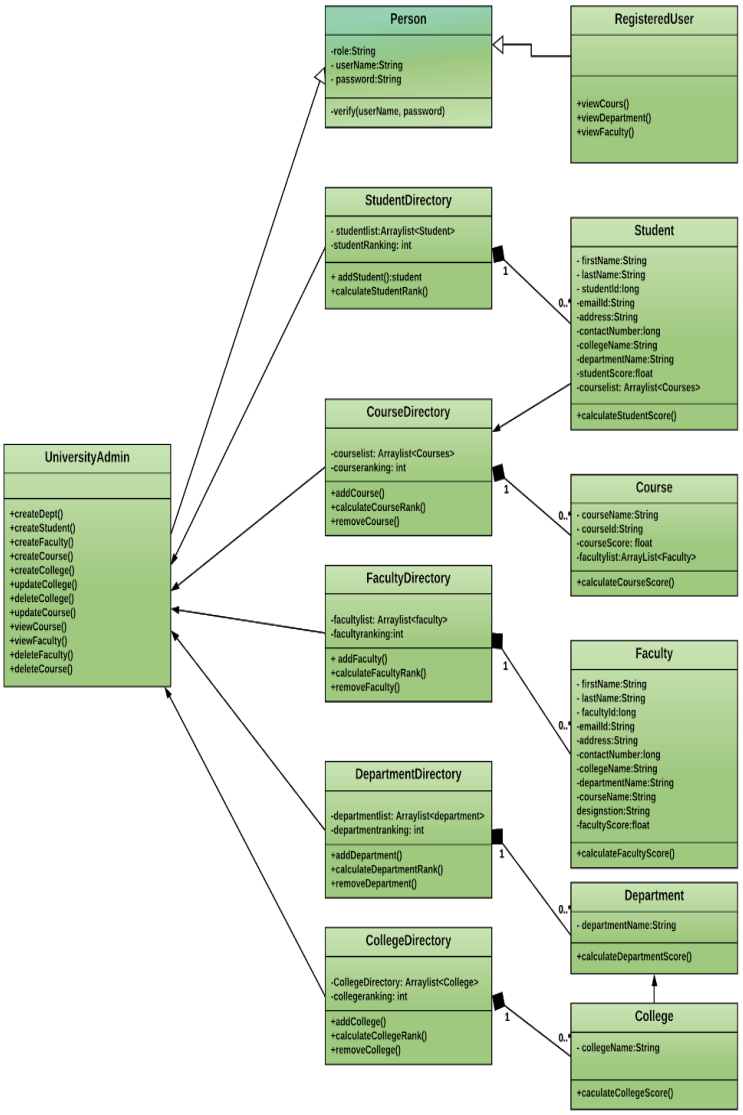
Student Score based on the feeback received:

Student	Company Score	Promotion Score	Designation Score	Patent Course	Unemployment Score	FRS	Job Score	Salary score	Graduate Score
ABC	1000	100	400	0	100	300	300	200	3200
DEF	500	500	250	200	300	400	350	400	3500
GHI	1000	100	400	0	100	300	300	200	3200
JKL	500	500	250	200	300	400	350	400	3500
MNO	1000	100	400	0	100	300	300	200	3200
PQR	500	500	250	200	300	400	350	400	3500
STU	1000	100	400	0	100	300	300	200	3200
VWX	500	500	250	200	300	400	350	400	3500

COURSE POPULARITY



UML OBJECT MODEL DIAGRAM

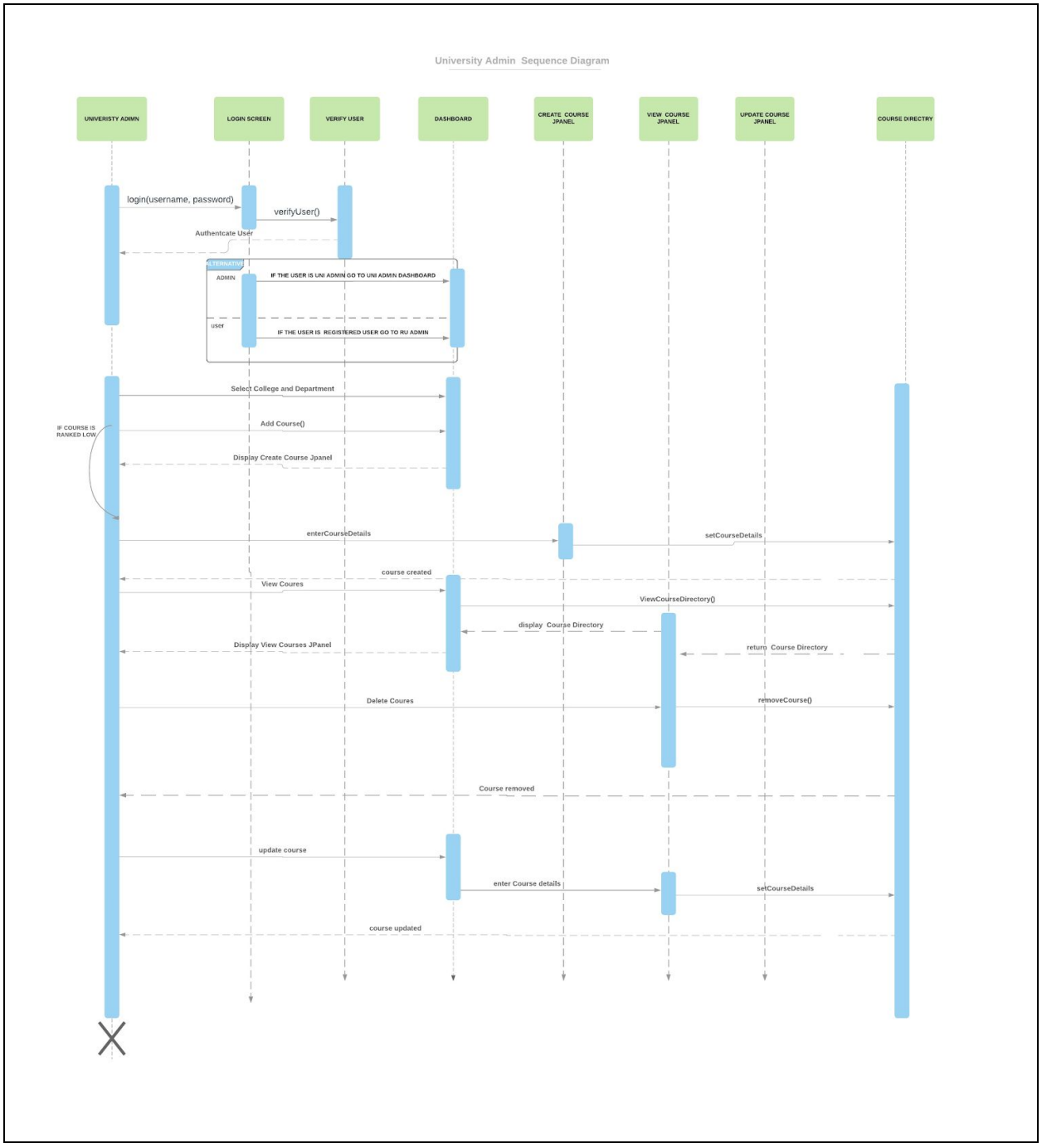


DEFINED METHODS AND ATTRIBUTES

Method/Attribute	Description
createStudent()	Admin function to create a new student in the database
createCourse()	Admin function to create a new course
createFaculty()	Admin function to create a new faculty
createDepartment()	Admin function to create a new department.
createCollege()	Admin function to create a new college.
viewCourse()	Admin and student function to view course details.
updateStudent()	Admin function to update student details.
updateCourse()	Admin function to update course details.
updateFaculty()	Admin function to update faculty details.
updateDepartment()	Admin function to update department details.
updateCollege()	Admin function to update student details.
deleteStudent()	Admin function to delete student detail.
deleteCourse()	Admin function to delete a course.
deleteFaculty()	Admin function to delete faculty detail.
deleteDepartment()	Admin function to delete a department.
deleteCollege()	Admin function to delete a college.
addStudent()	Add a student in student directory
addCourse()	Add a course in course directory
addFaculty()	Add a faculty in faculty directory
addDepartment()	Add a department in department directory
addCollege()	Add a college in college directory
calculateStudentRank()	To calculate student rank based on student score
calculateCourseRank()	To calculate course rank based on course score
removeCourse()	To remove a course from course directory.
calculateCourseCourse()	To calculate course score based on student feedback

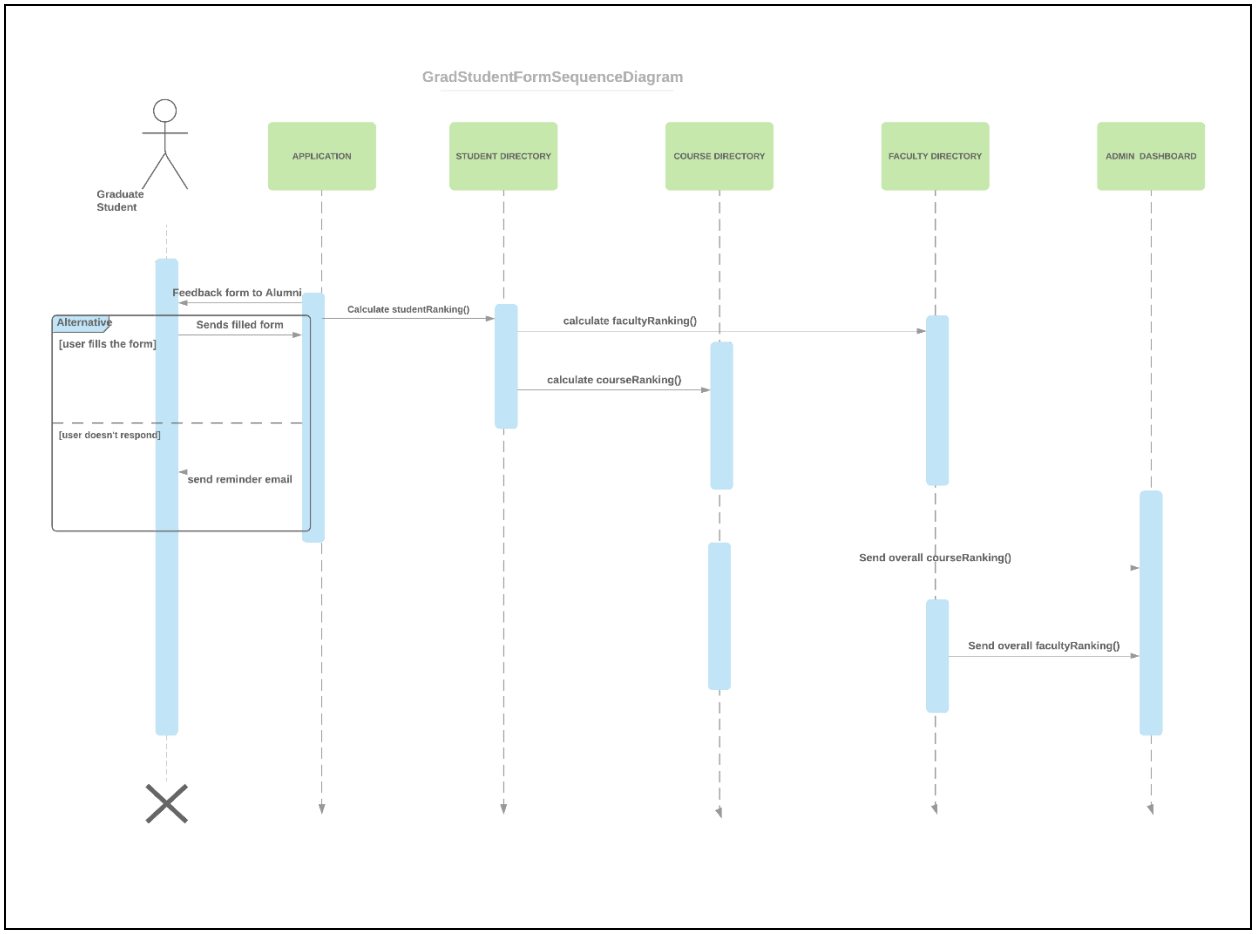
UML SEQUENCE DIAGRAM

The university Admin will be able to login to the Dashboard using the provided credentials. Once logged in, the admin will be able to perform all administrative operations such as view, create, update and delete of student, faculty or course records. The dashboard will also allow the admin to view the success score of each student, which will be used to populate and sort the Course Rank and Faculty Rank table which will then be displayed as a bar graph. Using this information, the admin can analyze and choose to drop a poorly performing course or faculty, or make other appropriate changes to improve the educational system.



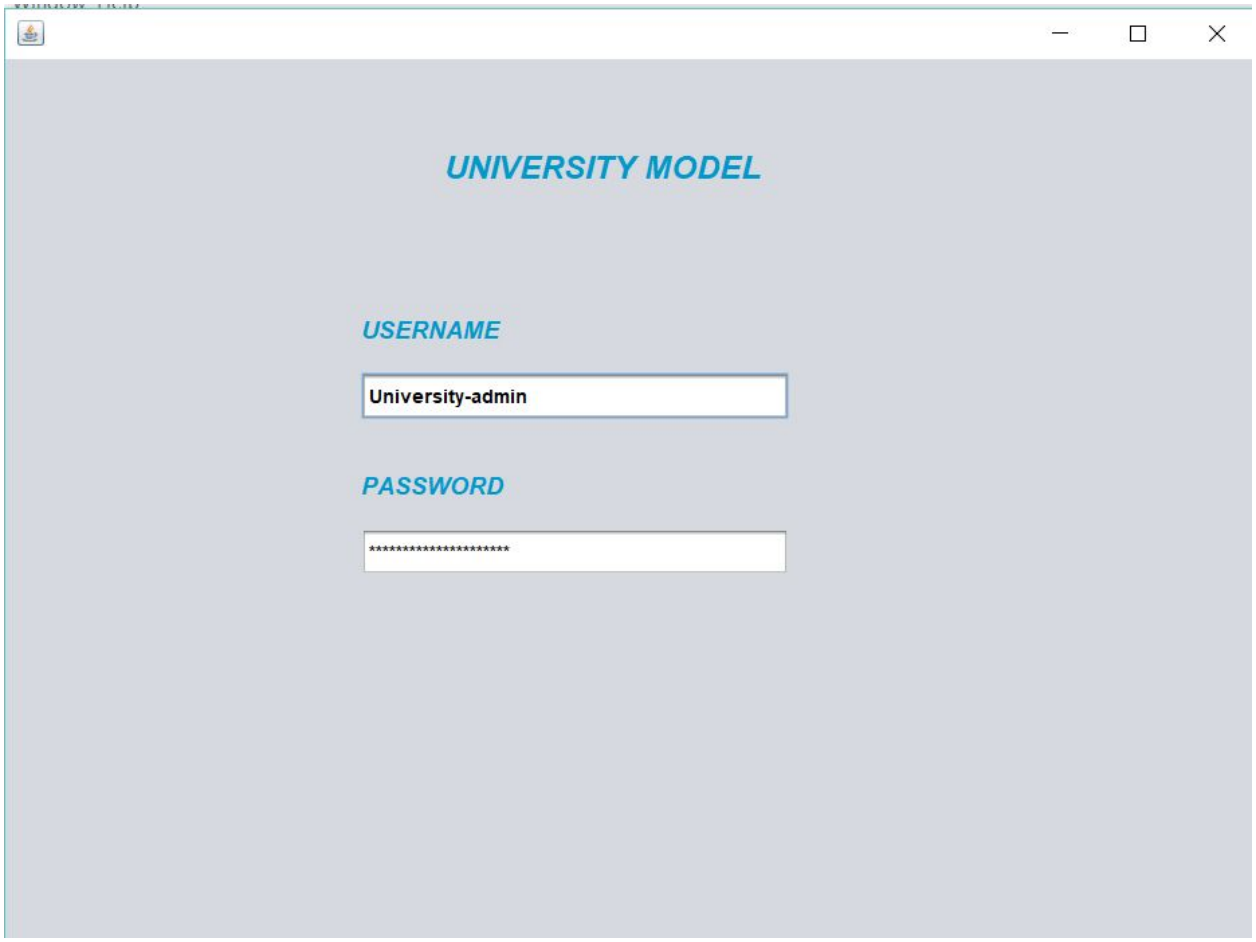
The Graduate student will be notified with an automated email which requests her to fill up a form with her personal information such as employment, salary, years of experience etc. The form will also allow the student to provide a 1-10 rating for the courses which she had taken based on the relevance of usefulness of that course to her current occupation. This details will then be stored in the student directory where the student score will be calculated using the defined algorithm.

This collection of student scores will then be grouped to calculate the course rank and the faculty rank using the respective directories, and it will be updated into the admin dashboard.

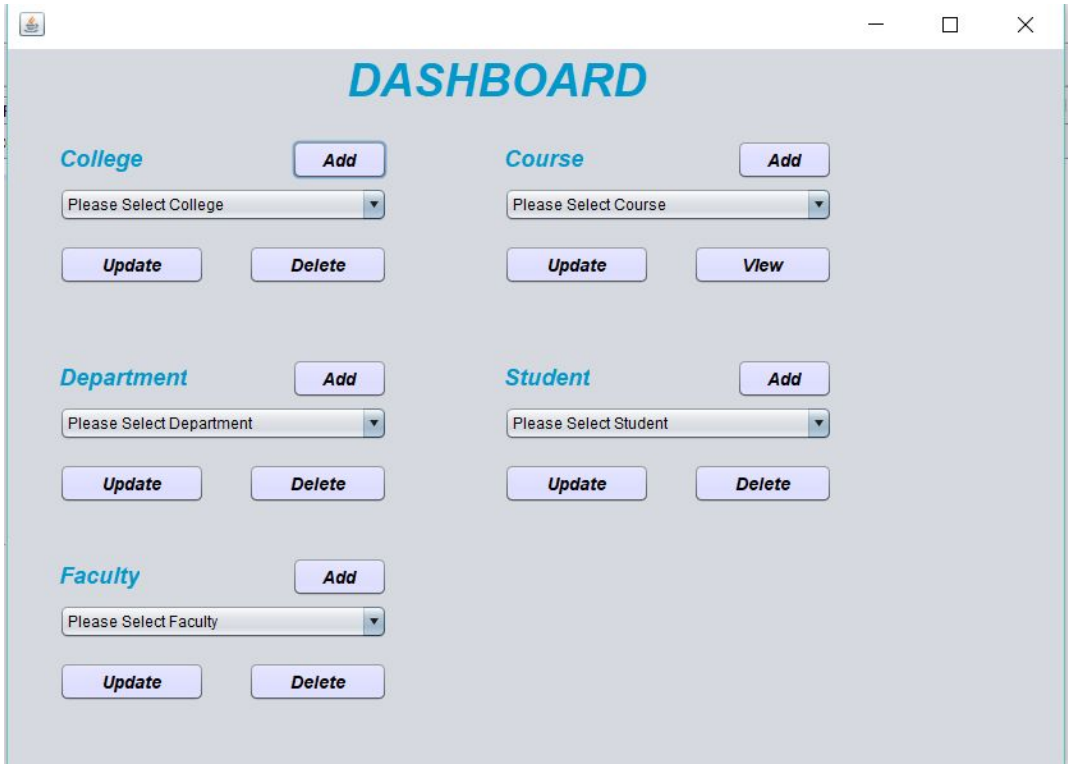


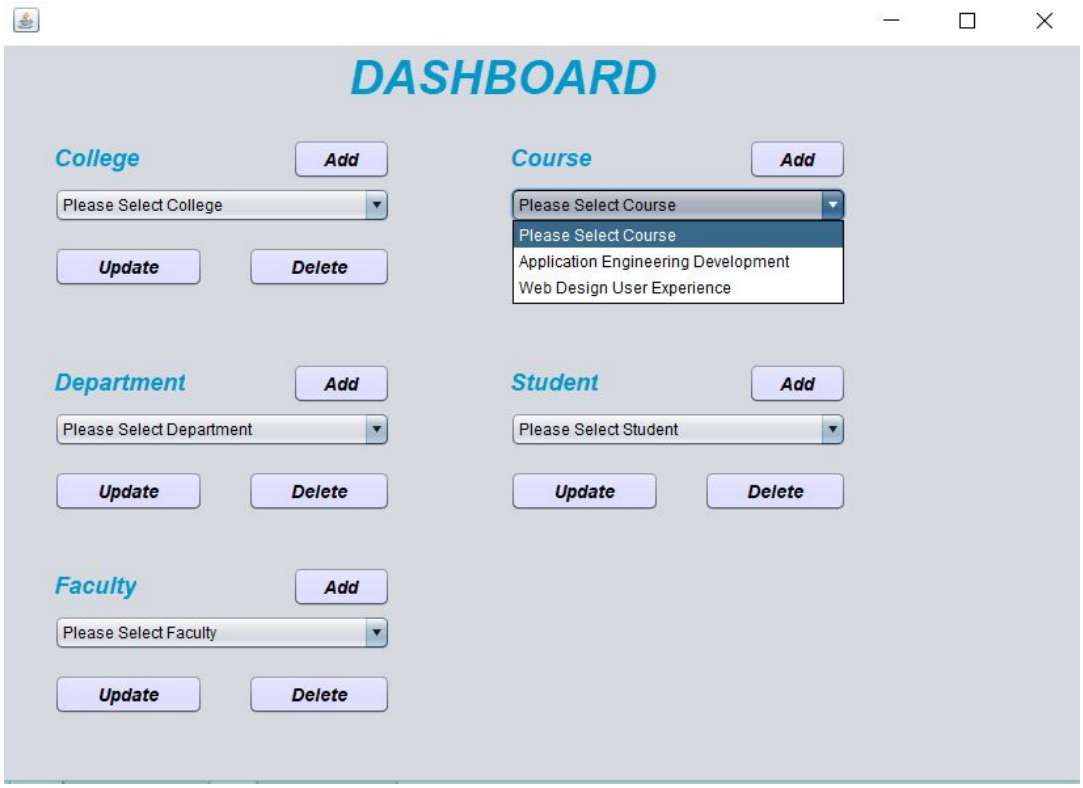
APPLICATION USER INTERFACE DESIGNS

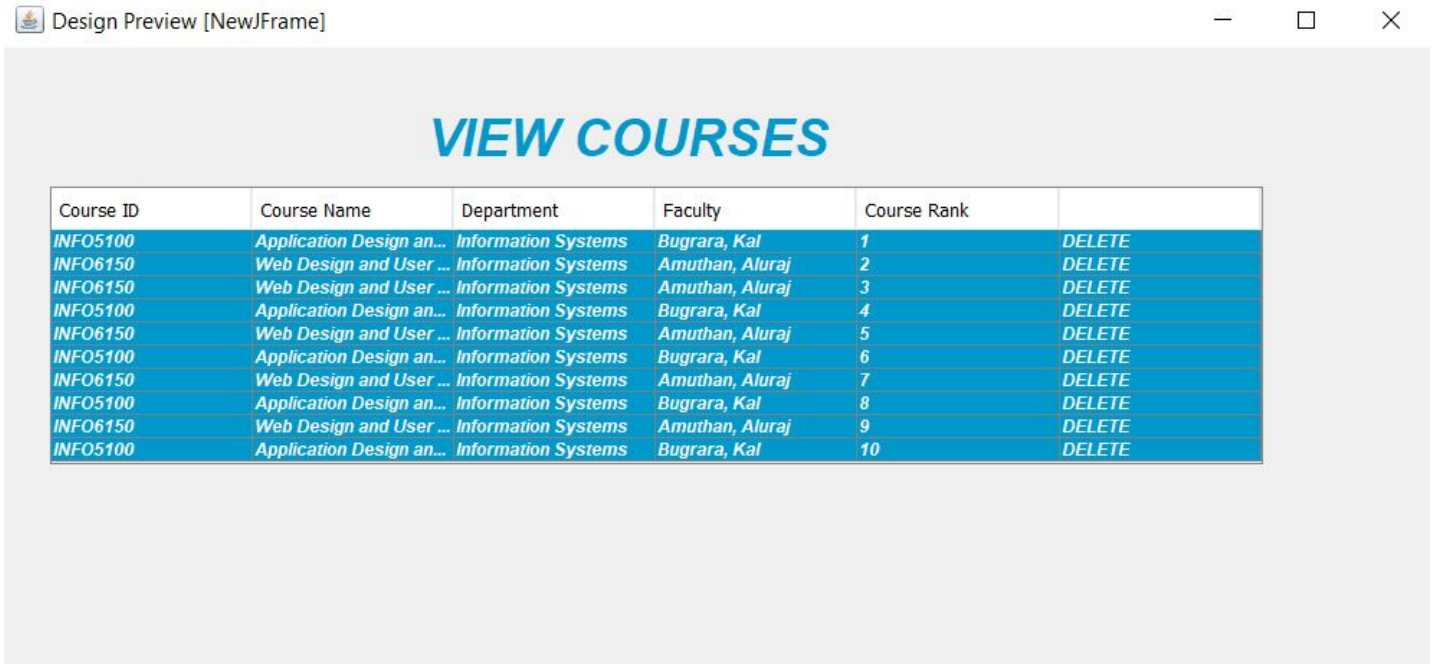
LOGIN SCREEN:



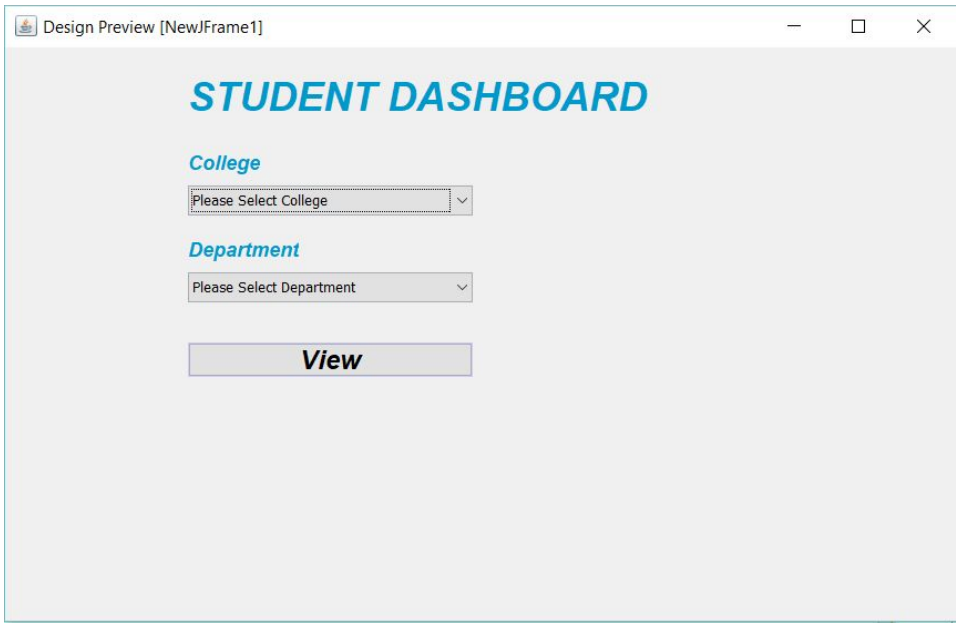
UNIVERSITY ADMIN DASHBOARD:

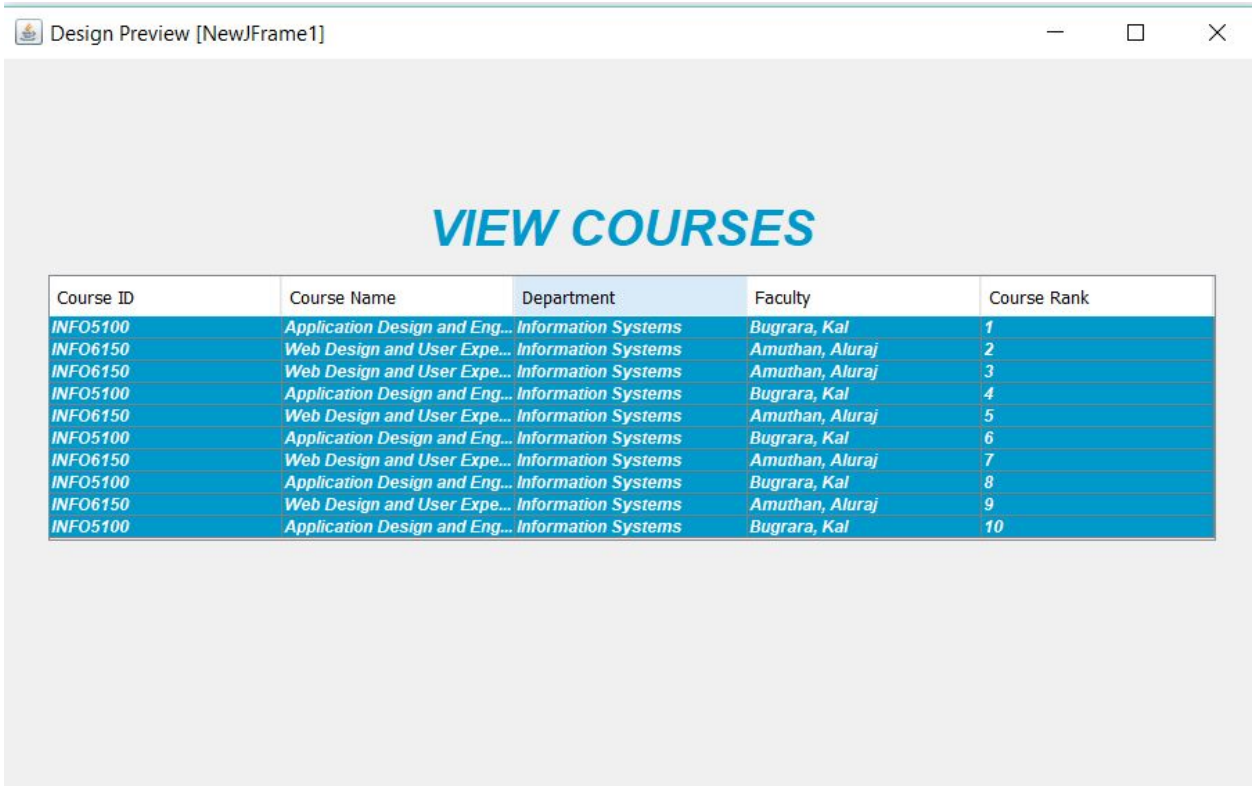

DASHBOARD
College
Add
Please Select College
Update
Delete
Course
Add
Please Select Course
Update
View
Department
Add
Please Select Department
Update
Delete
Student
Add
Please Select Student
Update
Delete
Faculty
Add
Please Select Faculty
Update
Delete


DASHBOARD
College
Add
Please Select College
Update
Delete
Course
Add
Please Select Course
Please Select Course
Application Engineering Development
Web Design User Experience
Department
Add
Please Select Department
Update
Delete
Student
Add
Please Select Student
Update
Delete
Faculty
Add
Please Select Faculty
Update
Delete



STUDENT DASHBOARD:





PROJECT BY

import teamName;

MANSI AGARWAL - 1356676

ANISH SINGH - 1388384

ASHWIN LAKSHMAN - 1353233