

University Model.



Abstract: To instill the techniques for turning an object model into a machine for information gathering and data aggregation. We want to use software engineering techniques to improve the quality of education anywhere and hold people accountable for improving the quality of life through education, learning to learn, and feedback. Your task is to study ways to create a performance measurement solution to enable universities to measure the quality of the education they deliver to their students. The approach will be to investigate how an educational system in terms of faculty and courses contribute to the growth of their graduates over a 5-year period. You must figure out ways to track the jobs and promotions graduates get over time and assign rankings accordingly. In addition, track the connection of courses and their relevance to graduates' growth.

Contents:

1. Detail Document
2. Sequence Diagram
3. Class Diagram

Group Members:

1. Yash Jain,
2. Parth Shah, 001006181

DEFINITIONS

In this section we will go through some synonyms and jargons used in this document.

) **University Model:** A university model is a system design which will help a university to build its own system to track various KPI'S for a successful and life changing aspect for their students and staff members.

) **Object Model:** An object model is a logical interface, software or system that is modeled using object-oriented techniques. It enables the creation of an architectural software or system model prior to development or programming.

More Info: https://en.wikipedia.org/wiki/Object_model

) **Sequence Diagram:** A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

More info: <https://www.lucidchart.com/pages/uml-sequence-diagram>

) **Class Diagram:** Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

More Info: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

University Model.

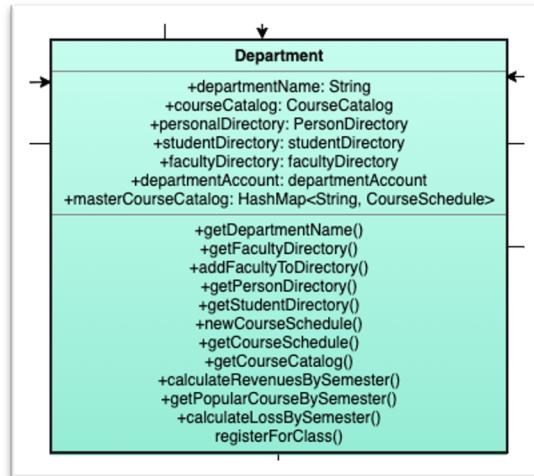
Report Outlines Proposed Model

University Model represents a dynamic structure for handling multiple user objects to handle university functionalities. We here propose a unique, simple model which will help universities to handle their operations from managing admissions, to taking feedbacks from their student employers in an easy and manageable step.

Main Objects:

- Department
- Student
- Transcripts
- Seat Assignments
- Course Offer
- Employer
- Faculty
- Course

Department:



Department is the main root of the class diagram where a department can have access to other modules.

Class Department have various methods which can gain knowledge from different classes.

Objective of Class Department: Department is responsible for handling all the 3 Modules and should have data related to students, Faculty, Employer. It should have a capability of gaining feedbacks on how to improve their system

Methods:

- 1) **addFacultyDirectory():** Department has the capability of adding faculty to the list to faculty directory. This helps in maintaining and keeping the track of all faculties
- 2) **newCourseSchedule():** If any new courses needed to be added or update any existing courses then department can do that with the help of these method.

University Model.



- 3) **calculaterevenuesBySemester()**: On the business side department also has the capability of calculating revenues and keeping the track of their progress over time.
- 4) **Addstudentdirectory()** : Department is also responsible for giving admits and accepting students to their department which can be done by this method.
- 5) **Addjob()**: Department can add jobs to which will help students to gain more knowledge.
- 6) **addscholarships()**: list of scholarships which students can have an access to if they meet the merit.

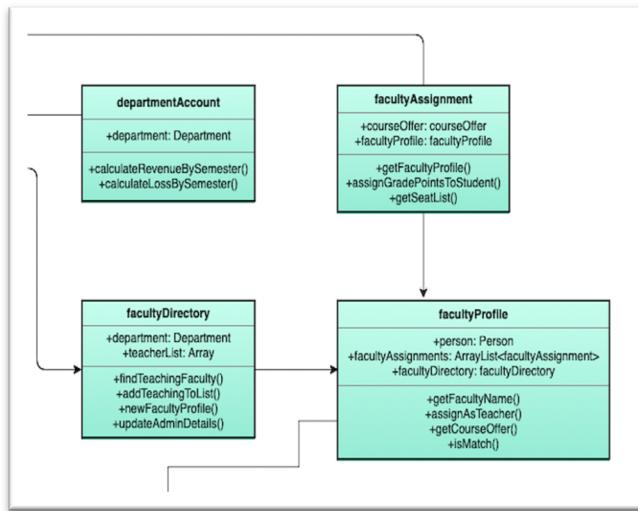
Note: There are more methods to the department which performs various functionalities which can be accessed by the class diagram.

Associations:

- 1) Department → Faculty (One to Many)
- 2) Department → Students (One to Many)
- 3) Department → College (One to One)
- 4) Department → University (One to One)
- 5) Department → Employers (One to Many)

Faculty:

Faculties are an important part of the university and having a good faculty member in the department is directly proportional to improving knowledge in students.



Faculty is a big module which we have divided in more granular classes.

Class Facultyprofile: This class handles all the information about faculties which can be accessed by department/students

Class Facultyassignment: Faculties should be able to manage their sessions with students and able to improve students' academic by keeping the track of their student grades.

- 1) **assignGradePointstoStudens()**: This method helps in assigning the grades to students by the faculty member.
- 2) **Givefeedbackstudent()**: It is very important for the university to have a feedbacks from the faculty members about students and to keep track of the trajectory in which students are going.

Class Facultydirectory: Contains all the details of faculties which are in the department. It consists of methods to get teaching faculties, add faculties, update faculties details.

University Model.

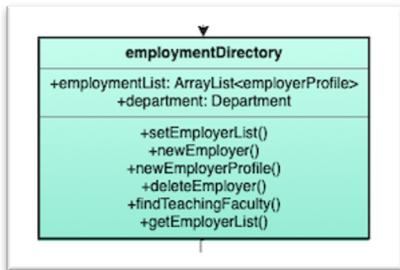


Associations:

- 1) Facultyprofile → Facultydirectory (One to One)
- 2) Facultyprofile → Department (Many to Many)
- 3) Facultyassignment → Facultyprofile (One to Many)

Employment Directory

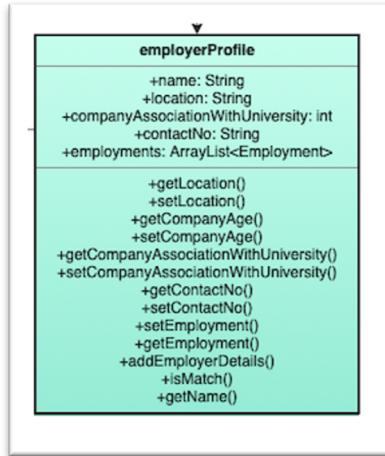
The Employer Directory is a list of the Companies that the students get hired to from each department.



To be an successful university and change lives of students, university should be able to get jobs to their students. University needs to have a list on employers which is managed by this methods.

- 1) **newEmployerProfile()** : If a student is hired by an employer which is not in the current list of employer this method can help in adding that.
- 2) **Findteachingfaculty()**: Often students gets hired depending on the subjects which they take in during their academics and how they perform in that subject. This method helps in keeping track if the teachers which are teaching great subjects.

Employer Profile



There are various domains out there where a student can be recruited, and a diverse portfolio of a university is a big yes for its success. So, we should know everything about the employer so that we can generate more analysis on it.

We can have various methods inside the employer profile which helps us in more details-oriented analysis in future.

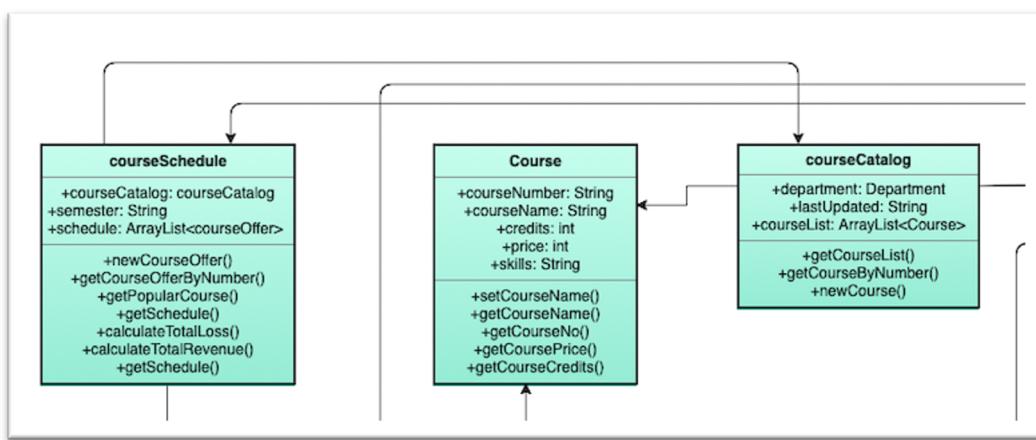
Associations:

- 1) **Employer Profile** → **Employment Directory**

University Model.



Courses:



In any university selection/patterns of courses are driving forces to get students enroll in their program. Without industry standard courses and recent trending topics no student will be attracted to the university. We have 3 Classes which can manage courses and its attributed in our system.

CourseCatalog: Course catalog class helps in keeping the track of all courses which are currently thought in the college it answers to some of the following questions:

- 1) How many students are taking that course?
- 2) How is that course relevant to current industry standards
- 3) How is that course different from other courses
- 4) Which faculty is taking that course?

Some of the methods which are included in the courseCatlog are

- 1) Newcourse(): This method can help in assigning new courses to the catalog.

CourseSchedule: The Course Schedule consists of all the courses offered, list of the schedules of that particular course in that particular semester. It consists of methods to course number in the Course Offering list and the revenue generated by the course in that semester.

Associations:

CourseSchedule → CourseOfferings (One to one)

Course: The Department of each college lists all the courses that are offered irrespective of semester.

The Course Consists of a name, it's number (Course Number), the credits given upon completion of the course and it's price

Associations:

CourseCatlog: The Course Load is the list of Courses that the student has taken in each semester. It consists of the semester number that the student has taken in the particular semester. It also consists of the Seat Assignment list.

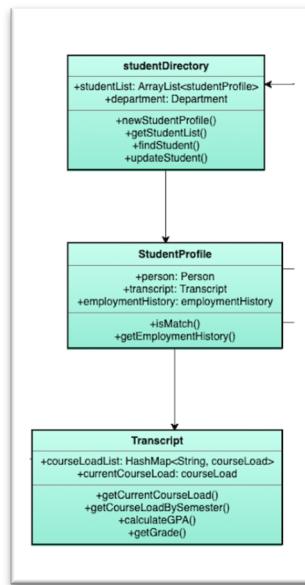
Associations:

CourseCatlog → SeatAssignment

University Model.



Class Student:



In any university the success factor is majorly derived from the students off that university. Student modules needs to be in-depth and a 2-way module where students also have a chance to give feedbacks to universities and give suggestions.

studentDirectory: The Student Directory is the list of students from a particular department. It contains the method to find a particular student from a particular department, it adds the student to the directory

Associations:

StudentDirectory → StudentProfile

Studentprofile: The Student profile contains the details of the student such as his transcript and employment history. It contains the methods to get/add the course load by semester, get the course list and fetching the transcript of the student

Transcript: The transcript is mapped to the concerned unique student profile and course list. It contains methods to the current course load by the student, total score and to get the course load based on the semester.

Associations:

Transcript → Transcript class.

Employment: The Employment contains attributes of the student such as the course load taken,it's weightage, quality of the job and the leakage between the current job and his/her next job.

Associations:

Employment → CourseOffer class

Employment → EmploymentHistory class.

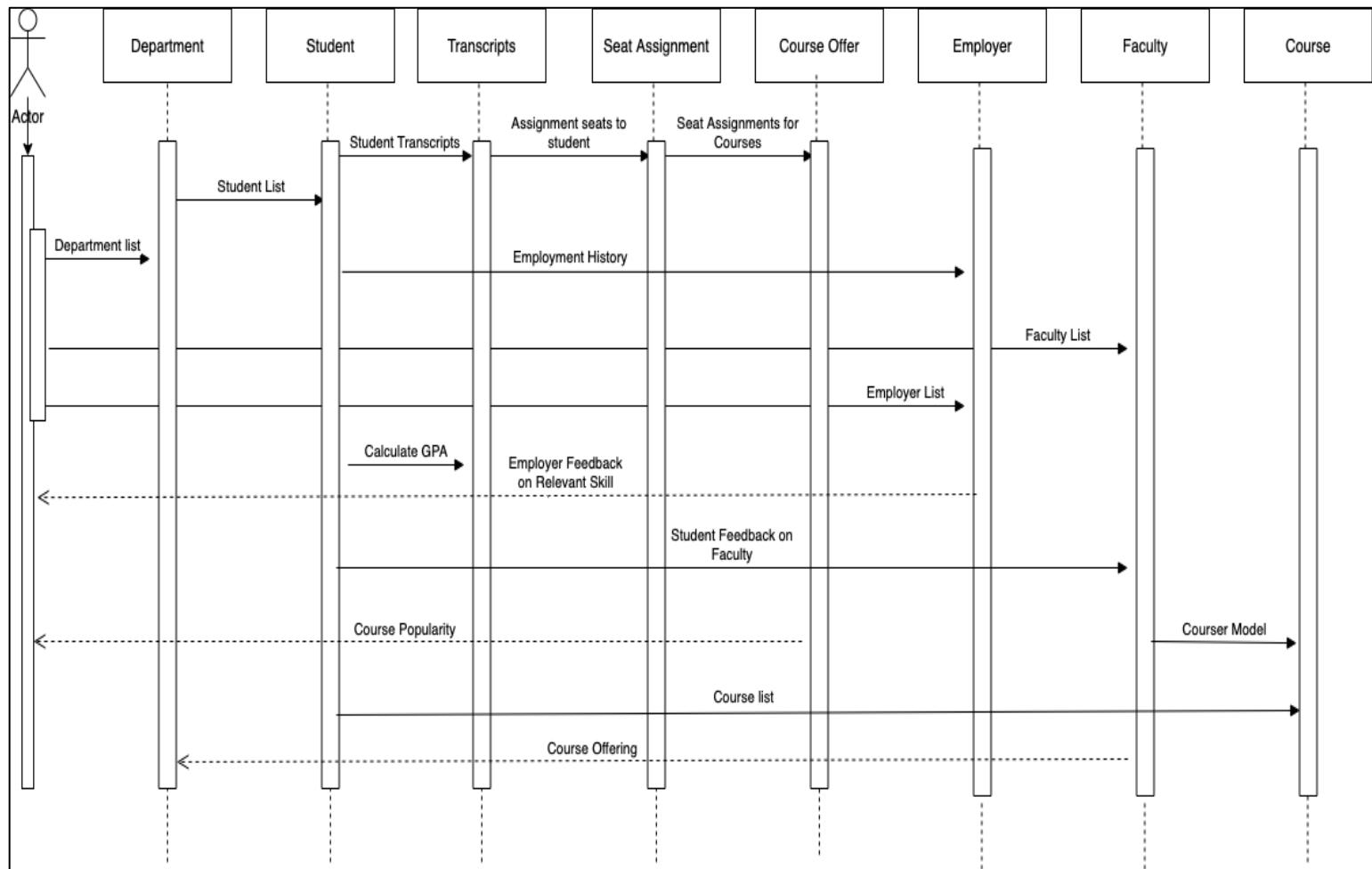
Employment History: The Employment History contains the list of companies that the student has been associated with

Associations:

Employment History → Employment Class.

University Model.

Sequence Diagram:



The sequence diagram helps us to understand what all steps are been followed when we traverse through the university model.
Like

- 1) Department can get student list from Class Student. To get this information Class Department needs to use the methods which are initialized in the class.
- 2) All the arrows are methods which can be used by the classes to gain some amount of information.
- 3) There are also reverse arrows which can be used for feedbacks.

This diagram can act like a pseudo code for the university Model.

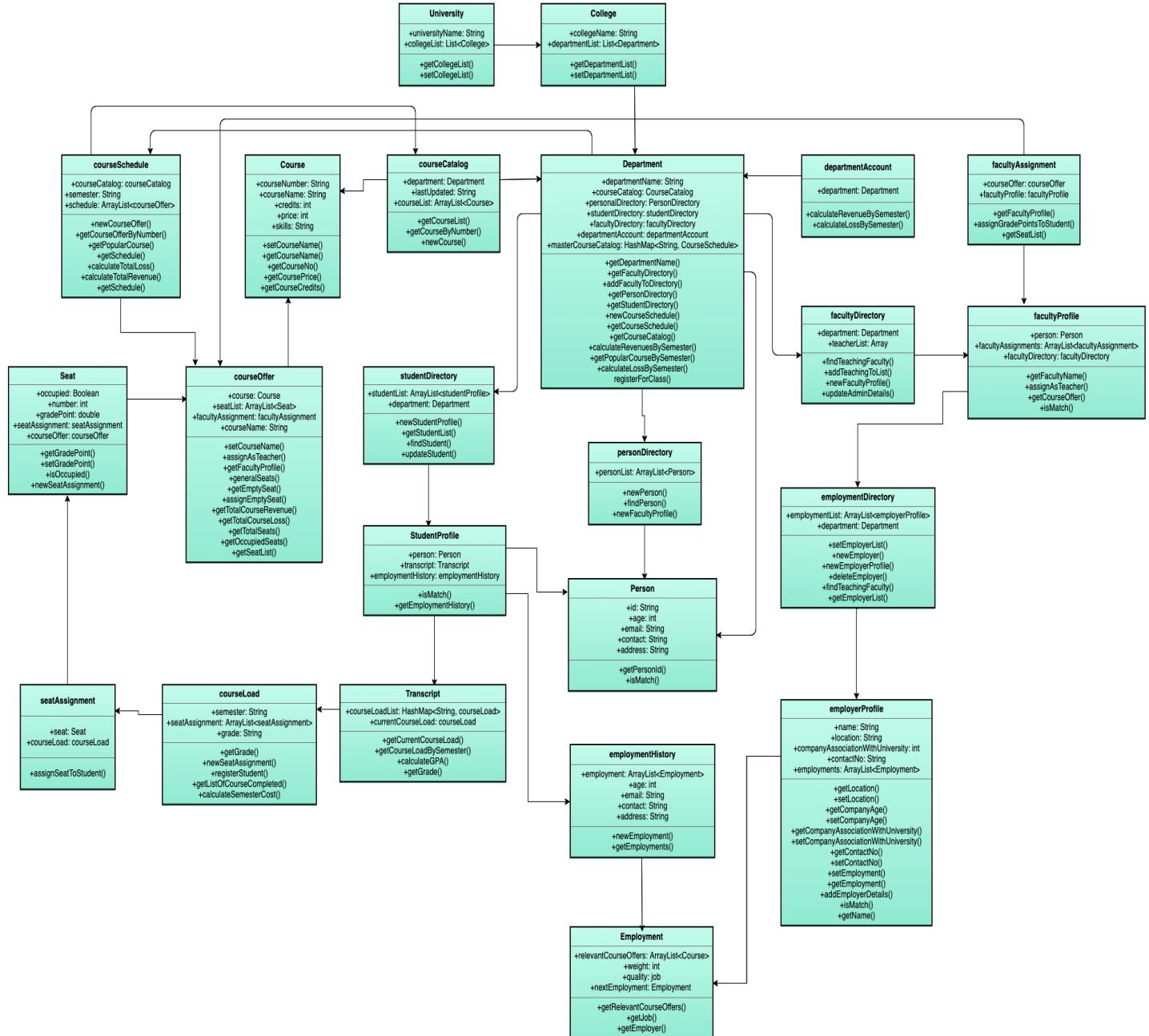
University Model.



Class Diagram

Visual Paradigm Online Free Edition

University Model Class Diagram



University Model.



User Interface Screenshots:

Admin Login View:

The Admin Login View shows a dark-themed interface with a top navigation bar featuring the Huskies logo and "STUDENT MANAGEMENT SYSTEM". Below the navigation is a login form with tabs for "Student", "Admin" (which is selected), and "Employer". The login form includes fields for "Username" (with placeholder "User") and "Password" (with placeholder "*****"). A teal "Login" button is at the bottom.

Admin Menu View:

The Admin Menu View displays a sidebar menu on the left with the following items: Admin, Profiles, Students, Faculty, Employer, Revenue, Department, Analysis, and Logout. The main area is currently empty.

View Update Admin:

The View Update Admin screen shows a "View / Update Profile" section. It contains fields for Admin Name (admin), Age (40.0), Email ID (admin@gmail.com), Contact Number (1233455678), and Address (Boston). At the bottom is a "Update Admin" button. The sidebar menu on the left includes Admin, Profiles, Students, Faculty, Employer, Revenue, Department, Analysis, and Logout.

University Model.



Add new Faculty View:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Add New Faculty

Faculty Name	<input type="text"/>
Age	<input type="text"/>
Email ID	<input type="text"/>
Contact Number	<input type="text"/>
Address	<input type="text"/>
Department	<input type="text" value="--None--"/>

Add Faculty

View Faculties

Name	Age	Email ID	Contact ...	Address	Depart...

Add new employer View:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Add an employer

Employer Name	<input type="text"/>
Contact No	<input type="text"/>
Associated with university	<input type="text"/>
Established since (in years)	<input type="text"/>
Address	<input type="text"/>

Add Employer

Delete an employer

Enter Employer Name

Delete Employer

View employers

Name	Contact	Established	Address	Associat...
Amazon	123-45...	5	Boston	4
Google	123-45...	10	California	5
facebook	123-45...	15	NewYork	13
Tesla	123-45...	20	Boston	2
ABC	123-45...	20	Boston	2
PFF	123-45...	30	Boston	2

Add new Student View:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Add New Student

Student Name	<input type="text"/>
Age	<input type="text"/>
Email ID	<input type="text"/>
Contact Number	<input type="text"/>
Address	<input type="text"/>
Department	<input type="text" value="--None--"/>

Add Student

View Students

Name	Age	Email ID	Contact ...	Address	Depart...

University Model.



Add Department View:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Add an department

Department Name	<input type="text"/>
Department Head	<input type="text"/>
Description	<input type="text"/>
Established since (in years)	<input type="text"/>

Add Department.

Delete a department

Enter Department Name

Delete Department.

View departments

Name	Department...	Established...	Description
Information...	Test 1	Desc 1	5.0
Computer S...	Test 2	Desc 2	5.0
Artificial Int...	Test 3	Desc 3	13.0
CSN	Test 4	Desc 4	4.0
Data Science	Test 5	Desc 5	3.0
Big Informati...	Test 6	Desc 6	A.0

Register for Course View:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Register a course

Semester:

Department:

Search **Clear**

Available Courses:

Add a course

View Course

STUDENT MANAGEMENT SYSTEM

HUSKIES

Courses

Course	Lecturer
Engineering of Big-Data Systems	Professor H
Business Analysis and Information Engin...	Professor M

University Model.



View hiring roles:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Percentage of hiring for different roles

No of Student Applications	Neu Students who cleared all rounds
Full Stack Developer	3
Data Analyst/Engineer	3
ML/ Data Science Engineer	3

Submit

SDE 100.0%
ML 100.0%
DA 100.0%

Student with none role were hired the most

Most Popular Course

The most popular course based on the percentage of students who have taken up this is app dev

Fill Employment Details:

STUDENT MANAGEMENT SYSTEM

HUSKIES

Employment Details

Employer Name :	test
Employment Type :	test
Job Title :	test
<input checked="" type="checkbox"/> I am currently working in this role	
Start Date :	10/24/2021
End Date :	Present

Submit

View Employment History

STUDENT MANAGEMENT SYSTEM

HUSKIES

My Employment History

EMPLOYER	JOB PROFILE	JOB TYPE	START DATE	END DATE
test	test	test	10/24/2021	Present

University Model.



View Grades:

The screenshot shows the 'Grades' section of the system. On the left is a vertical navigation bar with icons for Student, Profile, Courses, Register, Feedback, Grades, Job, Employment, and Logout. The main area has a dark header with 'HUSKIES' and a red 'N' logo. Below it is a 'STUDENT MANAGEMENT SYSTEM' title. The 'Grades' section contains a table with three rows:

COURSE	GRADE
app eng	F
app dev	F
app eng	F

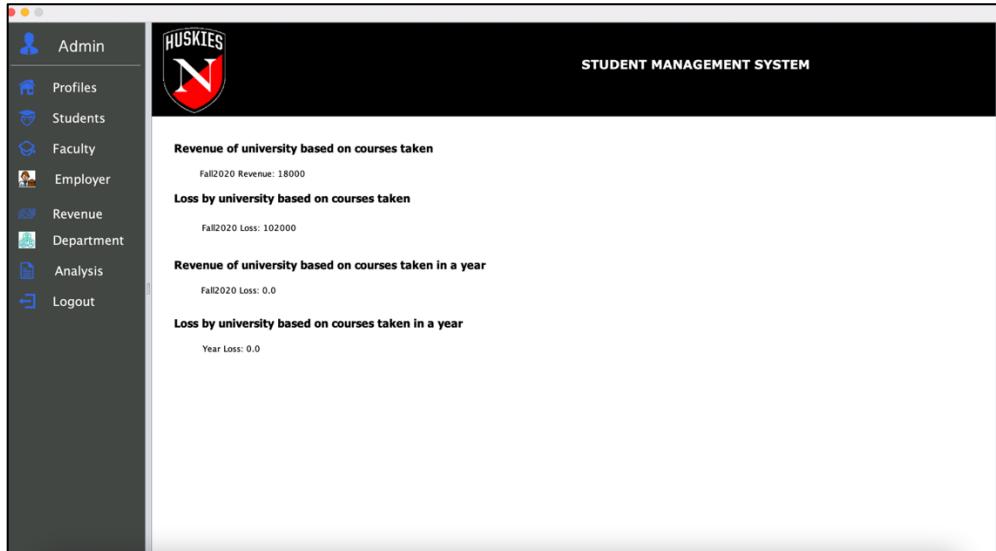
Below the table, it says 'Total GPA: 0.0'.

View/Update Student Profile

The screenshot shows the 'View / Update Profile' page. The left navigation bar is identical to the previous screenshot. The main area has a dark header with 'HUSKIES' and a red 'N' logo. Below it is a 'STUDENT MANAGEMENT SYSTEM' title. The 'View / Update Profile' section contains fields for Student Name (0112303), Age (26.0), Email ID (yash@gmail.com), Contact Number (123-345), and Address (Boston). At the bottom is a 'Update Student' button.

University Model.

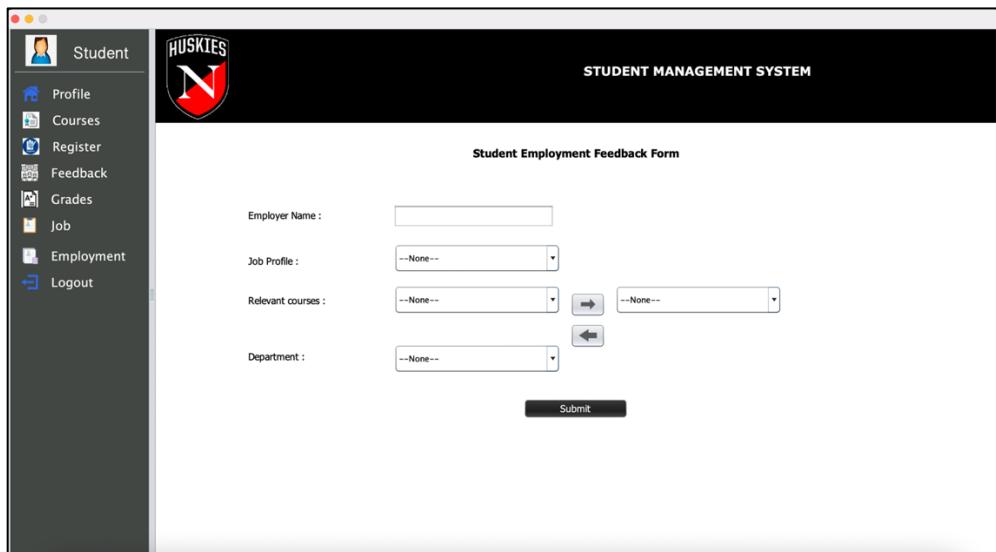
View Revenue by Semester:



The screenshot shows a window titled "STUDENT MANAGEMENT SYSTEM". On the left is a sidebar with icons for Admin, Profiles, Students, Faculty, Employer, Revenue, Department, Analysis, and Logout. The main area displays four sections of data:

- Revenue of university based on courses taken**
Fall2020 Revenue: 18000
- Loss by university based on courses taken**
Fall2020 Loss: 102000
- Revenue of university based on courses taken in a year**
Fall2020 Loss: 0.0
- Loss by university based on courses taken in a year**
Year Loss: 0.0

Student feedback form:

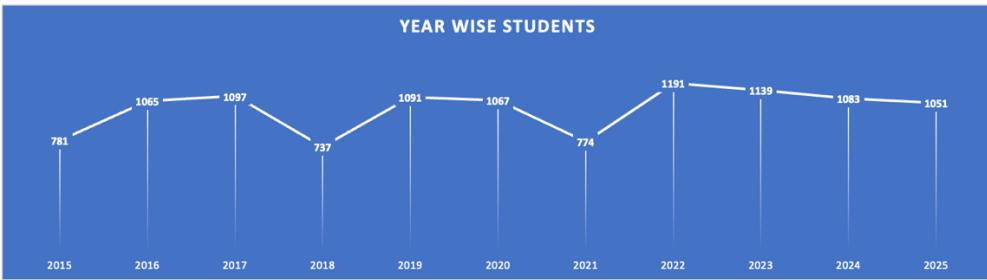
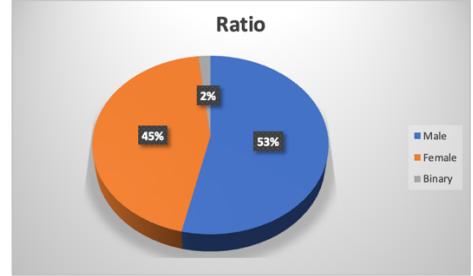
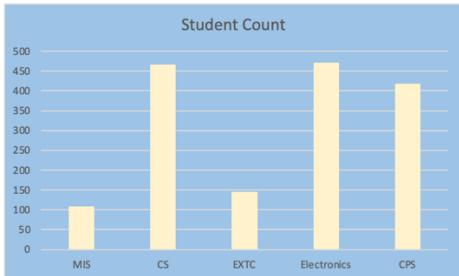


The screenshot shows a window titled "STUDENT MANAGEMENT SYSTEM". On the left is a sidebar with icons for Student, Profile, Courses, Register, Feedback, Grades, Job, Employment, and Logout. The main area is titled "Student Employment Feedback Form" and contains the following fields:

- Employer Name :
- Job Profile : --None--
- Relevant courses : --None-- --None--
- Department : --None--

A "Submit" button is located at the bottom.

University Model.



Conclusion: In this document we saw a detail version of how to build an university model with class, Sequence diagrams and added some UI Examples.