

AED ASSIGNMENT 3

PROFESSOR AS A SERVICE

Sai Supraja Kosuri – 002191644

Pranaya Chilamkuri – 002924536

Ruchita Yarlagadda – 002920198

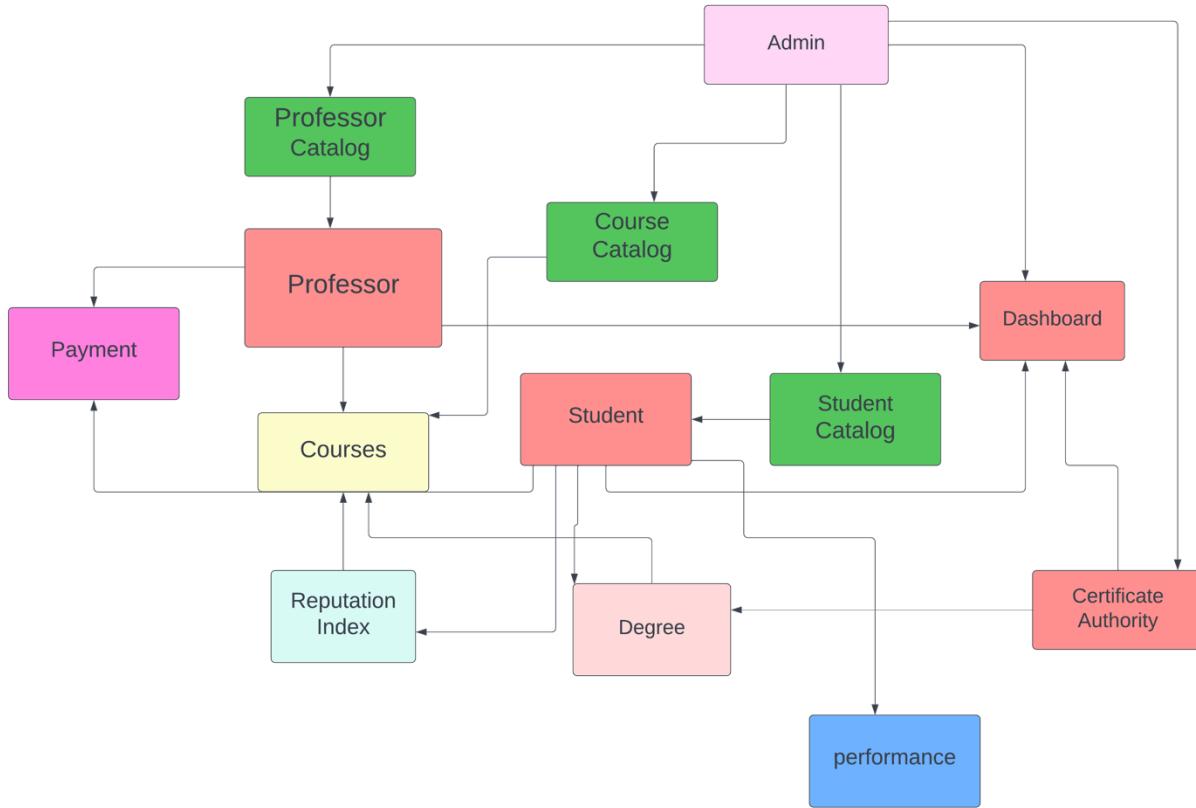
Objective:

To benefit the students from all around the world with high-quality education that is built using software engineering principles to reduce education costs while improving education quality. People are held accountable for their acts in this way, and pupils' quality is improved through instruction and feedback. We proposed a method that gathers data and generates a reputation index based on the input received. This proposed methodology also allows for the tracking of students' progress and the delivery of an appropriate degree from a third-party authority.

PROPOSED SOLUTION:

When it comes to course selection, the traditional university approach is rigid. The Professor as a Service approach provides students with the ability to select courses from a wide number of options and view them at their leisure. Professors will appreciate this model since it allows them to determine their own course costs and educate from a remote location. Students, professors, and the certifying authority are all monitored by Admin under this approach. When student's complete courses taught by instructors, the certification authority is responsible for awarding degrees to them. Students can use a reputation index to help them choose courses because it is based on student input. Students can see this rating when they register for a course. As a result, this model outperforms the traditional model.

ARCHITECTURE DIAGRAM FOR PROFESSOR AS A SERVICE:



The above diagram represents the architectural model for professor as a service, it is different from the traditional university model explained as follows:

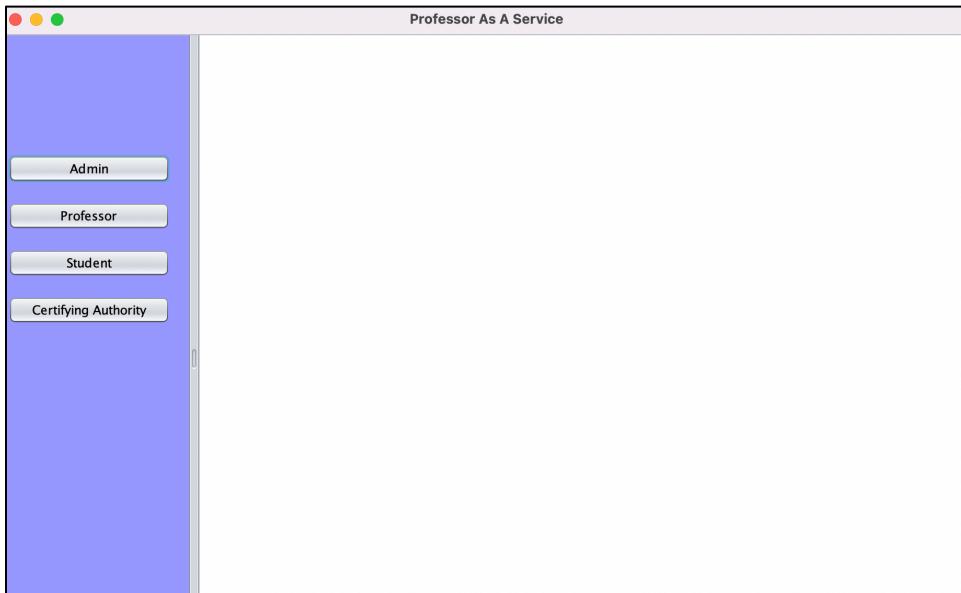
- The traditional university model mainly depends on the college and department classes whereas the new model is dependent on the “Professor” class. The Professor is the center of students’ learning instead of the university.
- In the traditional university model, we must keep track of classrooms assigned, classroom seating, and other infrastructure required to conduct a class by a professor/faculty. However, in the Professor as a Service model, the cost overhead can be cut off and the professors can conduct the class whenever and from wherever they want.
- In the traditional university model, the payment system is managed by the college or department entity but in the new model the payment made by the student directly goes to the professor and the professor must pay subscription fees for using the Portal.

- In the traditional university model, the respective department or the university awards the degree but in the new model degree is awarded by a certificate authority (Third-party certifiers)
- In the traditional university model, there is no concept called course reputation index but in the new model, there is a concept called reputation index. The reputation index consists of metrics collected from students based on the quality, of course, job opportunities increased and course. In the traditional university model, there is no feedback loop for the professor to improve the course quality but in the new system reputation index plays a key role to provide insights for both professor and the student about how a course is being conducted. It is helpful for the incoming students to select the best course for their stream.

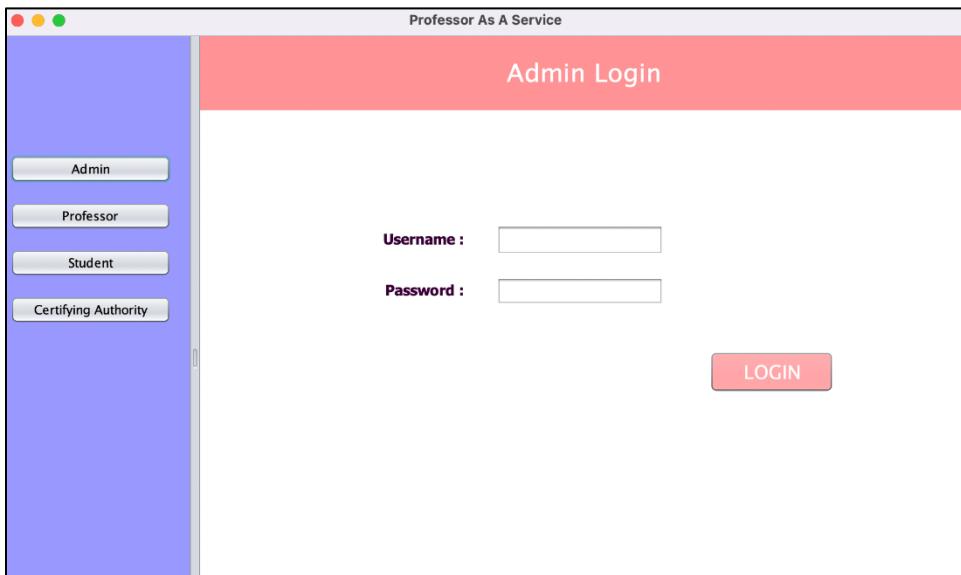
SCREEN DESIGNS

Professor_As_A_Service Dashboard

Initial Dashboard when application is launched.



Admin Login Dashboard: Username and Password to be given to login.



Browsing screen: Different catalogues available to browse in the amin view.



Professor Catalog

The screenshot shows a desktop application window titled "Professor As A Service". The main area is labeled "Professor Catalog" and contains the heading "Professor Details :". Below this, there is a table displaying professor information:

Professor Id	Professor Name	Professor Email
002924	Kalyan P	p.kalyan@neu.edu
003476	Ankith M	m.ankith@neu.edu
004842	Pooja S	s.pooja@neu.edu

At the bottom of the table, there are three red rectangular buttons: "VIEW PROFESSOR", "UPDATE PROFESSOR", and "DELETE PROFESSOR". On the left side, there is a vertical sidebar with four buttons: "Admin", "Professor", "Student", and "Certifying Authority".

Student Catalog

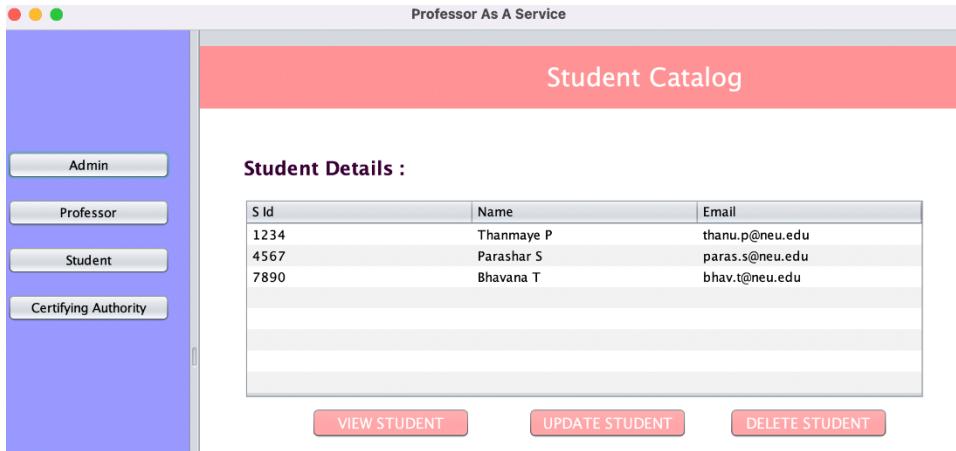
Professor As A Service

Student Catalog

Student Details :

S Id	Name	Email
1234	Thanmaye P	thanu.p@neu.edu
4567	Parashar S	paras.s@neu.edu
7890	Bhavana T	bhav.t@neu.edu

VIEW STUDENT **UPDATE STUDENT** **DELETE STUDENT**



Course Catalog

Professor As A Service

Course Catalog

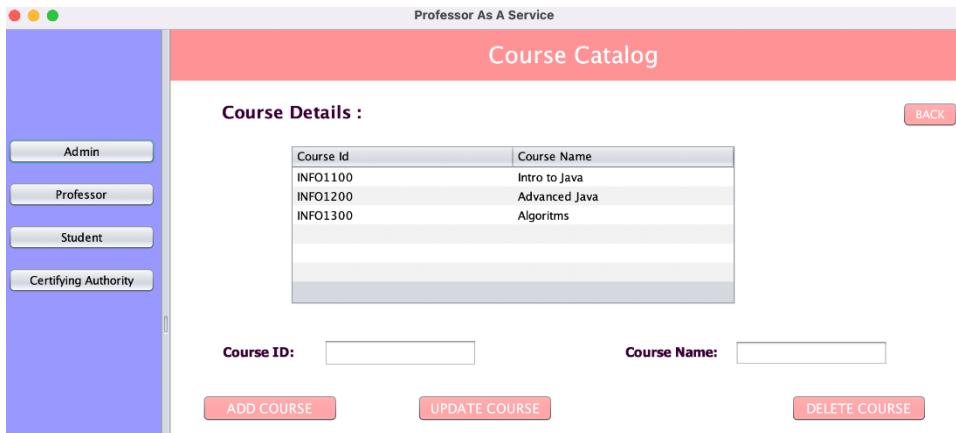
Course Details :

Course Id	Course Name
INFO1100	Intro to Java
INFO1200	Advanced Java
INFO1300	Algorithms

BACK

Course ID: **Course Name:**

ADD COURSE **UPDATE COURSE** **DELETE COURSE**



Certification Authority

Professor As A Service

Certification Authority

Certifying Authority Details :

Authority Name
ComptIA
Pearson VUE
Prometric
Kryterion

BACK

Update Authority Delete Authority

Admin
Professor
Student
Certifying Authority

Degree

Professor As A Service

Degree

Degree Details :

Degree Name	Degree Requirements	Course List
Ms Comp. Science	3.4 GPA, 32 Credits	Java, Python, C
Ms Data Analytics	3.2 GPA, 34 Credits	Python, Analytics, R
Ms Engg. Management	3 GPA 32 Credits	Analytics, Supply chain, operations

BACK

Degree Name : Degree Requirements :

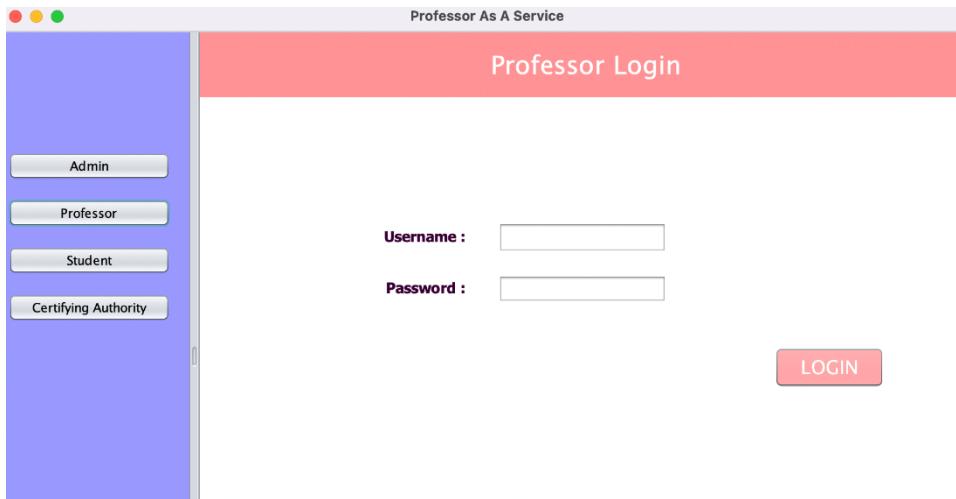
ADD DEGREE ADD COURSE

UPDATE DEGREE UPDATE COURSE

DELETE DEGREE DELETE COURSE

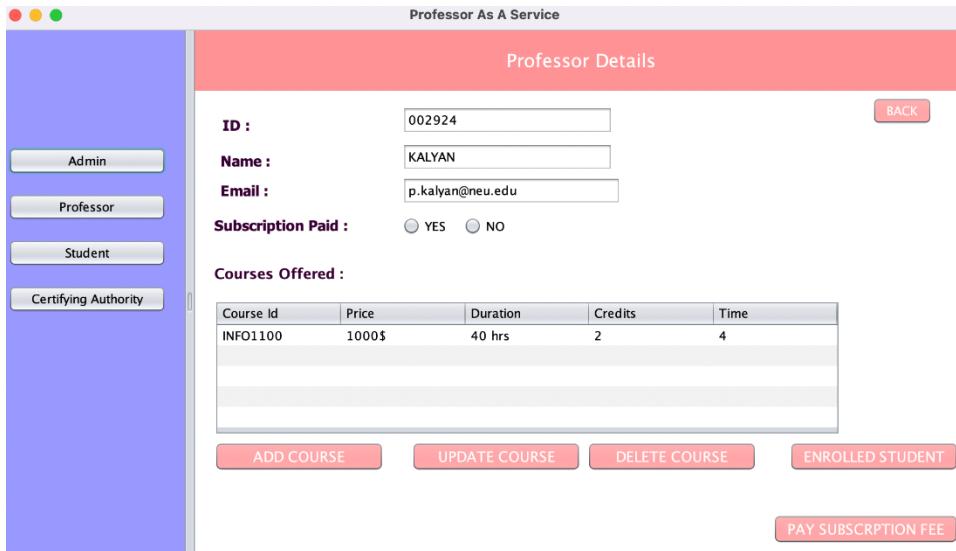
Admin
Professor
Student
Certifying Authority

Professor Login: Login page for professor to login.



The screenshot shows a window titled "Professor As A Service". On the left sidebar, there are four buttons: "Admin", "Professor" (which is highlighted in blue), "Student", and "Certifying Authority". The main area is titled "Professor Login". It contains two input fields: "Username :" and "Password :" with their respective labels. To the right of these fields is a red "LOGIN" button.

Professor Details: All the necessary details are present. Professor has to pay a subscription fee in order to teach a course to the students.



The screenshot shows a window titled "Professor As A Service". On the left sidebar, there are four buttons: "Admin", "Professor" (highlighted in blue), "Student", and "Certifying Authority". The main area is titled "Professor Details". It displays several input fields: "ID :" with value "002924", "Name :" with value "KALYAN", and "Email :" with value "p.kalyan@neu.edu". Below these is a "Subscription Paid :" section with two radio buttons: "YES" and "NO" (the "NO" button is checked). A "BACK" button is located to the right of the "ID" field. Under "Courses Offered:", there is a table showing one course: INFO1100, Price 1000\$, Duration 40 hrs, Credits 2, and Time 4. At the bottom are several buttons: "ADD COURSE", "UPDATE COURSE", "DELETE COURSE", "ENROLLED STUDENT", and a large red "PAY SUBSCRIPTION FEE" button.

Course Id	Price	Duration	Credits	Time
INFO1100	1000\$	40 hrs	2	4

Add Course Details: Professor can add course details and corresponding price and duration of the course.

Professor As A Service

Add Course Details

All Available Courses :

Course Id	Course Name
INFO1100	Intro to Java
INFO1200	Advanced Java
INFO1300	Algorithms

BACK

Course ID:

Course Name:

Price :

Duration :

Credits :

Time :

ADD COURSE

Enrolled Students Page: List of enrolled students are available for the professor to see. He can change the list by adding/deleting student(s).

Professor As A Service

Enrolled Students Details

Course Id : BACK

Course Name :

Enrolled Student List :

Student Id	Student Name	Student Email	Fees Paid	Grades
1234	Thanmaye P	thanu.p@neu.com	1000\$	B+

ADD STUDENT **DELETE STUDENT**

Student Login: Student login page where username and password are required.

Professor As A Service

Student Login

Admin

Professor

Student

Certifying Authority

Username :

Password :

LOGIN

Student Details Page: Information of the student is present. Options such as registration of a course or option to pay the fees is available to the students.

Professor As A Service

Student Details

BACK

ID : Degree Requirement :

Name : Total Credits Completed :

Degree Name : Degree Earned :

Email : Current GPA :

Courses Taken :

Course Id	Course Name	Professor Id	Professor Name	Fees Paid	Grades
INFO1100	Intro to Java	002924	Kalyan P	1000\$	A+

Register Course : Allows student to see different courses available with their corresponding reputation index which will allow students to choose courses better.

Professor As A Service

Course Registration

Search Course By :

Courses List :

Course Id	Course Name	Professor Id	Professor Name	Reputation Index	Credits
INFO1100	Intro to Java	002924	Kalyan P	4.2	3

BACK REGISTER

Requesting Degree Approval

Professor As A Service

Request For Degree Approval from Certifying Authority

Certifying Authority

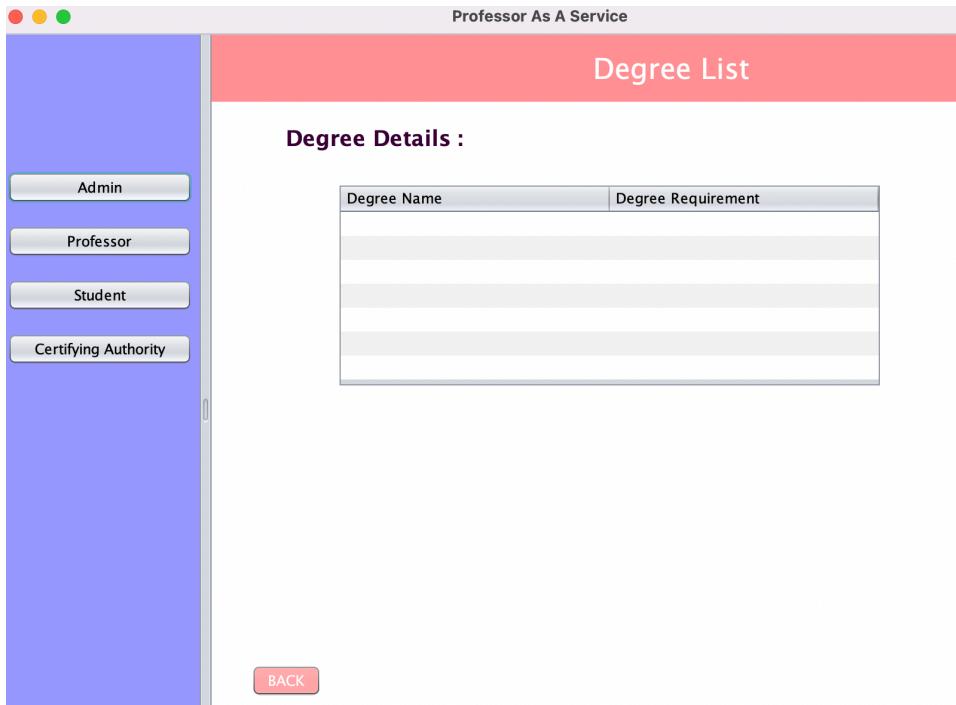
Authority Name

REQUEST

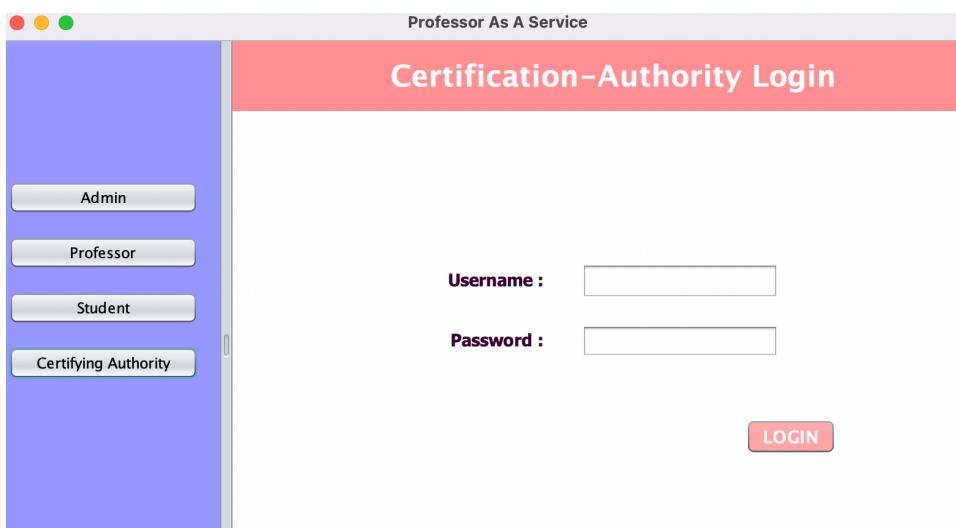
Degree Approved By

Authority Name

Degree list



Certification Login



Certifying Authority Details Page

Professor As A Service

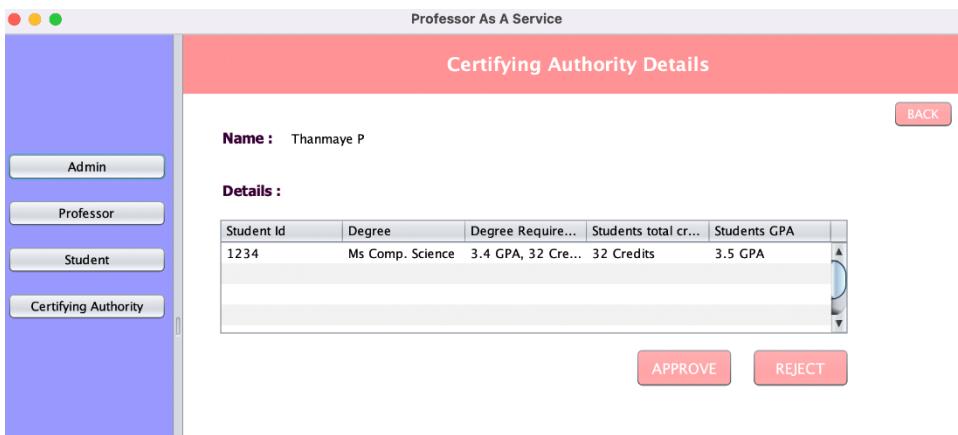
Certifying Authority Details

Name : Thanmaye P

Details :

Student Id	Degree	Degree Require...	Students total cr...	Students GPA
1234	Ms Comp. Science	3.4 GPA, 32 Cre...	32 Credits	3.5 GPA

BACK APPROVE REJECT



Statistics button to check Student Performance metrics

Professor As A Service

ADMIN VIEW

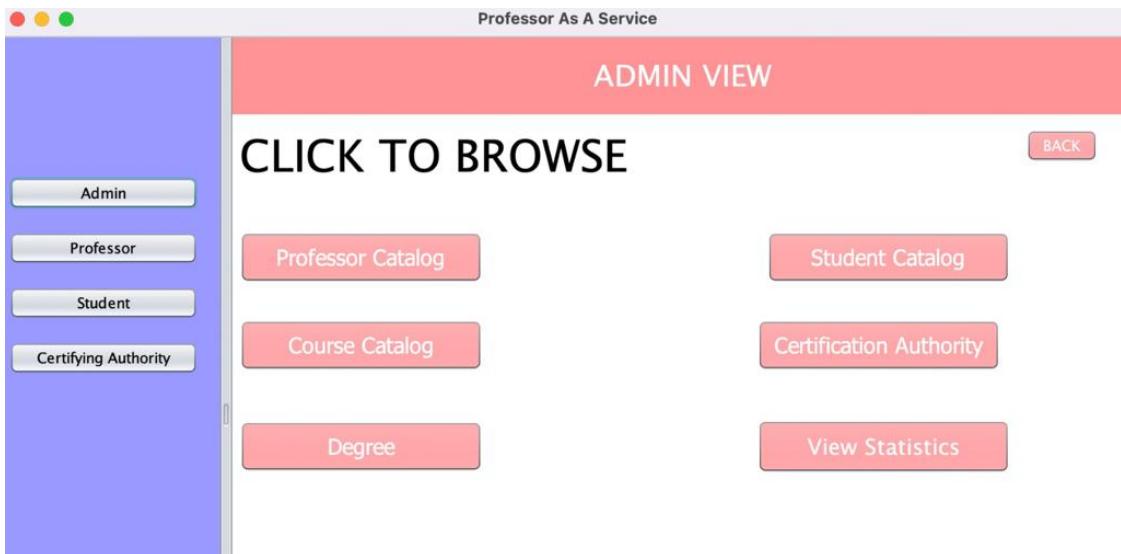
CLICK TO BROWSE

BACK

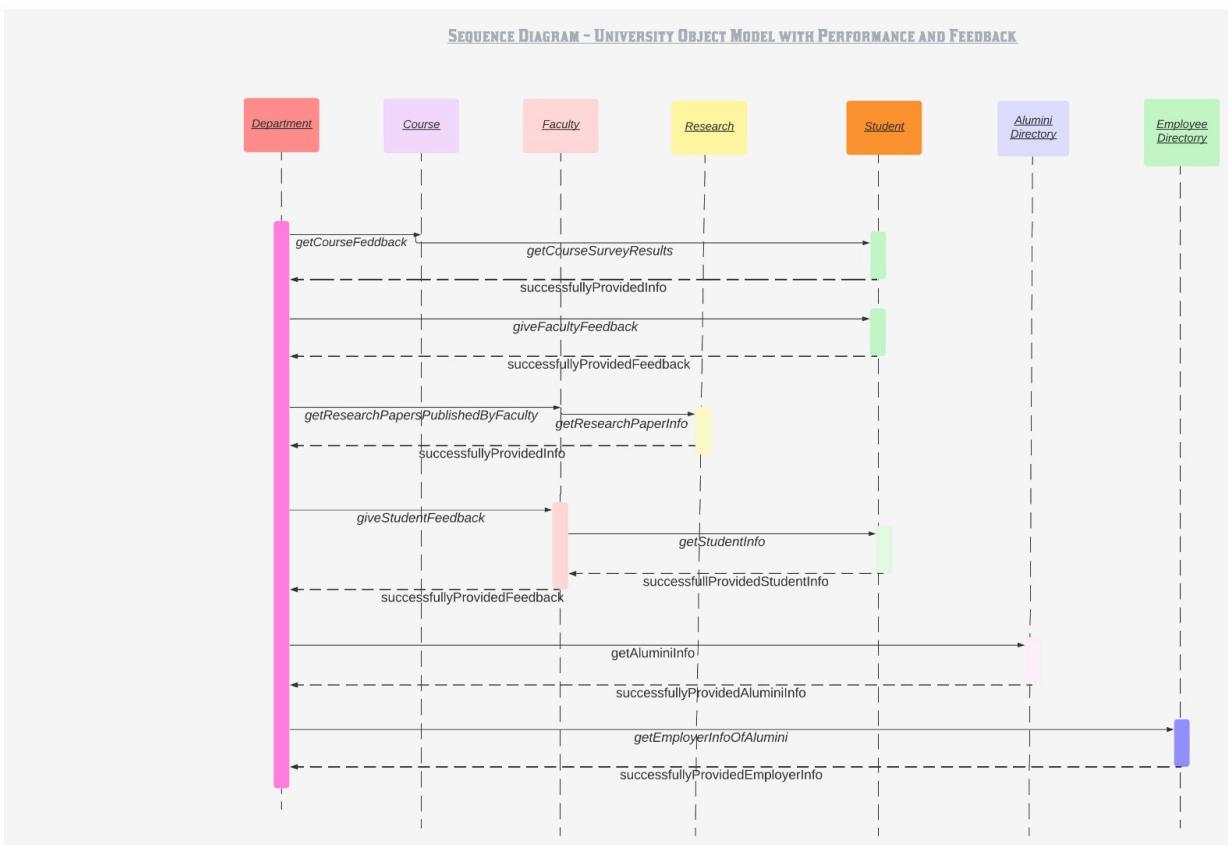
Admin Professor Student Certifying Authority

Professor Catalog Course Catalog Degree

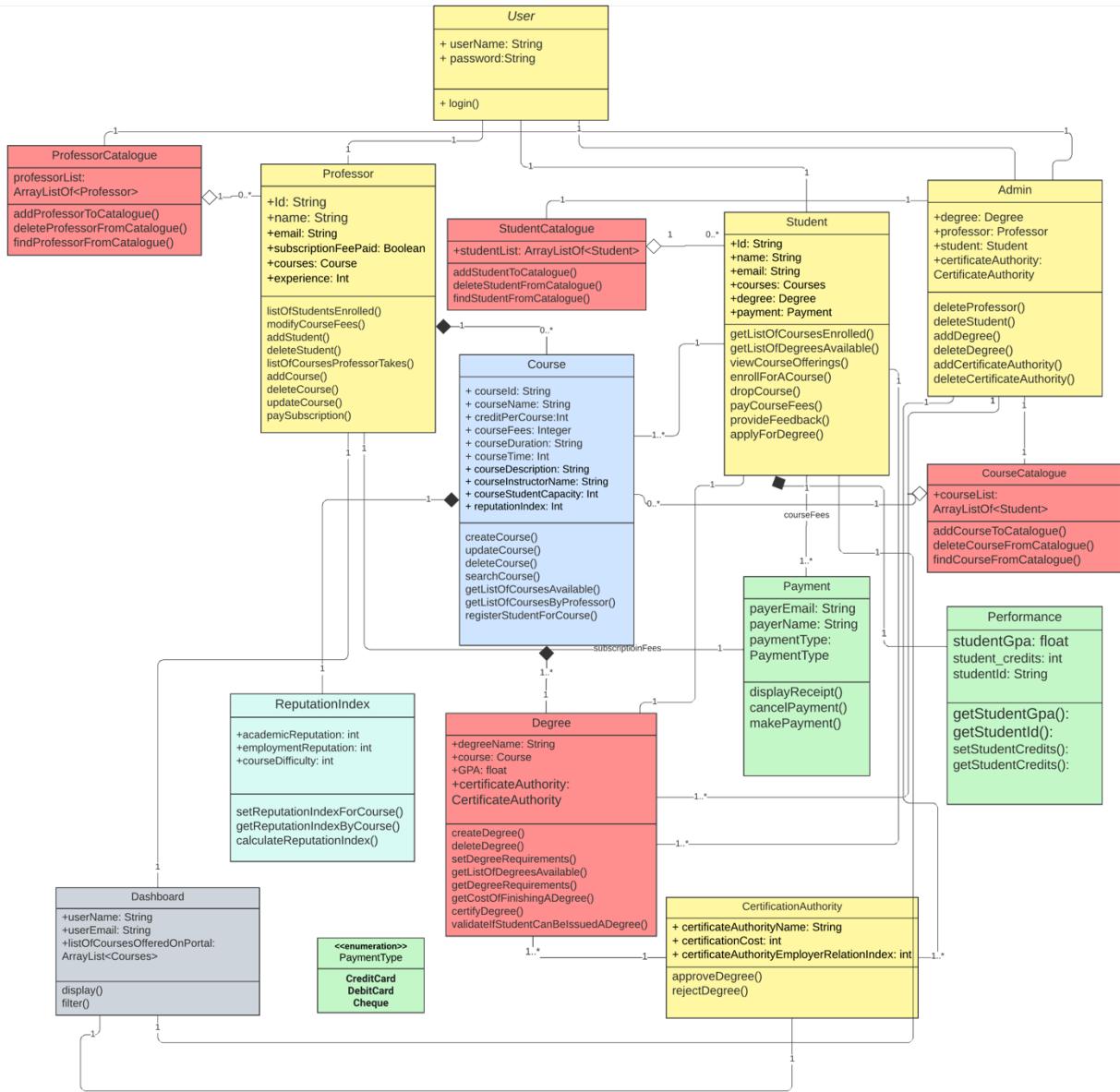
Student Catalog Certification Authority View Statistics



SEQUENCE DIAGRAM FOR UNIVERSITY OBJECT MODEL WITH PERFORMANCE AND FEEDBACK



CLASS DIAGRAM FOR PROFESSOR AS A SERVICE



USER:

Any person using the “Professor as a Service” model is a user. It can be a professor, student, third-party certification authority, or admin.

METHODS	DESCRIPTION
login()	Login to the model using the username and password.

PROFESSOR:

The Professor, who has full autonomy, is the sole authority of the Professor as a service model. He can offer courses and can control this system anywhere in the world and at any time. He pays the subscription fee for using the digital platform. Apart from modifying the fees and courses whenever required, the Professor can also add and delete students and courses.

METHODS	DESCRIPTION
listOfStudentsEnrolled()	Displays the details of the students who have enrolled under that Professor
modifyCourseFees()	Modifies the course fees that must be paid by the student
addStudent()	Adds a new student record
deleteStudent()	Deletes the record of an existing student
listOfCoursesProfessorTakes()	Displays the courses that the Professor teaches
addCourse()	Adds a new course
deleteCourse()	Deletes an existing course
updateCourse()	Makes necessary changes to the existing course
paySubscription()	Pays the fee to the digital platform

STUDENT:

A student is a user who takes courses from professors and certificates from third-party certification authorities. He can view all the course offerings and degrees available and can also enroll or drop the course. In the end, the student needs to give feedback for the course taken. in order to obtain a degree, students must take courses from various professors and fulfill the course requirements.

METHODS	DESCRIPTION
getListOfCoursesEnrolled()	Displays the courses the student has enrolled
getListOfDegreesAvailable()	Displays the available degrees
viewCourseOfferings()	Displays the various course offerings
enrollForACourse()	Lets the student enroll in any course
dropCourse()	Lets the student drop from any enrolled course
payCourseFees()	Payment for the course is done
provideFeedback()	Allows the student to give feedback on the course after the completion
applyForDegree()	Lets the student apply for a degree after the completion of the course

COURSE:

Courses are taught by the professor and the certification for these courses will be given by third-party certification authorities. All the courses, including the course fee are managed by the professor.

METHODS	DESCRIPTION
createCourse()	Creates a new course
updateCourse()	Updates an existing course
deleteCourse()	Deletes an existing course
searchCourse()	Searches for a required course
getListOfCoursesAvailable()	Displays the list of all the available courses
getListOfCoursesByProfessor()	Displays all the courses taught by a specific professor
registerStudentForCourse()	Registration of student for the course

DEGREE:

A degree is obtained by the student after completing all the courses. The third-party certification authority approves the degree for the students.

METHODS	DESCRIPTION
createDegree()	Creates a degree
deleteDegree()	Deletes an existing degree

setDegreeRequirements()	Sets all the requirements that a student needs to meet in order to obtain a degree
getListOfDegreesAvailable()	Displays all the available degrees
getDegreeRequirements()	Shows all the degree requirements
certifyDegree()	Certifies degree
getCostOfFinishingADegree()	Displays the fees needed to be paid to obtain a degree
validateIfStudentCanBeIssuedADegree()	Checks from all the degree requirements and decides if the student can obtain a degree or not

ADMIN:

Admin is a user who controls the entire system including its users.

METHODS	DESCRIPTION
deleteProfessor()	Removes the professor's record
deleteStudent()	Removes the student's record
addDegree()	Adds a new degree

DASHBOARD:

The dashboard is a display page. It displays the details of admin, student, professor and third-party certification authority.

METHODS	DESCRIPTION
display()	Displays all the details
filter()	Used for searching records

PAYMENT:

Payment is done by the professor for the use of the digital platform. Students also make payments for the courses to the professor. The various methods to make payment are also given.

METHODS	DESCRIPTION
displayReceipt()	Displays the payment receipt
cancelPayment()	Cancels the transaction
makePayment()	Makes payment

THIRD-PARTY CERTIFICATION AUTHORITY

The third-party certification authority is used to approve the degree.

METHODS	DESCRIPTION
approveDegree()	Approves the student degree after the student meets the degree requirements
rejectDegree()	Rejects the student degree if the student doesn't meet the degree requirements

REPUTATION INDEX

Reputation Index, for any course, is useful for the incoming students to check for courses based on rating. It is given by the students who have completed the course. It is given on a scale of 1 to 5, 1 being the lowest and 5 being the highest.

METHODS	DESCRIPTION
setReputationIndexForCourse()	Sets a reputation index for a course based on the feedback given by students who have completed the course
getReputationIndexByCourse()	Gets the reputation index by the course
calulateReputationIndex()	Calculates the reputation index

PROFESSOR CATALOGUE

It stores the details of all the professors.

METHODS	DESCRIPTION
addProfessorToCatalogue()	Adds a new professor record to the catalogue
deleteProfessorFromCatalogue()	Removes details of an existing professor from the catalogue.
findProfessorFromCatalogue()	Finds the details of a professor from the catalogue

STUDENT CATALOGUE

It stores the details of all the students.

METHODS	DESCRIPTION
addStudentToCatalogue()	Adds a new student record to the catalogue
deleteStudentFromCatalogue()	Removes details of an existing student from the catalogue.
findStudentFromCatalogue()	Finds the details of a student from the catalogue

COURSE CATALOGUE

It stores the details of all the courses.

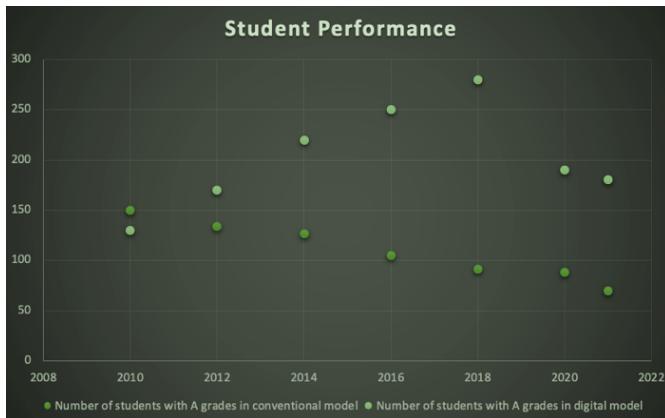
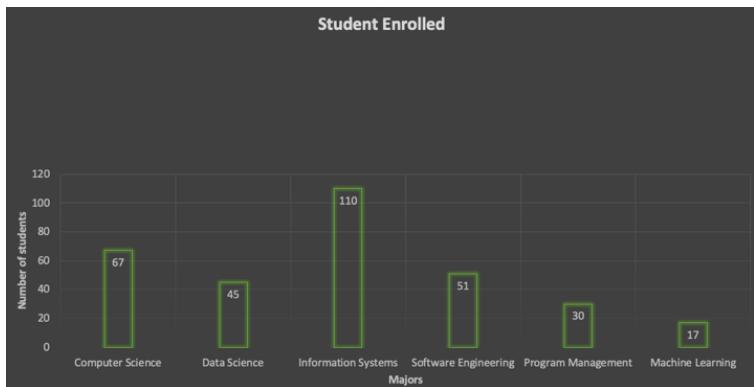
METHODS	DESCRIPTION
addCourseToCatalogue()	Adds a new course to the catalogue
deleteCourseFromCatalogue()	Removes details of an existing course from the catalogue.
findCourseFromCatalogue()	Finds the details of a course from the catalogue

PAYMENT TYPE

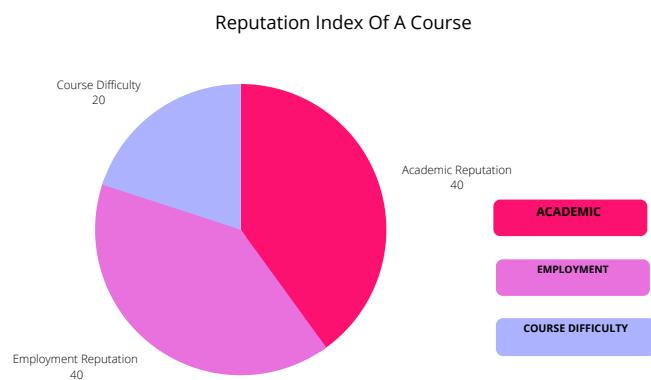
The payment can be made through three methods:

- Credit Card
- Debit Card
- Cheque

Admin's view of graphs for student performance metrics



The follow view can be seen by a student when they login to their dashboard to enroll in a course



Overall assessment and opinion

Scope of digital educational platforms is limitless. It provides students to access wide variety of courses. This model suggested will definitely help students and will allow them to experience convenient, innovative content through modern learning methods. Students can customize courses that they want to study and design their curriculum as per their own interest. This model can help students who cannot afford go to educational institutes to get a degree just with a smart phone, ipad or a tablet. They can now get access to education from anywhere in the world. Digital educational platforms will also provide students the option to review their performances and since this type of education is more engaging, studying is now more fun and as knowledge is available to students present in any corner of the world.