ASSIGNMENT 3
PROFESSOR AS A SERVICE

TEAM MEMBERS

Sanaa Rupani (001567616)
Sindhu Srinivas Prasad (002920301)
Shriya Dikshith (002921535)

PROBLEM STATEMENT

The goal is to reduce tuition costs while improving the quality of education around the world with software engineering techniques. In order to improve the quality of life for the masses, we want to hold people accountable through education, learning to learn, and feedback.

OBJECTIVE

To provide a decentralized system where professors hold rights to their talent and experience, not the educational institutions. The system was designed to analyze the statistical data obtained from observing students in a professional setting and analyzing the courses that helped students succeed in their career without having to pay higher tuition fees.

PROPOSED SOLUTION

- Each professor manages his or her own classes as well as when they will teach them.
- The professor will be able to log into the service from anywhere in the world and on any device.
- The professor has full control over the price per course.
- The student will be able to log into the service from anywhere in the world and on any device.
- Students may take courses from a variety of professors (located anywhere in the world) to earn their degree
- Students can take their preferred courses from their preferred instructor.
- Students are allowed to register for courses based on their willingness, there is no restriction imposed on the number of courses to be registered for any given period.
- Students are offered a waiver on the courses based on their highest degree grade which makes it easier for financially challenged students.
- Depending on demand the price per course can vary: if demand is higher than the price is increased by the professor by a negligible amount and, if demand is lower than the price is decreased by the professor by a negligible amount.
- Employers can get information about the courses learnt by a student by fetching data from student course history.

PERFORMANCE METRICS:

GPA and Grades will be evaluated based on the metrics below:

GPA	Percentage scored %	Grade
4.00	94% and 100%	A
3.75	90% and < 94%	A-
3.50	87% and < 90%	B+
3.25	84% and < 87%	B
3.00	80% and < 84%	B-
2.75	77% and < 80%	C+
2.50	74% and < 77%	C
2.25	70% and < 74%	C-
2.00	0% and < 70%	F

Fee waiver will be calculated based on:

Calculation of tuition fees with applied waivers for students based on the GPA obtained by them.

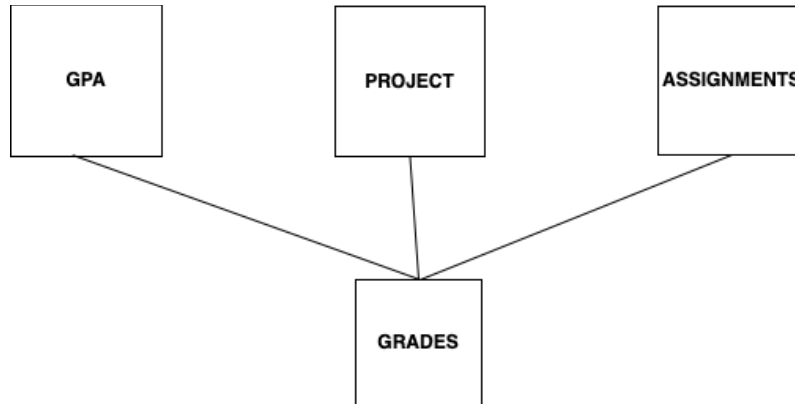| GPA obtained	Waiver offered on Tuition
4.00	65%
3.75	60%
3.50	55%
3.25	50%

Fee waiver calculations based on Annual Income:

Calculation of tuition fees with applied waivers for parents with annual income between $25000-$35000.

Annual Income	Waiver offered on Tuition
< $25,000	80%
$25,000 - $30,000	75%
$30,000 - $35,000	70%

Student Grading Metric:

This estimates student grades based on grades there's further waiver on tuition fees. Grades will be awarded based on Assignments, projects, and quizzes. The professor's course cost will be calculated based on this metric.

```
  ┌────────┐      ┌──────────┐      ┌──────────────┐
  │  GPA   │      │ PROJECT  │      │ ASSIGNMENTS  │
  └────────┘      └──────────┘      └──────────────┘
        \              │               /
         \             │              /
          \        ┌────────┐        /
           \───────│ GRADES │───────/
                   └────────┘
```
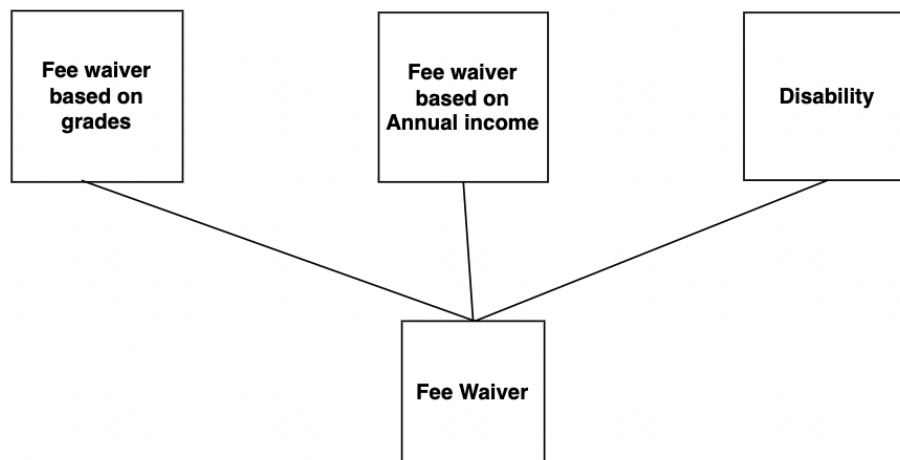
Professor Reputation Index:

This metric will help us calculate professor's performance based on the ratings and reviews given by the students.

```
  ┌──────────┐   getRatings()   ┌────────────┐
  │ STUDENT  │ ───────────────► │ PROFESSOR  │
  └──────────┘                  └────────────┘
```

Fee waiver Index:

We have offered different fee waivers based on financial status, grades, and disability.

```
┌──────────────┐      ┌──────────────┐      ┌──────────────┐
│  Fee waiver  │      │  Fee waiver  │      │  Disability  │
│   based on   │      │   based on   │      │              │
│    grades    │      │ Annual income│      │              │
└──────────────┘      └──────────────┘      └──────────────┘
         \                   │                    /
          \                  │                   /
           \                 │                  /
            \         ┌──────────────┐         /
             \────────│  Fee Waiver  │────────/
                      └──────────────┘
```

CONCLUSION

Equality states that every student, regardless of where they come from, should have equal access to a high-quality education. It is not accessible to everyone equally due to a few limits such as financial status and geographic location, and the only individuals who have access are those who have the above-mentioned. Our approach focuses on making education accessible to anybody, wherever in the globe, by making it virtual and eliminating all traditional ways by providing Professors complete control over course creation and changing the pricing range of courses/degrees offered based on GPA.

UI DESIGN

Register as a user -
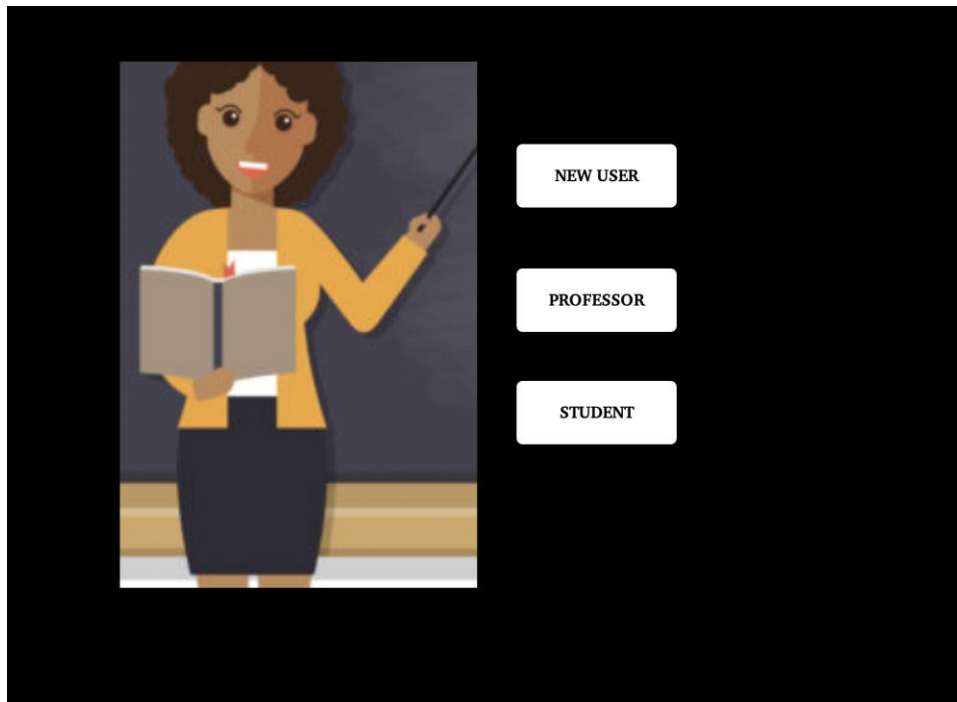
NEW USER

PROFESSOR

STUDENT
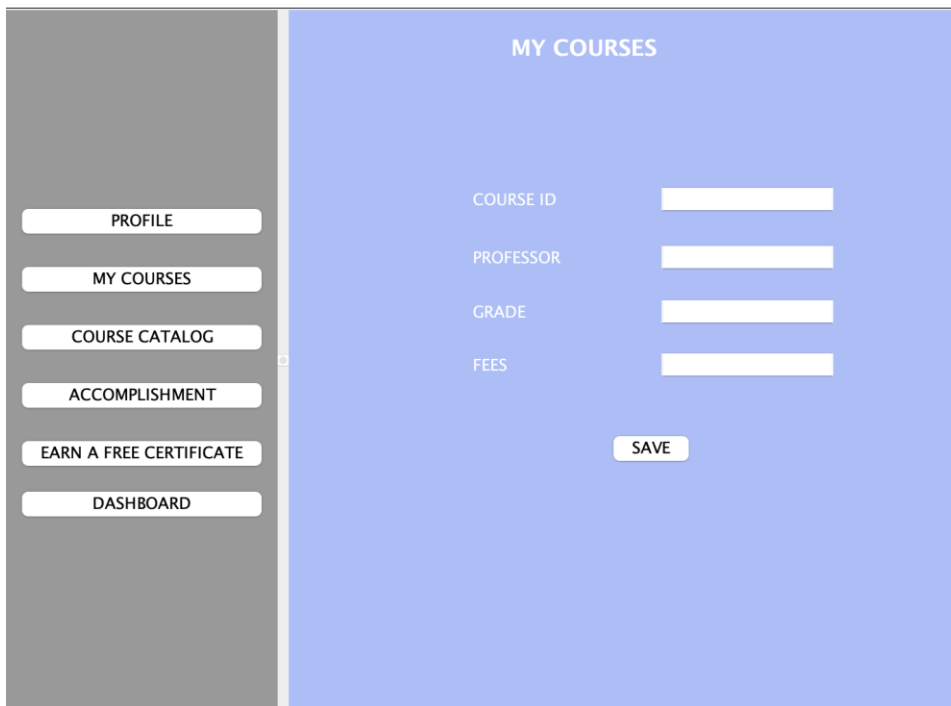
New User

page -

REGISTER

NEW USER

SIGN UP AS A STUDENT   SIGN UP AS A PROFESSOR

Student page -

MY COURSES

PROFILE

MY COURSES

COURSE CATALOG

ACCOMPLISHMENT

EARN A FREE CERTIFICATE

DASHBOARD

COURSE ID

PROFESSOR

GRADE

FEES

SAVE

Professor Page -

Subscription Payment -

UML DIAGRAM

Person
- username: String
- password: password
- emailAddress: String
- phone: double
- role: String
- signUp()
- signIn()

CertificationAuthority
- certificationId: int
- Grades grade
- Student student
- Course course
- Employer employer
- isCertified: boolean
- getStatus()
- editStatus()
- getEmployer()

StudentFeeWaiver
- Student student
- parentsIncome: int
- workingProfessional: boolean
- highestDegreeEarned: String
- gpa: float
- checkEligibility()

ProfessorDirectory
- ArrayList<Professor> professorList
- newProfessorProfile(User p)
- findProfessor(string ID)

Professor
- Id: int
- name: String
- age: int
- Skillset: ArrayList<String>
- Experience: int
- Country: String
- addSkills()
- addExperience()

Student
- User user
- Studendid: int
- Age: int
- Semester
- Student()

Employer
- employerId: int
- name: String
- positionsAvailable: int
- location: String
- baseSalary: double
- desginationOffered: String
- getAverageSalary()
- getIndustry()

StudentDirectory
- ArrayList<Student> StudentList
- newStudentProfile(User p)
- findStudent(string ID)
- getStudentList()

Grades
- Course course
- Student student
- grade: String
- addGrade()
- editGrade()
- getGrade()

CourseSchedule
- Course course
- days: String
- time: DateTime
- getCourseSchedule()

SlotAssignment
- Slot slot
- isOccupied()
- assignNewSlot()

StudentCourseHistory
- Student student
- Course course
- CertificationAuthority cert
- getEnrolledCourses()
- getCompletedCourses()
- getCourseStatus()

Course
- courseId: int
- courseName: String
- Professor professor
- fees: float
- duration: int
- rating: int
- getCourseDetails()
- setCourseFee()

EmployerDirectory
- ArrayList<Employer>
- getEmployerList()

Slot
- Course course
- occupied: boolean
- number: int
- SlotAssignment slotAssignment
- isOccupied()
- assignNewSlot()

Subscription
- subscriptionId: int
- Professor professor
- duration: int
- price: int
- addSubscription()
- removeSubscription()
- editSubscription()

CourseOffering
- Course course
- ProfessorAssign professorAssign
- assignProfessorToCourse(Professo

CourseCatalog
- ArrayList<Course> courseList
- getAllCourses()
- getCourseByName(String name)
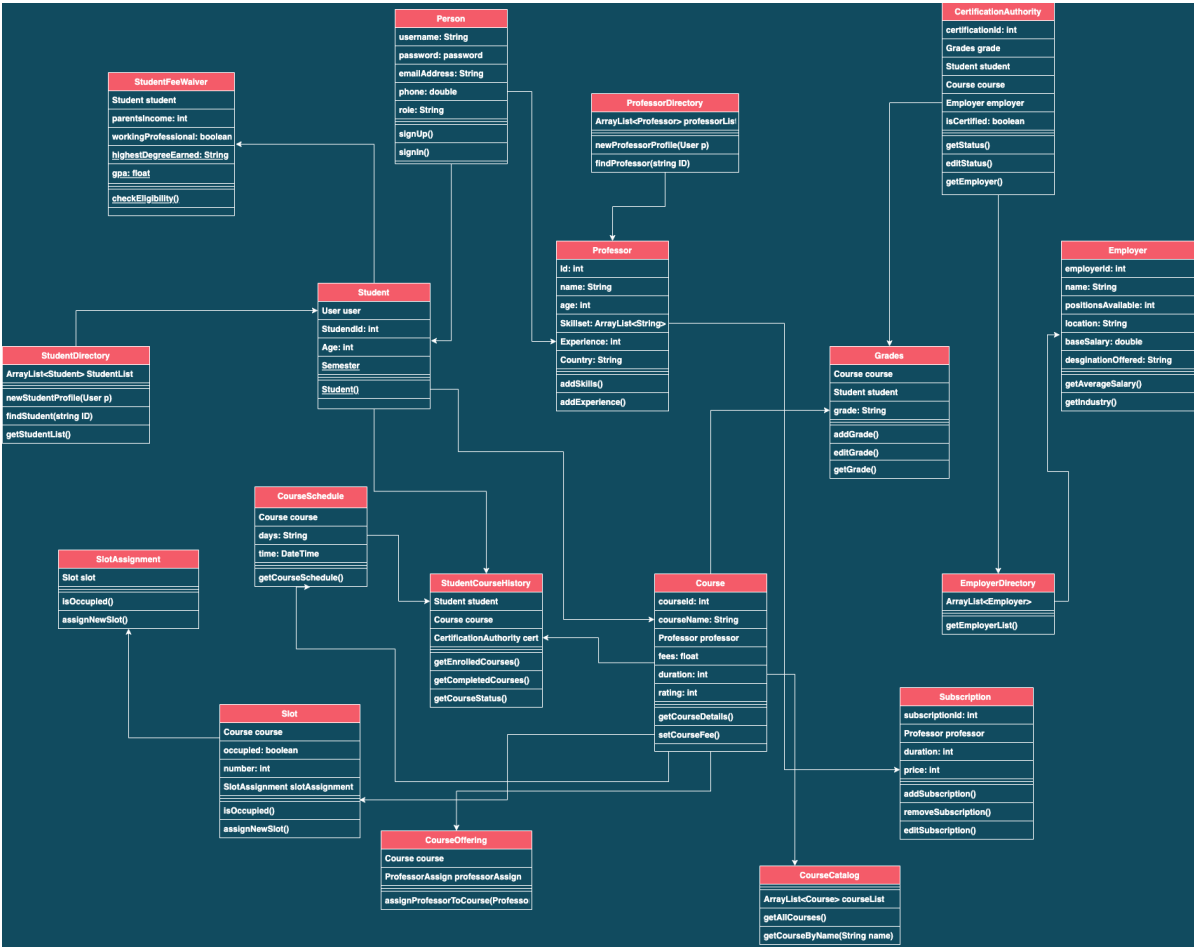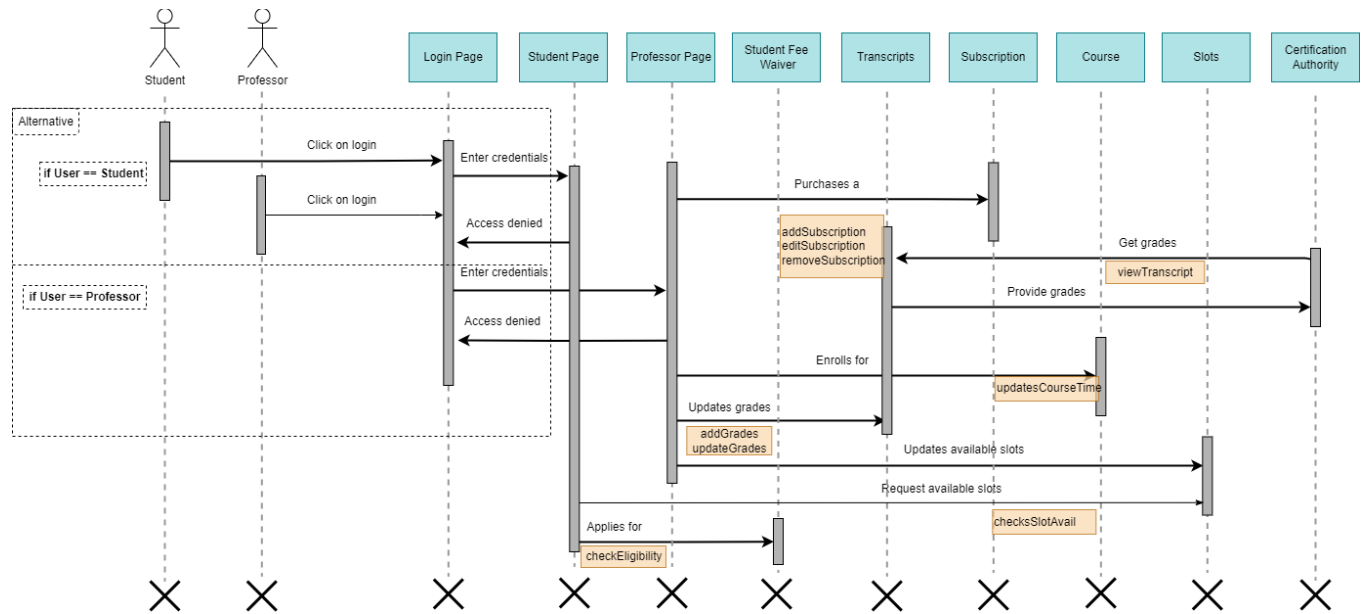
SEQUENCE MODEL

ARCHITECTURE OF THE MODEL:

The diagram below illustrates how the new business model differs from the traditional university system, as well as its architecture.