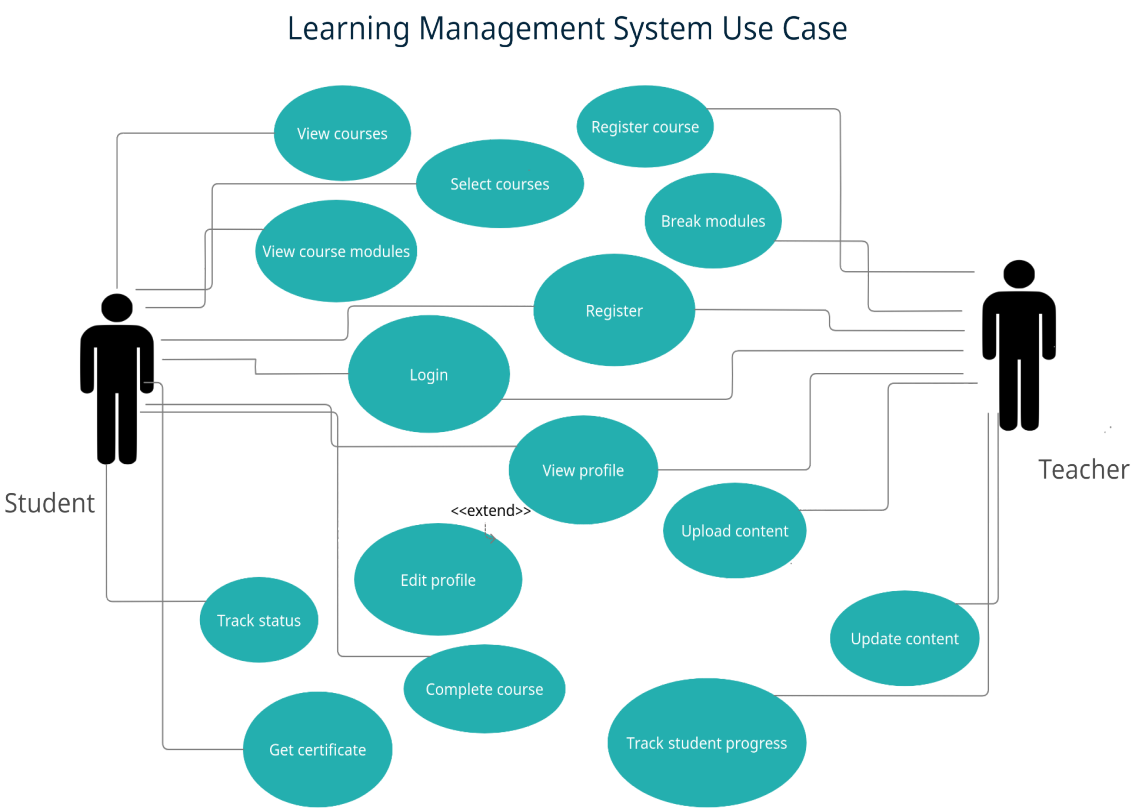
Product Design

Team Number: 45, Gautam Ghai, Yash Mehan, Manaswini Tharigopula, Mandyam Brunda

Design Overview

Architectural design

In our project there are 3 types of users: admin, teachers and students. After registering oneself as a teacher/student and then logging in, the dashboards will be different for both teacher and the student.
The student will be able to view all the courses he has taken and upon clicking a particular course, he'll see all the details of the course like course topics and modules for that course. He'll also be able to view what percentage of that course has been completed by him. Within a module there can be reading material, videos and a quiz for that module. Upon completing a course a certificate will be generated automatically which will be shown in the student's profile.
The teacher will have the ability to register a course, break that into modules and upload/update content for each module. The teacher can view the courses he is instructing and can see the progress of his students who are taking a particular course. Both the teacher and student can view/edit their respective profiles.
The admin will have the ability to provide staff permissions, configure portal and create and customise certificate

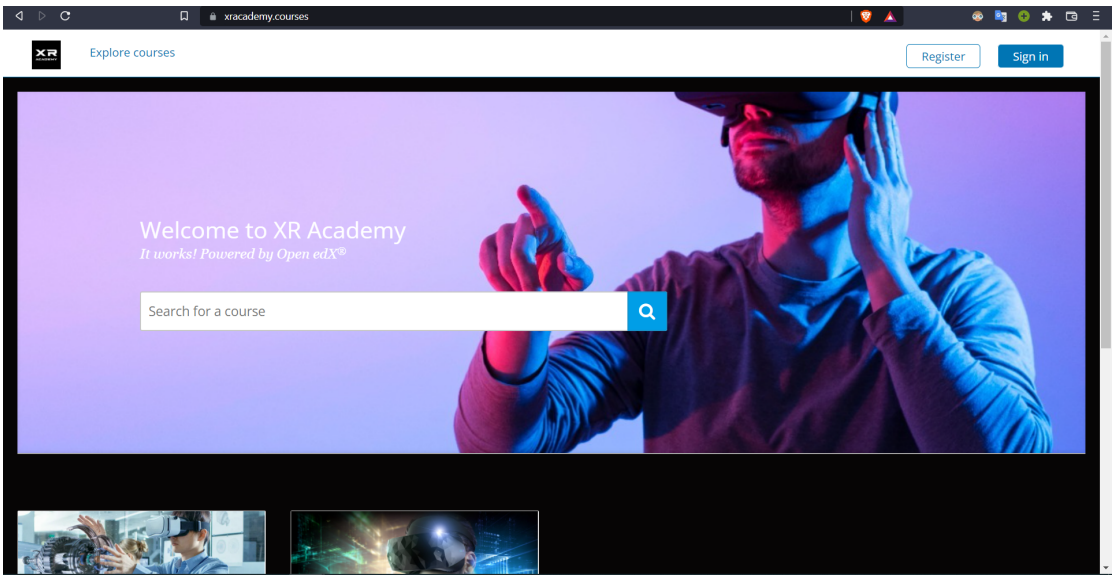


System Interfaces

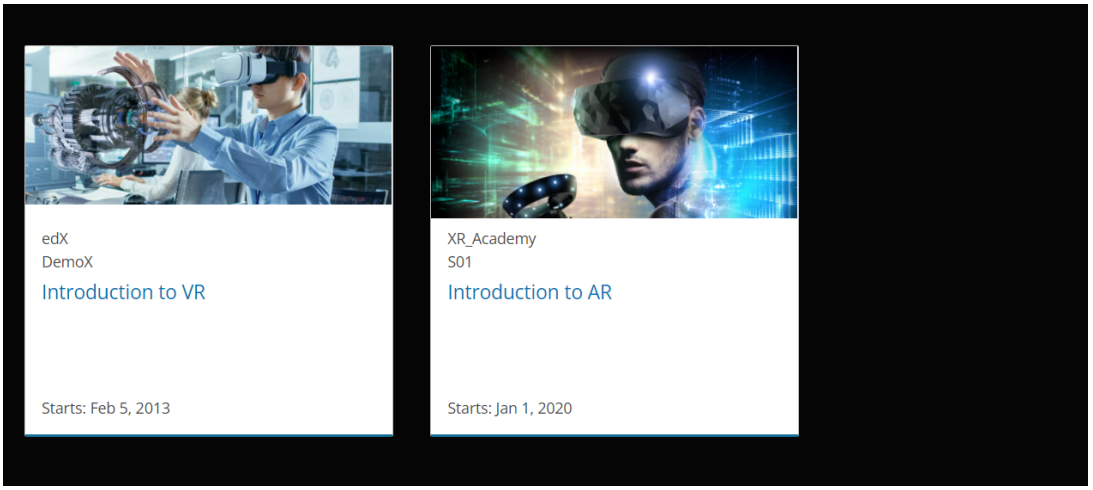
User Interface:

The system caters to 3 types of users: students, Staff and the administrator. Students can sign up, search for a course and register for it.The student can see the list of the courses he has taken and upon clicking a course he can see the topics, modules and also his progress in that course. After completing the quizzes in each course, they'll get a certificate that they can see in their profile. The student can edit the personal details in his profile as well. Teachers can sign up, register a course, break it into modules and upload/update the content for these modules. They can see the list of courses that they are instructing and on clicking a particular course they can see the list of students enrolled in that course and their progress. The teacher can edit the personal details in his profile as well.

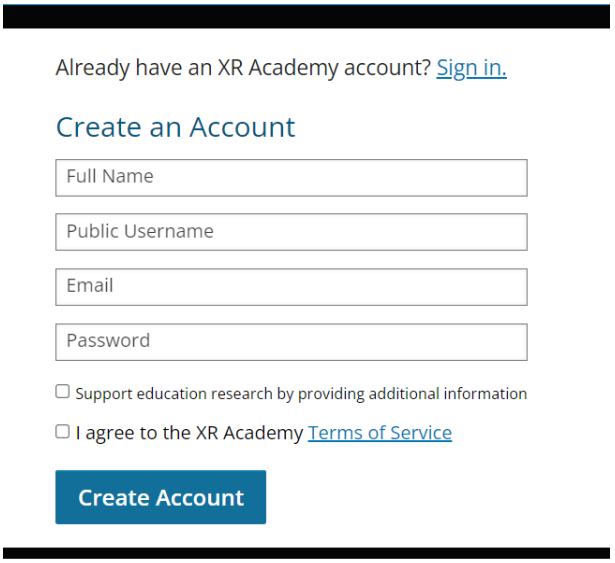
Below are the screenshots for register and login page:



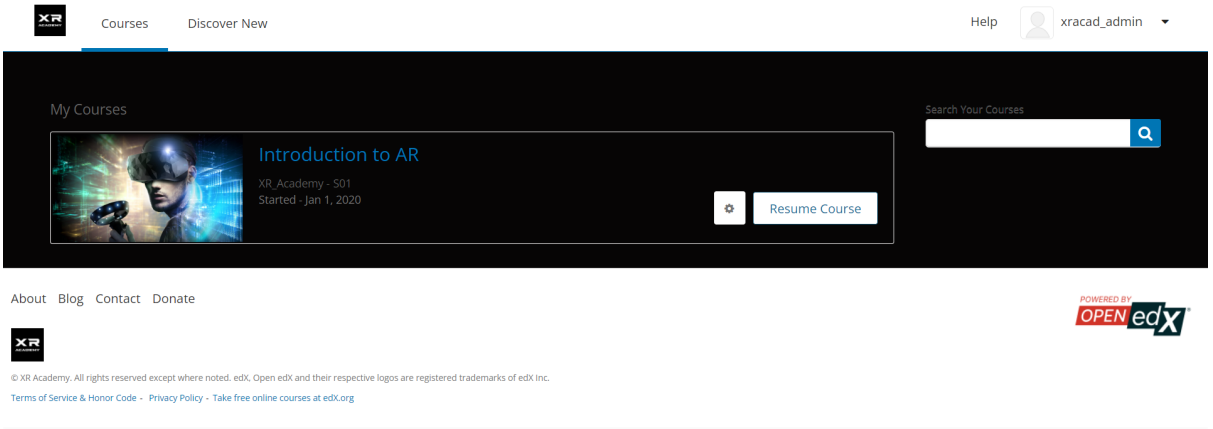
Course Listing



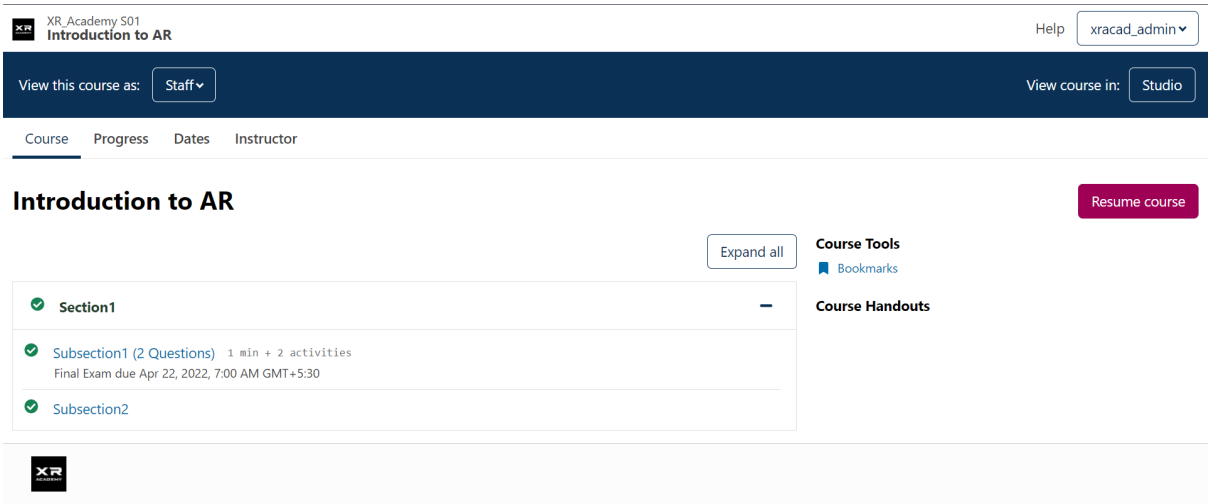
Authentication page



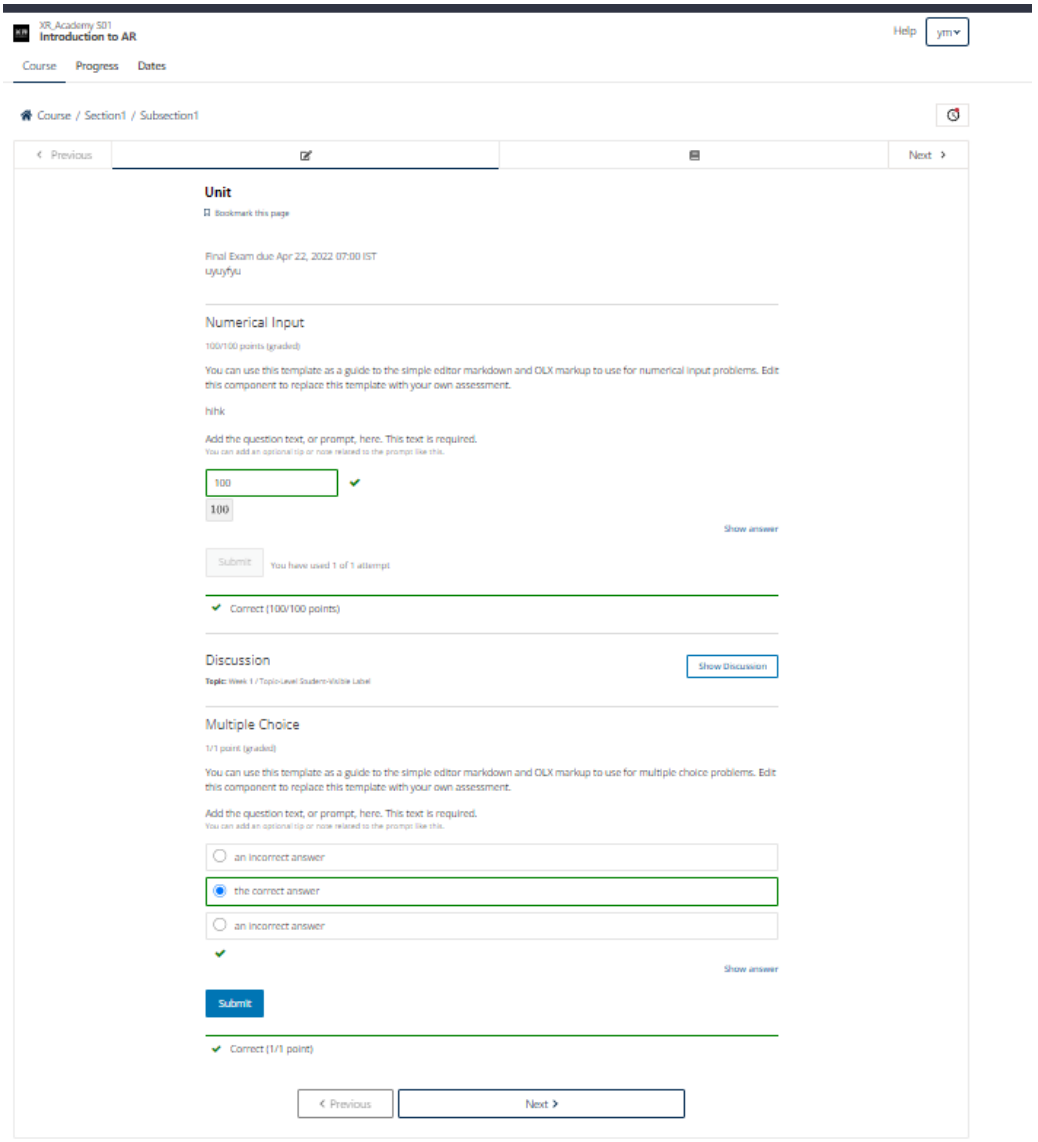
Landing page



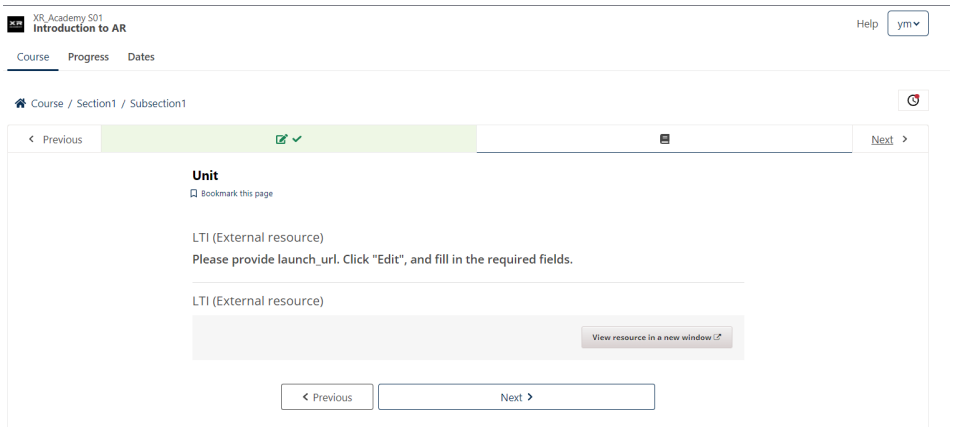
Overview of a course



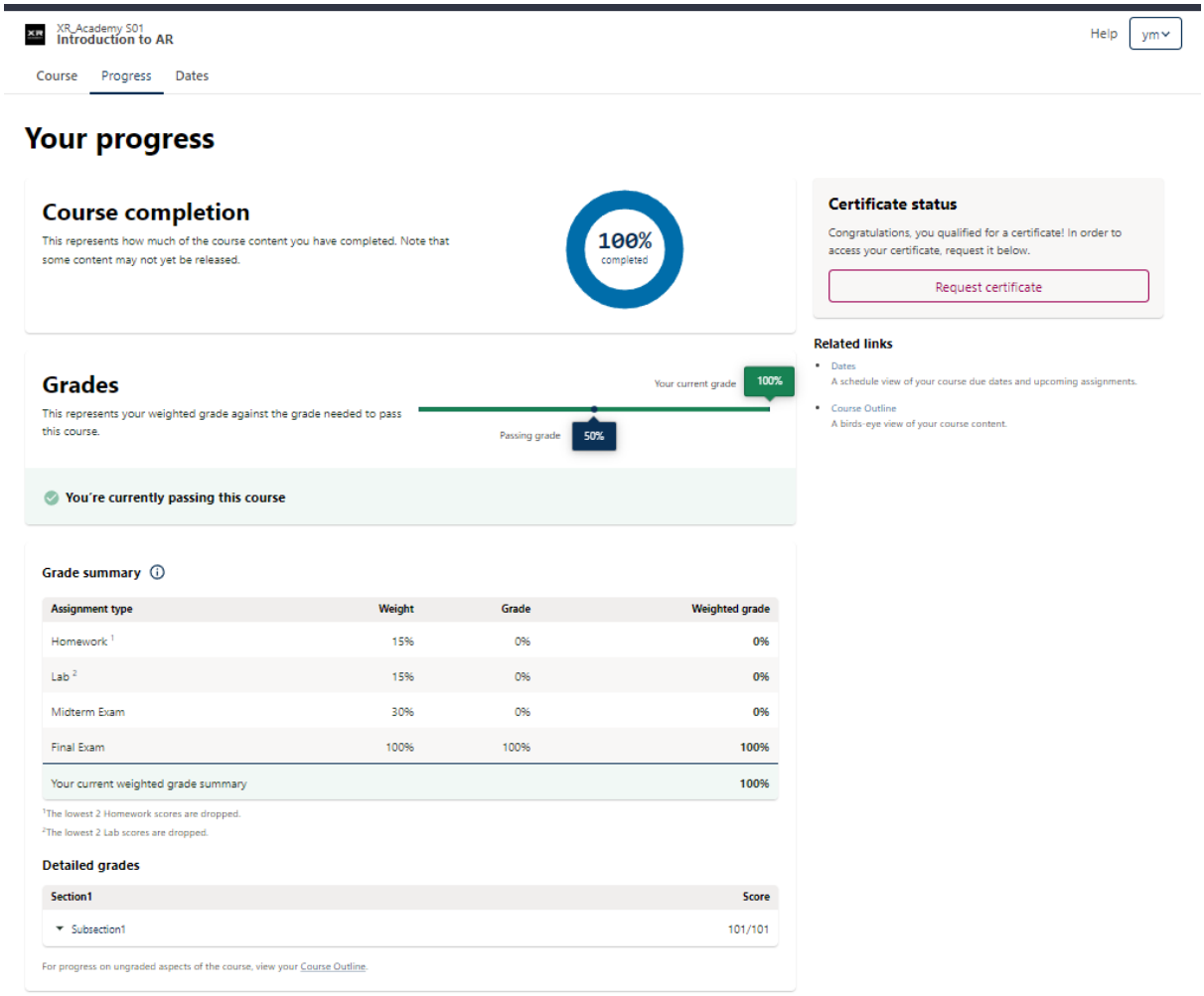
Assessment Module



Media Module

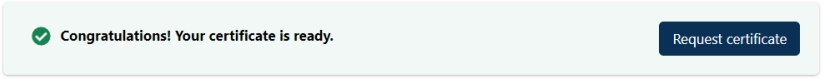


Progress Page

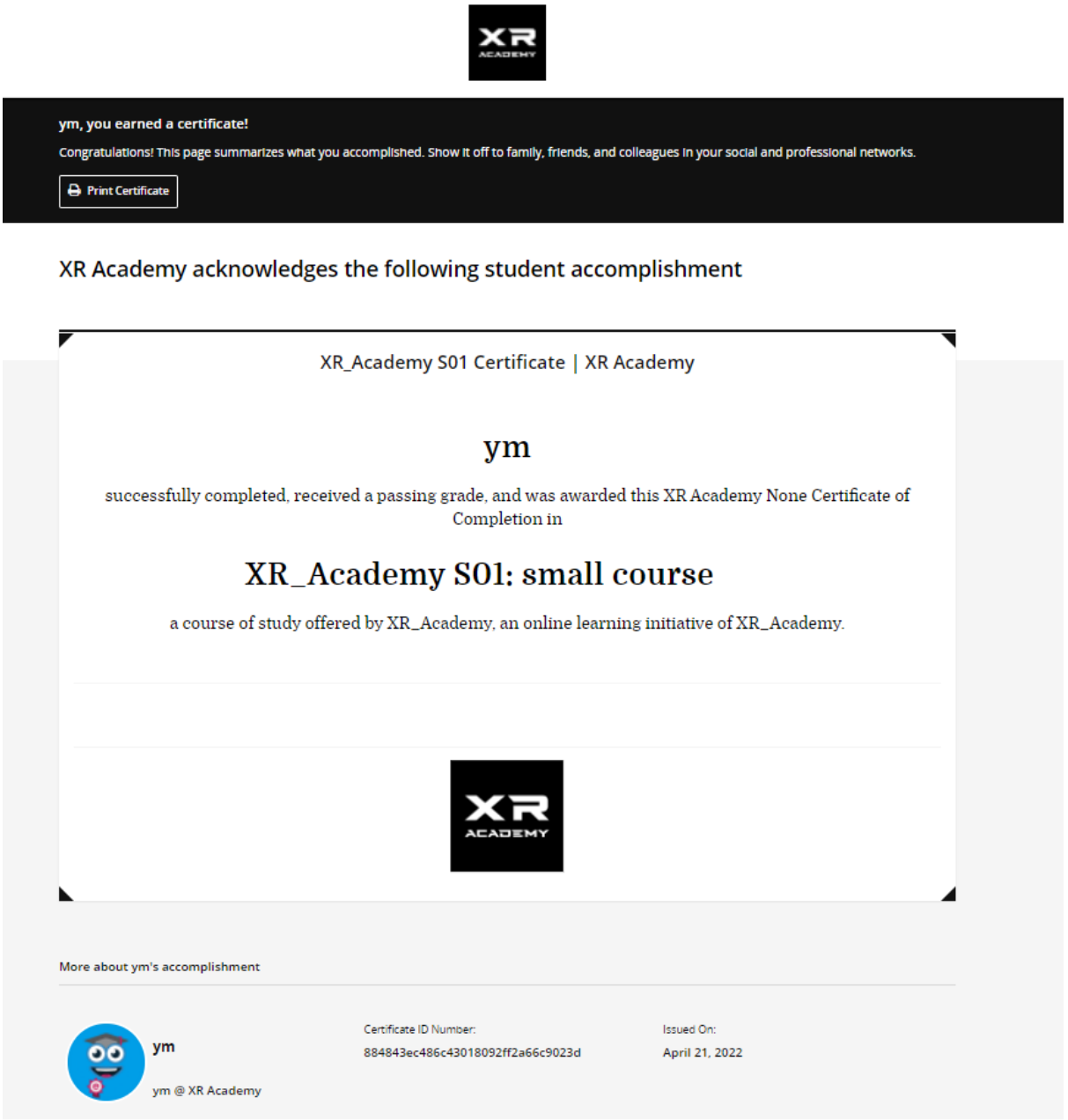


Certificate Generation

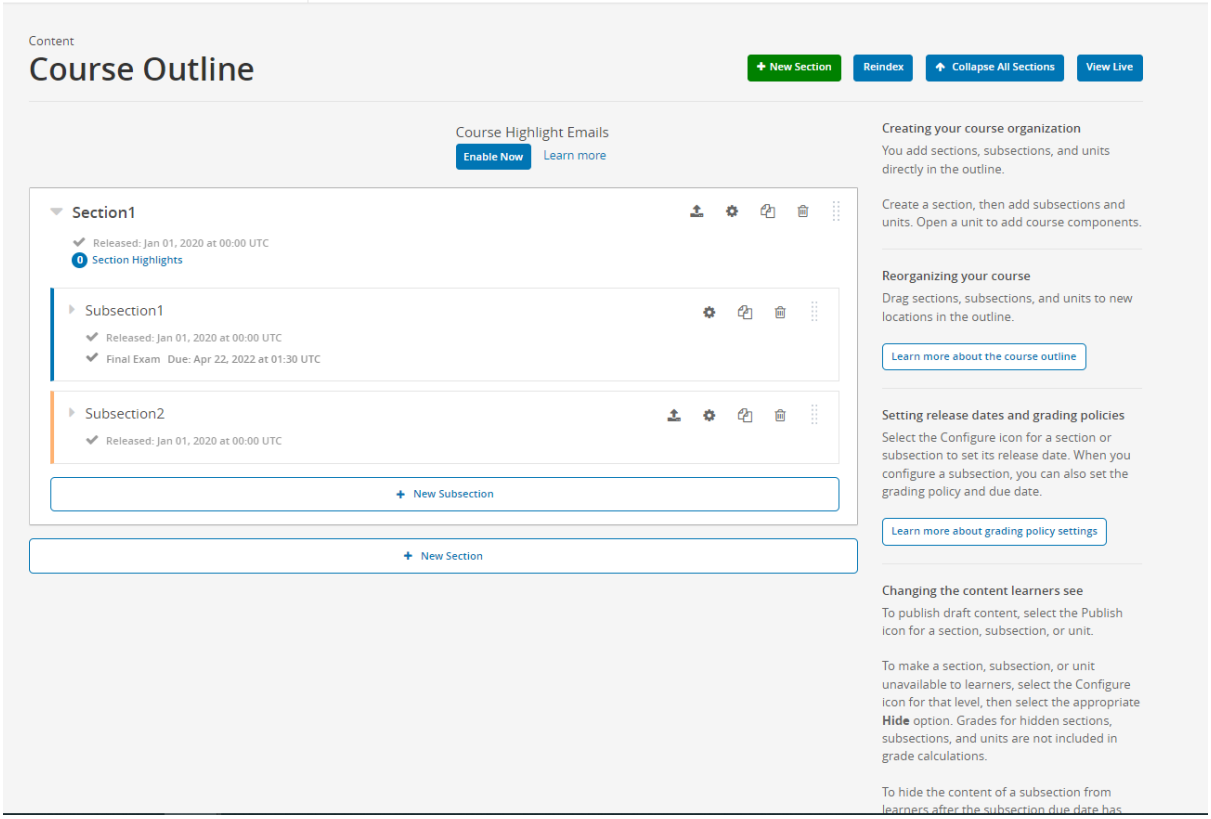
Introduction to AR



Generated certificate

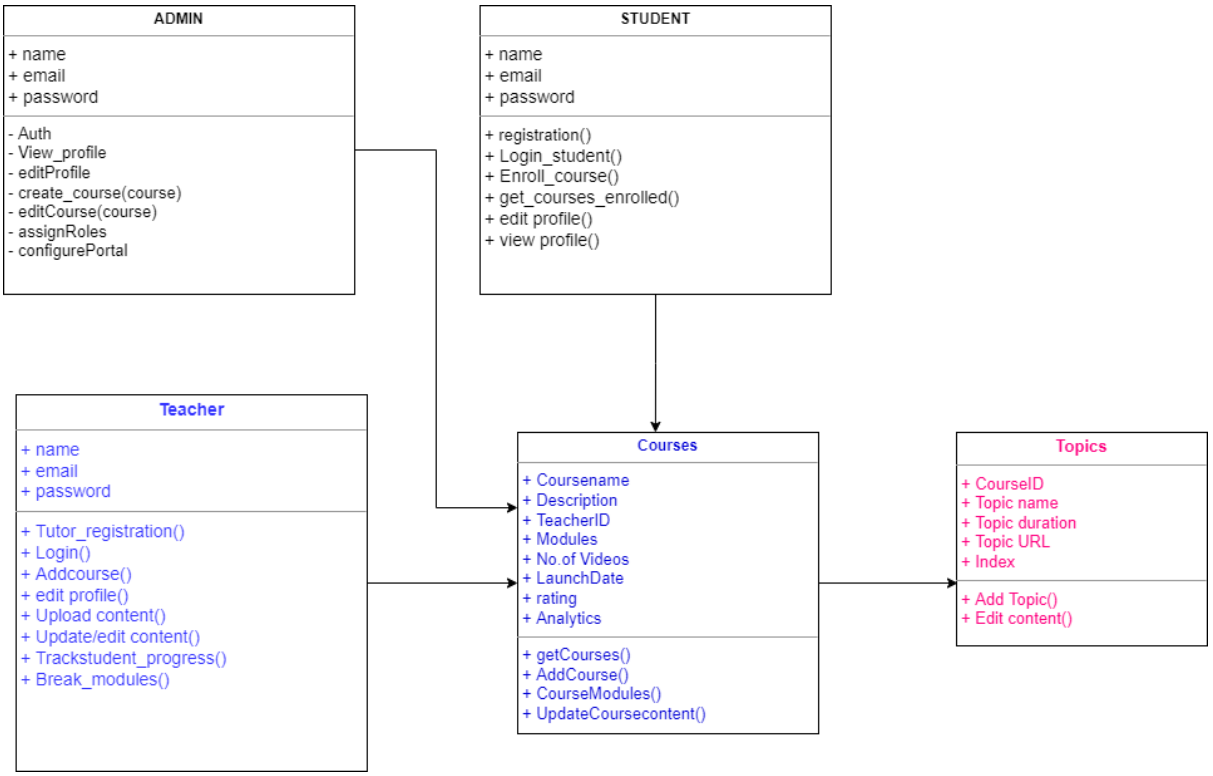


Studio Portal for managing individual Course



Model

Diagram link: https://app.diagrams.net/#G1OLpA_dE7SwuLELqETzPEvUowy0WoIM4c

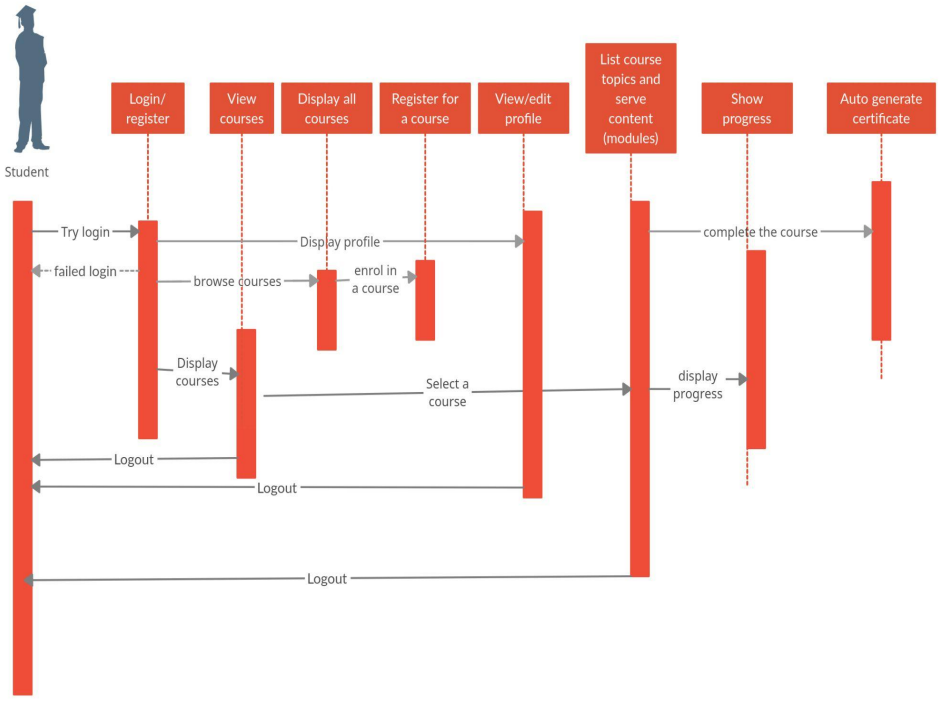


STUDENT	Class state • Name • Email Class Behaviour • Login() • Register() • Enroll_course() • get_courses_enrolled() • view/edit profile
TEACHER	Class state • Name • email Class Behaviour • Login() • Tutor_registration() • Trackstudent_progress() • Break_modules() • view/edit profile
ADMIN	Class State

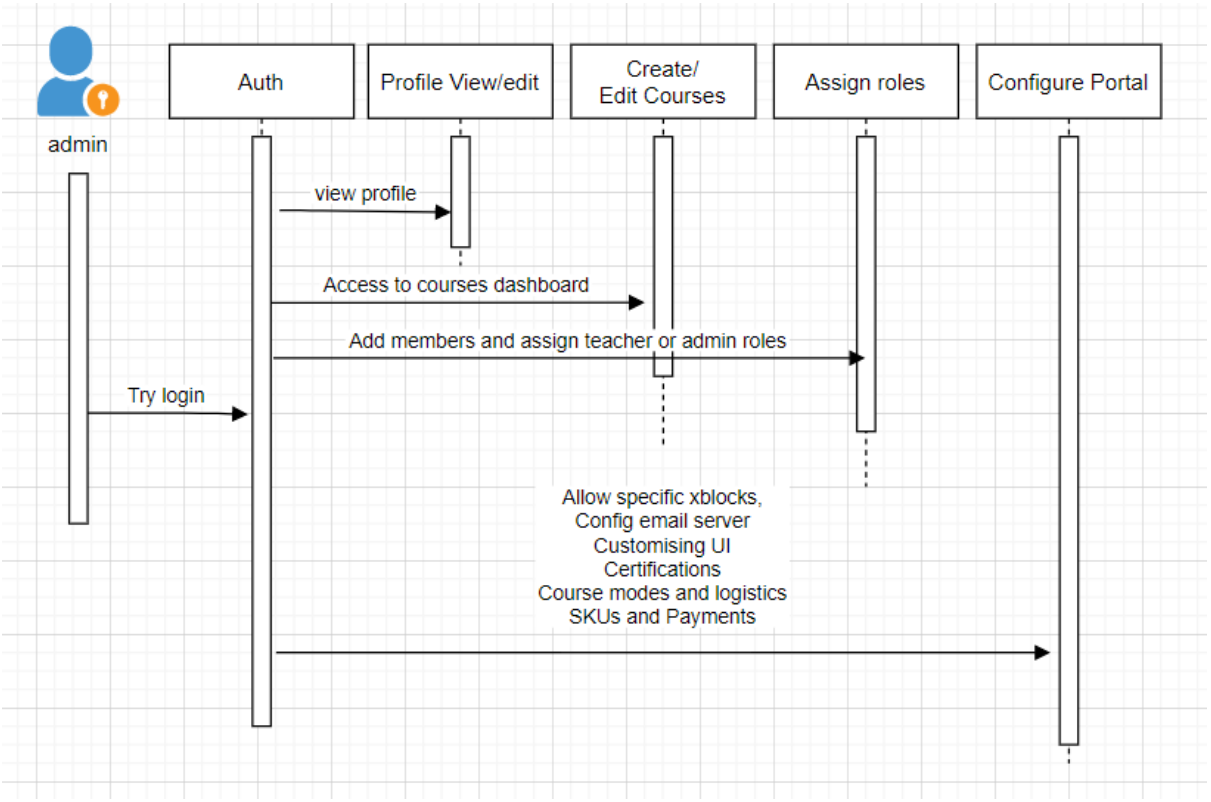
	• Name • email Class Behaviour • Auth • viewProfile • editProfile • createCourse • editCourse • addContent • assignRoles • configurePortal
COURSES	Class state • Course name • TeacherID • Description Class Behaviour • Course_modules() • AddCourses()
TOPICS	Class state • CourseID • TopicName • Topic URL Class Behaviour • AddTopic() • EditContent()

Sequence Diagrams

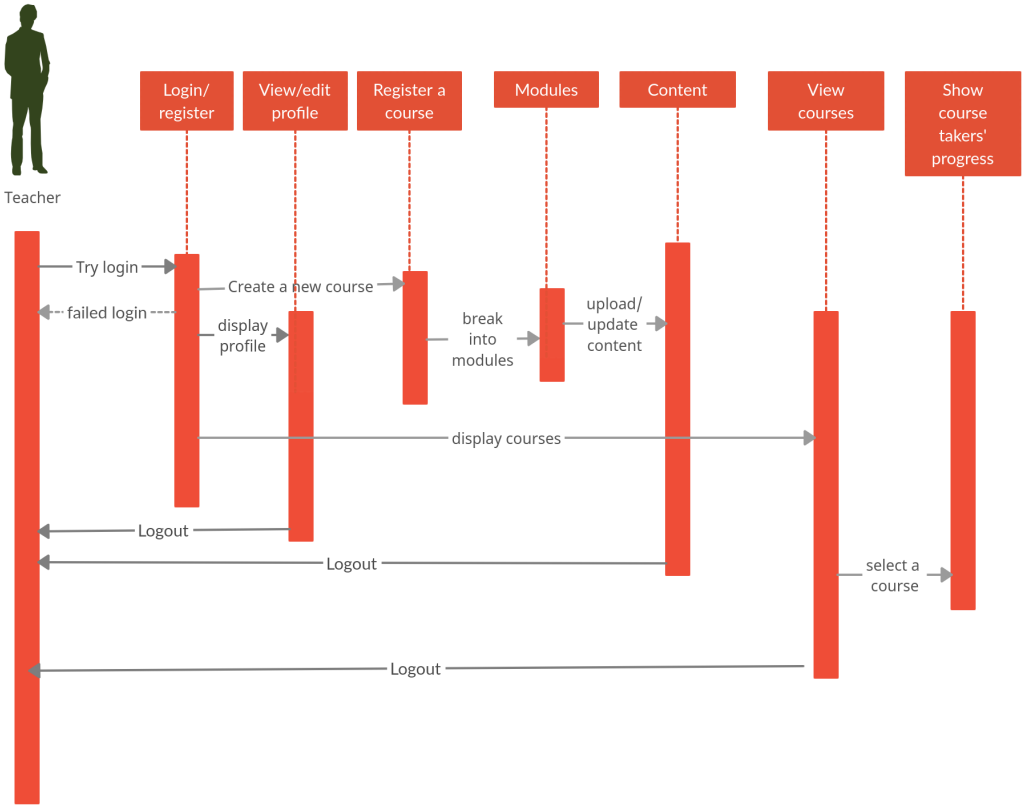
1. For student



2. For admin



3. Teacher



Design Rationale

1. Initially the client proposed that the backend be built using .NET. However, due to limited object relational support, the idea was dropped and instead node js was used for backend
2. Initially the client gave the option of using either SQL or Nosql databases but due more scalability and deep query ability, mongodb was chosen over SQL.
3. The client's initial proposal was that we build the entire LMS from scratch. However, later on the client proposed that we use an open source LMS instead so the final LMS has been built on tutor, a distribution of Openedx.