IT314 Software Engineering

Lab 6

Date: 23-03-2023

Group - 13

- **Group Members:**

ID No	Name
202001133	DANKHARA MANAN SHAILESHKUMAR
202001163	SHAH ISHA JATINKUMAR
202001149	BUCH KAVAN JAYESHBHAI
202001168	PRAJAPATI SHREYA
202001135	PATEL DHARVA YOGESHKUMAR
202001148	BRIJRAJSINH GOHIL
202001137	MAKWANA KIRTAN DHARMENDRABHAI
202001152	SANEPARA KHUSHI SURESHBHAI
202001155	RAMOLIYA HARSH MADHUKANTBHAI

❖ Domain Analysis Model:

1. Introduction:

The domain for the system is “Providing Educational content” to students.The motive behind developing such kind of system is to provide better quality educational content at one place with ease of use.

2. Glossary:

Most of the terms that are used are related to educational domain,but Some terminologies also include technical domain.

3. General Knowledge about the domain:

As a student he/she wants to see the different courses available and their Quality Content arranged in a staggered manner or you-tube videos for better understanding and as an educator he/she will be happy to share his/her knowledge to the whole student community.

System will be able to handle client from any regions and able to handle as much as large number of requests coming across the globe.

4. Users:

The users that are using this systems are mostly common students and instructors who want to share their valuable knowledge and grow up student community.

5. Competing Software:

There are many softwares are available across the world and our system should be able to compete with this existing systems in terms of scalability and reliability and availability.

❖Identify boundary, entity, control object:

1. Boundary Object:

- **Dashboard:**

It represents the interface between the user and the educational content system.

It provides content that user have created for different courses,and additionally to manage that content and how many views it is getting and what is rating of that content.it is a like a boundary between user and educational system.

- **Login Page:**

Login Page eventually provides user to log in and manage their personal information and manage their content,So In some sense Login Page is providing visual boundary between user and system.

2. Entity Object:

Admin: Admin will verify the user detail and create course and remove it

Student: Student can sign up/ log in on platform. s/he can access to any course content.

Educator: Educator can sign up/ log in on platform. s/he can upload course material as youtube link, documents (pdf).

Course: It has list of topic included in that. For each topic, there will be educator username, uploaded content by him and rating.

Discussion Forum: This will have question which uploaded by students, many user can reply to these questions.

3. Control Object:

- **Verification:** System verify the log-in which user have entered. This controls that correct user will be logged in.

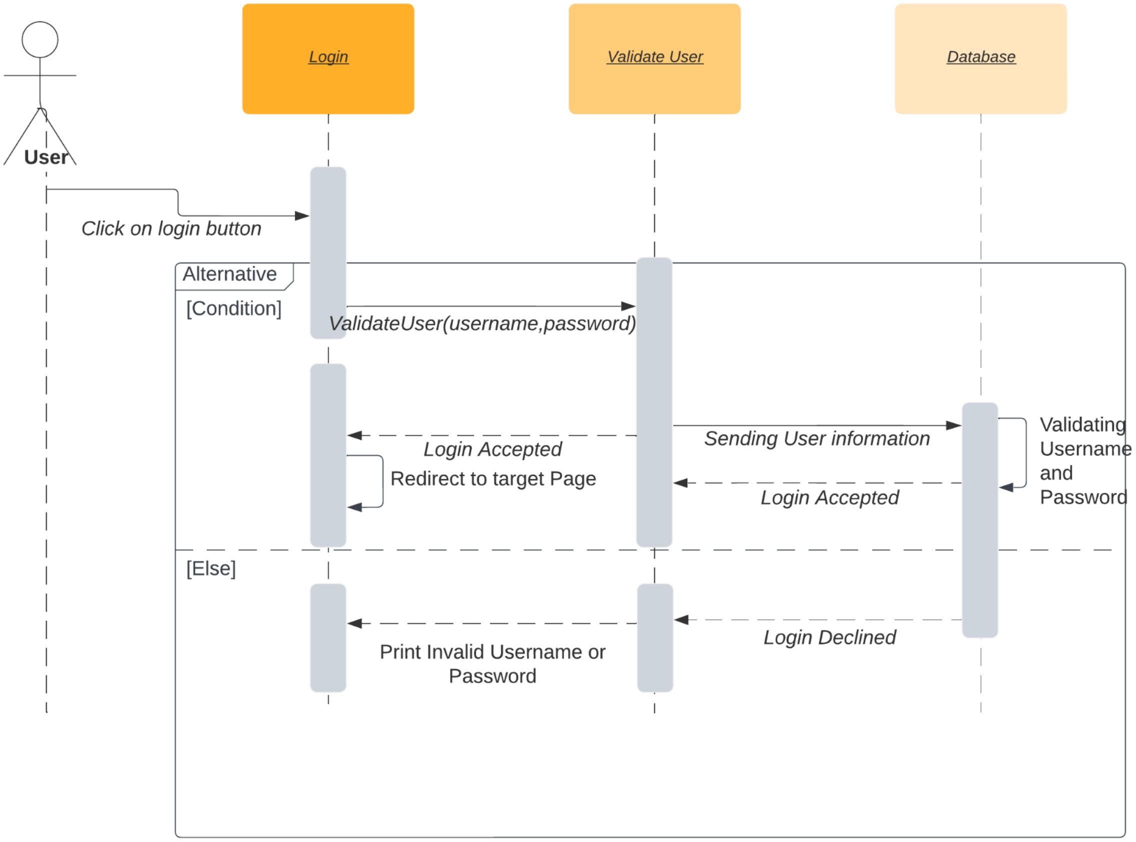
- **Search bar:** It allows users to search for a specific course. Because user control system’s behavior of what content to be shown.

- **Navigation:** This control object allows users to navigate to different sections of the platform such as courses, Q&A etc. It helps to make it easier for users to find what they are looking for.

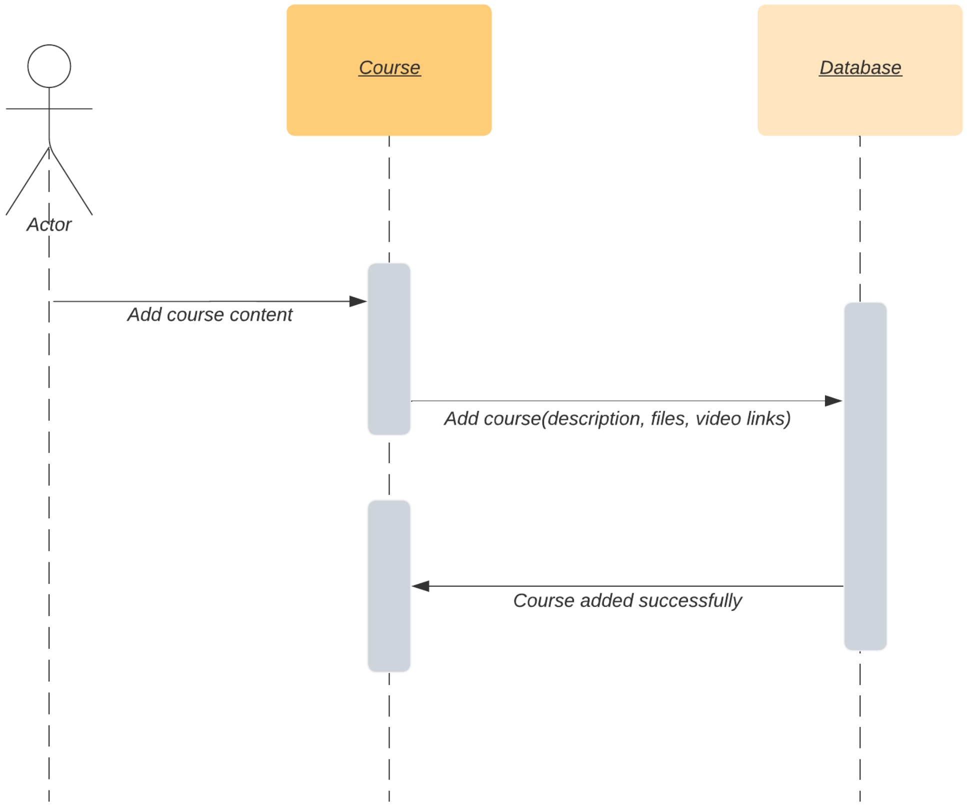
- **Rating system:** It controls the visibility and popularity of the content oon website.This allows users to rate the educational content available on platform.
- **Q&A section**: It controls relevance and quality of educational content.It allows users to ask doubts and answer questions.

- **Sequence Diagram:**

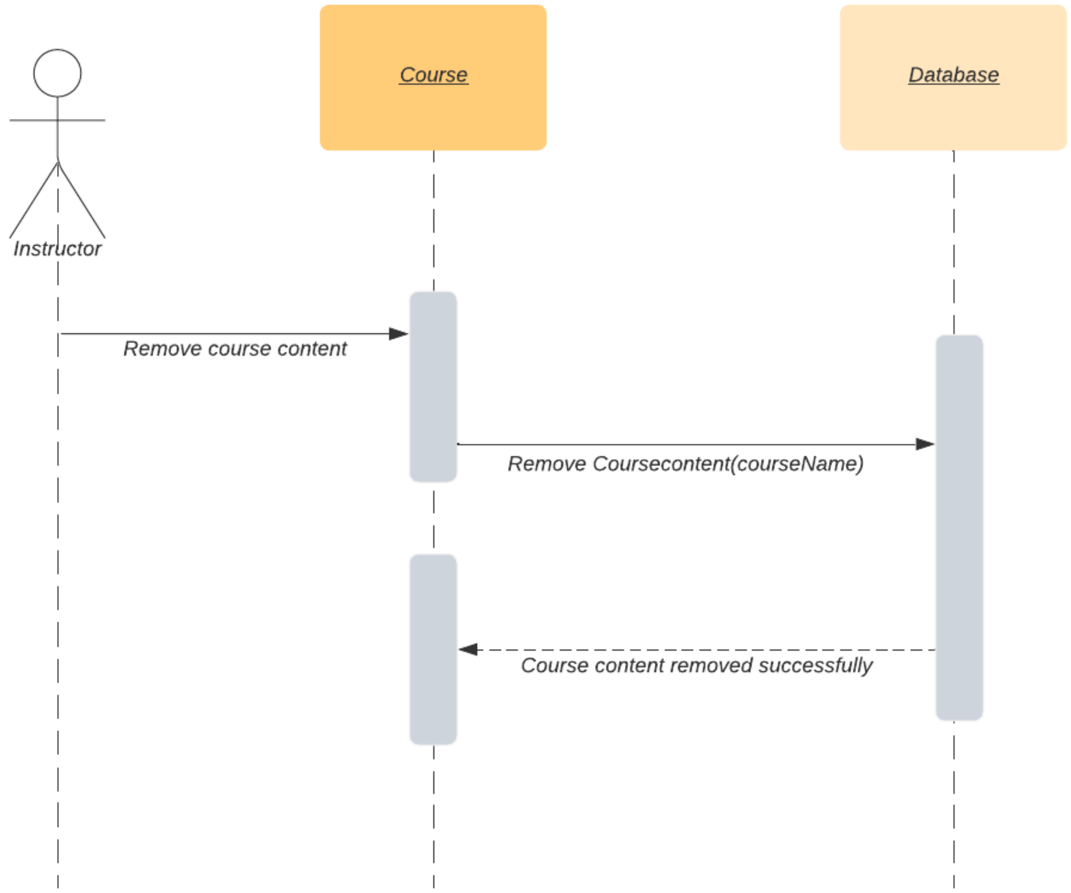
1. Login:



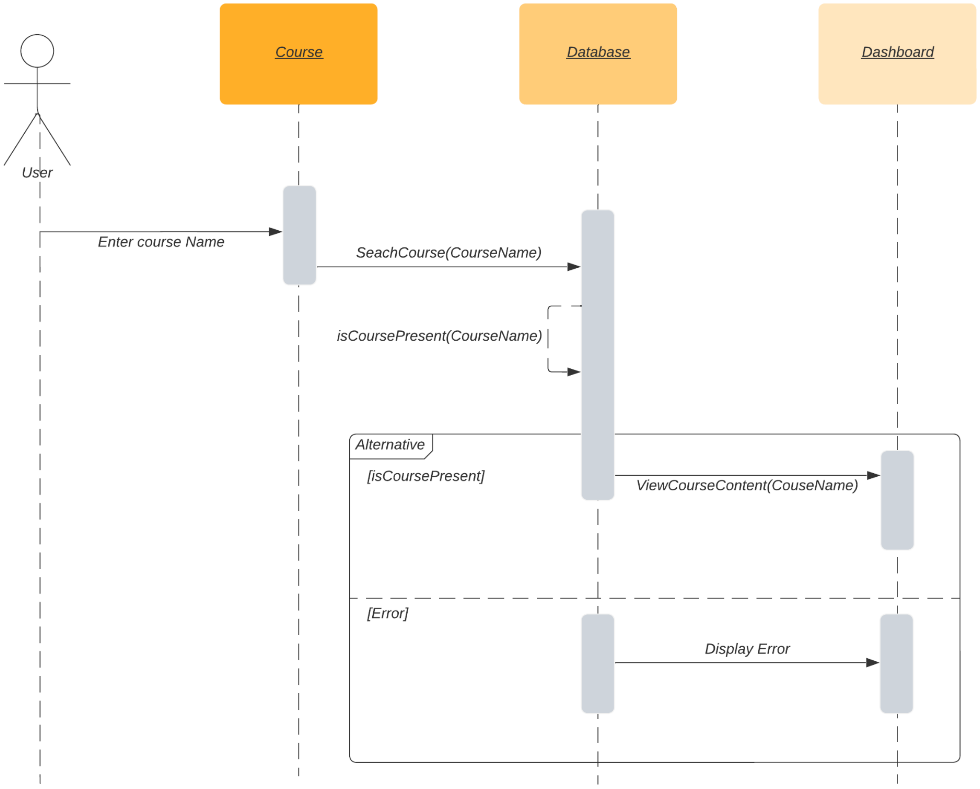
2. Adding Course content:


Course
Database
Actor
Add course content
Add course(description, files, video links)
Course added successfully

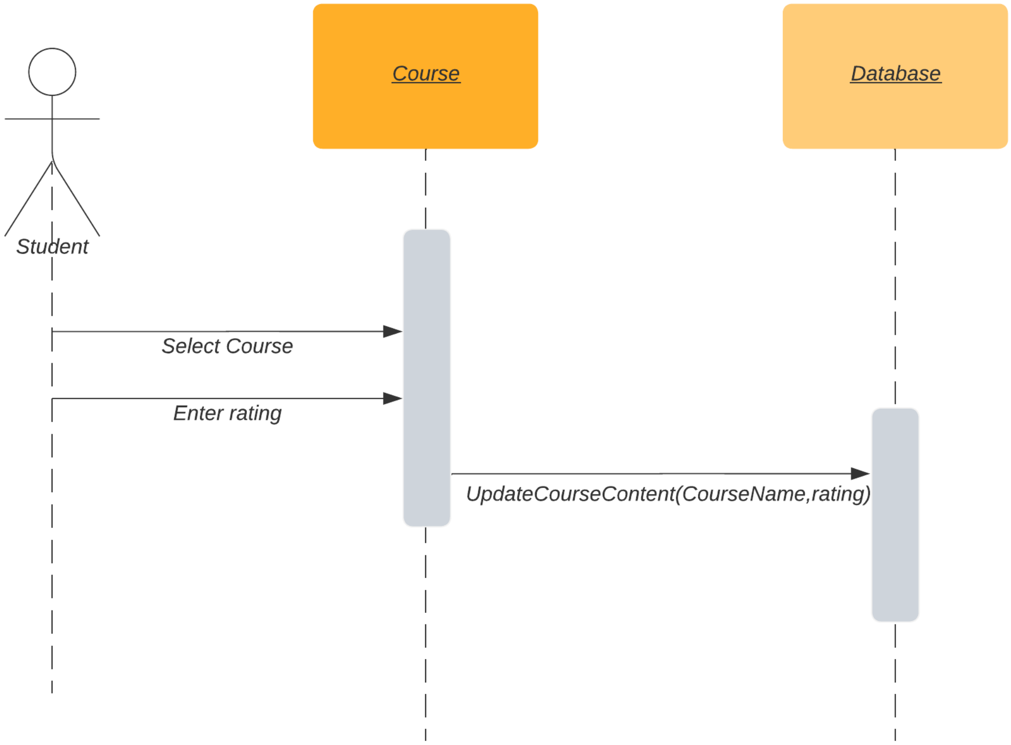
3. Remove course content:


Course
Database
Instructor
Remove course content
Remove Coursecontent(courseName)
Course content removed successfully

4. Search Course:

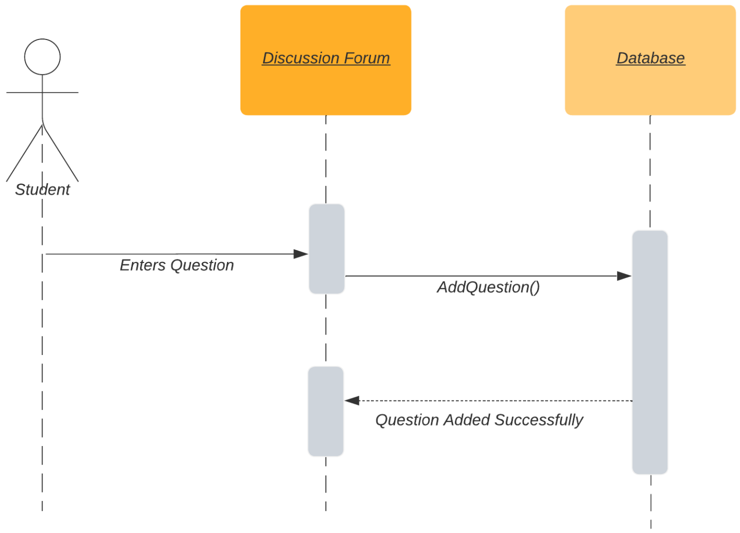


5. Rating & Feedback:

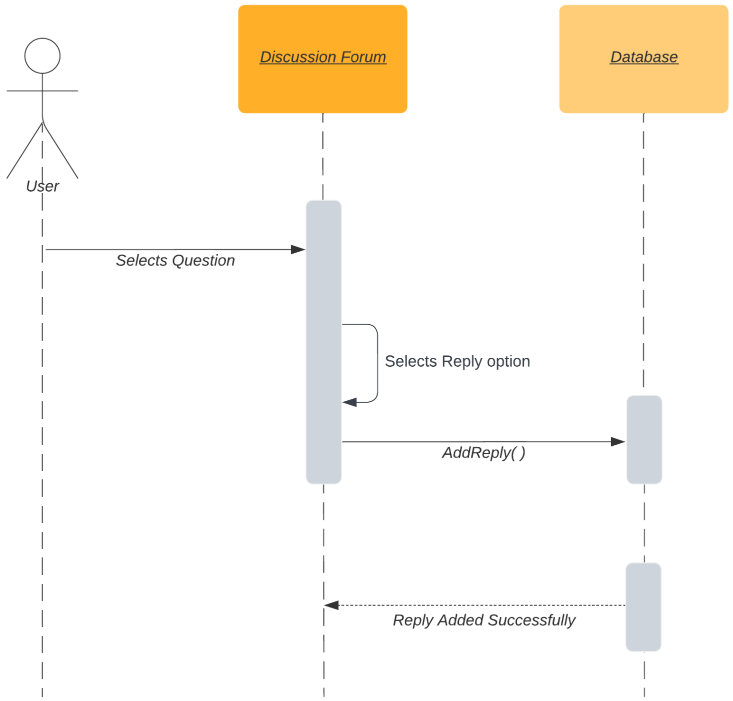


6. Discussion Forum:

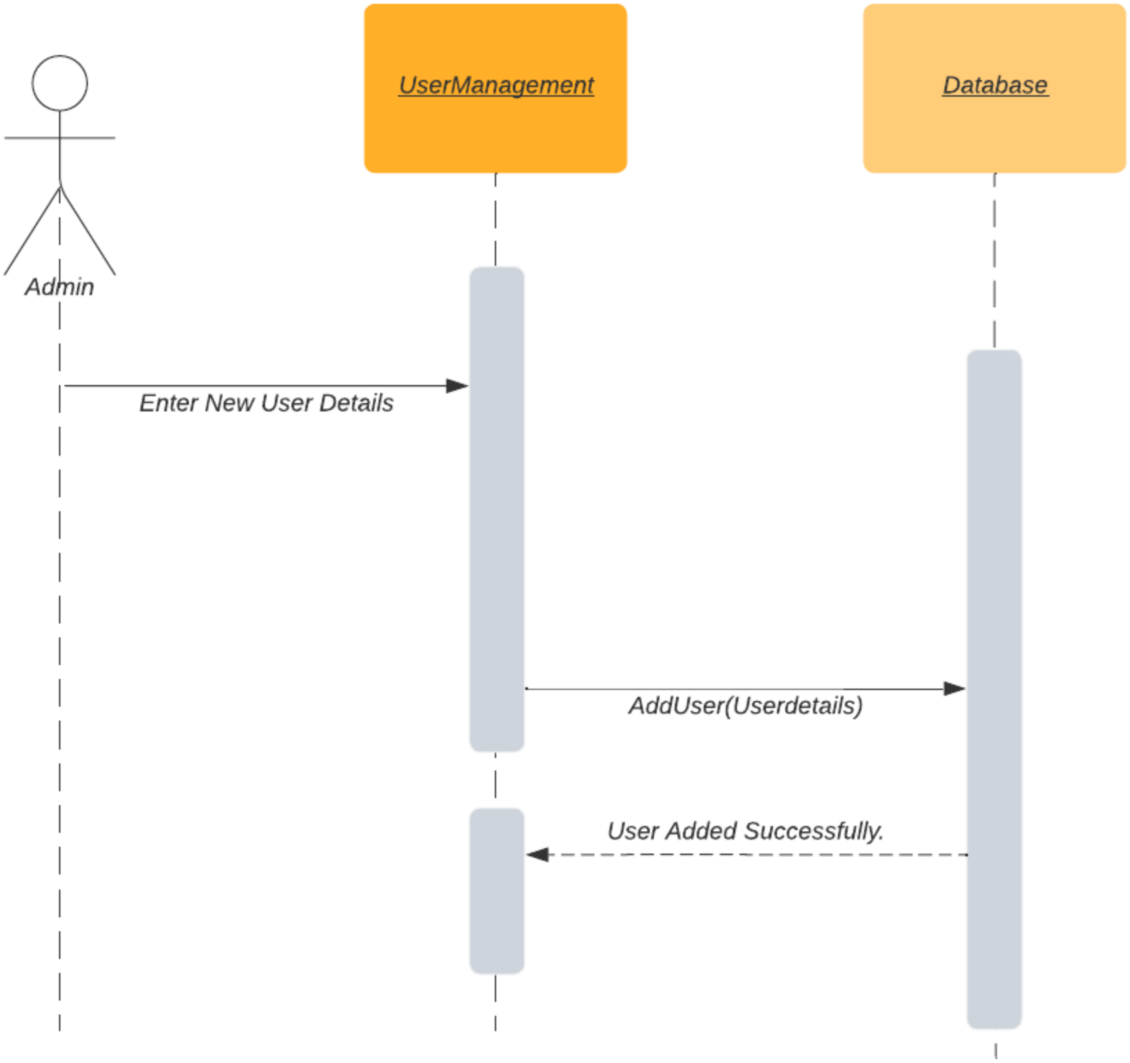
(i) Posting Doubts



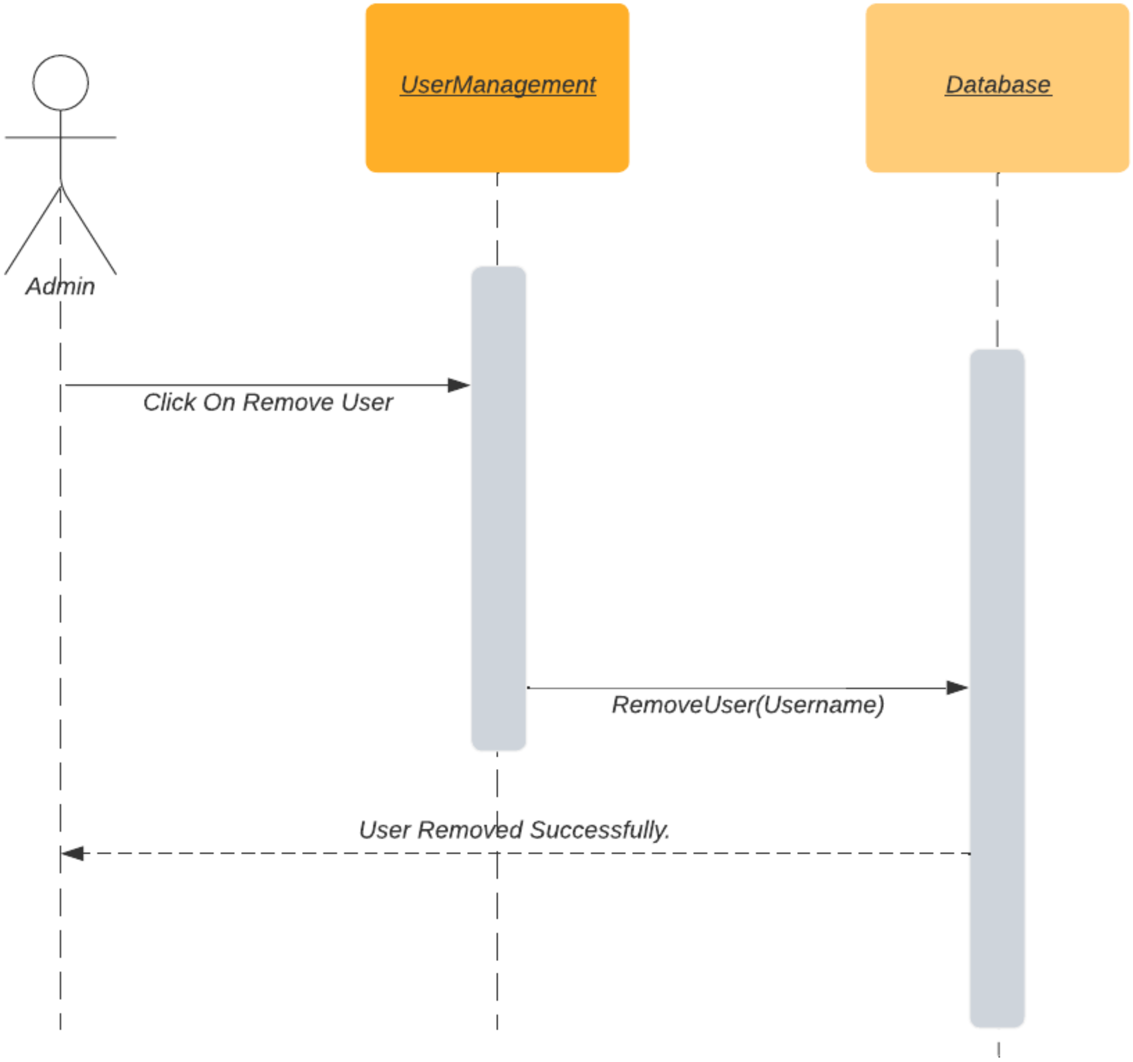
(ii) replying doubts



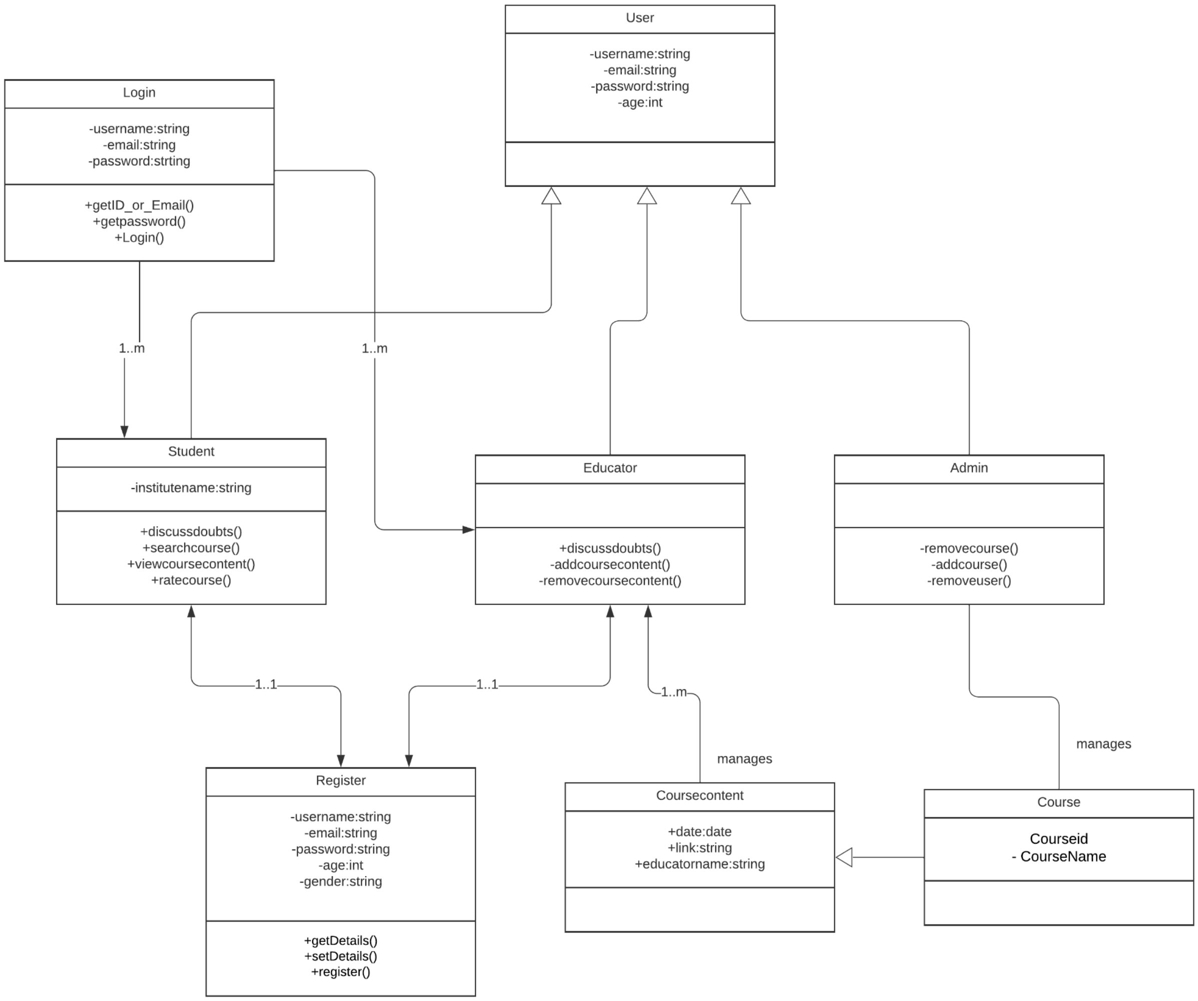
7. Add User:


UserManagement
Database
Admin
Enter New User Details
AddUser(Userdetails)
User Added Successfully.

8. Remove User:



- **UML class diagram:**



❖Design Goals

- **Maintenance and Dependability Criteria:**

    - **Functionality:** The software should have all the necessary features to allow all required stakeholders (like admin) to add a new course, remove users and (like users) to add course content, rate the course etc..

    - **Availability:** The system should run constantly, with little downtime for upgrades and maintenance.

    - **Scalability:** The system can adapt to change with increasing number of user requests without any lag time.

    - **Security:** The system should secure sensitive data like users password from unauthorized access and it should be authenticated using suitable safeguards with proper encryption.

    - **Completeness:** The design had all components such as data structures, modules, external interfaces etc.

    - **Robustness:** Errors and incorrect inputs should be handled very effectively by the system during system execution.
    - **Flexibility:** If at a later time, the number of users using the system increases, the system should be flexible for a wide range of user interfaces.

- **Performance Criteria:**

    - **Response Time:** The system should respond to users actions immediately within no time .

    - **Throughput:** System should be able to complete certain tasks in some stipulated time without making users wait much longer.

- **End- User Criteria:**

    - **Utility:** System should be such that it can easily handle the end-user's work.

    - **Usability:** The interface of the system should be simple and attractive which can be used by any user without requiring any special training or technical knowledge regarding that, making them easy to use.

❖ High Level System Design

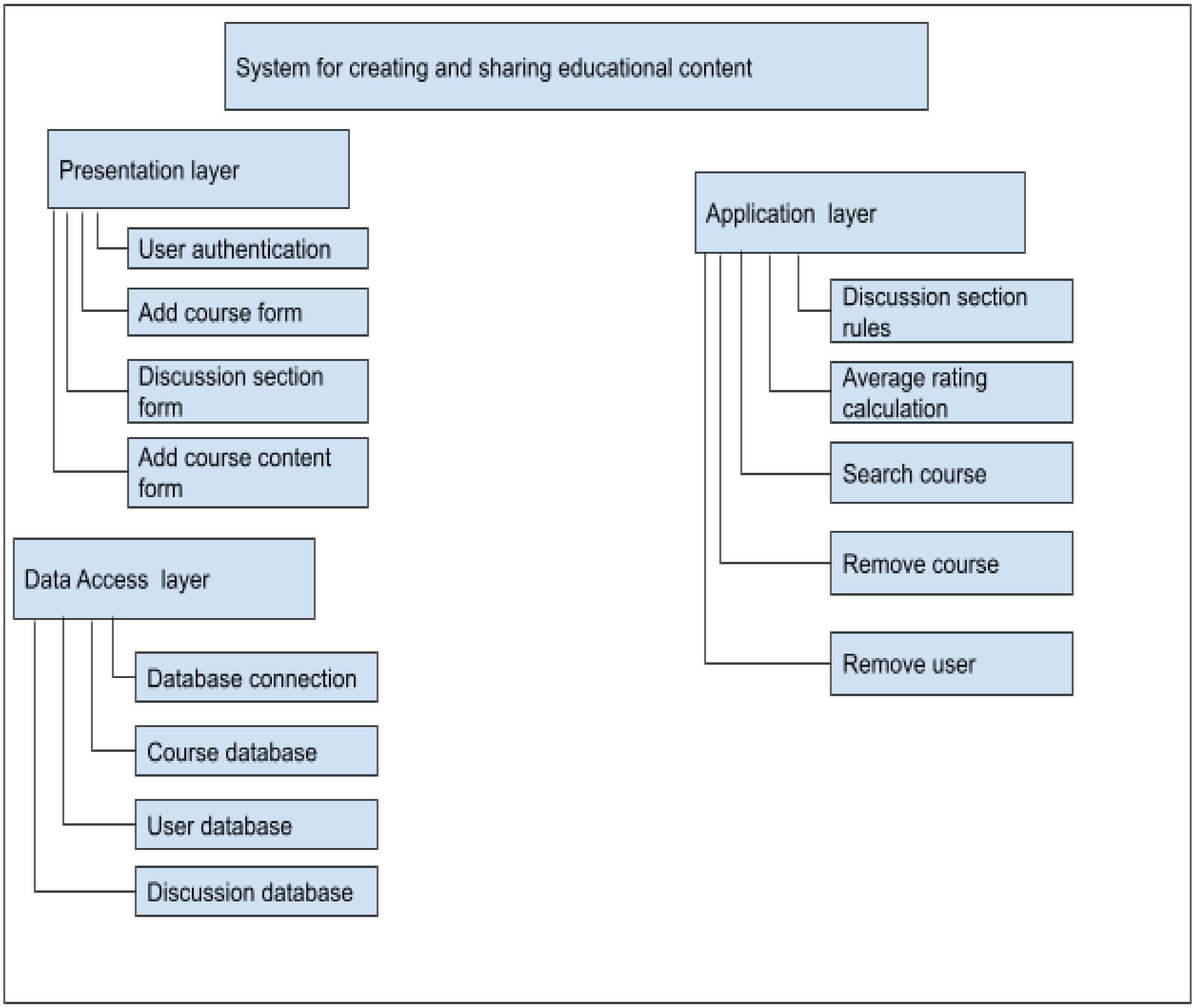
→ For creating and sharing education content, we are using **three-tier architecture** according to our model with three layers as:

- Presentation layer
- Business layer
- Data access layer

❖ Why three-tier architecture?

Three-tier architecture is well established software application architecture that organizes applications into three separate logical and physical layers. Here, each tier runs on its own infrastructure, each tier can be developed simultaneously and can be updated or scaled as needed without impacting any other tier.

Three-tier architecture model for our system.



- **The subsystems for three-tier architecture:**

  1. **Presentation layer:**

     → This tier is the user interface and communication layer of the application where the end user interacts with the application. Its main purpose is to collect information from users and give output to users.
     → This subsystem handles user authentication and viewing different sections like course content page, discussion section and many more.
     → It includes user authentication which verifies user credentials, the add course form which allows admin to add course, add course content form which allows user to add content to particular course and discussion section which allows any user to ask doubts or give reply to the asked doubts.

  2. **Application Layer**

     → In this tier, information collected from the presentation layer is processed against some rules within the data tier. It can also add, modify or remove any data.
     → It includes discussion section rules which appends answers of the same question beneath that only, average rating calculation rule which always creates an average rating when any user adds their feedback, search course that handles searching process for particular course and remove course,

remove user which handles removing any course or user from existing database.

3. **Data Access Layer:**
   → This layer is sometimes called a database layer where information processed by the application is stored and managed. Here, we are using the MongoDb database to store data.
   → It mainly includes user data where all information of the user is stored, a course database where all information regarding a particular course like their creators, rating etc are stored. It also contains a discussion database where every doubt is stored with their multiple answers.

Overall, this architecture provides a clear separation of concerns and allows for easy maintenance and scalability of the system.