

# CourseMind

## Team Members

Parshwa Shah (MLOps)

Foram Shah (Backend Development)

Yashodhan Jaltare (Backend Development)

Rucha Keni (Frontend Development)

Hitakshi Shirude (Frontend Development)

## System Description:

- CourseMind is a web application which is created to support professors in managing courses, assignments and quizzes. It provides a unified platform with easy interactions that enable professors to manage the planning of courses and deliver course materials efficiently.
- This application will simplify the creation of quizzes, assignments, given the course outlines and materials, saving professors time while ensuring consistency and quality in their content delivery.
- In this way, CourseMind frees the professors from these administrative tasks so they could pay more attention to teaching and meaningful engagement with their students.

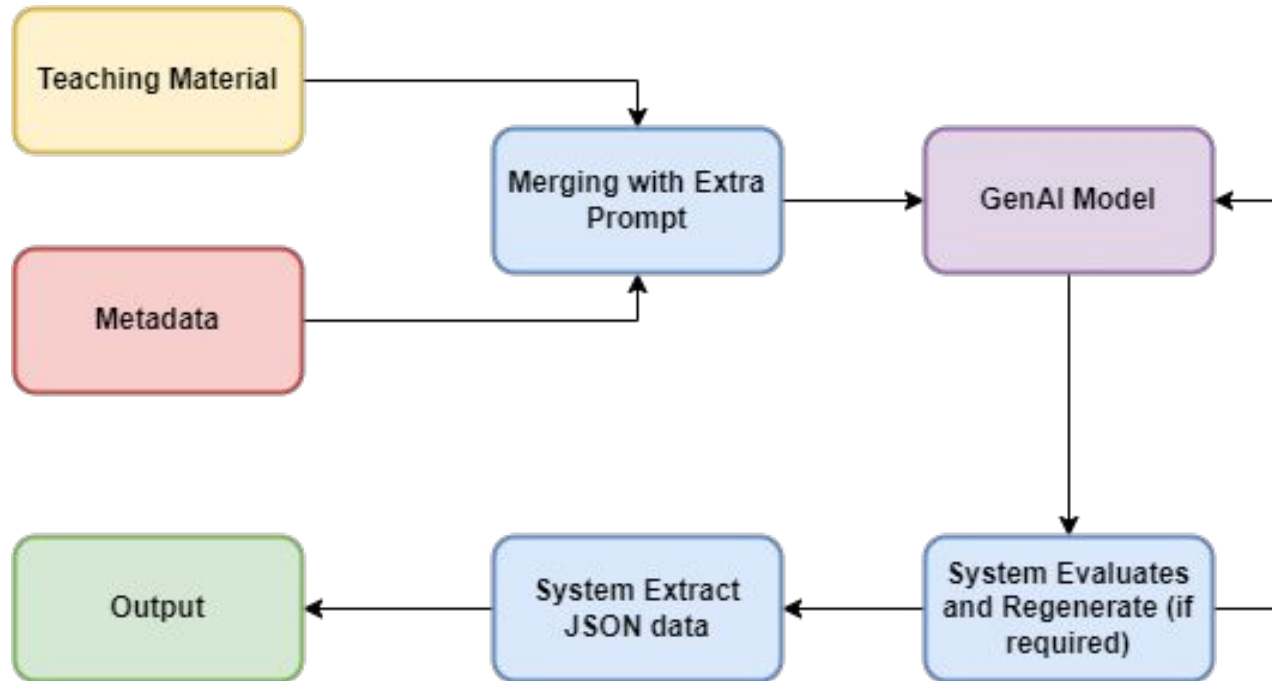
## Scope:

- Develop a user-friendly interface for professors to create and manage their courses.
- Build a question bank for quizzes and assignments with the ability of auto-generation.
- Enable professors to add, update, delete new course assignments and quizzes.
- Allow course content and assignments to be exported to share with others.

## Objectives:

- Provide an easy-to-use dashboard for course management.
- Build a dynamic question generation system for quizzes and assignments to assist the professor.
- Allow export of course quizzes and assignments to share with others.
- Ensure basic authentication and user roles (professor only).

## How we actually generate this?



# User Stories

## 1. Feature: Generate examples for students to understand topics User Stories

**As a teacher,**

**I want to,** generate examples of a particular topic from particular material,

**So that,** I can teach some topic easily to students

## 2. Feature: Generate questions for students to assess them on various topics User Stories

**As a teacher,**

**I want to,** generate questions of a particular type from particular material,

**So that,** I can assess students with variety of questions

## 3. Feature: Generate questions for students to assess them on various topics User Stories

**As a teacher,**

**I want to,** generate short assignments from particular material,

**So that,** I can give all students different assignments for better assessment quality.

# User Stories

## 4. Feature: Manage Courses

**As a teacher,**

**I want to,** add, update and delete courses

**So that,** I can manage my courses on the system

## 5. Feature: Authentication

**As a Teacher,**

**I want** to keep my courses and my materials separate than others',

**So that** I can manage my data easily with other fellows.

## 6. Feature: Select Courses

**As a teacher,**

**I want to,** select a course from the list of courses I am teaching,

**So that,** I can open the course material

# User Stories

## 7. Feature: Manage Quizzes

**As a** teacher,

**I want to**, select a particular course from the list of courses I am teaching,  
**that**, I can manage that course material

**So**

## 8. Feature: Managing Assignments

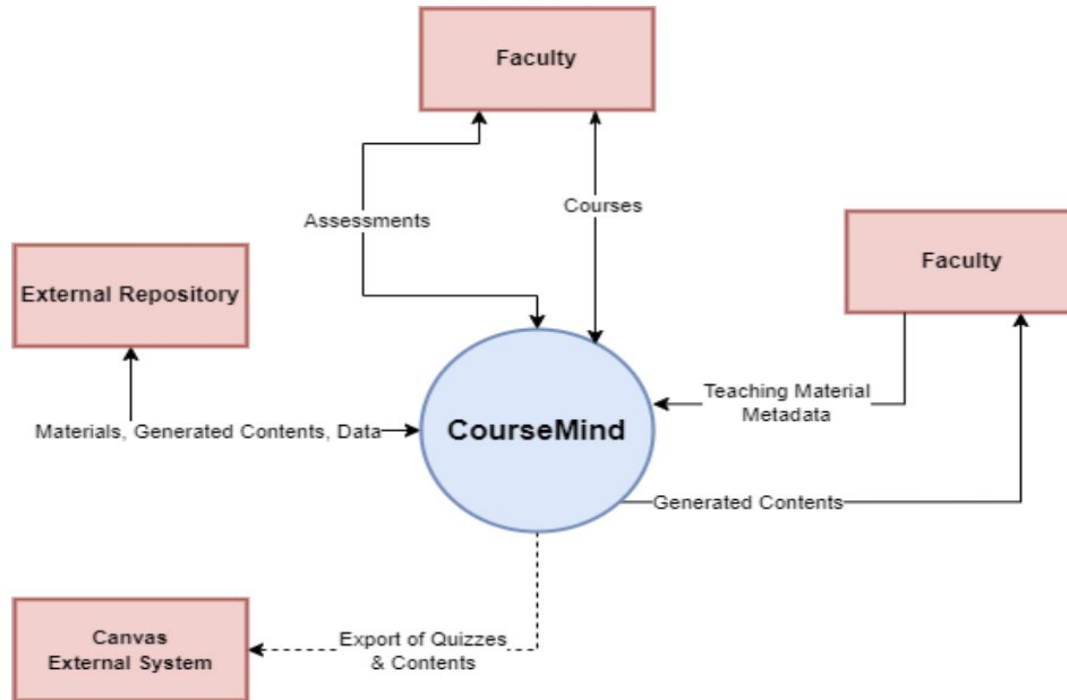
**User Story:**

**As a** Teacher,

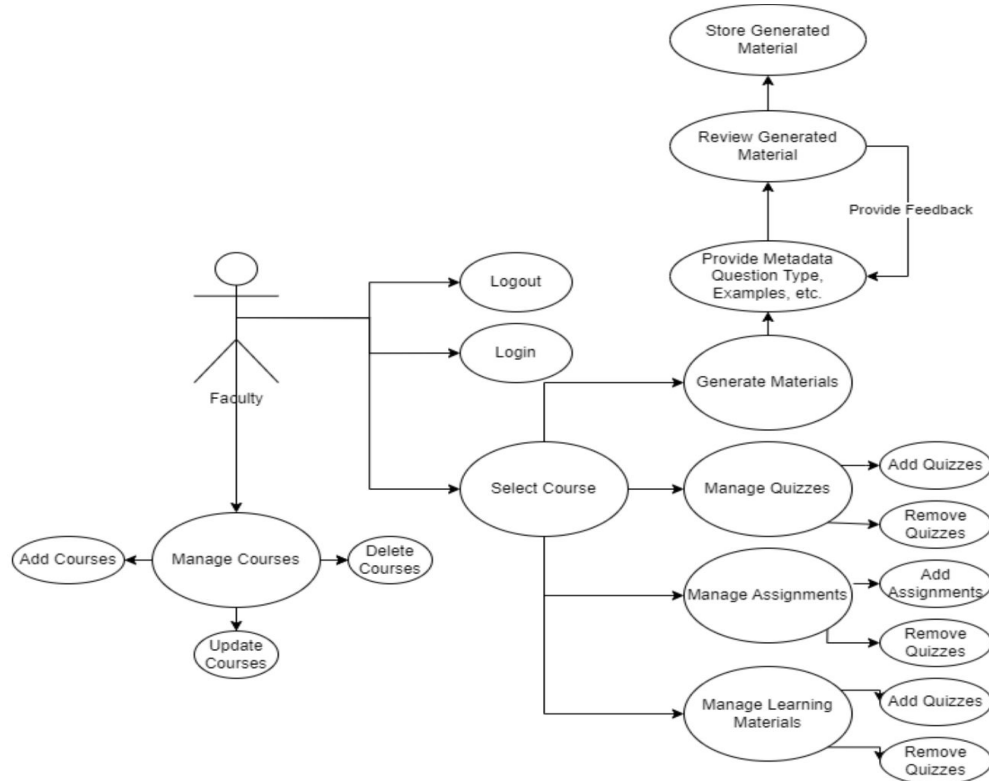
**I want** to manage assignments for my courses

**so that** I can add, delete and update assignments to keep the course materials up to date.

# Context Diagram

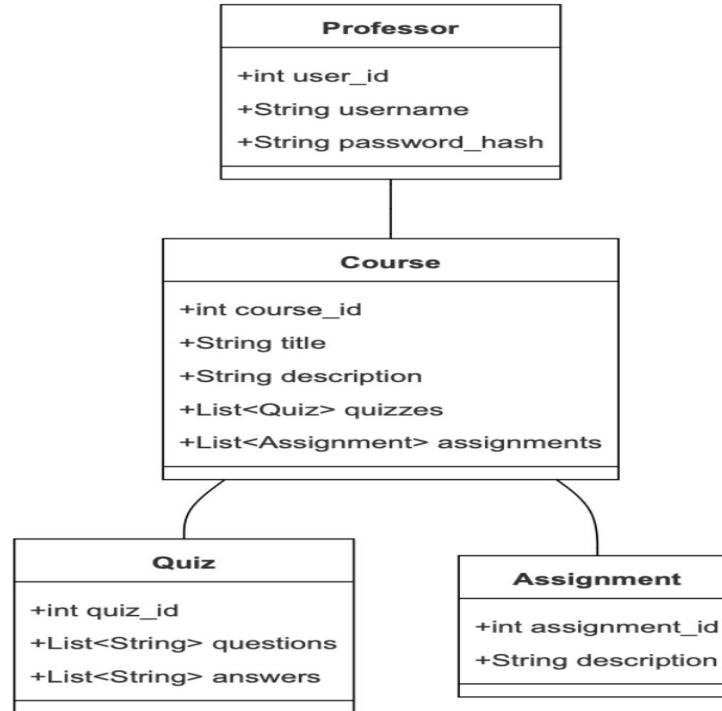


# Use Case Diagram





# Database Diagram



# Database Model

## 1. Users Collection

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for each user
user_type	String	Type of user (e.g., "professor", "student")
name	String	Name of the user
email	String	Email of the user
password	String	Hashed password
courses	Array of ObjectIds	List of course IDs the user is associated with

## 2. Courses Collection

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for each course
professor_id	ObjectId	Reference to the professor (from Users)
course_name	String	Name of the course
description	String	Course description
modules	Array of ObjectIds	List of module IDs associated with the course
created_at	ISODate	Date the course was created

## 3. Modules Collection

Field Name	Data Type	Description
_id	ObjectId	Unique identifier for each module
course_id	ObjectId	Reference to the course (from Courses)
module_name	String	Name of the module
description	String	Description of the module
documents	Array of Objects	List of documents (e.g., name and URL)

Field Name	Data Type	Description
assignments	Array of ObjectIds	List of assignment IDs associated with the module
quizzes	Array of ObjectIds	List of quiz IDs associated with the module
created_at	ISODate	Date the module was created

# Database Model

## 4. Assignments Collection

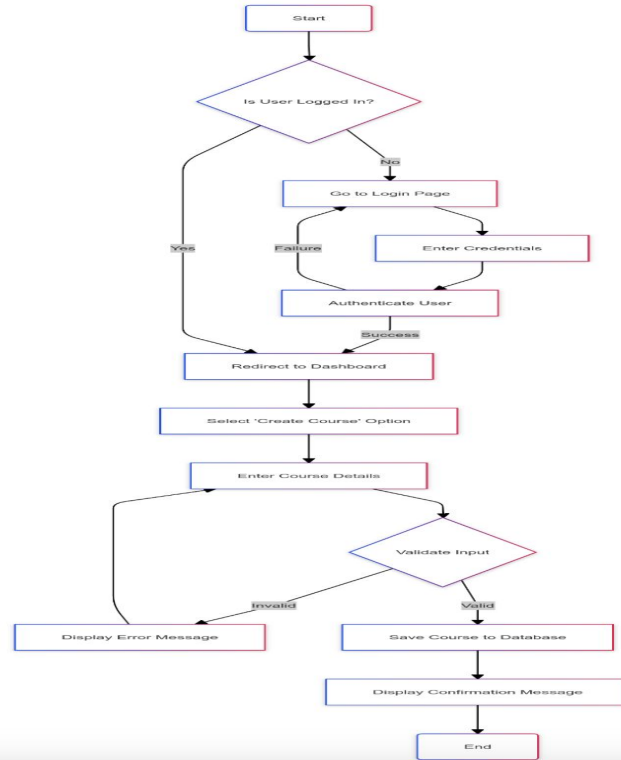
Field Name	Data Type	Description
_id	ObjectId	Unique identifier for each assignment
module_id	ObjectId	Reference to the module (from Modules)
title	String	Title of the assignment
description	String	Description of the assignment
due_date	ISODate	Due date of the assignment
file_url	String	URL to the assignment file
created_at	ISODate	Date the assignment was created

## 5. Quizzes Collection

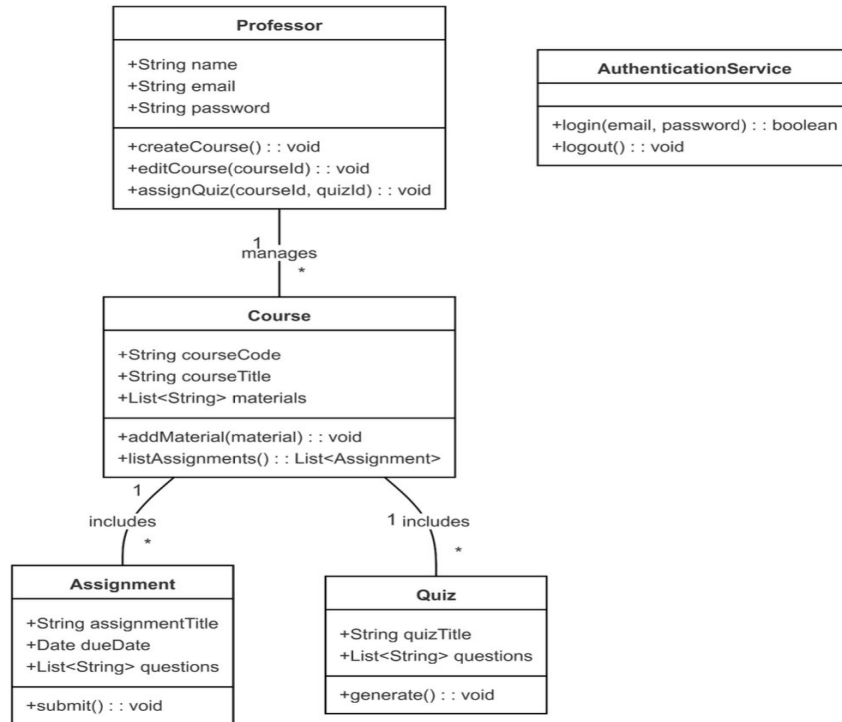
Field Name	Data Type	Description
_id	ObjectId	Unique identifier for each quiz
module_id	ObjectId	Reference to the module (from Modules)
quiz_name	String	Name of the quiz
questions	Array of Objects	List of questions, options, and correct answers
created_at	ISODate	Date the quiz was created

# Other UML Diagrams

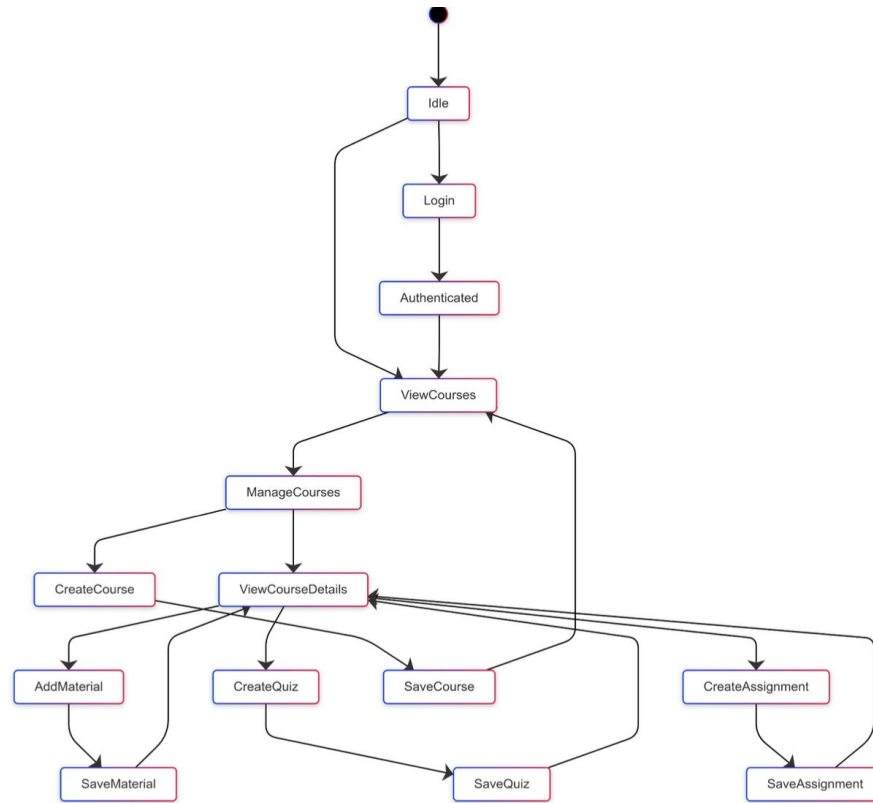
# Activity Diagram



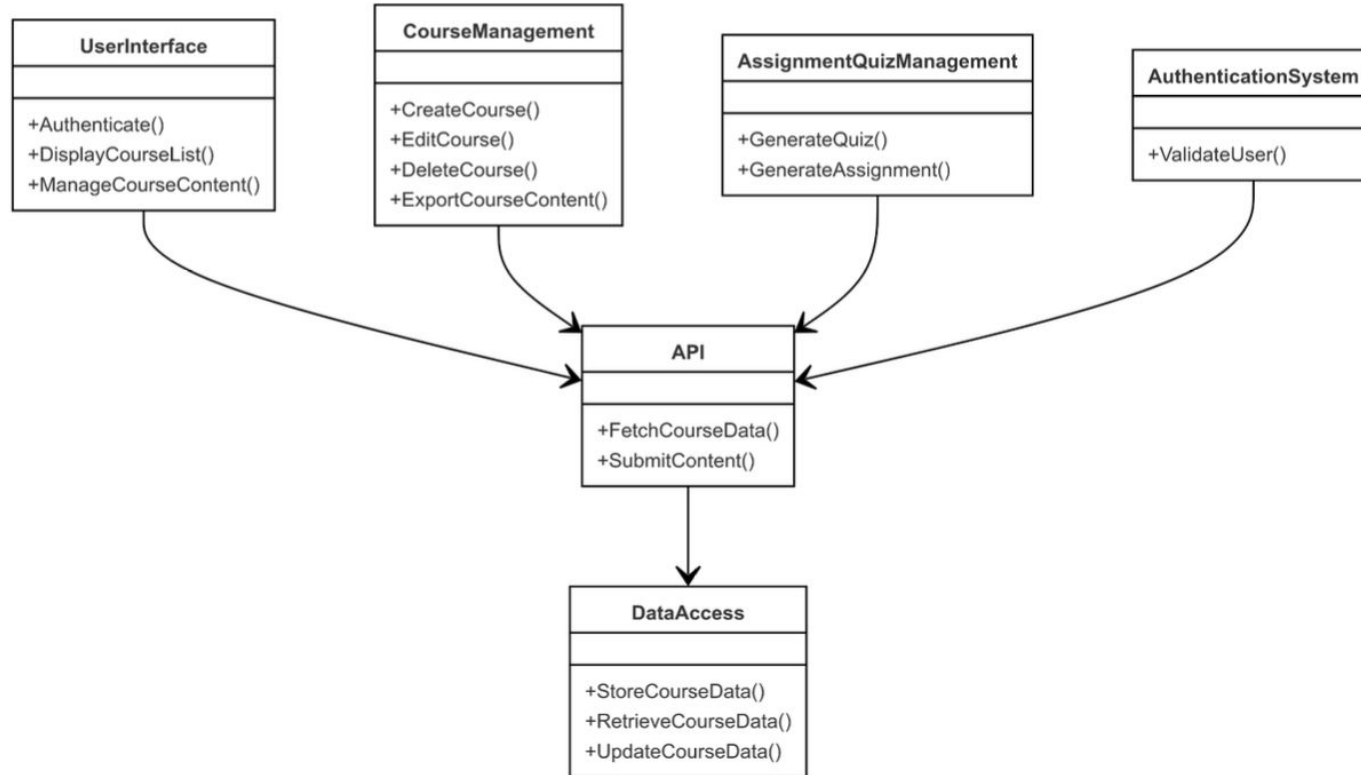
# Class Diagram



# State Diagram

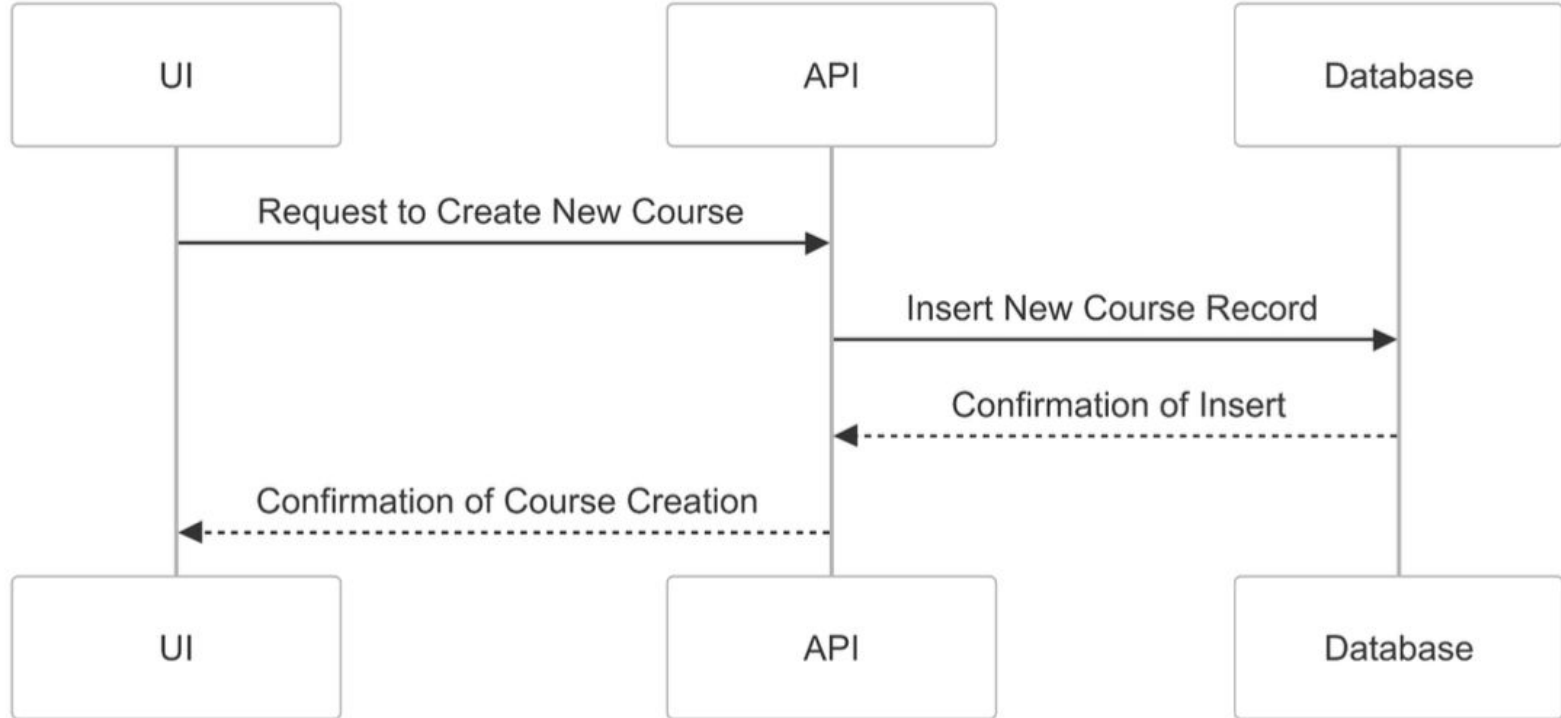


# Component Diagram





# Collaboration Diagram



State of development using product backlog:  
How many sprints are done so far  
how many requirements are implemented ?

## TimeLine Chart

**This is our timeline chart, how the project work is allocated and the timeline when it should be completed**



# Development Status

**Sprint 1:** Includes the following tasks:

- Task 1: Authentication - UI
- Task 2: Dashboard with list of courses - UI
- Task 3: Inner page for list of assignments and quizzes for each course - UI
- Task 4: Authentication - backend API
- Task 5: Integration of Authentication UI & API
- Task 6: Get a model to create questions from given context - AI Model (POC)

**Sprint 2:** Includes the following tasks:

- Task 7: Create a new course - UI
- Task 8: CRUD for courses - backend API
- Task 9: Integration of Courses CRUD UI & API
- Task 10: Create new quizzes - UI
- Task 11: Create new assignment - UI
- Task 12: CRUD for quizzes - backend API
- Task 13: CRUD for assignment - backend API
- Task 14: Integration of Quizzes CRUD UI & API
- Task 15: Create database schema - backend (structuring)

## Development Status

**Sprint 3:** Includes the following tasks:

- Task 16: Add new materials for course - UI
- Task 17: CRUD for materials for course - backend API
- Task 18: Use data chunks to find key information in chunks - AI Model
- Task 19: Integration of Assignments CRUD UI & API

**Sprint 4:** Includes the following tasks:

- Task 20: Convert document to data chunks - AI Model
- Task 21: Get a model to generate assignments or short projects problems for course - AI Model
- Task 22: Integration of AI Model to feed new materials
- Task 23: Integration of AI Model to generate new thin

## How many sprints are done so far?

**Total Tasks Implemented and Sprint Completed So Far are the following:**

- **Sprint 1:** 6 tasks
- **Sprint 2:** 9 tasks
- **Sprint 3:** 4 tasks

**Number of sprints done so far:** 3 sprints (Sprint 1, 2 and 3).

# Lessons learnt

## **1. Project Planning and Scope Management-**

Lesson Learnt: The project scope should be clearly defined and then divided into decent tasks. A few of them took longer than expected; hence, this caused a delay in the sprint timeline.

In our project, The AI models took longer to integrate because it was quite a complicated process in trying to ensure the model would generate meaningful content from course materials. Much of this would have gone better because of better scoping and an earlier focus on research into AI.

## **2. User-Interface Design and Usability-**

Lesson Learned: During the design phase, feedback from people who would actually be using the device was needed. In our project, The prototype of the course dashboard UI was functional but not intuitive. The feedback received after presenting it to users indicated that simplification of the interface to only provide focus to vital functionalities would facilitate better use. In other words, a design had to be iterated based upon user input.

# Lessons learnt

## **3. AI Model Development-**

Lesson Learned: Development of AI models involves iterative testing and fine-tuning with regard to relevance in terms of delivered results.

In our project, While the initial model for AI-generated questions worked, it actually required a number of iterations and fine-tuning in order to return useful questions. Herein, we learned that testing and feedback loops are quite essential in refining features driven by AI.

## **4. API Integration-**

Lesson Learned: Clear and consistent API documentation is key in smooth integration, particularly since UI, backend, and AI are three intertwined systems here.

In our project, In API integrations between frontend and backend, mismatches in data structure and format of requests/responses led to some delays. In this regard, early documentation and testing of APIs between teams could have smoothened the process.



# Next Step

**Feature Additions as per Sprint 3**

**Feature integration**

**ML Model integration**

**Testing**

# GitHub Link and Demo Link

GitHub Link: <https://github.com/ParshwaS/CourseMind>

Demo - [https://drive.google.com/file/d/1WGWUU2Rz6pms-hKPUG4urdp4DORvbZZ6/view?usp=drive\\_link](https://drive.google.com/file/d/1WGWUU2Rz6pms-hKPUG4urdp4DORvbZZ6/view?usp=drive_link)