

# CourseMind

**Group 2**

**Date: 10/08/2024**

---

**Authors:**

Parshwa Shah

Foram Shah

Yashodhan Jaltare

Hitakshi Shirude

Rucha Keni

USE CASE NAME:	Generating Examples from teaching materials	USE CASE TYPE	
USE CASE ID:	1	Business Requirements:	<input type="checkbox"/>
PRIORITY:	1	System Analysis:	<input type="checkbox"/>
SOURCE:	Teacher	System Design:	<input checked="" type="checkbox"/>
PRIMARY BUSINESS ACTOR	Teacher who wants to teach some topic to students		
PRIMARY SYSTEM ACTOR	CourseMind System (working on generating examples)		
OTHER PARTICIPATING ACTORS:	<ul style="list-style-type: none"><li>Generative AI Model (part of CourseMind)</li></ul>		
OTHER INTERESTED STAKEHOLDERS:	<ul style="list-style-type: none"><li></li></ul>		
DESCRIPTION:	Focuses on generating examples to teach students from given teaching materials		
PRE-CONDITION:	Teacher has uploaded some teaching materials for the selected course from which Teacher wants CourseMind to create examples.		
TRIGGER:	Teacher wants to teach something and seeks to generate examples		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: Teacher navigates to selected course	Step 4: Gives examples for faculty to filter through	
	Step 2: Selects uploaded material from which teacher wants to create examples		
	Step 3: Gives some details about what type of examples teacher wants		
	Step 5: Selects and exports these examples in JSON		
ALTERNATE COURSES:	If Teacher doesn't like the examples generated by CourseMind, teacher can provide feedback to CourseMind which allows teacher to get different examples		
	If user CourseMind doesn't get any good examples it would return with error message		
CONCLUSION:	Teacher will have examples for particular topic ready to help students with studying		
POST-CONDITION:	Teachers will have examples generated and stored into a database as well as JSON of all examples exported.		
BUSINESS RULES	<ul style="list-style-type: none"><li>Teacher should be logged in</li><li>Should have uploaded some reference material</li></ul>		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"><li>Only authorized users can access</li><li>It creates examples on 1 topic at a time</li></ul>		
ASSUMPTIONS:	<ul style="list-style-type: none"><li>Teachers will provide high quality teaching material to create examples from</li></ul>		
OPEN ISSUES:			

USE CASE NAME:	Generating Question of different types from teaching materials	USE CASE TYPE	
USE CASE ID:	2	Business Requirements:	<input type="checkbox"/>
PRIORITY:	1	System Analysis:	<input type="checkbox"/>
SOURCE:	Teacher	System Design:	<input checked="" type="checkbox"/>
PRIMARY BUSINESS ACTOR	Teacher who wants to assess student’s knowledge on some topic		
PRIMARY SYSTEM ACTOR	CourseMind System (working on generating examples)		
OTHER PARTICIPATING ACTORS:	<ul style="list-style-type: none"><li>Generative AI Model (part of CourseMind)</li></ul>		
OTHER INTERESTED STAKEHOLDERS:	<ul style="list-style-type: none"><li></li></ul>		
DESCRIPTION:	This feature allows teachers to generate questions from a given teaching materials so that they can assess students’ performance		
PRE-CONDITION:	Teacher has uploaded some teaching materials for the selected course from which Teacher wants CourseMind to create questions		
TRIGGER:	Teacher wants to generate quiz questions to assess students		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: Teacher navigates to selected course	Step 4: CourseMind will give question based on metadata and material given	
	Step 2: Selects uploaded material from which the teacher wants to create questions from.		
	Step 3: Gives type of question and number of questions to be generated.		
	Step 5: Selects and exports the complete quiz in QTI format		
ALTERNATE COURSES:	If Teacher doesn’t like the questions generated by CourseMind, teacher can provide feedback to CourseMind which allows teacher to get different questions based on their feedback		
	If user CourseMind doesn’t get the required number of questions it will give however many it was able to generate and add an error saying that was only able to generate some of it.		
CONCLUSION:	Teacher will have a quiz which can be used to assess students’ knowledge on certain topic		
POST-CONDITION:	Teacher will have questions generated and stored into a database as well as exported to Canvas for assessment.		
BUSINESS RULES	<ul style="list-style-type: none"><li>Teacher should be logged in</li><li>Should have uploaded some reference material</li><li>Should provide valid type of question</li></ul>		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"><li>Only authorized users can access</li></ul>		
ASSUMPTIONS:	<ul style="list-style-type: none"><li>Teachers will provide high quality teaching material to create questions from</li></ul>		
OPEN ISSUES:			

USE CASE NAME:	Generating Assignments for courses from teaching materials	USE CASE TYPE  Business Requirements: <input type="checkbox"/> System Analysis: <input type="checkbox"/> System Design: <input checked="" type="checkbox"/>	
USE CASE ID:	3		
PRIORITY:	1		
SOURCE:	Teacher		
PRIMARY BUSINESS ACTOR	Teacher who wants to give students personalized assignment for particular topic		
PRIMARY SYSTEM ACTOR	CourseMind System (working on generating examples)		
OTHER PARTICIPATING ACTORS:	<ul style="list-style-type: none"><li>Generative AI Model (part of CourseMind)</li></ul>		
OTHER INTERESTED STAKEHOLDERS:	<ul style="list-style-type: none"><li></li></ul>		
DESCRIPTION:	This feature allows teachers to generate different assignments from a given teaching materials so that they can assess students’ performance in much detailed manner		
PRE-CONDITION:	Teacher should have some topic and content to create short personalized assignments for students		
TRIGGER:	Teacher wants to generate quiz questions to assess students		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: Teacher navigates to selected course	Step 4: Generates assignments based on information and sample given	
	Step 2: Selects uploaded material from which the teacher wants to create questions from.		
	Step 3: Gives the idea and sample of assignment and number of assignments required		
	Step 5: Selects and exports the JSON of created assignments		
ALTERNATE COURSES:	If Teacher doesn’t like the assignments generated by CourseMind, teacher can provide feedback to CourseMind which allows teacher to get different assignments based on their feedback		
	If user CourseMind doesn’t get the required number of assignments it will give however many it was able to generate and add an error saying that it was only able to generate some of it.		
CONCLUSION:	Teacher will get examples generated for particular topic ready to help students with studying using existing teaching materials		
POST-CONDITION:	Teachers will have questions generated and stored into a database as well as exported to Canvas for assessment.		
BUSINESS RULES	<ul style="list-style-type: none"><li>Teacher should be logged in</li><li>Should have uploaded some reference material</li><li>Should provide valid type of question</li></ul>		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"><li>Only authorized users can access</li></ul>		
ASSUMPTIONS:	<ul style="list-style-type: none"><li>Teachers will provide high quality teaching material to create assignments from</li></ul>		
OPEN ISSUES:			

USE CASE NAME:	Authentication	USE CASE TYPE Business Requirements: <input type="checkbox"/> System Analysis: <input type="checkbox"/> System Design: <input checked="" type="checkbox"/>	
USE CASE ID:	4		
PRIORITY:	1		
SOURCE:	Teacher		
PRIMARY BUSINESS ACTOR	The person uploading and managing course content.		
PRIMARY SYSTEM ACTOR	The system where the materials are stored and managed.		
OTHER PARTICIPATING ACTORS:	<ul style="list-style-type: none"><li>Teaching Assistants and Canvas</li></ul>		
OTHER INTERESTED STAKEHOLDERS:	<ul style="list-style-type: none"><li>None</li></ul>		
DESCRIPTION:	This feature allows teacher to login and keep their data separated from others so that it doesn't get mixed up		
PRE-CONDITION:	Teacher has internet access and a compatible device		
TRIGGER:	To create an account or log into existing one		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: Teacher navigates to login portal	Step 2: System validates the credentials	
	Types in his credentials in the system or register with all the details	Saves their account details and allows access to the system with just their own data	
	Teacher can logout of their account	Clears the cookies and authentication key from cache and navigate them back to login portal	
ALTERNATE COURSES:	If the Teacher email does not exist, it will show the SignUp page.		
	If CourseMind can not validate the credentials it will return an error saying Invalid Credentials.		
CONCLUSION:	Atlast, the teacher will be able to perform CRUD requests for their courses		
POST-CONDITION:	If the credentials were correct then the teacher would be redirected to the dashboard with their data.		
BUSINESS RULES	<ul style="list-style-type: none"><li>Teacher must have an account or sign up with all the details</li></ul>		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"><li>All the details provided by the user, mainly emailId, must be correct.</li></ul>		
ASSUMPTIONS:	<ul style="list-style-type: none"><li>Only users logged in/ registered can be authenticated</li></ul>		
OPEN ISSUES:	<ul style="list-style-type: none"><li>None</li></ul>		

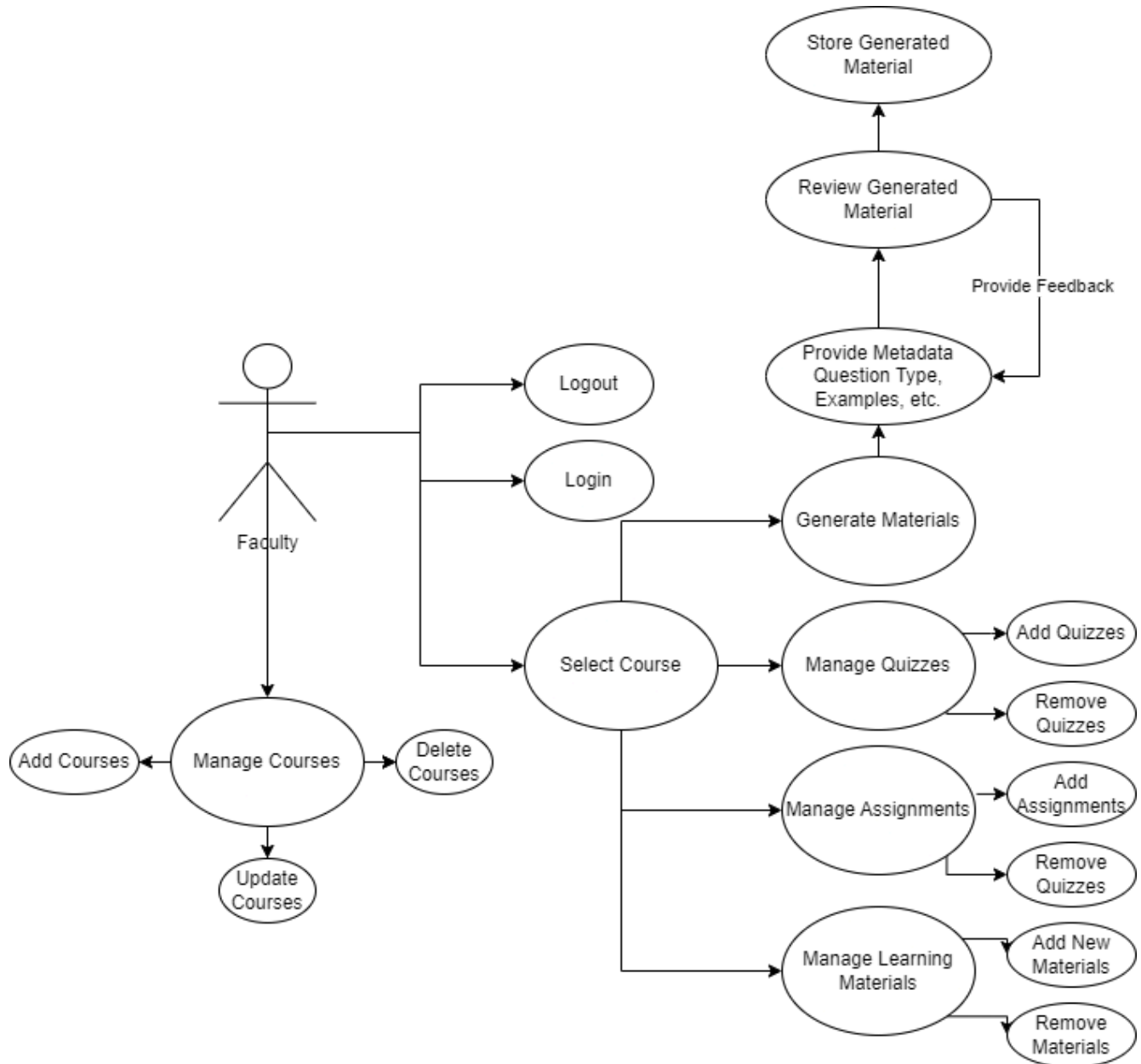
USE CASE NAME:	Manage all courses	
USE CASE ID:	5	
PRIORITY:	1	
SOURCE:	Teacher	
PRIMARY BUSINESS ACTOR	Teacher who wants to manage all his courses	
PRIMARY SYSTEM ACTOR	CourseMind (The backend system for managing all the user courses and CRUD requests)	
OTHER PARTICIPATING ACTORS:	<ul style="list-style-type: none"><li>Teaching Assistants and Canvas</li></ul>	
OTHER INTERESTED STAKEHOLDERS:	<ul style="list-style-type: none"><li>None</li></ul>	
DESCRIPTION:	This feature allows teachers to create, update, read and delete all their courses.	
PRE-CONDITION:	Teacher has internet access and a compatible device and is authenticated.	
TRIGGER:	To manage their courses in one page	
TYPICAL COURSE OF EVENTS:	Actor Action	
	Step 1: Teacher sees the dashboard and their added courses(if any)	
	Adds new course name, code, description and start and end date and save	
	Teachers can edit any listed course, make changes and save.	
	Teacher can delete any course and all his materials. A confirmation dropbox will popup for delete.	
ALTERNATE COURSES:	If Teacher is not authenticated, it will show the LogIn/SignUp page.	
	If the added course details are of incorrect type or missing, it will return an error message and won't add it to the dataset.	
	If CourseMind faces any error in deleting the course or course material from the database, it will return with error message	
CONCLUSION:	Atlast, the teacher will be able to perform CRUD requests for their courses	
POST-CONDITION:	Teacher will have all the courses added, updated and listed on the dashboard.	
BUSINESS RULES	<ul style="list-style-type: none"><li>User/Teacher/Professor must be logged in</li><li>Course that is being added should have all required information</li></ul>	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"><li>Only authorized users are granted read permission</li></ul>	
ASSUMPTIONS:	<ul style="list-style-type: none"><li>Only users seeing this page must be authorized</li></ul>	
OPEN ISSUES:	<ul style="list-style-type: none"><li>None</li></ul>	

<b>USE CASE NAME:</b>	Selecting course from a list of courses I am teaching and taught in the past		<b>USE CASE TYPE</b>	
<b>USE CASE ID:</b>	6		Business Requirements:	<input type="checkbox"/>
<b>PRIORITY:</b>	1		System Analysis:	<input type="checkbox"/>
<b>SOURCE:</b>	Teacher/Professor		System Design:	<input checked="" type="checkbox"/>
<b>PRIMARY BUSINESS ACTOR</b>	Teacher/Professor (who wants to enter into an individual course from portal)			
<b>PRIMARY SYSTEM ACTOR</b>	CourseMind (The backend system)			
<b>OTHER PARTICIPATING ACTORS:</b>	•			
<b>OTHER INTERESTED STAKEHOLDERS:</b>	•			
<b>DESCRIPTION:</b>	Focuses on selecting a course from existing courses list for opening course material			
<b>PRE-CONDITION:</b>	User has internet access and a compatible device and has at least one course added in the database			
<b>TRIGGER:</b>	User wants to enter into an individual course webpage for further work			
<b>TYPICAL COURSE OF EVENTS:</b>	<b>Actor Action</b>	<b>System Response</b>		
	<b>Step 1:</b> User navigates to the Courses page	<b>Step 3:</b> Upon successful completion of the process, the system will display– i. User enters inside a course page where assignments and quizzes are displayed.		
	<b>Step 2:</b> User clicks on “Open” button and enters into a course by filling			
<b>ALTERNATE COURSES:</b>	If error occurs during the course select process, the error message will be displayed.			
<b>CONCLUSION:</b>				
<b>POST-CONDITION:</b>	User will view individual course webpage, displaying course notes, assignments and quizzes			
<b>BUSINESS RULES</b>	<ul style="list-style-type: none"> <li>• User/Teacher/Professor must be logged in</li> <li>• There should be at least one course added in the system</li> </ul>			
<b>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</b>	<ul style="list-style-type: none"> <li>• Only authorized users are granted read permission</li> </ul>			
<b>ASSUMPTIONS:</b>	<ul style="list-style-type: none"> <li>• Only authorized users are granted read permission</li> </ul>			
<b>OPEN ISSUES:</b>	<ul style="list-style-type: none"> <li>• Courses are already present in the database to begin with</li> </ul>			

USE CASE NAME:	Managing Assignments for courses from teaching materials	USE CASE TYPE	
USE CASE ID:	7	Business Requirements:	<input type="checkbox"/>
PRIORITY:	1	System Analysis:	<input type="checkbox"/>
SOURCE:	Teacher	System Design:	<input checked="" type="checkbox"/>
PRIMARY BUSINESS ACTOR	Teacher who is responsible for managing course assignments		
PRIMARY SYSTEM ACTOR	CourseMind System (helps teachers to manage course assignments)		
OTHER PARTICIPATING ACTORS:	<ul style="list-style-type: none"><li>Generative AI Model (part of CourseMind)</li></ul>		
OTHER INTERESTED STAKEHOLDERS:	<ul style="list-style-type: none"><li></li></ul>		
DESCRIPTION:	This feature focuses on how teachers manage the course assignments for a particular course including the ability to add, remove and review assignments.		
PRE-CONDITION:	Teacher should have successfully logged into the system and has selected a course for which they want to manage assignments.		
TRIGGER:	Teacher wants to manage assignments for students		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: Teacher navigates to selected course’s dashboard	ii) Remove Assignment: Teacher selects an existing assignment to remove. The system asks for confirmation before removing the assignment. Teacher confirms, and the assignment is removed.	
	Step 2: Selects “Manage Assignments” option and the system displays all the existing assignments for selected course	iii) Review Assignment: Teacher selects an existing assignment to review. The system displays the details of the selected assignment. Teachers can update the details or leave them unchanged.	
	Step 3: Teacher can choose to do		
	i) Add Assignments: Teacher selects Add Assignment The system prompts for assignment details (title, due date, description, etc)		
ALTERNATE FLOW:	If Teacher attempts to add an assignment without providing all required information (e.g., missing title or due date), the system will prompt them to complete the missing fields before submission.		
	If Teacher tries to remove an assignment that has active student submissions, the CourseMind system will prompt them with a warning that removal will also delete the submissions, requiring confirmation.		
CONCLUSION:	Teacher will be able to add, delete and update assignments to keep the course materials up to date and relevant		
POST-CONDITION:	If the assignment is added or removed successfully, the course assignments list is updated accordingly. If there is an issue with adding or removing assignments (e.g., network error), an error message is displayed to the faculty member.		
BUSINESS RULES	<ul style="list-style-type: none"><li>Teacher should be logged in</li><li>Should have assignments to be managed</li></ul>		



IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	<ul style="list-style-type: none"> <li>Only authorized users can access</li> </ul>
ASSUMPTIONS:	<ul style="list-style-type: none"> <li>Teachers will perform high quality management to manage assignments</li> </ul>
OPEN ISSUES:	



github repo link: <https://github.com/ParshwaS/CourseMind>