

《问道：基于课程教学数据的内容推荐和个性化智能
答疑系统》

软件设计规格说明书

版本	章节	负责人		修订时间
V1.0	引言、体系结构模式	何思成 (1 3,1 2)	龚政源 (3)	2025 4 22
	系统界面设计	陶恒轩 (1 2)	侯宜辰 (1)	
	系统详细设计	侯宜辰 (3)	何思成 (1 2)	

一、引言

1. 编写目的

本软件设计规格说明书旨在为“视频答疑系统”的开发提供详细的设计指导依据，确保系统的实现能够满足预先定义的功能性与非功能性需求。该文档的撰写旨在将前期需求分析阶段形成的系统需求转化为可操作的设计规范，以指导后续的系统架构设计、模块开发、接口定义、数据库设计与系统部署等工作环节。

本说明书面向项目的所有相关参与者，包括开发团队成员、指导教师、测试人员以及后续可能的运维人员。通过提供结构化、清晰、可验证的设计内容，本说明书将为项目的高效开发、测试与后期维护奠定基础。

该文档还将作为评审系统设计可行性与合理性的依据，确保系统架构在可扩展性、可维护性、用户体验与技术实现方面具备良好的综合表现。同时，该文档也将成为项目归档与项目交付的重要组成部分，为后续的系统升级与迁移提供技术参考。

2. 读者对象

指导教师与评审人员：理解系统的整体设计与技术实现方案，评估其合理性与可行性。

开发人员：依据文档明确的模块职责与接口协议，实施代码开发与集成。

测试人员：依据设计结构与流程制定测试用例，开展系统功能与性能的验证工作。

后续运维人员：在系统部署后进行版本维护、故障修复与升级支持。

3. 软件项目概述

项目名称：视频答疑系统

用户单位：高校教师、学生、教学辅助平台

当前高校教学场景中，学生通过回看教学视频进行复习的需求日益增长。然而，传统的教学视频内容未经过结构化处理，导致学生在回看过程中难以快速定位重点内容，且无法根据自身疑问获得个性化解答，影响学习效率和体验。与此同时，教师也缺乏手段了解学生在学习过程中的知识点掌握情况和疑难点分布。

在此背景下，依托自然语言处理、大语言模型（如 GPT）、OCR/ASR 技术与知识图谱构建能力，“视频答疑系统”应运而生。该系统旨在通过自动结构化教学资源、支持智能问答与内容总结，构建一个集视频搜索、个性化问答、教学总结、关键帧定位于一体的教学辅助平台。

本软件实现了以下功能：教学资源结构化与文本提取处理：支持视频、课件、作业等教学资源的上传与自动解析，结合 OCR/ASR 技术提取文字信息，形成可用于后续问答与总结的结构化文本数据；课程智能问答服务：结合 RAG (Retrieval Augmented Generation) 架构与权限校验机制，为拥有访问权限的学生提供基于教学内容的自然语言答疑功能；课堂内容总结生成：通过定时任务分析教学资源内容，提取教学主旨、关键词与时间戳，生成课堂摘要信息并与视频关键帧绑定；总结查看与智能跳转问答：学生可在视频页面查看教学总结，点击跳转对应片段，提出问题并得到个性化 AI 答复；个性化推荐。

本软件旨在提升学生复习效率：通过智能摘要与语义问答帮助学生快速定位知识点与解答疑问；增强教师教学管理：帮助教师了解教学资源的核心内容分布与学生关注点；优化交互体验：通过结构化内容导航与问答交互界面提升系统可用性与满意度。

4. 文档概述

本软件设计规格说明书包括以下部分：

系统设计约束：包括设计目标、开发原则、平台限制等。

系统体系结构设计：明确系统的整体架构及模块划分。

详细用例设计：涵盖所有核心功能点的交互流程与参与者。

类设计：对于系统中各个功能和数据类进行阐述。

二、数据设计：描述系统中涉及的数据结构、数据库表结构等。

三、体系结构模式

1. 面向服务的体系结构 (SOA)

本部分旨在阐述系统的整体体系结构设计。基于前期需求分析阶段所明确的功能需求、非功能需求（性能、约束等）以及系统的核心目标，构建一个集成了智能问答、个性化推荐、动态知识图谱等多功能于一体的智慧教学辅助平台，我们对多种主流体系结构模式进行了评估。为了确保系统具备高内聚、低耦合、易于扩展和维护的特性，并能有效支撑复杂业务逻辑和多角色应用场景，我们最终选择了面向服务的体系结构 (Service Oriented Architecture SOA) 作为本次设计的核心指导模式。

经过对代理模式、MVC 模式、管道过滤器模式、客户端/服务器模式、点对点模式、面向服务的体系结构模式和发布/订阅模式的细致分析与比对，结合本系统的具体需求，我们确定采用 **面向服务的体系结构 (SOA)**。

虽然系统在宏观交互层面体现为典型的客户端/服务器模式 (特别是前后端分离的实现)，但 SOA 更能精确地描述系统后端内部的组织方式、模块划分和服务交互机制，是指导后端详细设计的更优选择。

2. 选择 SOA 的原因

我们选择面向服务的体系结构 (SOA) 主要基于以下几点考虑：

契合后端模块化与服务化设计：我们在需求分析中明确了系统后端包含数据处理、内容推荐、智能问答、知识图谱构建等多个相对独立的核心功能模块。SOA 的核心思想是将这些功能单元封装为独立的服务（如问答服务、推荐服务、数据处理服务等），每个服务承担明确职责。这种方式与我们分层、分模块的设计思路高度一致，有利于实现功能的高内聚和模块间的低耦合。

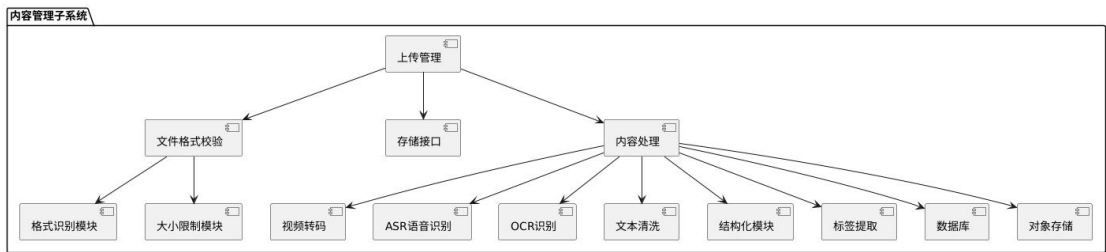
强调标准接口与互操作性： 系统要求通过标准化的 API (服务层) 向 Web 前端、移动端乃至其他第三方教育平台提供服务。SOA 正是通过定义清晰的服务接口和契约来实现不同服务消费者与提供者之间的互操作性。这不仅满足了当前多端接入的需求，也为未来系统集成和功能扩展奠定了良好基础。

提升系统的可扩展性与可维护性： 将系统拆分为一系列独立的服务，使得各服务可以独立开发、测试、部署、升级和扩展。例如，计算密集型的智能问答服务（依赖 LLM 和 RAG）可以根据实际负载独立进行资源扩充，或者通过调用第三方 API 实现，而不影响推荐服务或数据处理服务。这种灵活性显著提高了系统的整体可维护性，降低了未来变更的风险和成本，满足了“高扩展性”和“灵活性”的设计目标。

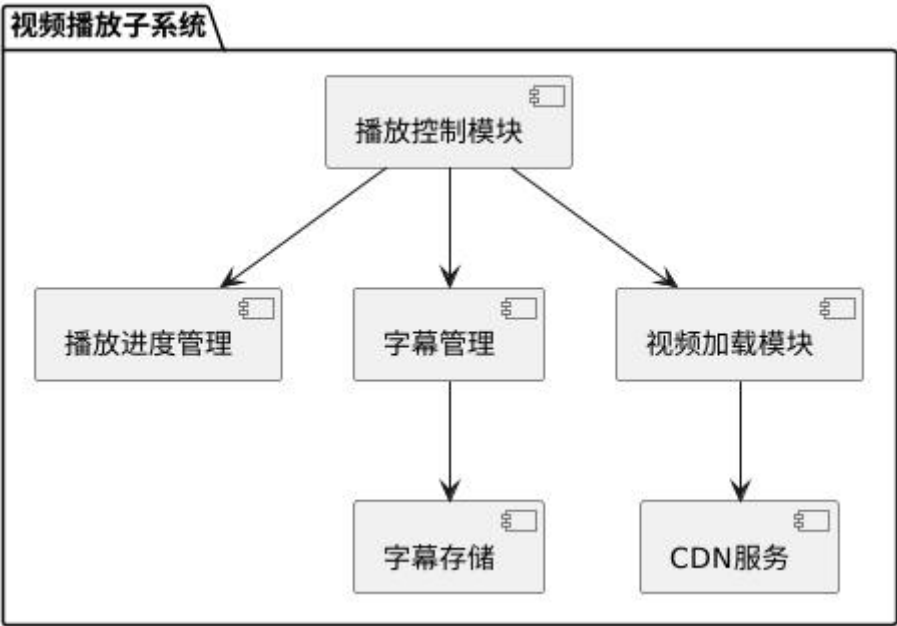
有效管理复杂性： 本系统涉及多种数据源处理、复杂的 AI 算法集成 (LLM, RAG, 推荐模型, 知识图谱) 以及多角色权限管理。SOA 通过将复杂问题分解为更小、更易于管理的服务单元，有助于降低整体系统的复杂度，使开发和管理更加高效有序。

3.软件体系结构

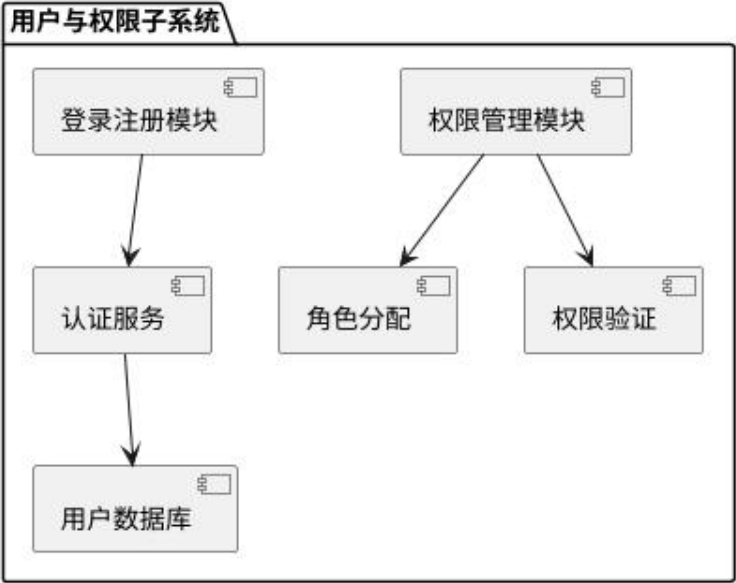
内容管理子系统:



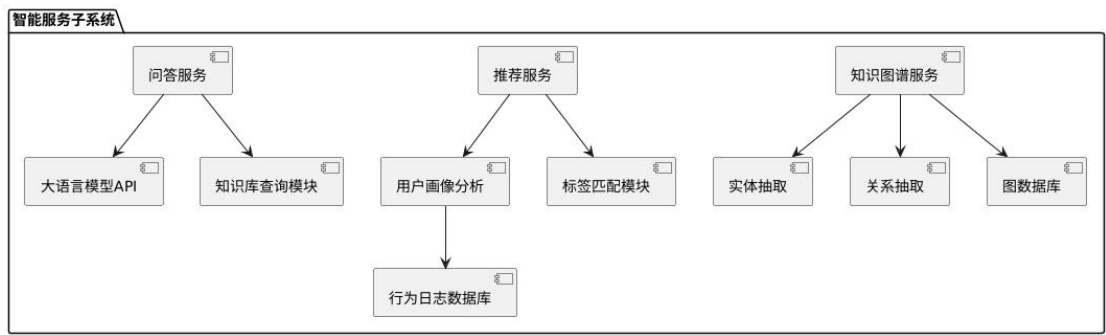
视频播放子系统:



用户与权限子系统:



智能服务子系统:

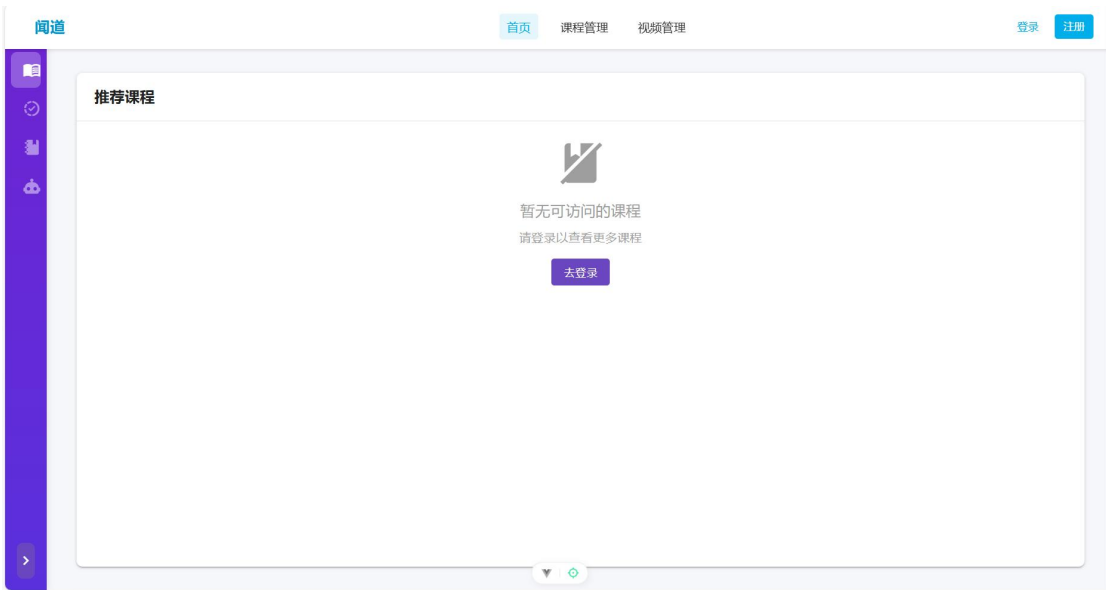


四、模块分解设计时遵循高内聚、低耦合的原则，每个子系统职责明确、功能聚焦，通过服务接口进行通信，避免模块之间直接调用造成的耦合问题。系统结构层次分明，分为表现层、应用层、服务层和数据层，便于后期扩展，如新增考试模块或替换推荐算法模块等。各功能服务采用解耦设计，智能问答、内容推荐、OCR/ASR 等功能均封装为独立服务，具备良好的复用性，可在其他教育平台中灵活集成。

五、系统界面设计

1. 原型设计

学生端首页：未登录时显示公开课程，登录后除公开课程外还有教师导入的课程



闻道

首页课程管理视频管理

登录注册

推荐课程

学习进度

笔记本

AI助手

推荐课程

暂无可访问的课程

请登录以查看更多课程

去登录

闻道

首页课程管理视频管理

登录注册

推荐课程

学习进度

笔记本

AI助手

12

在学课程

48

学习时长(小时)

5

已获证书

4.8

平均评分

当前学习进度

搜索课程

软件工程导论

已完成 8/12 课时

66%

2小时前

剩余 15 天

继续学习

计算机网络

已完成 10/12 课时

83%

昨天

剩余 20 天

继续学习

数据结构与算法

已完成 6/15 课时

40%

3天前

剩余 30 天

继续学习

闻道

首页课程管理视频管理

登录注册

推荐课程

学习进度

笔记本

AI助手

我的笔记

+ 新建

搜索笔记

软件工程第一章笔记
软件工程导论 · 2小时前

计算机网络协议层次
计算机网络 · 8小时前

数据结构-树的遍历
数据结构与算法 · 3天前

算法设计与分析笔记

保存

分支限界法

1. 基本思想

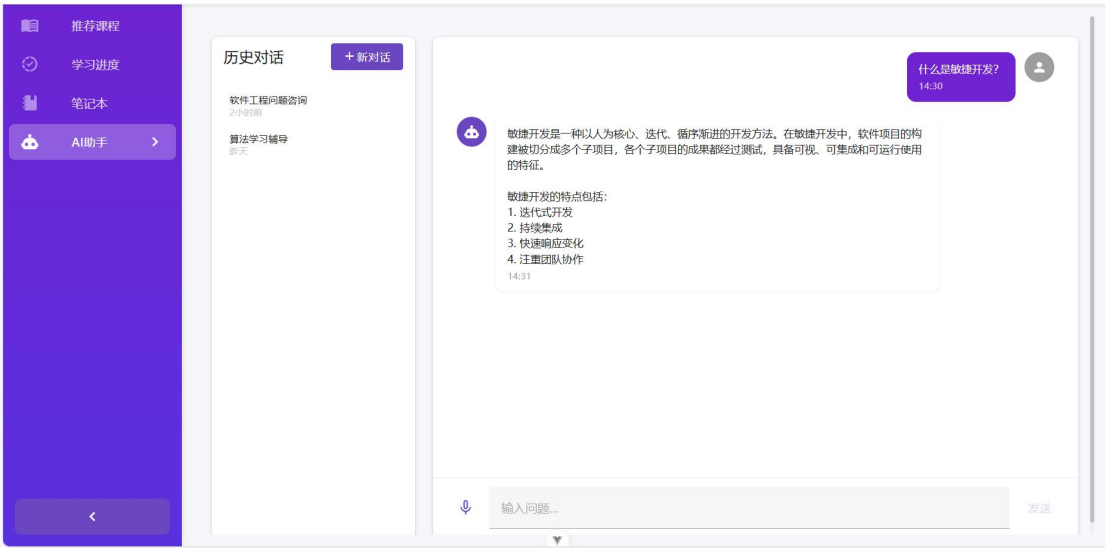
分支是使用广度优先策略，依次生成扩展结点的所有分支。

限界是在结点扩展过程中，计算结点的上界，搜索的同时剪掉某些分支。

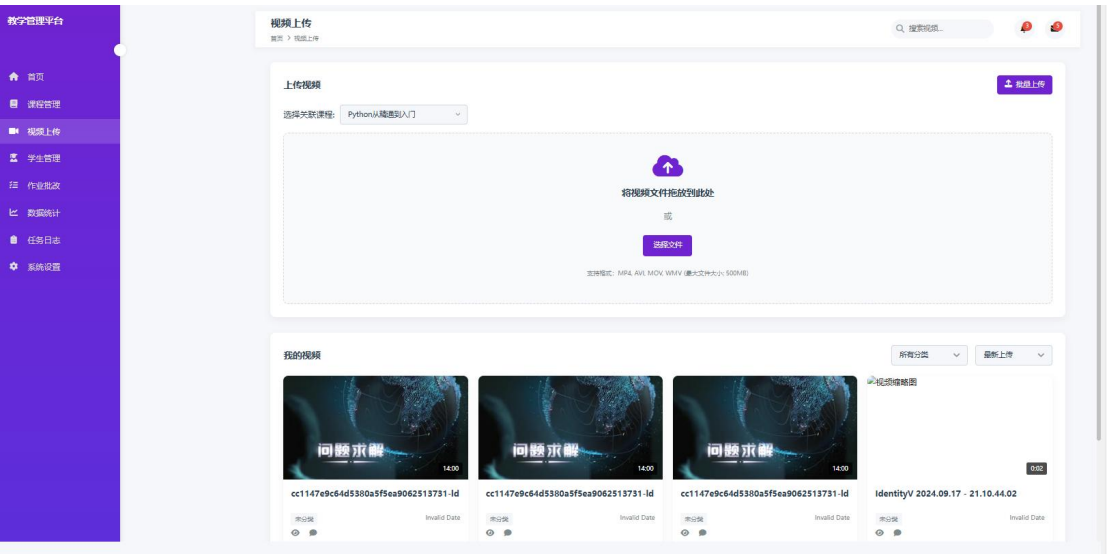
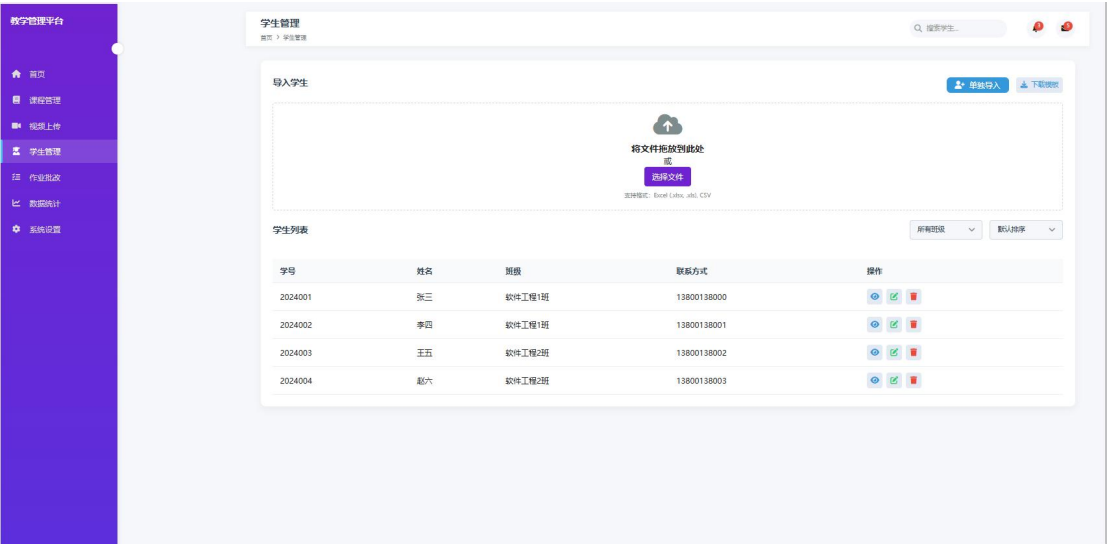
分支限界法就是把问题的可行解展开，再由各个分支寻找最佳解。

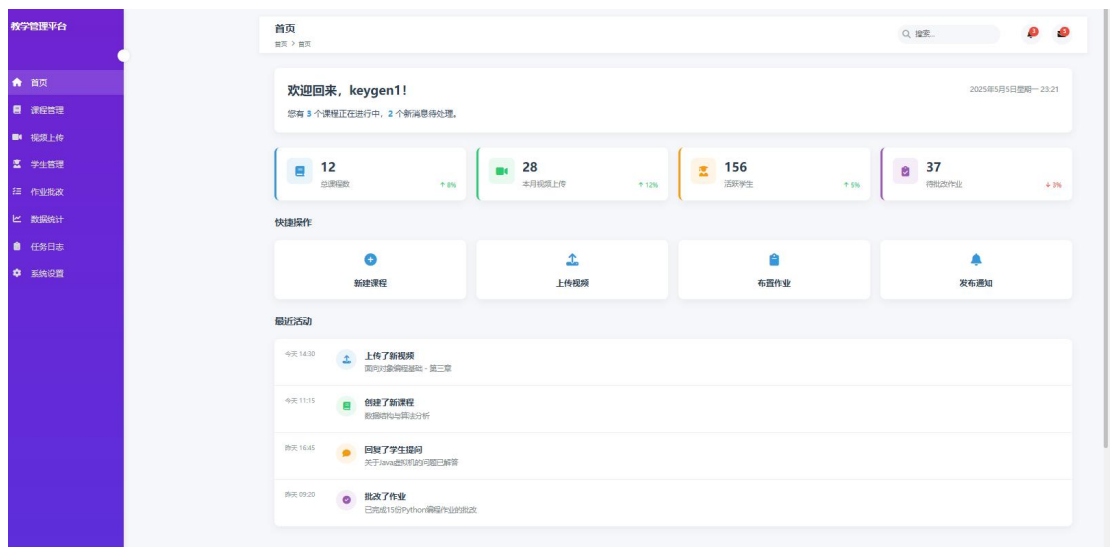
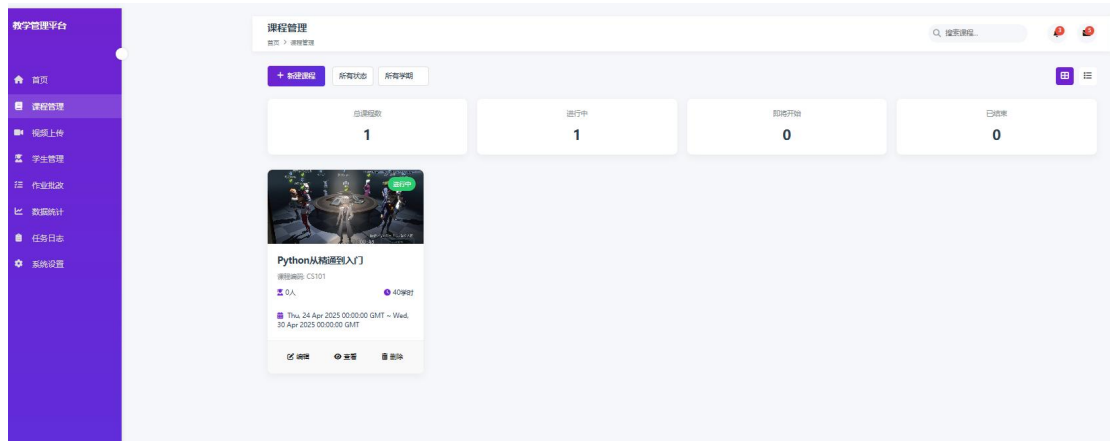
与回溯法类似，分支限界法也是在解空间中搜索得到解；

不同的是，分支限界法会生成所有扩展结点，并舍弃不可能通向最优解的结点，然后根据广度优先/最小耗费优先，从活结点中选择一个作为扩展结点，使搜索向解空间上有最优解的分支推进。



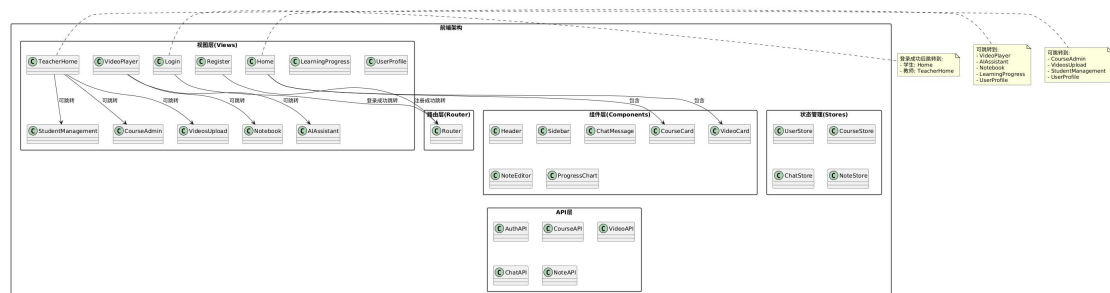
教师端各项管理功能：



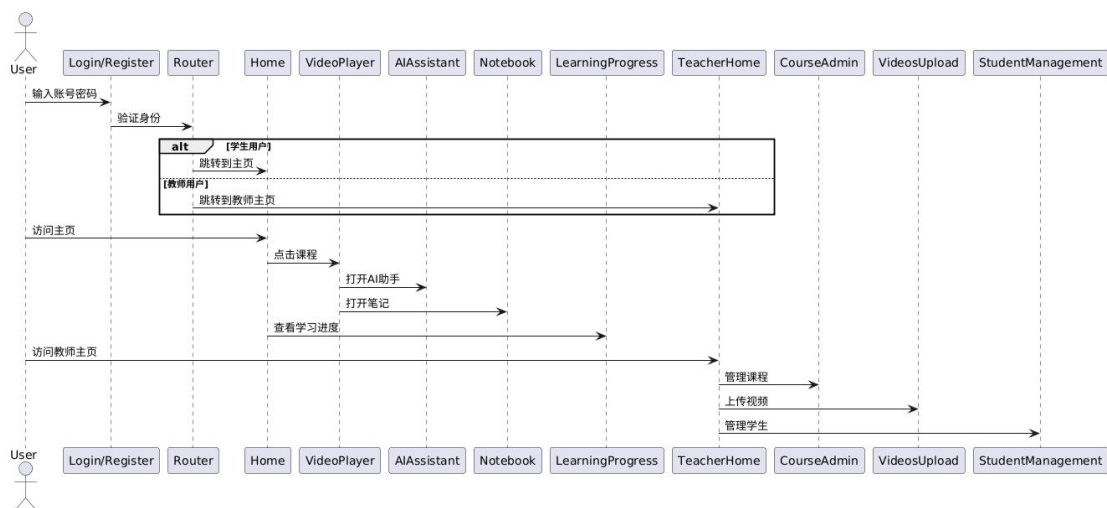


2. 类图和顺序图

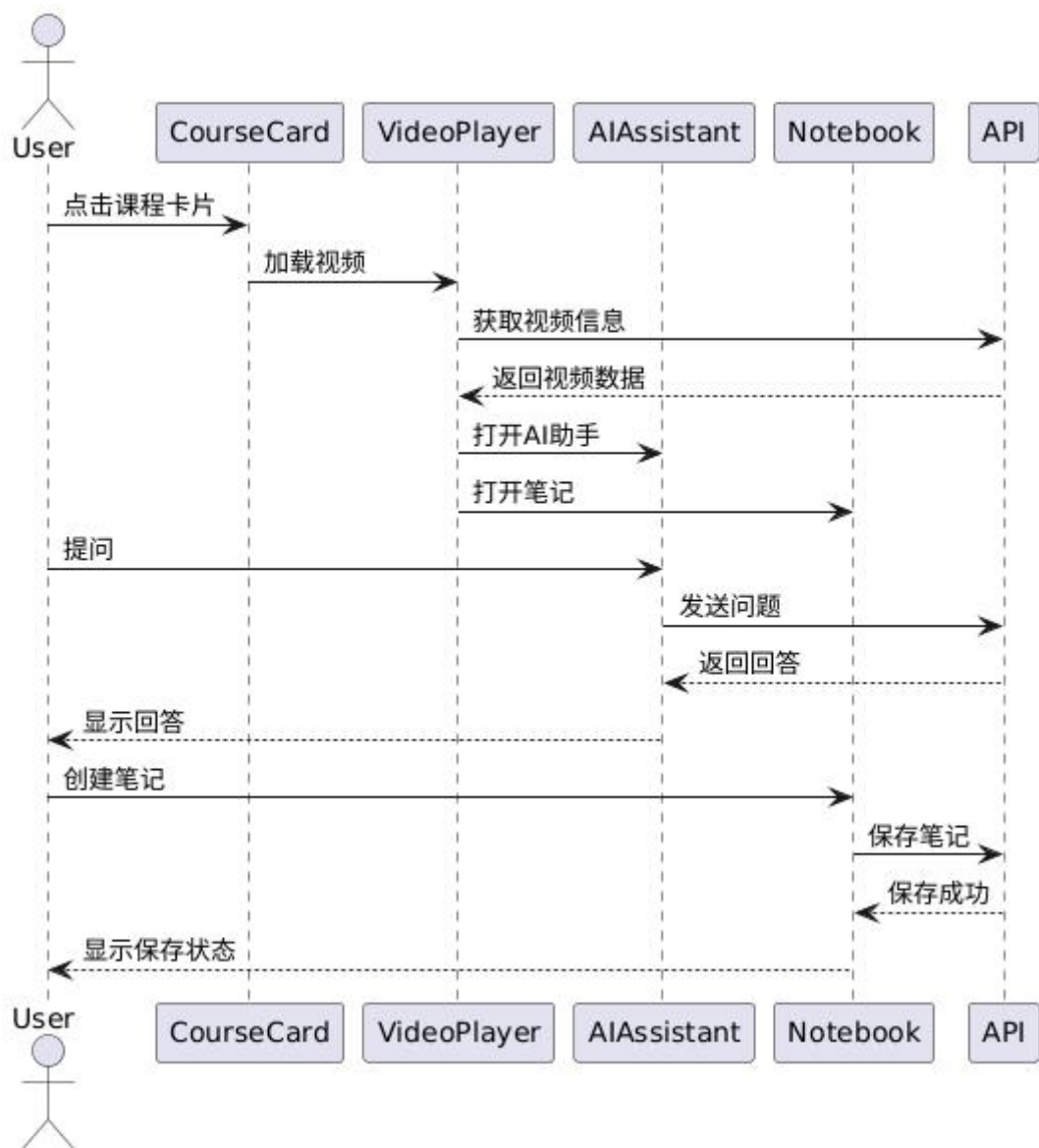
前端架构:



页面跳转顺序图:



组件交互顺序图：

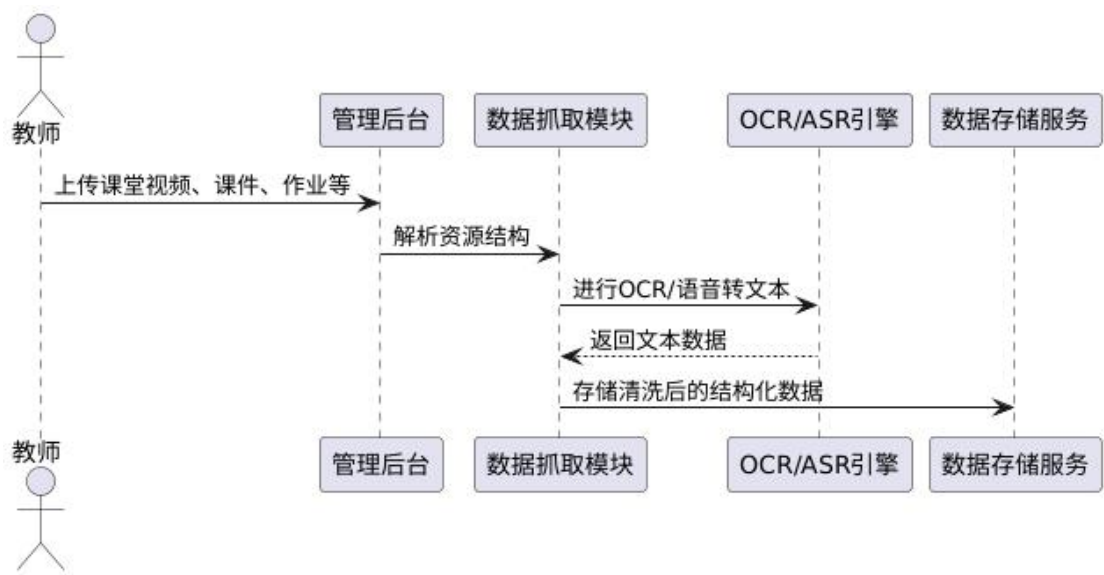


六、系统详细设计

1. 用例设计

1.1 视频答疑功能需求

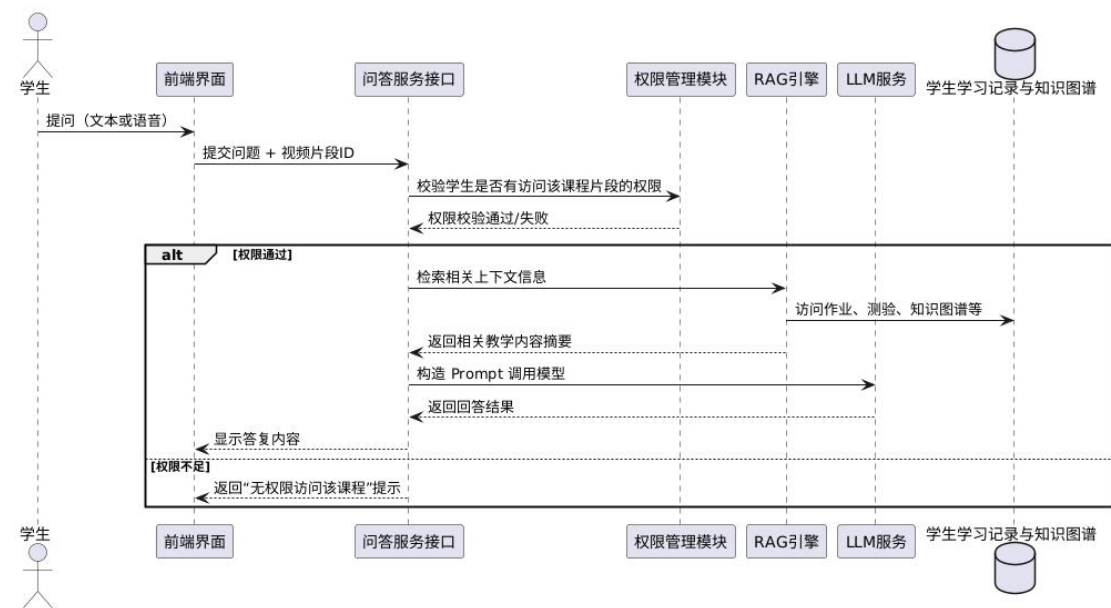
1.1.1 课堂数据抓取和清洗



用例名	教学资源结构化与文本提取处理
用例描述	将教师上传的课堂视频、课件、作业等教学资源，自动解析为结构化数据，并通过OCR/ASR 技术提取文本内容，供后续存储与使用
参与者	教师、系统模块（管理后台、数据抓取模块、OCR/ASR 引擎、数据存储服务）

过程	<ol style="list-style-type: none">1. 教师通过管理后台上传课堂视频、课件、作业等教学资源。2. 管理后台解析上传的资源结构，将资源交由数据抓取模块处理。3. 数据抓取模块将资源发送至 OCR/ASR 引擎进行文本提取（包括图像文字识别与语音转文字）。4. OCR/ASR 引擎返回识别后的文本数据至数据抓取模块。5. 数据抓取模块清洗并结构化文本数据。6. 清洗后的结构化数据被存储至数据存储服务中。7. 该流程可重复执行，支持多次教学资源的上传与处理。
----	---

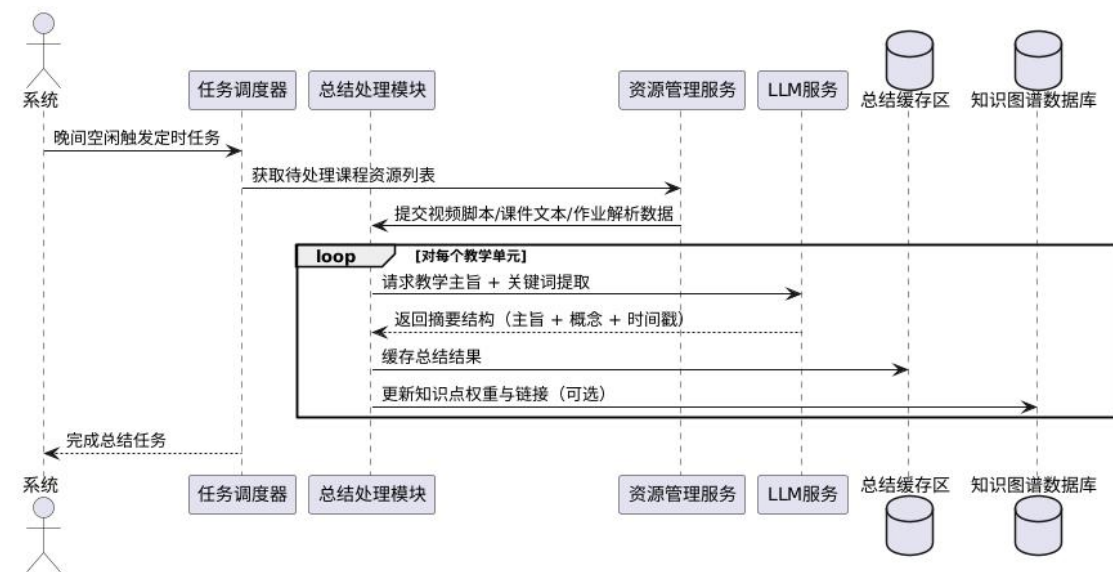
1.1.2 课程智能问答



用例名	基于权限的视频问答服务
用例描述	学生通过提问（文本或语音）获取视频相关课程内容的答复，系统在权限校验通过后调用大模型生成回答。
参与者	学生、问答服务接口、权限管理模块、RAG引擎、LLM服务
过程	1. 学生通过前端界面提出问题(文本或语音)

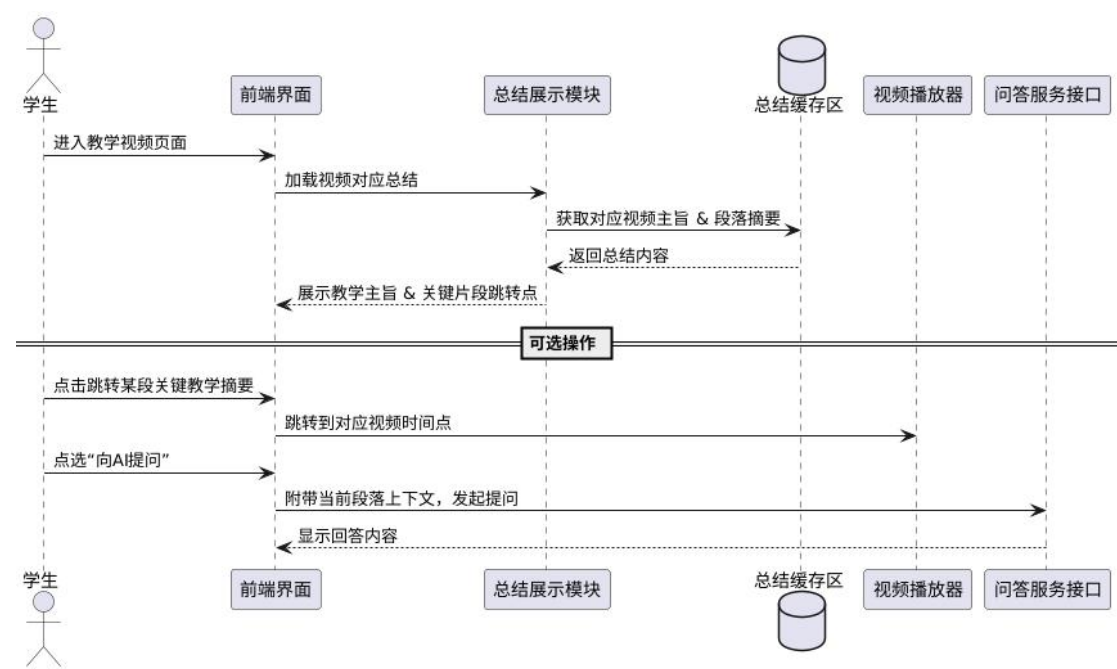
	<ol style="list-style-type: none">2. 问答服务接口接收到问题和视频片段 ID3. 权限管理模块验证学生是否拥有该课程片段的访问权限4. 若权限校验失败，返回“无权限访问该课程”提示，结束流程5. 若权限校验通过，RAG 引擎访问学生学习记录、作业、测验、知识图谱等，检索相关上下文信息6. 返回相关学内容摘要7. 构造 Prompt 并调用 LLM 服务8. 返回回答结果9. 前端界面显示答复内容
重复性	学生可多次发起问答请求，系统重复上述流程处理

1.1.3 课堂智能总结



用例名	课堂内容总结生成
用例描述	系统在预设时间触发任务，对教学视频资源进行分析，提取教学要点并生成总结内容。
参与者	系统、任务调度器、总结处理模块、资源管理服务、LLM 服务、知识图谱数据库、总结缓存区
过程	1. 系统在时间点到达时触发处理任务

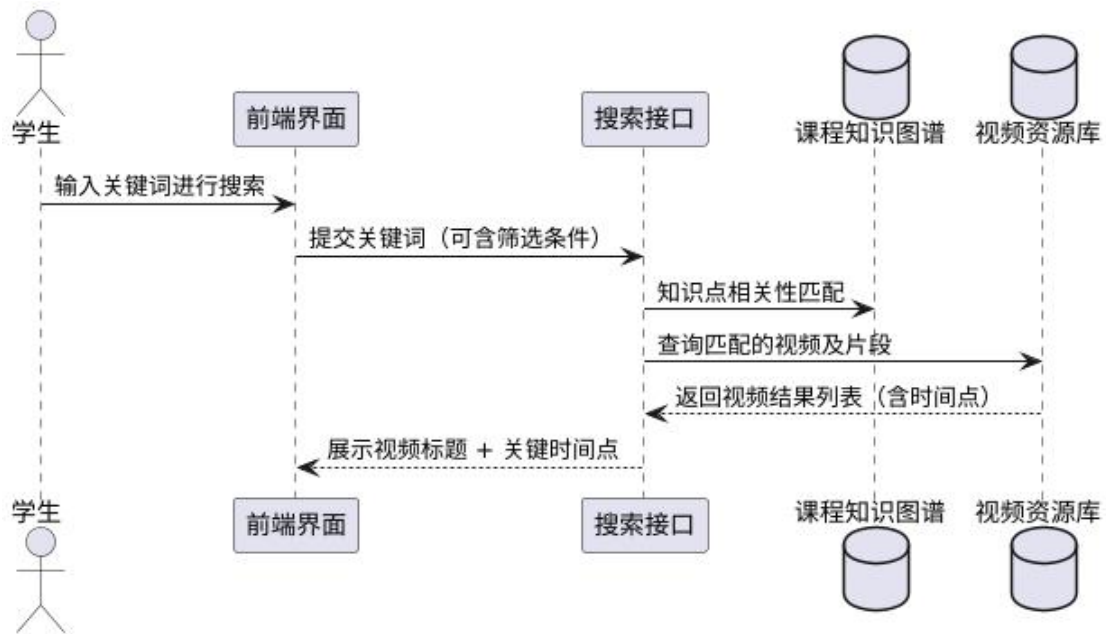
	<div>2. 任务调度器通知总结处理模块</div> <div>3. 总结处理模块向资源管理服务请求待处理资源列表</div> <div>4. 提取视频脚本/课件文本/作业解析数据</div> <div>5. 对每个教学单元循环执行以下操作:<div><div>a. 请求教学主旨 + 关键词提取</div><div>b. LLM 返回教学结构 (主旨 + 概念 + 时间戳)</div><div>c. 缓存总结结果</div><div>d. (可选) 更新知识图谱重构与链接</div></div></div> <div>6. 完成总结任务后通知系统</div>
重复性	可根据课程更新周期定期自动执行



用例名	学生查看教学视频总结并提问
用例描述	学生在教学视频界面查看系统生成的教学总结，支持点击跳转关键片段和向 AI 提问获取答案。
参与者	学生、前端界面、总结展示模块、总结缓存区、视频播放器、问答服务接口
过程	<div>1. 学生进入教学视频页面</div> <div>2. 前端界面请求加载该视频的总结内容</div> <div>3. 总结展示模块从缓存中获取教学主旨 &</div>

	<p>关键摘要</p> <p>4. 前端展示总结 + 跳转锚点</p> <p>5. 学生点击跳转按钮，播放器跳转到对应时间点</p> <p>6. 学生选择“向 AI 提问”，系统附带段落上下文提交问题</p> <p>7. 问答服务接口处理并返回答复</p> <p>8. 前端展示回答内容</p>
重复性	每个视频段落可重复查看总结、重复提问

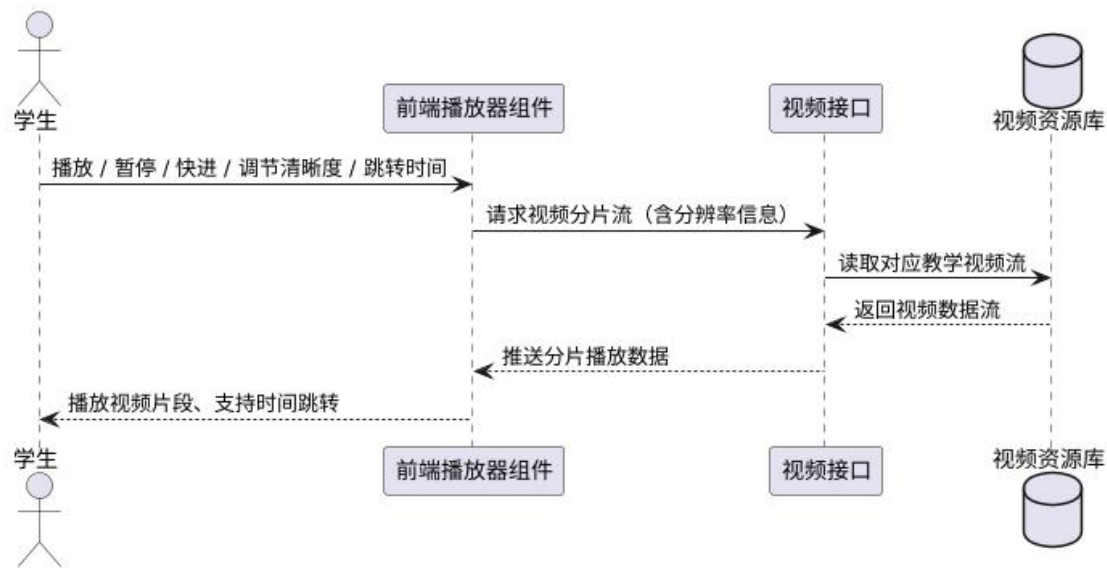
1.1.4 视频列表及搜索功能



用例名	学生搜索并查看教学视频内容
用例描述	学生通过输入关键词进行搜索，系统匹配课程知识点并返回相关视频及片段时间点供查看。
参与者	学生、前端界面、搜索接口、课程知识图谱、视频资源库
过程	<ol style="list-style-type: none"> 1. 学生在前端界面输入关键词发起搜索请求 2. 前端界面将关键词提交给搜索接口（可包含筛选条件） 3. 搜索接口基于关键词与课程知识点进行相关性匹配 4. 检索匹配的视频及片段信息

	5. 返回包含视频标题与关键时间点的视频结果列表 6. 前端界面展示查询结果，学生可点击查看
重复性	学生可根据不同关键词多次搜索，系统持续响应返回匹配视频信息

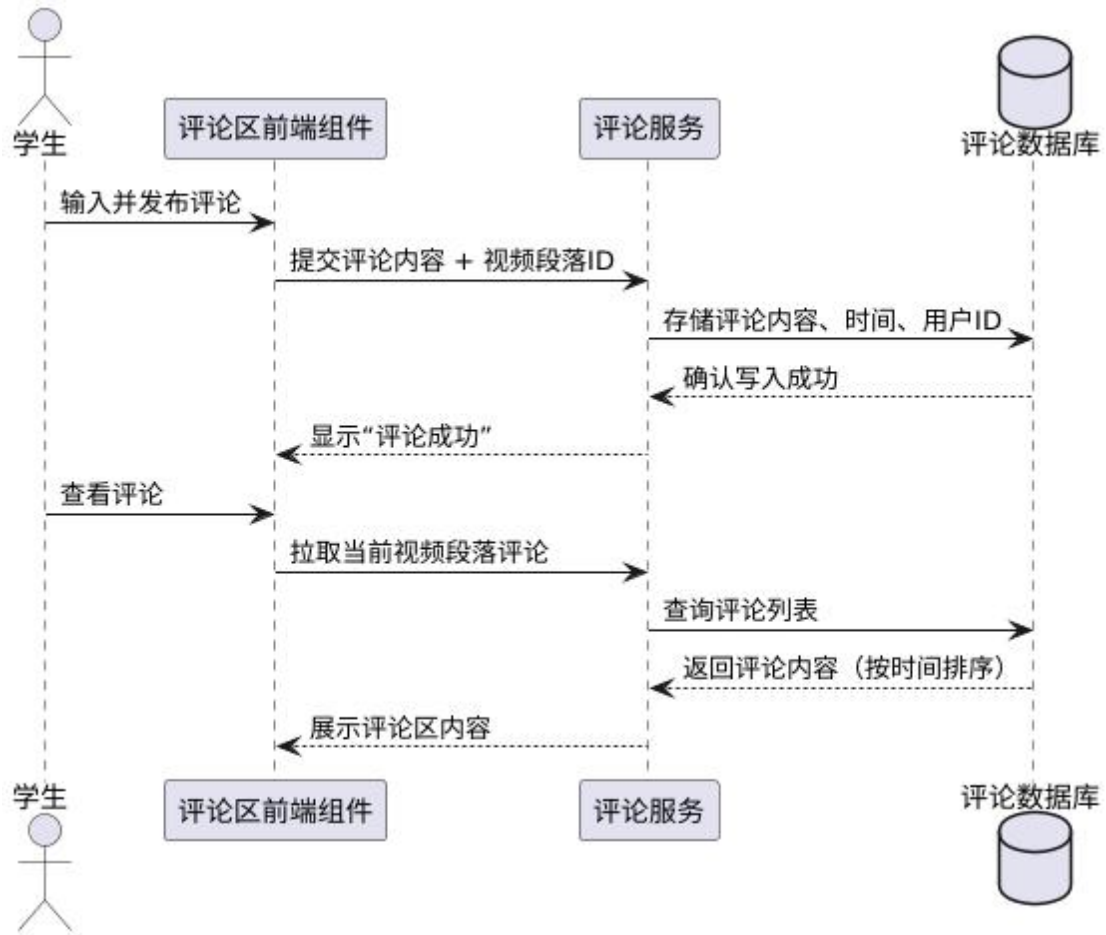
1.1.5 视频播放功能



用例名	学生观看教学视频
用例描述	学生通过前端播放器观看教学视频，支持清晰度选择和播放控制
参与者	学生
过程	<ol style="list-style-type: none"> 1. 学生通过前端播放器组件发起播放请求（包括播放、暂停、快进、跳转时间、选择清晰度等操作） 2. 播放器组件向视频接口请求分片视频流（含分辨率信息） 3. 视频接口从视频资源库读取对应教学视频流 4. 视频接口将分片视频数据返回给播放器组件 5. 播放器组件推送视频数据至学生端进行播放，支持时间跳转等操作 6. 学生继续播放视频，或进行其他播放控制

	操作
--	----

1.1.6 留言板

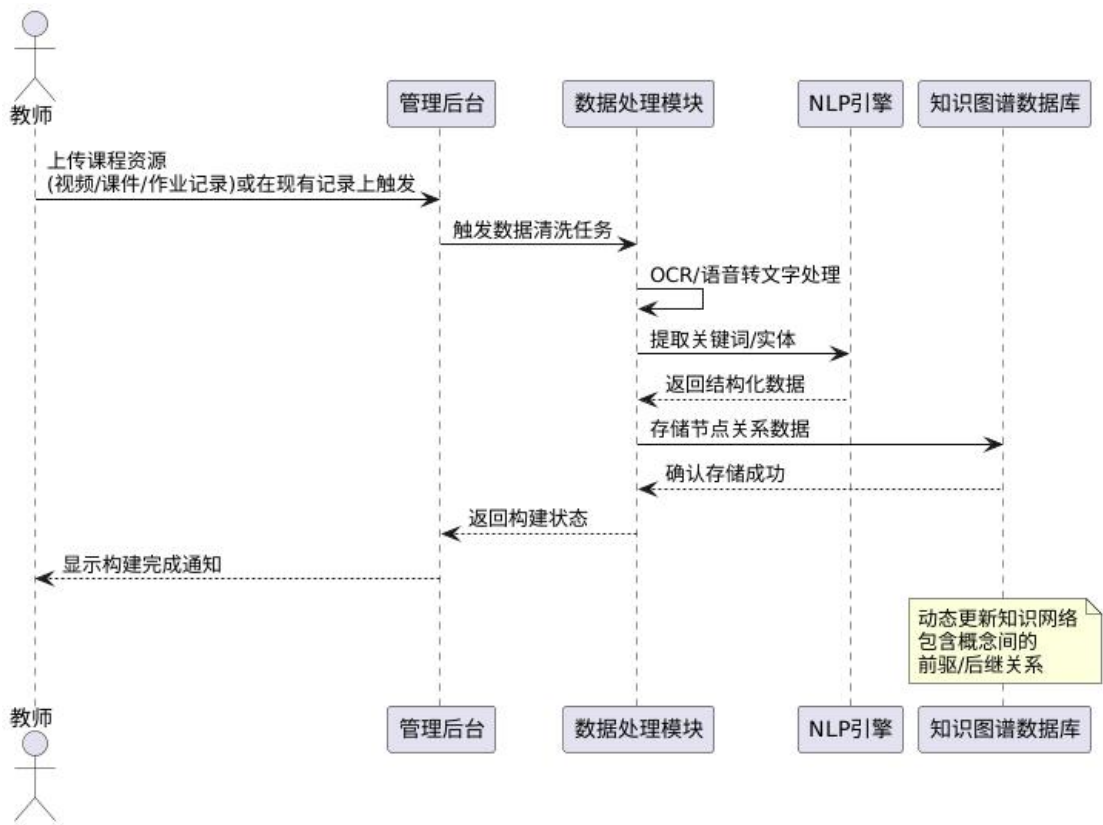


用例名	学生评论视频片段
用例描述	学生在观看教学视频过程中，可以对视频片段发表评论并查看他人评论
参与者	学生
过程	<ol style="list-style-type: none"> 1. 学生在评论区前端组件中输入并发布评论内容 2. 前端将评论内容与视频片段 ID 一并提交给评论服务 3. 评论服务将评论内容、时间、用户 ID 存入评论数据库 4. 数据库返回写入成功，评论服务反馈“评论成功”给前端 5. 学生查看评论时，前端拉取当前视频段落

	的评论 6. 评论服务从数据库查询相关评论列表，并按时间排序返回 7. 前端展示评论内容给学生
--	---

1.2 知识图谱和个性化学习需求

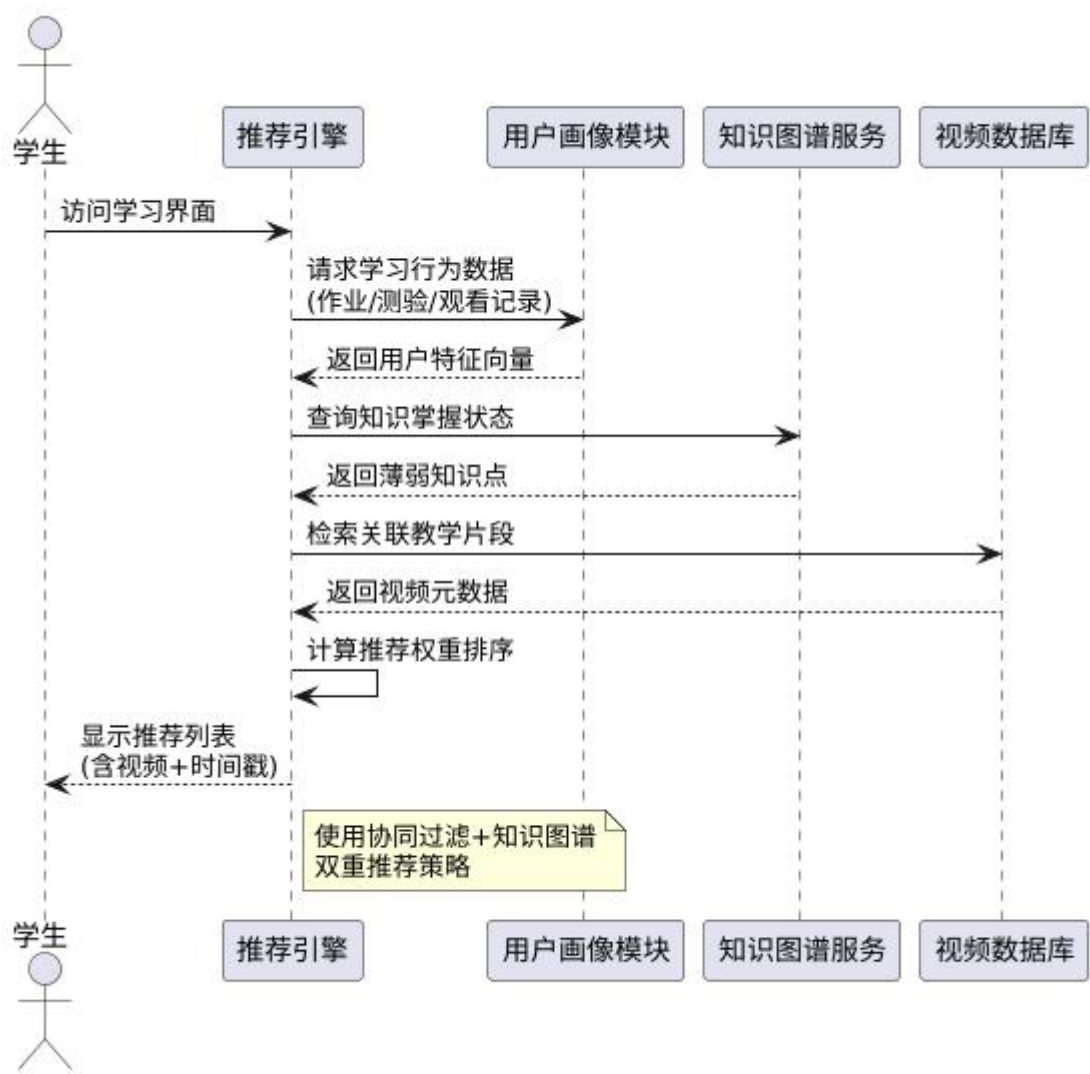
1.2.1 课程知识图谱构建



用例名	课程知识图谱构建
用例描述	教师上传课程资源后，系统自动清洗数据、提取关键词并构建知识图谱
参与者	教师
过程	1. 教师上传课程资源（视频/作业/作业记录）或选择已有记录上触发构建 2. 管理后台触发数据清洗任务 3. 数据处理模块进行 OCR/语音转文字处理 4. 提取关键词和实体信息 5. 返回结构化数据

	<div>6. 存储知识图谱节点及其关系到数据库中</div> <div>7.NLP 引擎确认存储成功并返回状态</div> <div>8. 管理后台将构建状态返回给教师</div> <div>9. 教师收到构建完成通知</div>
--	--

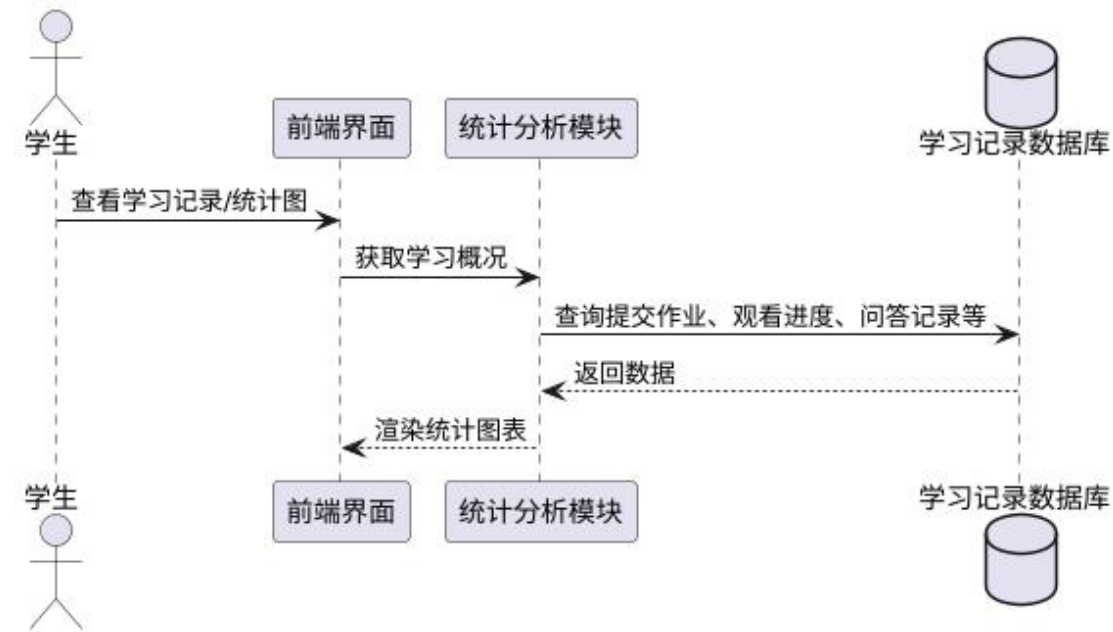
1.2.2 个性化学习推荐



用例名	个性化学习推荐（双重推荐策略）
用例描述	系统通过协同过滤与知识图谱双重策略，结合学生的行为与掌握状态，推荐适合的学习视频资源
参与者	学生

过程	<ol style="list-style-type: none">1. 学生访问学习界面，触发推荐流程2. 推荐引擎请求学生的学习行为数据（如作业、测试、观看记录）3. 用户画像模块返回用户特征向量4. 推荐引擎查询学生知识掌握状态5. 知识图谱服务返回学生薄弱知识点6. 推荐引擎检索相关教学片段7. 视频数据服务返回视频元数据信息8. 推荐引擎基于双重策略（协同过滤 + 知识图谱）计算权重排序9. 向前端展示推荐列表（含视频播放时长等信息）10. 学生点击播放继续学习
----	---

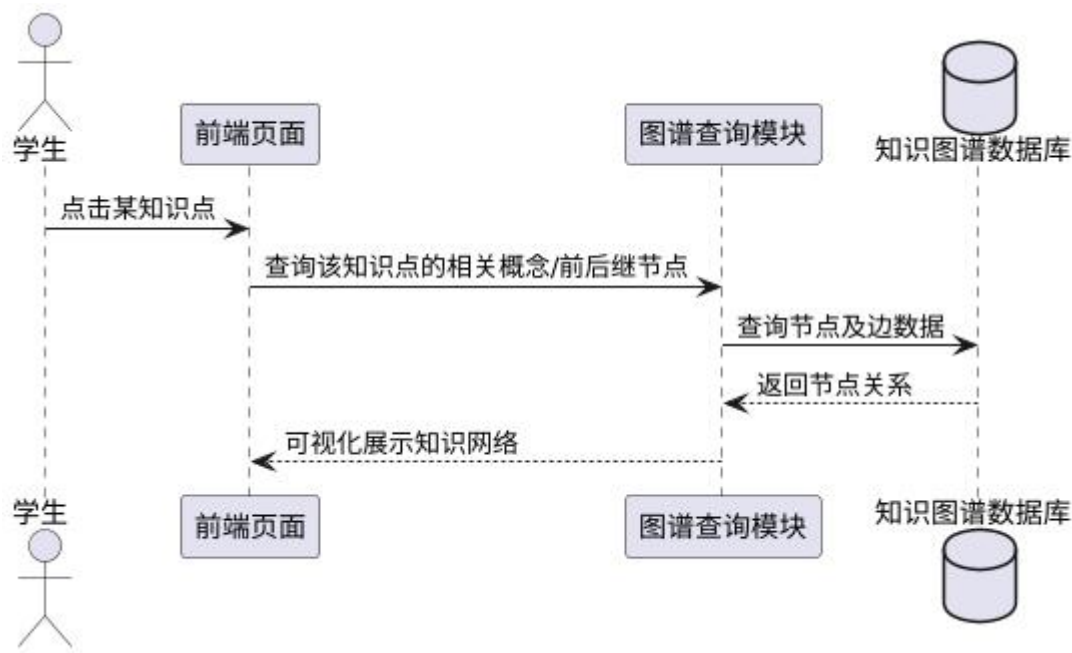
1.2.3 学习数据统计



用例名	学习数据统计
用例描述	系统统计学生的学习记录数据，并以图表方式展示学生学习情况，包括作业完成情况、观看进度、提问记录等
参与者	学生
过程	<ol style="list-style-type: none">1. 学生访问前端界面，点击查看学习记录/统计图2. 前端发送“获取学习概况”请求至统计分析模块3. 统计分析模块查询学习记录数据库中的数据（包括作业、观看、提问等）

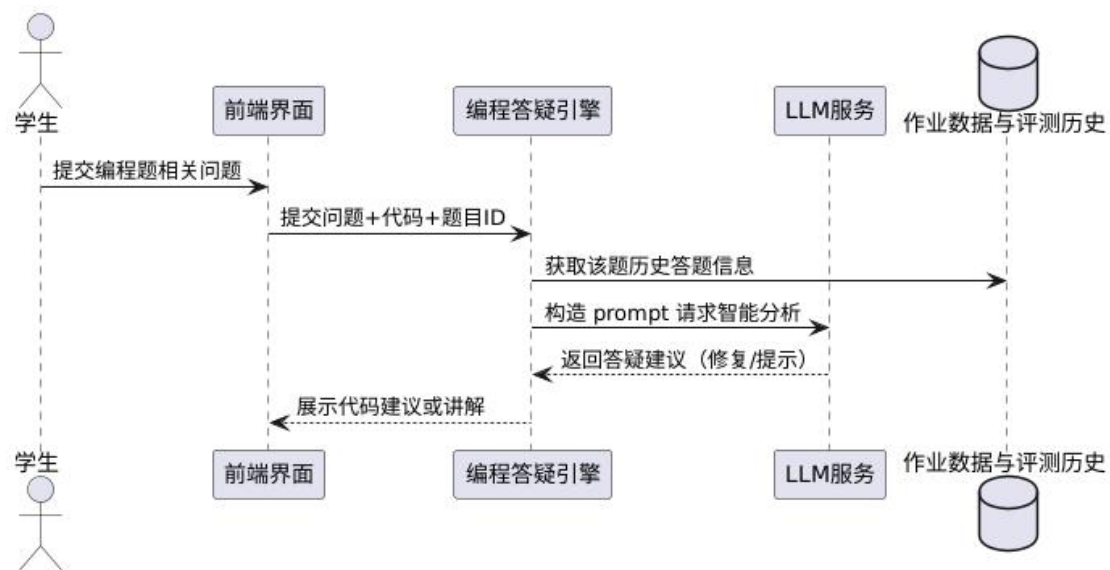
	4. 数据库返回学生相关学习记录数据 5. 统计分析模块对数据进行整理和分析 6. 返回图表所需统计数据给前端 7. 前端渲染并展示统计图表给学生查看
--	--

1.2.5 知识图谱展示



用例名	知识图谱展示
用例描述	学生点击某知识点后，系统查询该点相关概念与前后继节点，并可视化呈现知识图谱网络
参与者	学生
过程	1. 学生在前端页面点击某个知识点 2. 前端发送请求至图谱查询模块，查询该知识点的相关概念与上下游节点 3. 图谱查询模块向知识图谱数据库发起查询请求，获取相关节点及边数据 4. 数据库返回节点关系数据 5. 图谱查询模块将结构化数据返回给前端 6. 前端根据数据可视化呈现知识网络图

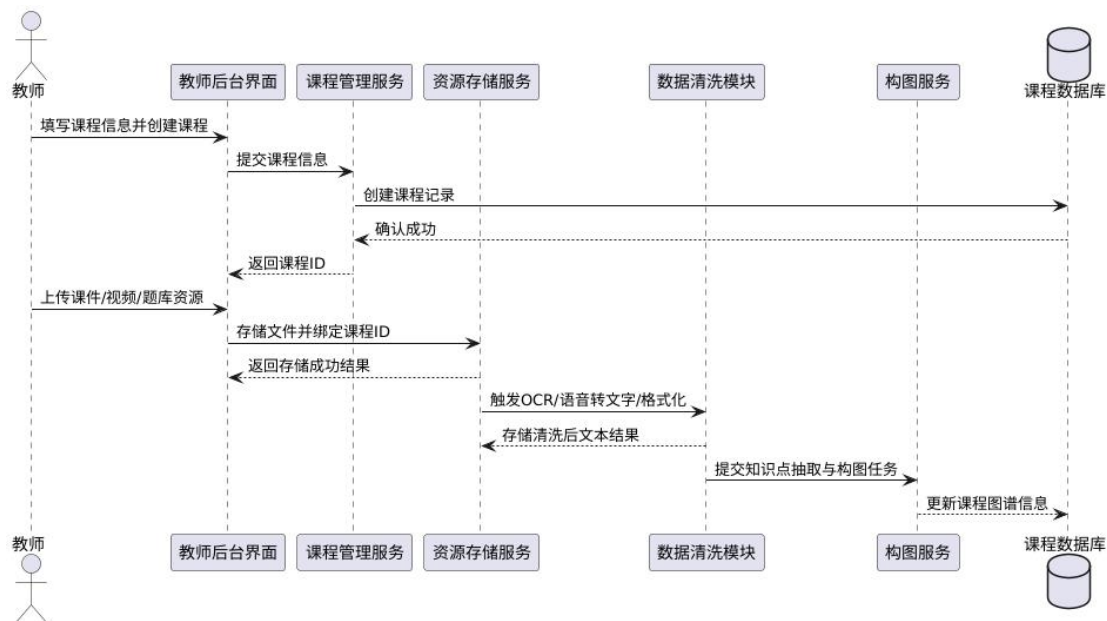
1.2.6 作业个性化答疑（编程题）



作业个性化答疑	用例名	作业个性化答疑
	用例描述	学生提交编程题相关问题后，系统结合历史作业记录与 LLM 服务，生成个性化答疑内容提供给学生
	参与者	学生
	过程	<ol style="list-style-type: none"> 1. 学生在前端界面提交编程题相关问题（包括代码与题目 ID） 2. 编程答疑引擎接收请求并获取该题目的历史答题信息 3. LLM 服务结合作业数据与评测历史，分析问题并构建 prompt 4. LLM 服务返回智能答疑建议（如修改建议或代码提示） 5. 编程答疑引擎返回答疑结果至前端 6. 前端将答疑建议展示给学生查看与理解

1.3 教师管理后台功能需求

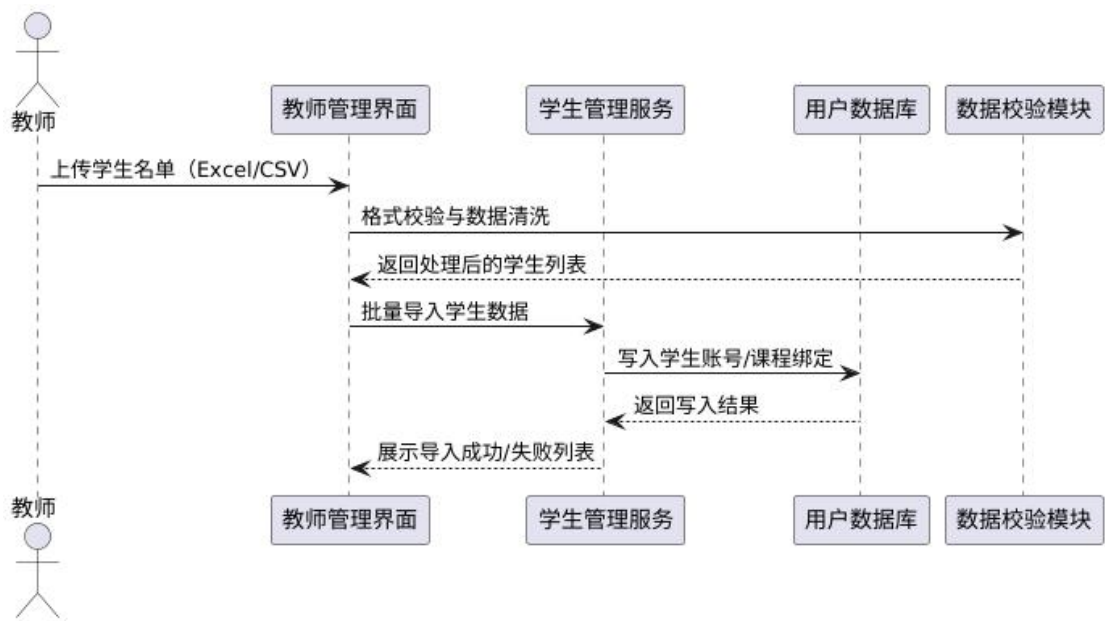
3.1.1 课程创建与资料导入



课程创建与资料导入用例表

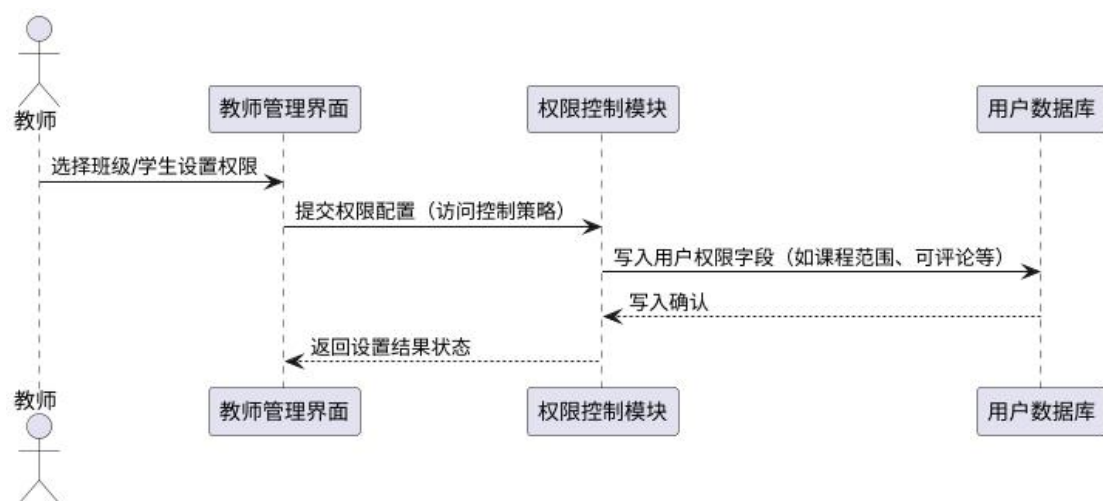
用例名	课程创建与资料导入		
用例描述	教师在后台填写课程信息并上传资源，系统完成课程创建并将资料保存，触发数据清洗及构建任务		
参与者	教师		
过程	步骤	操作	系统响应
课程创建	1	教师填写课程信息并点击创建课程	
	2	教师后台界面提交请求至课程管理服务	3. 课程管理服务创建课程记录并返回课程 ID
	4	教师上传课程资料（视频/作业/题目）	
	5	教师后台将文件上传至资源存储服务并附带课程 ID	6. 资源存储服务返回存储结果
数据处理	7	同时触发数据清洗模块进行 OCR/语音转文本/格式化处理	
数据更新	8	清洗后结果用于构建知识图谱任务，更新课程数据库信息	

3.1.2 导入学生



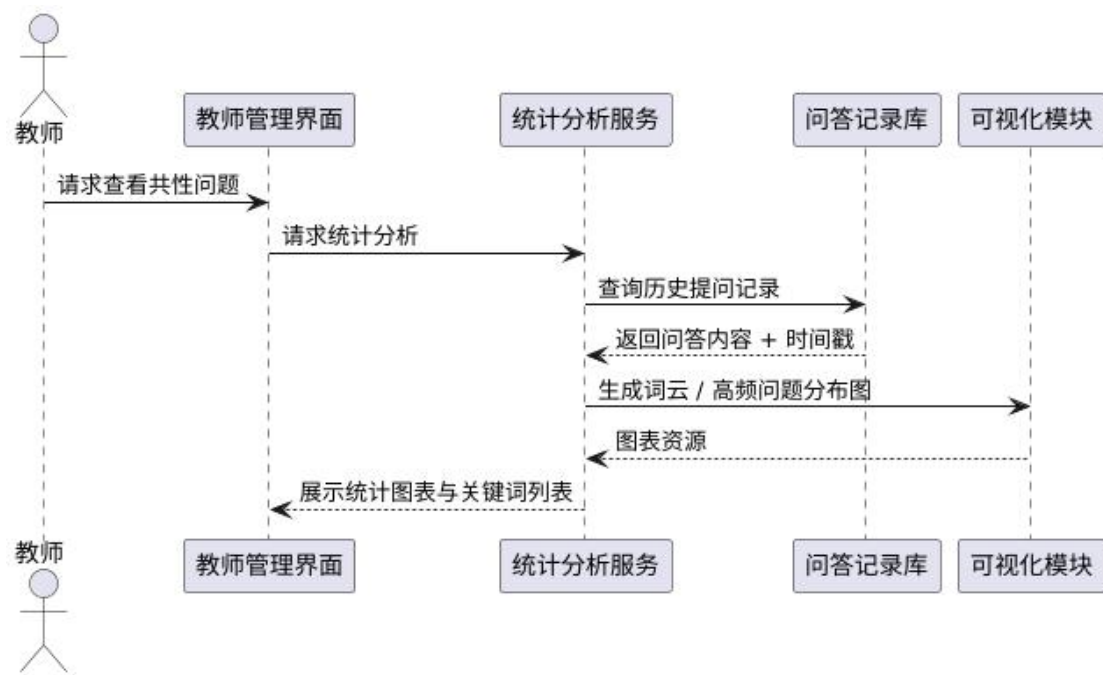
用例名	学生导入模块
用例描述	教师通过上传学生名单文件（Excel/CSV），系统对数据进行校验与清洗后，批量导入学生信息并完成课程绑定
参与者	教师
过程	<div>1. 教师在教师管理界面上传学生名单（Excel/CSV 格式）</div> <div>2. 学生管理服务调用数据校验模块，执行格式校验与数据清洗</div> <div>3. 数据校验模块返回处理后的学生列表</div> <div>4. 教师管理界面展示处理结果</div> <div>5. 教师确认后发起批量导入请求</div> <div>6. 学生管理服务写入学生账号和课程绑定信息至用户数据库</div> <div>7. 用户数据库返回写入结果</div> <div>8. 学生管理服务返回写入成功/失败列表</div> <div>9. 教师界面展示最终导入结果</div>

1.3.3 权限设置



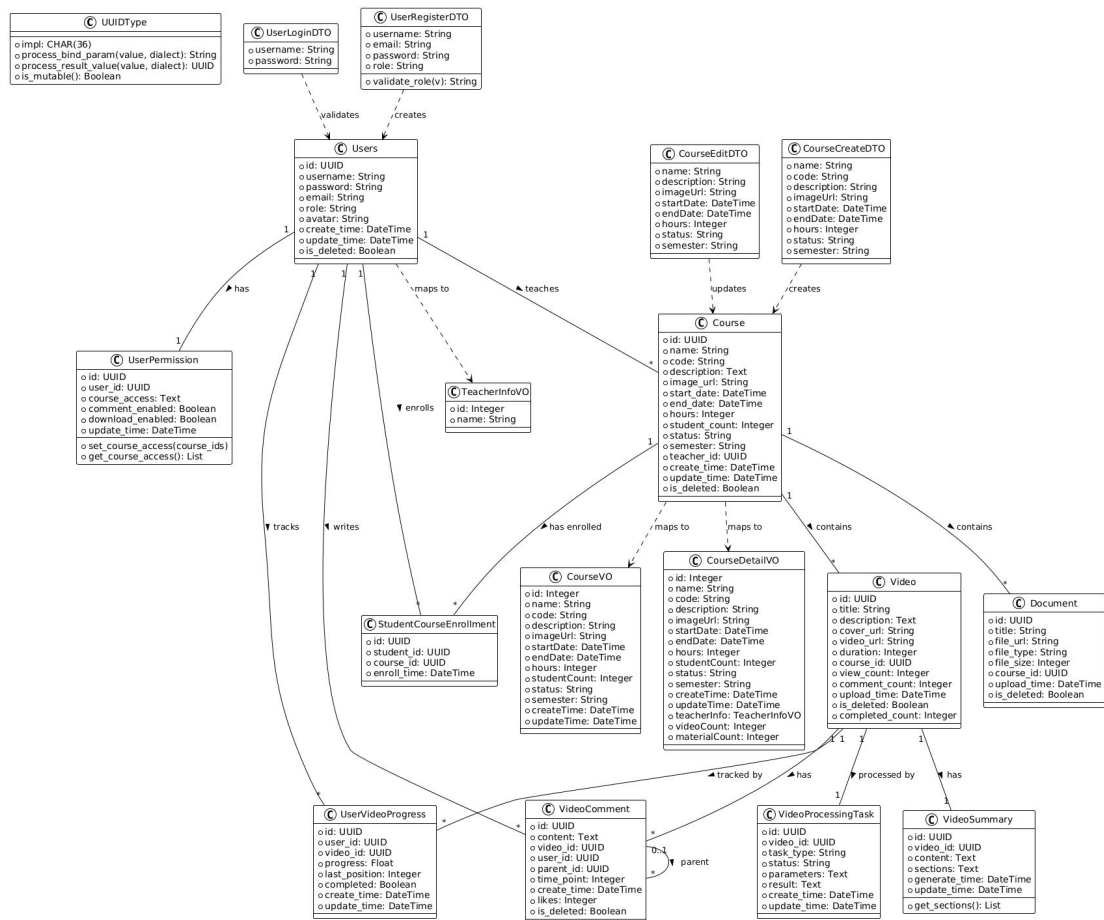
用例名	权限设置
用例描述	教师通过管理界面为特定班级或学生设置权限，系统将权限配置写入用户权限字段（如课程范围、评论权限等）
参与者	教师
过程	<ol style="list-style-type: none"> 1. 教师在教师管理界面中选择班级或学生，并设置权限配置 2. 教师管理界面提交权限配置（访问控制策略） 3. 权限控制模块接收配置请求，写入用户权限字段（如课程范围、评论权限等） 4. 权限控制模块将权限数据写入用户数据库 5. 用户数据库返回写入确认 6. 权限控制模块将设置结果状态返回给教师管理界面，教师收到反馈

1.3.4 共性问题统计与诊断



项目	内容
用例名	共性问题统计与诊断
用例描述	教师通过后台发起共性问题分析请求，系统基于学生的答题记录与提问内容，统计常见错误、高频问题并生成可视化图表
参与者	教师
过程	<ol style="list-style-type: none"> 1. 教师在教师管理界面请求查看共性问题 2. 教师管理界面向统计分析服务发送请求 3. 统计分析服务查询答题记录数据库获取问题与回答内容 4. 返回答题内容与时间戳 5. 统计分析服务生成词云/高频问题分布图等图表 6. 图表生成依赖可视化模块完成渲染资源 7. 最终返回图表资源给教师管理界面展示统计图表与关键词列表

2. 类设计



基础类型

- UUIDType
 - 作用：自定义的 SQLAlchemy 类型转换器，用于处理 UUID 和数据库字符串之间的转换。
 - 功能：确保 UUID 能正确存储在数据库中，并在查询时正确转换回 Python 的 UUID 对象。

用户相关类

- Users
 - 作用：存储系统中所有用户的基本信息。
 - 类型：包括教师和学生两种角色。
 - 关系：
 - 一个用户可以教授多个课程。

- 一个用户可以选修多个课程。
- 一个用户可以发表多个评论。
- 一个用户拥有多条学习进度记录。
- UserPermission
 - 作用：管理用户的权限设置。
 - 功能：控制用户可访问的课程、评论权限、下载权限等。
 - 特点：使用 JSON 格式存储课程访问权限列表。

课程相关类

- Course
 - 作用：存储系统中所有课程的信息。
 - 内容：包含课程名称、代码、描述、时间安排、学时数等基本信息。
 - 关系：
 - 一个课程由一名教师教授。
 - 一个课程包含多个视频资源。
 - 一个课程包含多个文档资源。
- StudentCourseEnrollment
 - 作用：记录学生选课信息，建立学生和课程之间的多对多关系。
 - 功能：跟踪哪些学生选修了哪些课程以及选课时间。

视频相关类

- Video
 - 作用：存储课程的视频资源。
 - 内容：包含视频标题、描述、链接、时长、观看数等信息。
 - 功能：提供完成率计算、关联评论和学习进度。
 - 关系：
 - 属于特定的课程。
 - 拥有多个评论。
 - 拥有多个学习进度记录。
- VideoComment
 - 作用：存储用户对视频的评论。
 - 特点：支持评论回复功能（通过自引用关系）。

- 功能：记录评论在视频中的时间点，支持点赞计数。
- UserVideoProgress
 - 作用：记录用户观看视频的进度。
 - 内容：存储观看进度百分比、最后位置、是否完成。
 - 功能：支持断点续看和学习进度追踪。
- VideoSummary
 - 作用：存储视频的智能摘要信息。
 - 内容：包含主要观点、关键词、章节结构等。
 - 功能：提供 JSON 格式的章节信息存取方法。
- VideoProcessingTask
 - 作用：管理视频处理任务的状态和进度。
 - 类型：支持转码、缩略图生成、字幕提取等处理类型。
 - 功能：跟踪处理状态、进度、错误信息等。

文档相关类

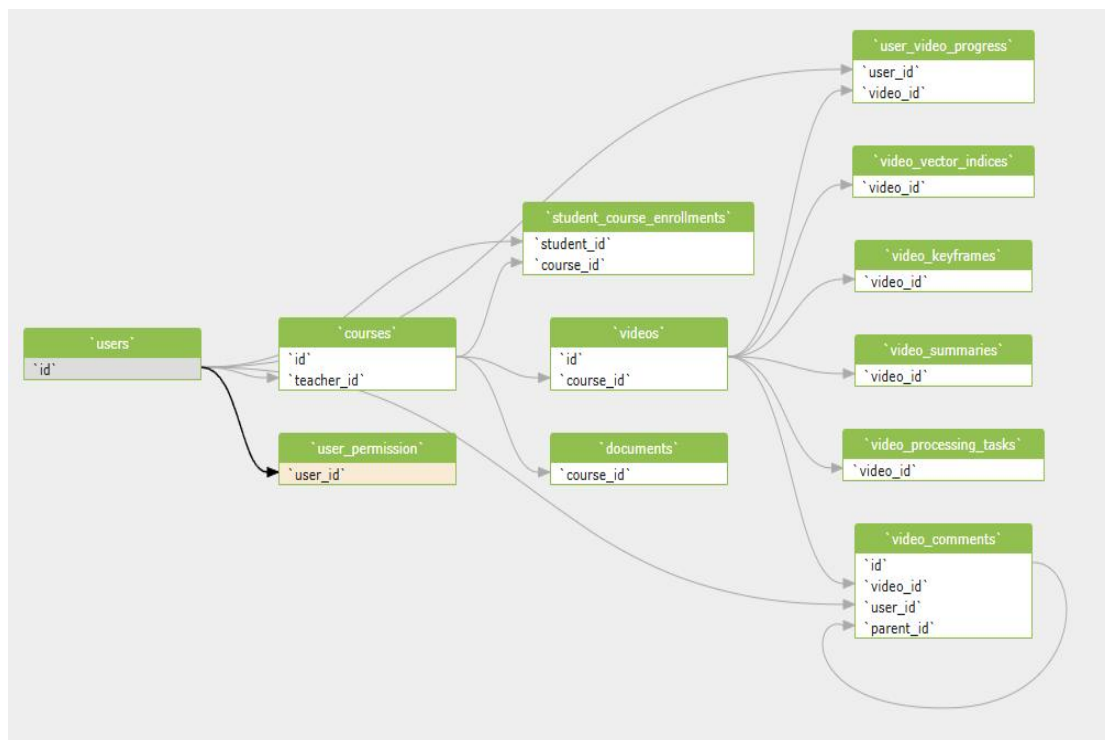
- Document
 - 作用：存储课程的文档资料。
 - 内容：包含文件标题、URL、类型、大小等基本信息。
 - 关系：属于特定的课程。

这些类共同构成了完整的在线教育平台数据模型，通过精心设计的关系网络，实现了：

- 课程管理：教师创建和管理课程内容。
- 学习资源组织：视频和文档等多种学习资源的管理。
- 用户互动：评论系统支持学习交流。
- 学习进度追踪：记录并分析用户的学习行为。

智能内容处理：自动生成视频摘要、关键词等。

3. 数据设计



数据关联关系如上图，图中只展示了具有关联的数据字段。

1. 数据库设计概述

闻道平台是一个融合大型语言模型的智能学习平台，其数据设计围绕用户、课程、视频和 AI 功能等核心业务实体展开。系统采用关系型数据库 MySQL 作为数据存储基础，通过 SQLAlchemy ORM 框架进行数据操作，实现了业务逻辑与数据访问的解耦。

数据库架构设计遵循以下原则：

实体完整性：每个表都有唯一标识的主键，使用 UUID 确保全局唯一性

引用完整性：通过外键约束维护实体间关系

业务完整性：使用约束和触发器保证业务规则实现

可扩展性：模块化设计，便于未来功能扩展

数据追溯：关键操作记录创建和更新时间

2. 核心数据实体

2.1 用户数据模型

用户是系统的基础，分为教师和学生两种角色，通过 Users 表实现统一管理。

用户数据模型包含基本信息、认证信息和状态信息。用户与其他实体的关联通

过外键和关系映射实现。

特点:

统一用户表, 通过 role 字段区分用户角色

邮箱和用户名唯一性保证

支持逻辑删除, 保留历史数据

UUID 主键提高安全性

字段设计:

id: UUID 类型主键, 全局唯一标识用户

username: 用户登录名, 唯一

password: 加密存储的密码

email: 电子邮箱, 唯一

role: 用户角色 (teacher/student)

avatar: 用户头像 URL

create_time/update_time: 创建和更新时间戳

is_deleted: 逻辑删除标记

2.2 课程数据模型

课程是教学内容的组织单元, 通过 Course 表实现。课程与教师是一对多关系, 一个教师可以教授多门课程。课程与学生是多对多关系, 通过 StudentCourseEnrollment 关联表实现。

特点:

丰富的课程元数据, 便于检索和展示

灵活的状态管理, 支持课程生命周期

学期与课时管理, 符合教学实际

统计数据缓存, 如学生数量

字段设计:

id: UUID 类型主键

name: 课程名称

code: 课程代码, 便于检索

description: 课程描述

image_url: 课程封面图

start_date/end_date: 课程时间段

hours: 课时数
student_count: 选课人数
status: 课程状态 (active/inactive)
semester: 学期信息
teacher_id: 教师外键
create_time/update_time: 时间戳
is_deleted: 逻辑删除标记

2.3 视频数据模型

视频是平台核心的学习内容，通过 Video 表实现。每个视频关联到特定课程，并包含丰富的元数据和统计信息。

特点：

- 完整的视频元数据支持
- 智能分析结果存储
- 用户交互数据统计
- 多媒体资源管理

字段设计：

id: UUID 类型主键
title: 视频标题
description: 视频描述
cover_url: 封面图
video_url: 视频文件 URL
duration: 视频时长(秒)
course_id: 关联课程外键
view_count: 观看次数
comment_count: 评论数量
completed_count: 完成观看人数
upload_time: 上传时间
is_deleted: 逻辑删除标记

2.4 学习进度模型

学习进度是个性化学习体验的基础，通过 UserVideoProgress 表记录每个学生对每个视频的学习状态和进度。

特点:

- 精确记录学习位置
- 支持断点续播
- 学习完成状态跟踪
- 多维度学习数据分析

字段设计:

- id: UUID 类型主键
- user_id: 用户外键
- video_id: 视频外键
- progress: 进度比例(0 1)
- last_position: 上次观看位置(秒)
- completed: 是否完成观看
- update_time: 更新时间

3. 智能 AI 特性数据模型

3.1 视频摘要模型

系统自动生成视频内容摘要，通过 VideoSummary 表存储。每个视频对应一条摘要记录，包含主要知识点、关键词和章节信息。

特点:

- 结构化知识点提取
- 关键词索引支持
- 章节划分逻辑保存
- JSON 格式灵活存储

字段设计:

- id: UUID 类型主键
- video_id: 视频外键
- main_points: 主要观点文本
- keywords: 关键词列表
- sections: JSON 格式章节摘要
- generate_time: 生成时间

3.2 视频关键帧模型

视频关键帧是 AI 分析的基础单元，通过 VideoKeyframe 表存储。系统提取视频

关键帧, 并进行 OCR(光学字符识别)和 ASR(语音识别)处理。

特点:

- 多模态内容提取
- 结构化存储文本内容
- 精确时间点标记
- 支持智能检索

字段设计:

- id: 自增主键
- video_id: 视频外键
- frame_number: 帧编号
- time_point: 时间点(秒)
- time_formatted: 格式化时间
- file_name: 关键帧图片文件名
- ocr_result: OCR 识别结果(JSON)
- asr_texts: 语音识别文本
- create_time: 创建时间

3.3 向量索引模型

向量索引是实现语义搜索和智能问答的基础, 通过 VideoVectorIndex 表管理。系统为每个视频创建向量索引, 支持基于内容的相似度检索。

特点:

- 支持高维向量存储
- 模型版本管理
- 索引文件路径记录
- 向量统计信息

字段设计:

- id: 自增主键
- video_id: 视频外键
- index_path: 索引文件路径
- embedding_model: 嵌入模型名称
- total_vectors: 向量总数
- create_time/update_time: 时间戳

3.4 视频处理任务模型

视频智能处理涉及多个异步任务，通过 VideoProcessingTask 表跟踪任务状态和进度。

特点：

任务状态生命周期管理

进度实时更新

错误追踪与恢复

多种处理类型支持

字段设计：

id: UUID 类型主键

video_id: 视频外键

task_id: 任务唯一标识

status: 任务状态(pending/processing/completed/failed)

processing_type: 处理类型

progress: 处理进度(0 100)

error_message: 错误信息

start_time/end_time: 开始和结束时间

4. 交互与社交数据模型

4.1 视频评论模型

视频评论实现师生和生生之间的交流，通过 VideoComment 表存储。支持对视频特定时间点的评论和评论的多级回复。

特点：

分层评论结构设计

时间点关联评论

点赞统计支持

字段设计：

id: UUID 类型主键

content: 评论内容

video_id: 视频外键

user_id: 用户外键

parent_id: 父评论 ID(用于回复)

time_point: 视频时间点(秒)

create_time: 创建时间

likes: 点赞数

is_deleted: 逻辑删除标记

4.2 课程选修模型

学生选修课程的数据通过 StudentCourseEnrollment 表维护, 实现了用户与课程的多对多关系。

特点:

唯一约束确保不重复选课

选课时间记录

支持选课状态扩展

简洁高效的关系设计

字段设计:

id: UUID 类型主键

student_id: 学生外键

course_id: 课程外键

enroll_time: 选课时间

5. 权限与访问控制

5.1 用户权限模型

系统实现了细粒度的权限控制, 通过 UserPermission 表管理用户对特定资源的访问权限。

特点:

基于课程的访问控制

功能级权限设置

JSON 格式灵活权限记录

动态权限调整支持

字段设计:

id: UUID 类型主键

user_id: 用户外键

course_access: 可访问课程列表(JSON)

comment_enabled: 评论权限

download_enabled: 下载权限

update_time: 更新时间

6. 数据关系设计

系统通过外键约束和 ORM 关系映射, 实现了实体间的有机关联。主要关系包括:

6.1 一对多关系

教师(Users) 1:N 课程(Course): 一个教师可以教授多个课程

课程(Course) 1:N 视频(Video): 一个课程包含多个视频

课程(Course) 1:N 文档(Document): 一个课程包含多个文档

用户(Users) 1:N 评论(VideoComment): 一个用户可发表多个评论

视频(Video) 1:N 评论(VideoComment): 一个视频有多个评论

视频(Video) 1:N 关键帧(VideoKeyframe): 一个视频有多个关键帧

评论(VideoComment) 1:N 回复(VideoComment): 评论支持多级回复

6.2 一对一关系

视频(Video) 1:1 摘要(VideoSummary): 一个视频有一个摘要

用户(Users) 1:1 权限(UserPermission): 一个用户有一个权限记录

6.3 多对多关系

学生(Users) M:N 课程(Course): 通过 StudentCourseEnrollment 关联表实现

用户(Users) M:N 视频(Video): 通过 UserVideoProgress 关联表实现进度记录

7. 数据库优化设计

系统在数据库设计层面进行了多项优化:

7.1 性能优化

合理的索引策略: 主键、外键和常用查询字段建立索引

使用 UUID 作为主键, 避免自增 ID 的局限性

数据分区与分表设计, 为大规模数据做准备

使用逻辑删除代替物理删除, 提高数据完整性

7.2 数据安全

密码加密存储

数据访问权限控制

敏感数据隔离

数据备份与恢复机制

7.3 扩展性设计

JSON 字段存储灵活结构数据

预留扩展字段

松耦合的表关系设计

模块化的数据架构

8. 数据流设计

系统主要数据流包括:

8.1 课程内容数据流

教师 → 课程创建 → 视频上传 → 视频处理 → 内容分析 → 向量索引生成
→ 学生访问

8.2 学习进度数据流

学生 → 视频播放 → 进度记录 → 数据分析 → 学习报告 → 个性化推荐

8.3 交互数据流

用户 → 提交评论/问题 → 数据存储 → AI 分析 → 智能回复/推荐