

同伴课堂

概要设计说明书

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | 第X组-同伴课堂-SD |
| 当前版本： | 1.0 |
| 作 者： |  |
| 完成日期： | 2024/10/15 |

卓越工程综合实践课程

X班第X组

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| 1.0 |  |  | 2024/10/09至2024/10/15 |  |

目 录

[0. 文档介绍 5](#_Toc183949220)

[0.1 文档目的 5](#_Toc183949221)

[0.2 文档范围 5](#_Toc183949222)

[0.3 读者对象 5](#_Toc183949223)

[0.4 参考文献 5](#_Toc183949224)

[0.5 术语与缩写解释 5](#_Toc183949225)

[1. 系统概述 6](#_Toc183949226)

[1.1 系统定义 6](#_Toc183949227)

[1.2 主要功能 6](#_Toc183949228)

[2. 设计约束 6](#_Toc183949229)

[2.1 需求约束 6](#_Toc183949230)

[2.2 技术约束 6](#_Toc183949231)

[2.3 操作约束 6](#_Toc183949232)

[2.4 法律与标准约束 7](#_Toc183949233)

[2.5 隐含约束 7](#_Toc183949234)

[3. 设计策略 7](#_Toc183949235)

[3.1 扩展策略 7](#_Toc183949236)

[3.2 复用策略 7](#_Toc183949237)

[3.3 折衷策略 8](#_Toc183949238)

[3.4 可维护性策略 8](#_Toc183949239)

[4. 系统总体结构 8](#_Toc183949240)

[4.1 系统架构图 8](#_Toc183949241)

[4.2 系统设计框架 9](#_Toc183949242)

[4.3 技术设计框架 9](#_Toc183949243)

[4.3.1概述 9](#_Toc183949244)

[4.3.2移动端 10](#_Toc183949245)

[4.3.3网页端 10](#_Toc183949246)

[4.3.4后端 11](#_Toc183949247)

[4.4功能框架图 12](#_Toc183949248)

[5子系统功能 12](#_Toc183949249)

[6用户界面设计 13](#_Toc183949250)

[6.1学生端 13](#_Toc183949251)

[6.1.1个人认证部分 13](#_Toc183949252)

[6.1.2课程部分 13](#_Toc183949253)

[6.1.3论坛部分 14](#_Toc183949254)

[6.1.4课堂部分 14](#_Toc183949255)

[6.1.5课前预习测验部分 15](#_Toc183949256)

[6.1.6上课测验答题部分 15](#_Toc183949257)

[6.1.7添加课程部分 16](#_Toc183949258)

[6.1.8个人信息部分 16](#_Toc183949259)

[6.1.9消息推送部分 17](#_Toc183949260)

[6.2教师端 17](#_Toc183949261)

[6.2.1课堂签到 17](#_Toc183949262)

[6.2.2.发布题目 18](#_Toc183949263)

[6.2.3数据统计 19](#_Toc183949264)

[6.2.4学生管理 19](#_Toc183949265)

[6.2.5课程管理 20](#_Toc183949266)

[6.2.6.新建课程 20](#_Toc183949267)

[6.2.7讨论区管理 21](#_Toc183949268)

[6.2.8预习资料 21](#_Toc183949269)

[6.2.9预习测试 22](#_Toc183949270)

[6.2.10题库管理 22](#_Toc183949271)

[6.2.11添加题目 23](#_Toc183949272)

[6.3管理员端 23](#_Toc183949273)

[6.3.1课程管理 23](#_Toc183949274)

[6.3.2查看课程数据 24](#_Toc183949275)

[6.3.3封禁/解封课程 25](#_Toc183949276)

[6.3.4学生管理 25](#_Toc183949277)

[6.3.5查看学生信息 26](#_Toc183949278)

[6.3.6封禁/解封学生 26](#_Toc183949279)

[6.3.7教师管理 27](#_Toc183949280)

[6.3.8查看教师信息 27](#_Toc183949281)

[6.3.9封禁/解封教师 28](#_Toc183949282)

[7. 开发环境的配置 28](#_Toc183949283)

[8. 运行环境的配置 29](#_Toc183949284)

[9. 测试环境的配置 29](#_Toc183949285)

# 0. 文档介绍

## 0.1 文档目的

本概要设计说明书旨在详细阐述同伴课堂的体系结构设计，包括系统的总体架构、子系统的划分、模块设计以及关键的设计决策。本文档为项目团队提供了一个清晰的技术蓝图，确保开发过程中的一致性和高效性，同时也为未来的维护和升级工作打下基础。

## 0.2 文档范围

本文档覆盖了同伴课堂的体系结构设计，包括但不限于系统概述、设计约束、设计策略、系统总体结构、子系统结构与功能、开发环境配置以及运行环境配置。它详细描述了从系统设计到部署的各个阶段，以及每个阶段的具体活动和预期成果。

## 0.3 读者对象

本文档的主要读者包括项目团队成员（开发人员、测试人员、项目经理）以及最终用户（学生和教师）。文档提供了足够的信息，使读者能够理解平台的技术架构和设计决策，以及如何满足他们的需求。

## 0.4 参考文献

[项目计划.docx]：描述了同伴课堂项目的实施框架、目标、范围、时间表、资源分配、 风险管理以及质量保证措施等。

[产品需求规格说明书.docx]：描述了同伴课堂项目的功能性需求和非功能性需求。

## 0.5 术语与缩写解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| SPP | 精简并行过程，Simplified Parallel Process |
| SD | 系统设计，System Design |
| WBS | 工作分解结构，Work Breakdown Structure |

# 1. 系统概述

## 1.1 系统定义

同伴课堂是一个在线教育平台，专为支持和增强同伴教学法在实际教学中的应用而设计。该平台利用现代信息技术，为教师和学生提供一个便捷、高效、互动的学习环境。

## 1.2 主要功能

* 用户管理：包括学生和教师的账户创建、管理和认证。
* 课程管理：允许教师创建和管理课程，学生加入和参与课程。
* 课堂互动：提供实时问答和讨论区等功能。
* 学习监控：教师可以监控学生的学习进度和课堂表现。

# 2. 设计约束

## 2.1 需求约束

* 标准与规范遵守：平台必须遵循教育技术标准和国家教育政策，确保内容和功能符合教育改革方向。
* 环境兼容性：平台需在不同的教育机构的网络环境和设备条件下良好运行。
* 接口/协议约束：系统各组件间的数据交换应遵循定义好的API和通信协议。
* 用户界面约束：界面设计需直观易用，适应不同用户群体的操作习惯。
* 软件质量约束：系统需满足正确性、健壮性、可靠性、性能、易用性、安全性、可扩展性、兼容性、可移植性等质量属性。

## 2.2 技术约束

* 技术选型：前端移动端采用Flutter框架，网页端使用Vue 3，后端使用Spring Boot，确保技术栈的现代性和维护性。
* 开发工具：使用Apifox进行API文档管理和测试，确保开发工具链的高效性。

## 2.3 操作约束

* 用户操作：系统应支持教师和学生的主要操作，如课程管理、课堂互动、成绩跟踪等，且操作流程应简洁明了。
* 数据操作：数据的增删改查（CRUD）应遵循教育数据管理的最佳实践，确保数据的一致性和完整性。

## 2.4 法律与标准约束

* 数据保护：遵守个人信息保护法律法规，如GDPR或相关国家法律，确保用户数据的安全和隐私。
* 内容规范：教学内容需符合教育部门的教学大纲和质量标准，不得包含违法或不适当的内容。

## 2.5 隐含约束

* 用户技能假设：设计时假设用户具备基本的计算机操作技能，能够使用平台进行日常的教学和学习活动。
* 硬件依赖：平台设计需考虑对主流硬件的依赖，如智能手机、平板电脑和个人电脑，确保广泛的可访问性。

# 3. 设计策略

## 3.1 扩展策略

为了确保同伴课堂能够适应未来教育需求的变化和技术的演进，我们将采取以下扩展策略：

* 模块化设计：系统各功能模块化，便于未来根据需要添加新模块或更新现有模块。
* 技术选型：选用具有良好社区支持和发展前景的技术栈，以便于长期维护和升级。

## 3.2 复用策略

* 组件库和框架：利用现有的组件库和框架来加速开发过程，减少重复工作。
* 代码复用：在不同模块和子系统中复用通用代码，确保一致性并减少错误。
* 设计模式：采用业界认可的设计模式来解决常见问题，提高代码的可维护性和可扩展性。

## 3.3 折衷策略

* 性能与资源：在系统性能优化和资源消耗之间找到平衡点，确保系统既能提供流畅的用户体验，又能高效利用服务器和网络资源。
* 用户体验与系统复杂性：在追求优秀的用户体验的同时，控制系统复杂性，避免过度设计导致难以维护。
* 创新与稳定性：在引入创新功能和保持系统稳定性之间做出合理折衷，确保新功能的引入不会对现有系统的稳定性造成负面影响。

## 3.4 可维护性策略

* 文档完整性：提供完整的开发和用户文档，确保所有系统组件都有清晰的说明。
* 代码质量：实施代码审查和持续集成，确保代码质量满足项目标准。
* 模块化：通过模块化设计，降低系统各部分之间的耦合度，提高系统的可维护性。

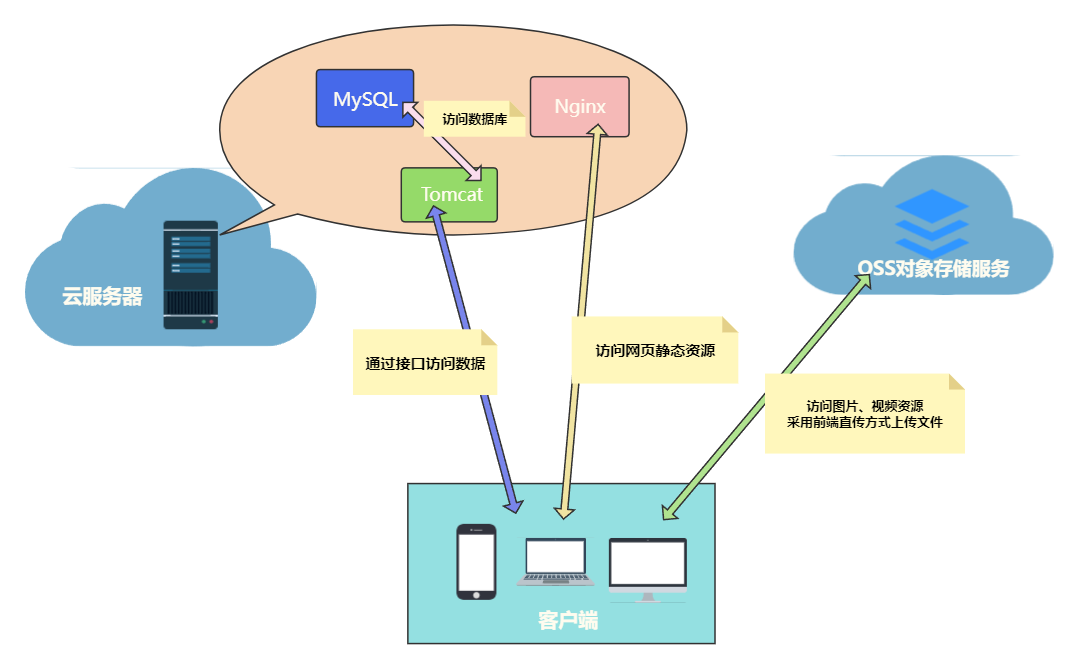
# 4. 系统总体结构

## 4.1 系统架构图

* 逻辑架构图：展示系统的主要组件及其相互关系，包括用户界面、业务逻辑层、数据访问层等。



* 物理架构图：描述系统的物理部署视图，包括服务器、数据库、网络设备等硬件资源的配置和布局。



## 4.2 系统设计框架

同伴课堂采用分层架构设计，以确保系统的可维护性、可扩展性和灵活性。系统主要分为以下几个层次：

* 用户层：提供网页端和移动客户端的用户界面，允许学生、教师和管理员与系统交互。
* 接入层：处理用户认证、反向代理、后端接口和资源访问，作为前端和后端之间的桥梁。
* 应用层：实现系统的核心功能，包括课程管理、学生管理、课堂互动、数据统计等。
* 依赖层：包含系统运行所依赖的第三方库和框架，如Spring Boot、JPA、JWT等。
* 数据层：负责数据的持久化，使用MySQL作为关系型数据库，以及OSS对象存储服务。

## 4.3 技术设计框架

### 4.3.1概述

系统分为学生端、教师端和管理员端三部分。其中学生端采用移动APP形式，教师端和管理员端采用网页形式。

### 4.3.2移动端

移动端采用Flutter框架。Flutter是谷歌推出的一款高性能、跨平台的UI框架，它允许开发者通过一套代码同时支持iOS、Android、Windows、Mac以及Linux等多个平台，且能达到原生应用的性能。

Flutter主要基于Dart语言进行开发。Dart是一种类型安全、支持JIT（即时编译）和AOT（提前编译）的编程语言，其优点如下：

* 跨平台应用开发：

Flutter非常适合用于开发需要同时在多个平台上运行的应用。

通过一套代码即可实现跨平台发布，大大降低了开发成本和时间。

* 原生性能体验：

尽管Flutter是跨平台框架，但它能够提供接近原生应用的性能体验。

这使得Flutter成为开发高性能应用的理想选择。

* 丰富的UI表现力：

Flutter提供了丰富的Widget组件库和自定义Widget的能力。

开发者可以创建出具有独特风格和高度交互性的UI界面。

### 4.3.3网页端

网页端的开发采用 Vue 3 作为主框架，配合 Vite 进行开发和构建，使用 Pinia 作为状态管理工具，并借助 Apifox 进行 API 文档管理和测试。Vue 3 是一个用于构建用户界面的现代 JavaScript 框架，特别适合创建交互性强的单页应用（SPA）。Vue 3 非常适合实时互动的应用，通过与后端的WebSocket或REST API通讯，学生和教师可以进行高效的课堂互动、问题讨论和数据实时展示。通过Vue 3实现前后端分离，前端与后端通过API接口进行通信，提高了项目的可维护性和扩展性，具体来说其优点如下：

* 组件化开发：

Vue 3 允许将页面拆分为独立的组件，方便代码复用和维护。

通过组合式API，开发者可以更灵活地组织和管理状态。

* 响应式数据绑定：

Vue 3 提供了强大的响应式系统，可以根据数据的变化自动更新用户界面，提升开发效率。

* 虚拟DOM：

使用虚拟DOM的机制，Vue 3 可以减少不必要的DOM操作，提升性能。

* 单文件组件（SFC）：

开发者可以在一个文件中同时编写HTML、JavaScript 和 CSS，便于开发和调试。

* 性能优化：

按需加载，使用 Vue 的 路由懒加载，只加载用户访问的页面组件。配合 Vite 插件，实现自动化的路由懒加载。

* API文档与接口一致：

保证前后端接口的一致性，一键测试接口，简化测试流程，提升接口联调效率。

### 4.3.4后端

后端采用 Spring Boot 框架。Spring Boot 是一个基于 Spring 框架的快速开发框架，旨在简化 Java 应用程序的构建和配置。它提供了自动配置、内嵌服务器等功能，极大地提高了开发效率，具体来说其优点如下：

* 快速开发与高效配置：

Spring Boot 通过自动配置和约定优于配置的理念，开发者可以迅速搭建应用，而无需进行繁琐的配置。这使得开发周期大大缩短，特别适合教育系统这种需要快速迭代的项目。

* 强大的生态系统：

Spring Boot 与 Spring 生态系统无缝集成，开发者可以利用 Spring Security、Spring Data、Spring Cloud 等丰富的工具和库，轻松实现安全性、数据访问和微服务架构等功能。

* 内嵌服务器：

Spring Boot 支持内嵌服务器（如 Tomcat、Jetty），开发者可以直接运行应用，而无需单独部署到外部服务器。这简化了部署过程，适合开发和测试阶段的快速验证。

* 易于测试：

Spring Boot 提供了良好的测试支持，能够通过 JUnit 和 Mockito 等工具方便地进行单元测试和集成测试，确保系统的稳定性和可靠性。

* 社区支持与文档丰富：

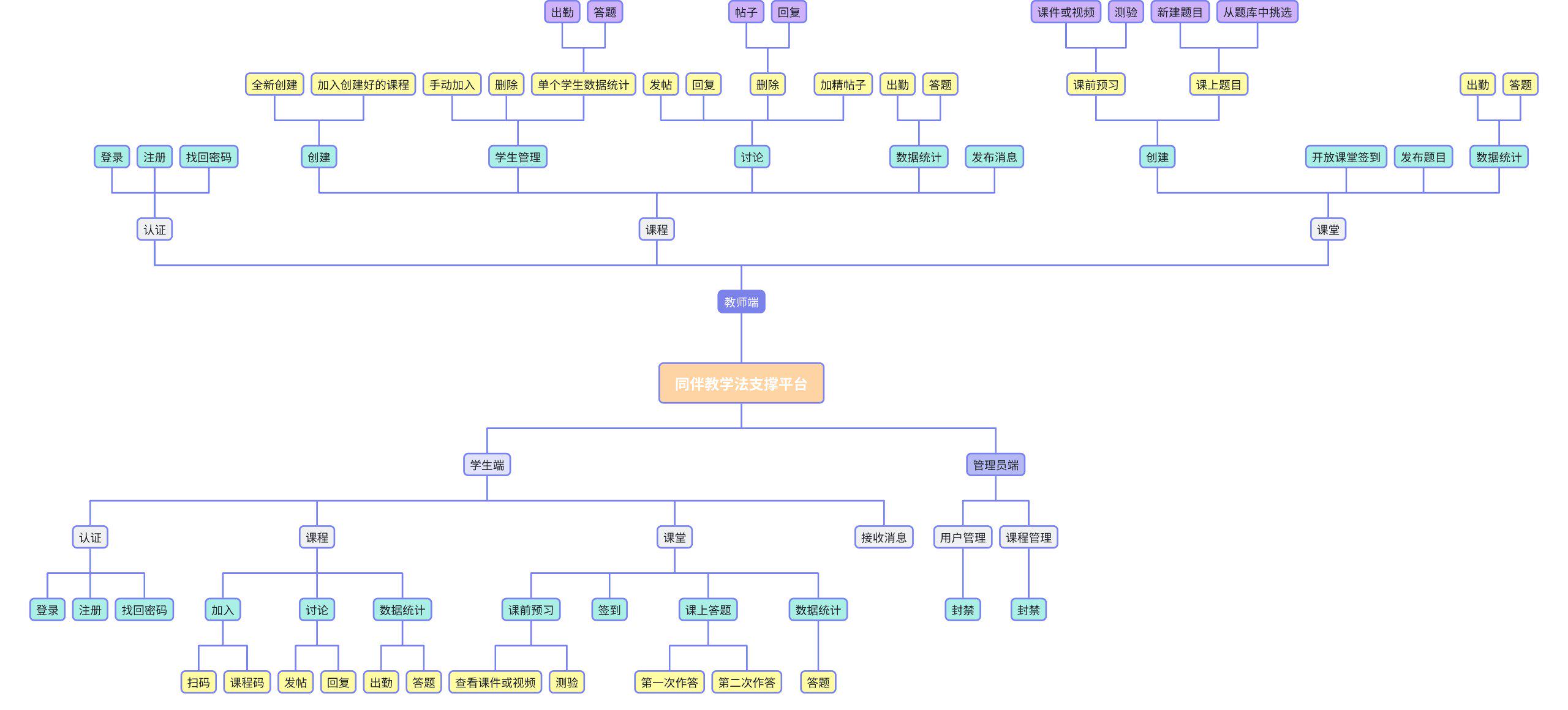
Spring Boot 拥有活跃的社区和大量的文档资源，开发者可以轻松找到所需的资料和解决方案，减少学习成本。

* 可扩展性：

Spring Boot 设计上非常灵活，支持微服务架构的构建，使得教学系统可以根据需求进行扩展，适应未来可能增加的功能和用户量。

Spring Boot 以其高效、灵活和强大的生态系统，非常适合用于开发教学系统的后端。通过使用 Spring Boot，我们可以更专注于业务逻辑的实现，而不是基础架构的搭建。

## 4.4功能框架图



功能框架图详细描述了平台的主要功能模块和子系统，以及它们之间的关系。以下是根据功能框架图的描述：

* 用户认证模块：包括登录、注册和找回密码功能，确保用户身份的合法性。
* 课程管理模块：允许教师创建、管理和加入课程，学生可以加入和查看课程。
* 课堂管理模块：包括课前预习、签到、课上答题和讨论区，增强课堂互动性。
* 数据统计模块：提供出勤率、答题正确率等统计信息，帮助教师和学生了解学习情况。
* 消息发布模块：允许教师发布课程公告，学生接收和查看消息。

# 5子系统功能

根据功能框架图，我们可以将系统分解为以下几个子系统：

* 学生端子系统：

提供学生登录、注册和找回密码的功能。

允许学生加入课程、参与课前预习、课堂签到和答题。

学生可以在讨论区发帖和回复，查看数据统计。

* 教师端子系统：

允许教师创建、管理课程以及管理课程学生。

教师可以发布课堂问题，进行签到管理，以及发布公告。

提供教师查看课堂数据分析，包括学生答题情况和正确率。

教师可以在讨论区发帖和回复。

教师可以管理题库，增、删、改、查题目。

* 管理员端子系统：

管理员可以管理用户账户，封禁或解封用户。

管理员可以管理课程，查看课程数据和进行课程封禁。

# 6用户界面设计

## 6.1学生端

### 6.1.1个人认证部分

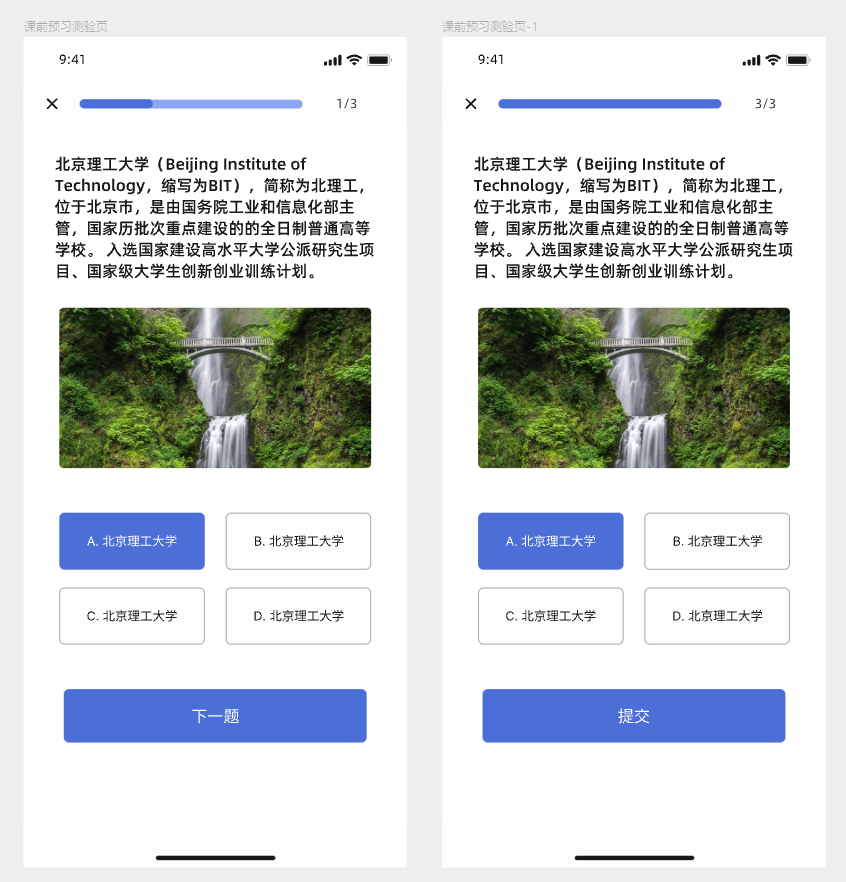
### 6.1.2课程部分

### 6.1.3论坛部分

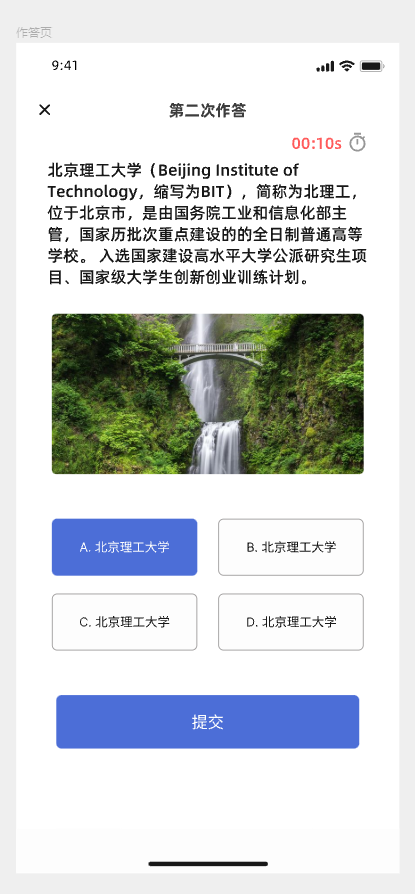
### 6.1.4课堂部分



### 6.1.5课前预习测验部分



### 6.1.6上课测验答题部分



### 6.1.7添加课程部分

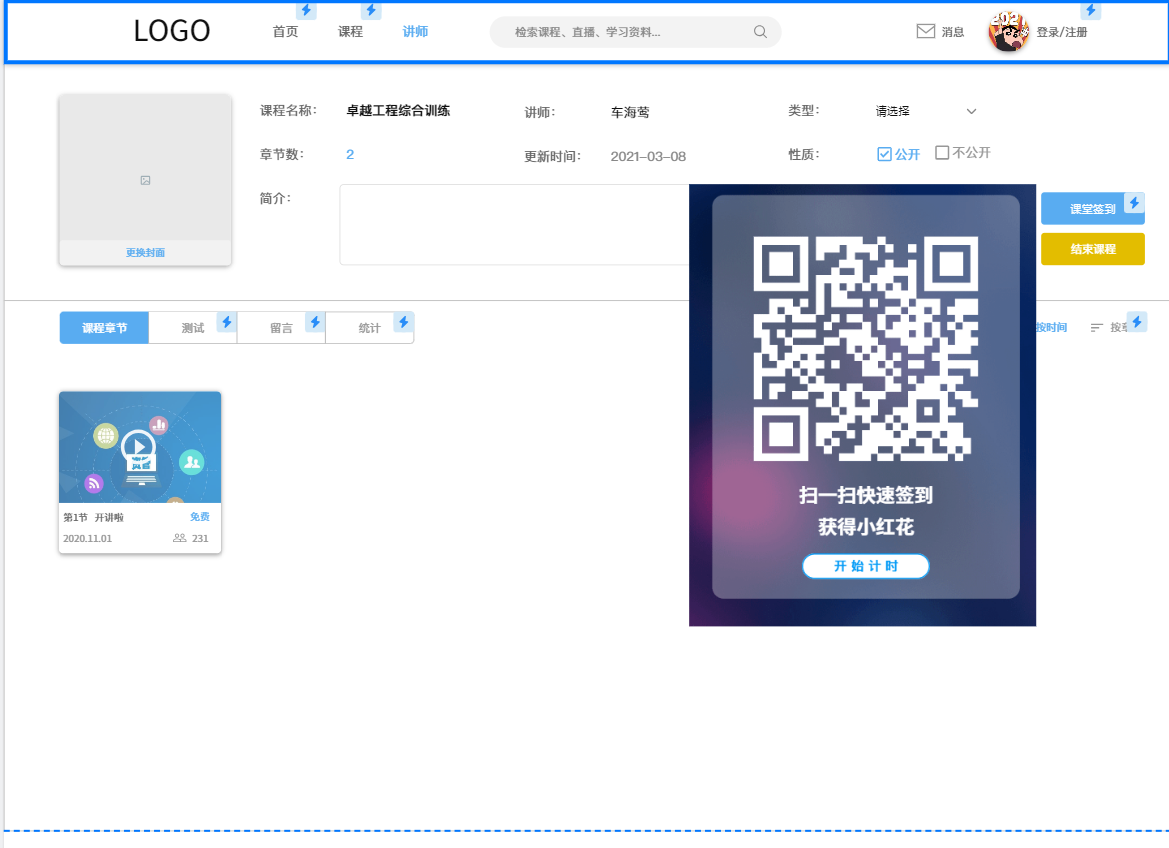
### 6.1.8个人信息部分

### 6.1.9消息推送部分

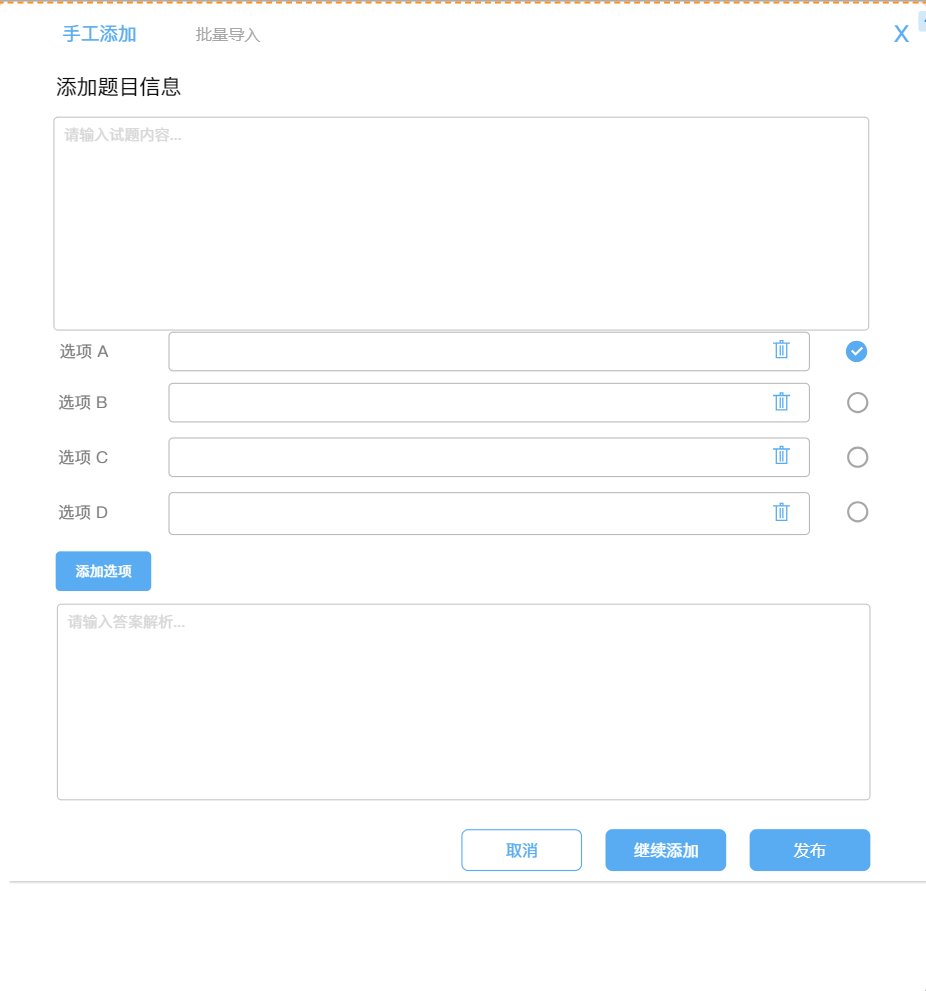
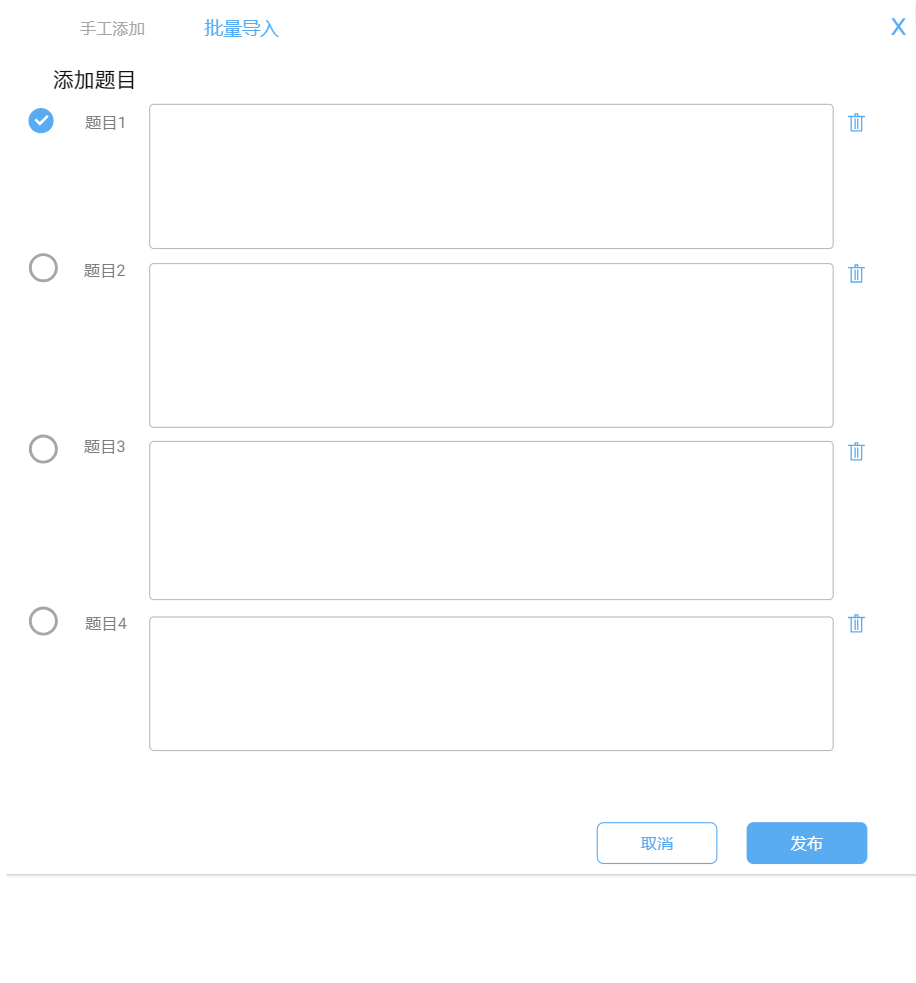


## 6.2教师端

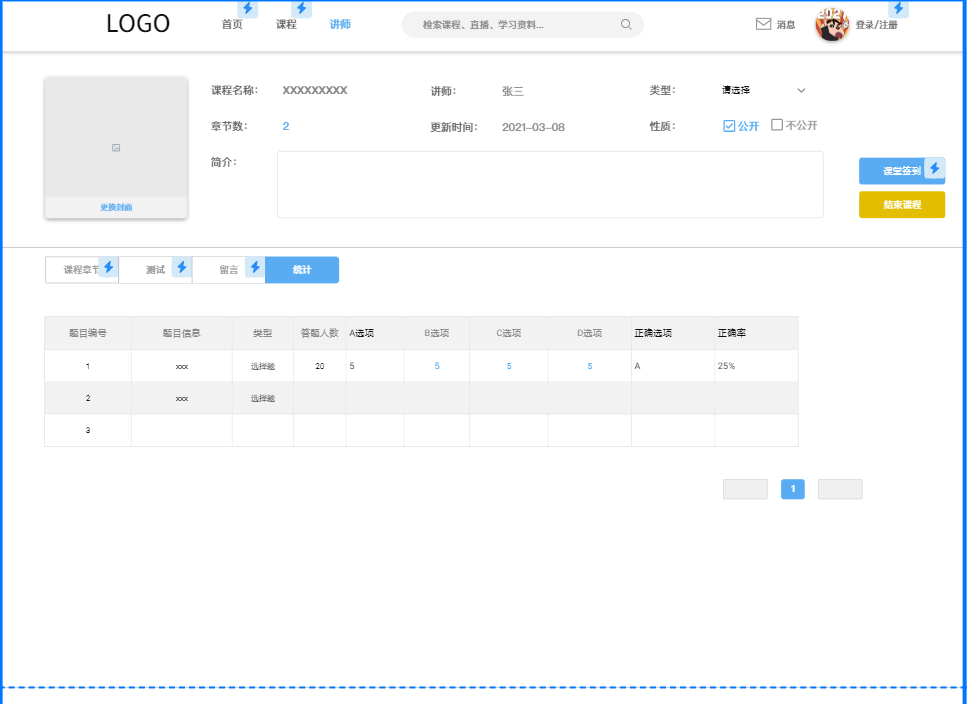
### 6.2.1课堂签到



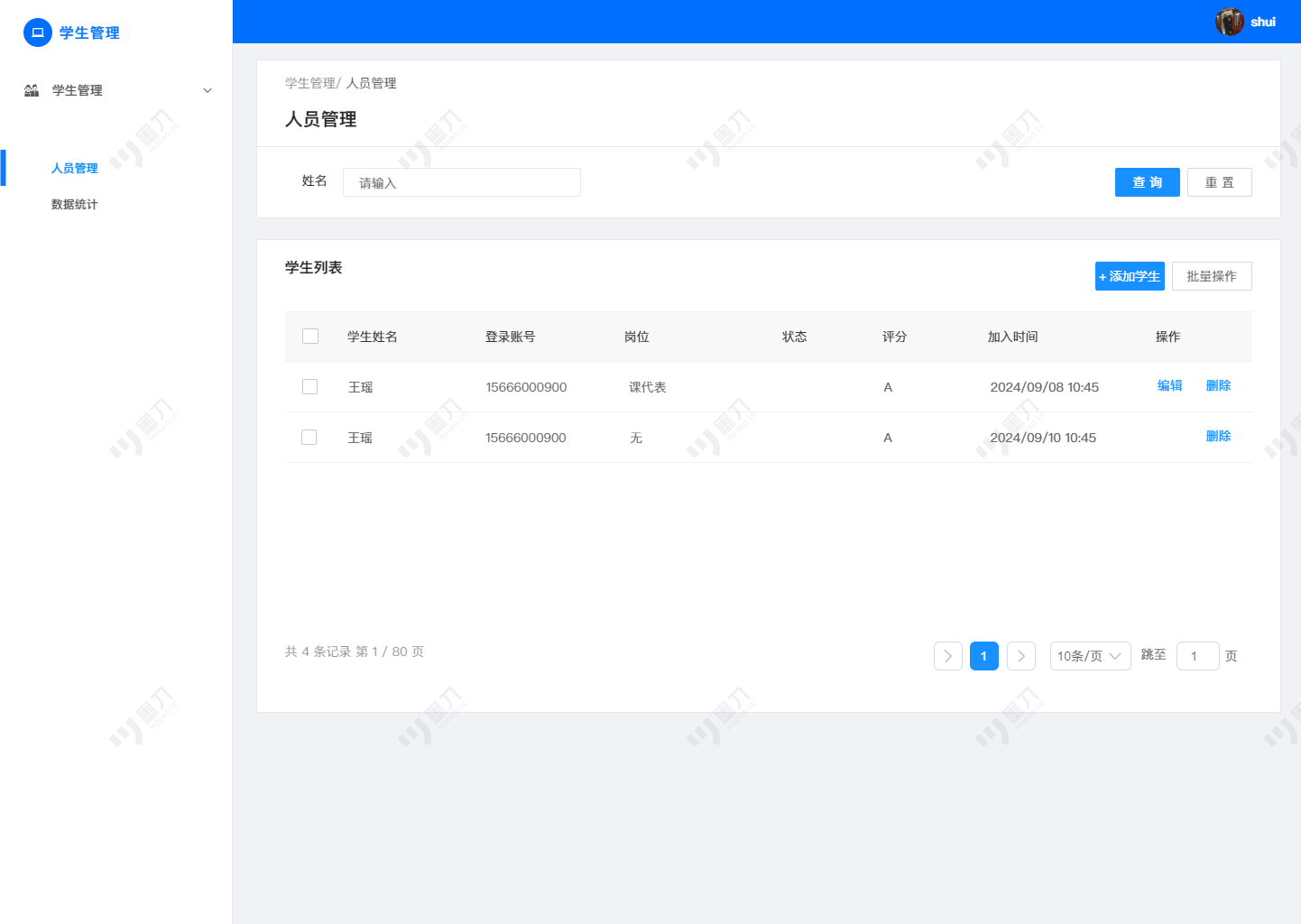
### 6.2.2.发布题目



### 6.2.3数据统计



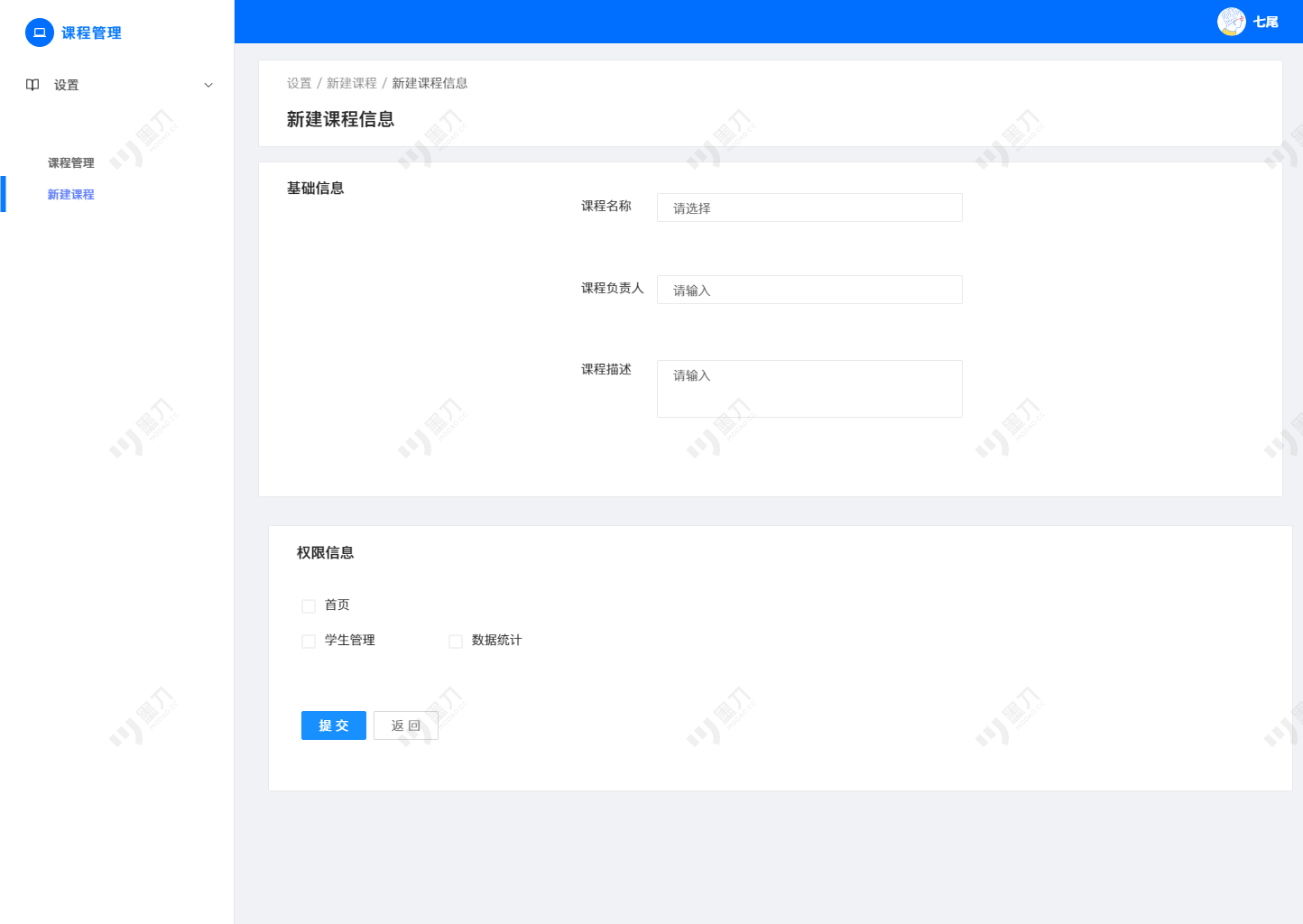
### 6.2.4学生管理



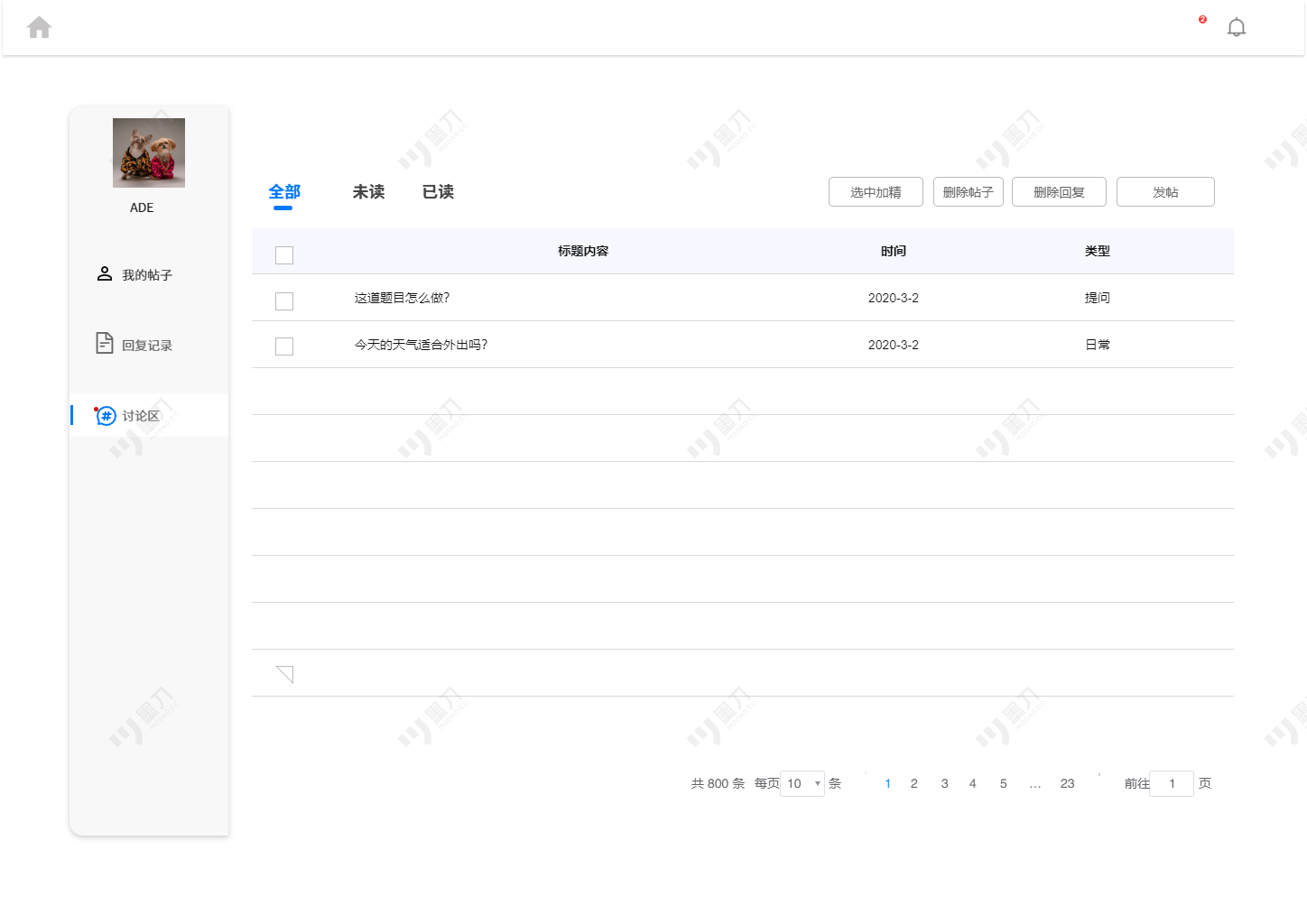
### 6.2.5课程管理



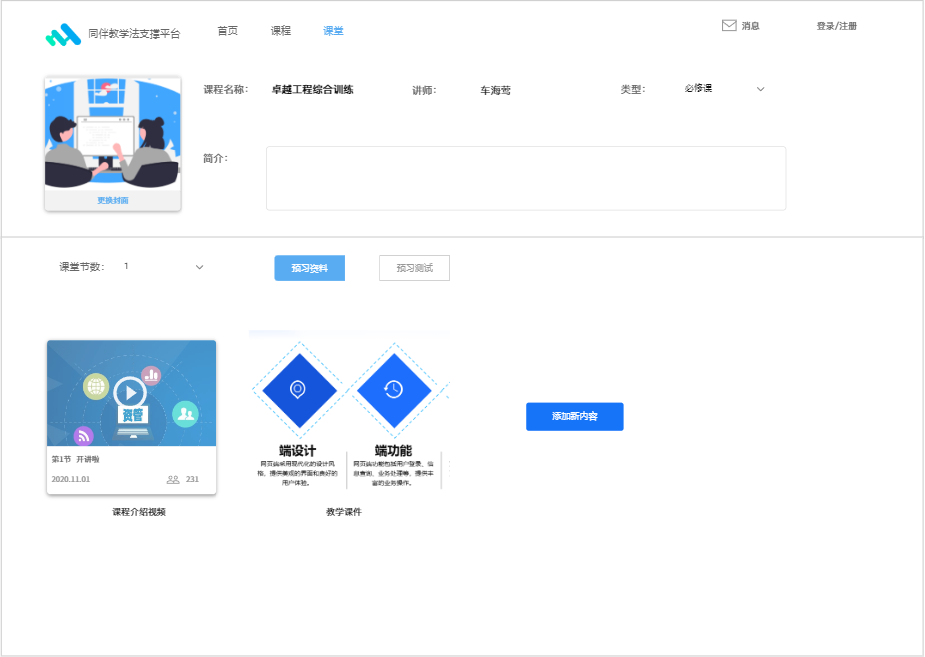
### 6.2.6.新建课程



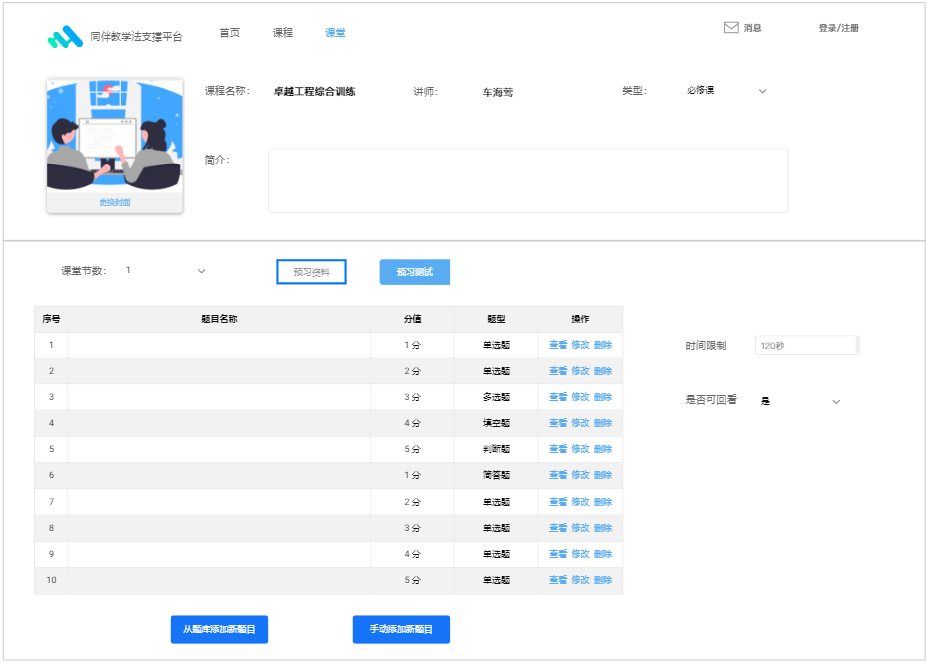
### 6.2.7讨论区管理



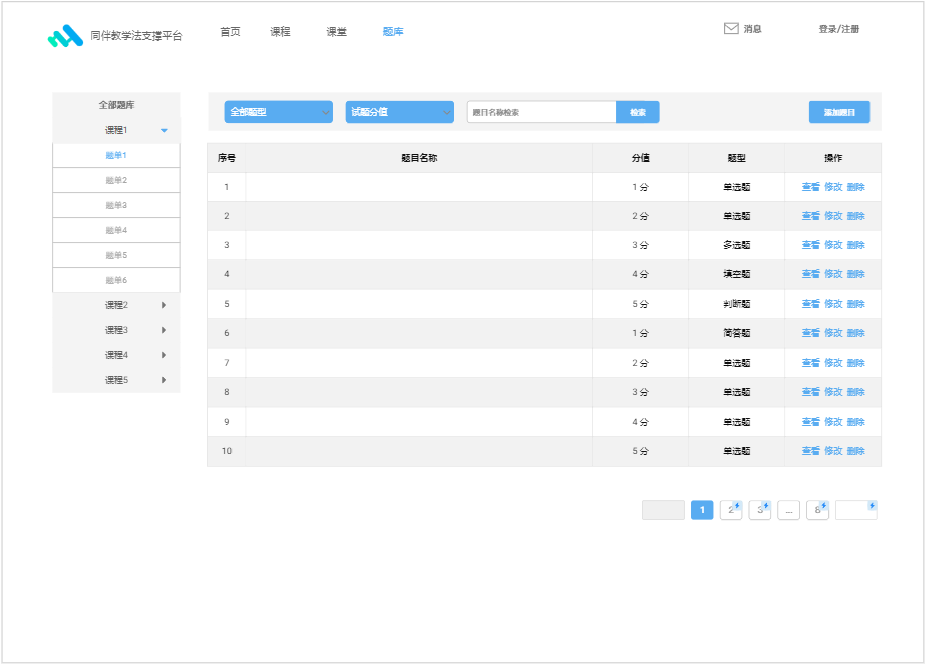
### 6.2.8预习资料



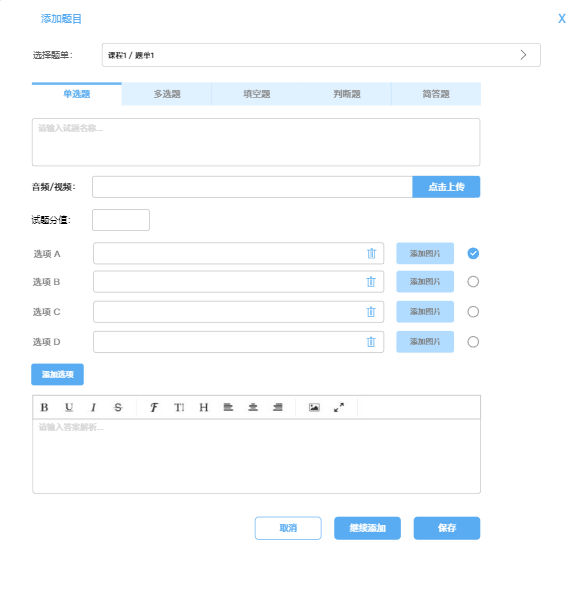
### 6.2.9预习测试



### 6.2.10题库管理

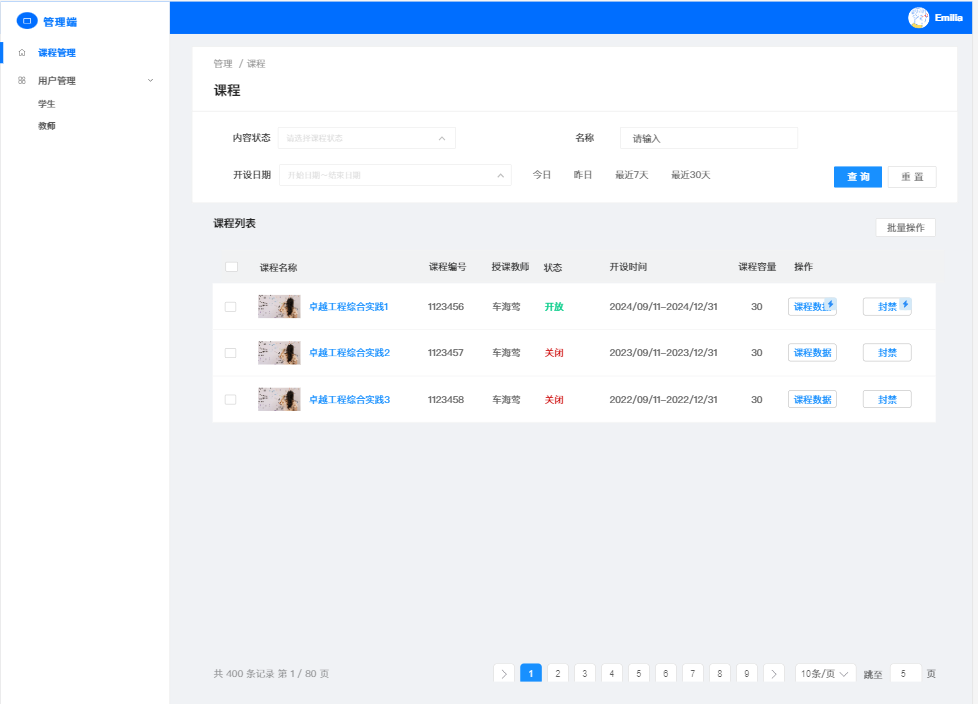
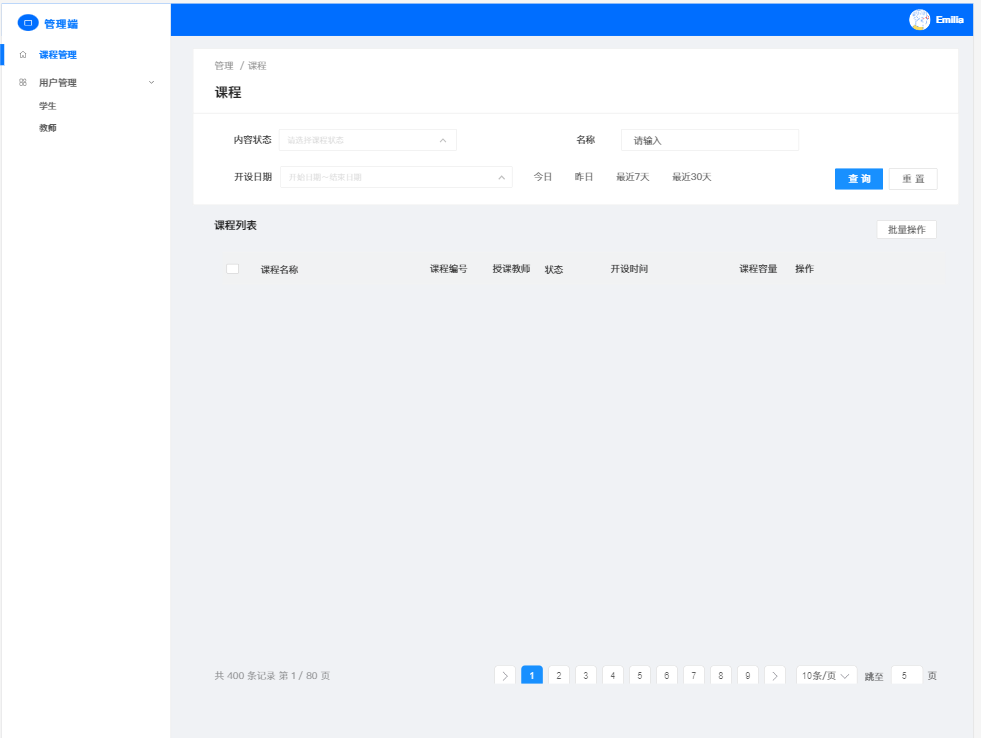


### 6.2.11添加题目



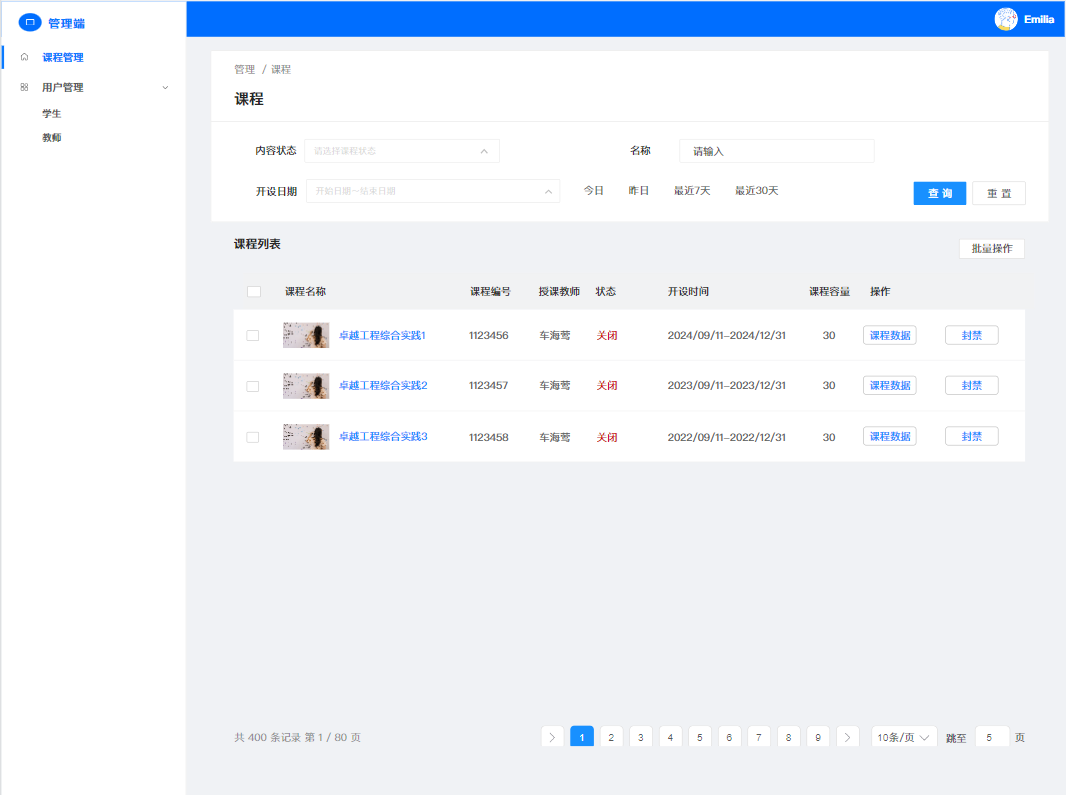
## 6.3管理员端

### 6.3.1课程管理

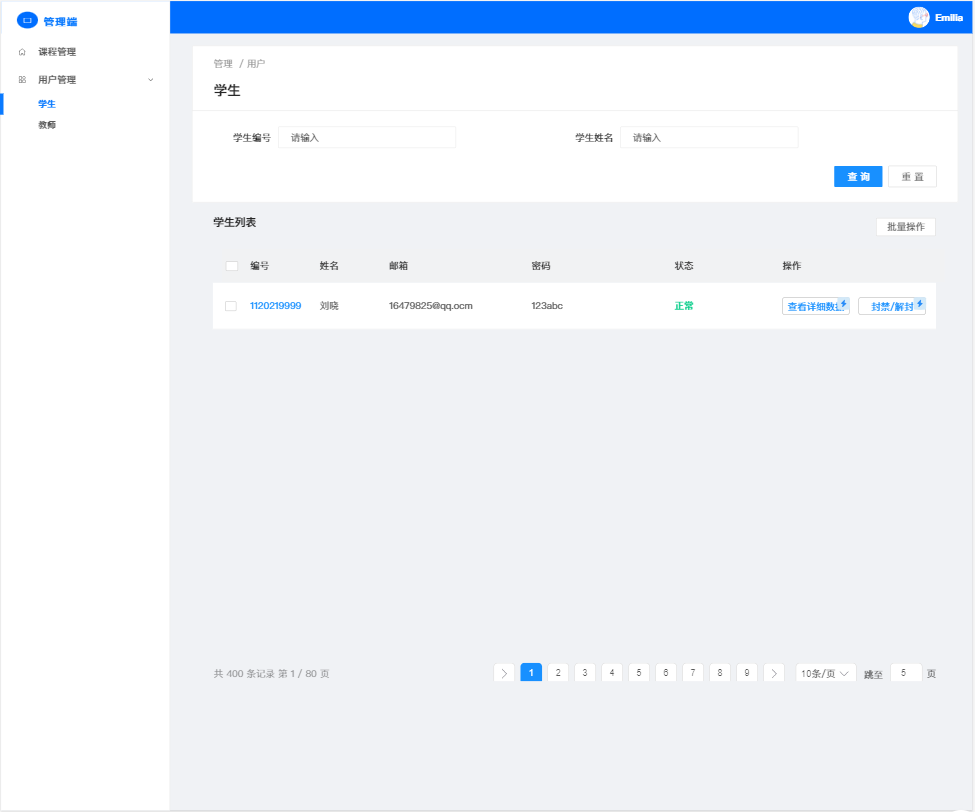


### 6.3.2查看课程数据

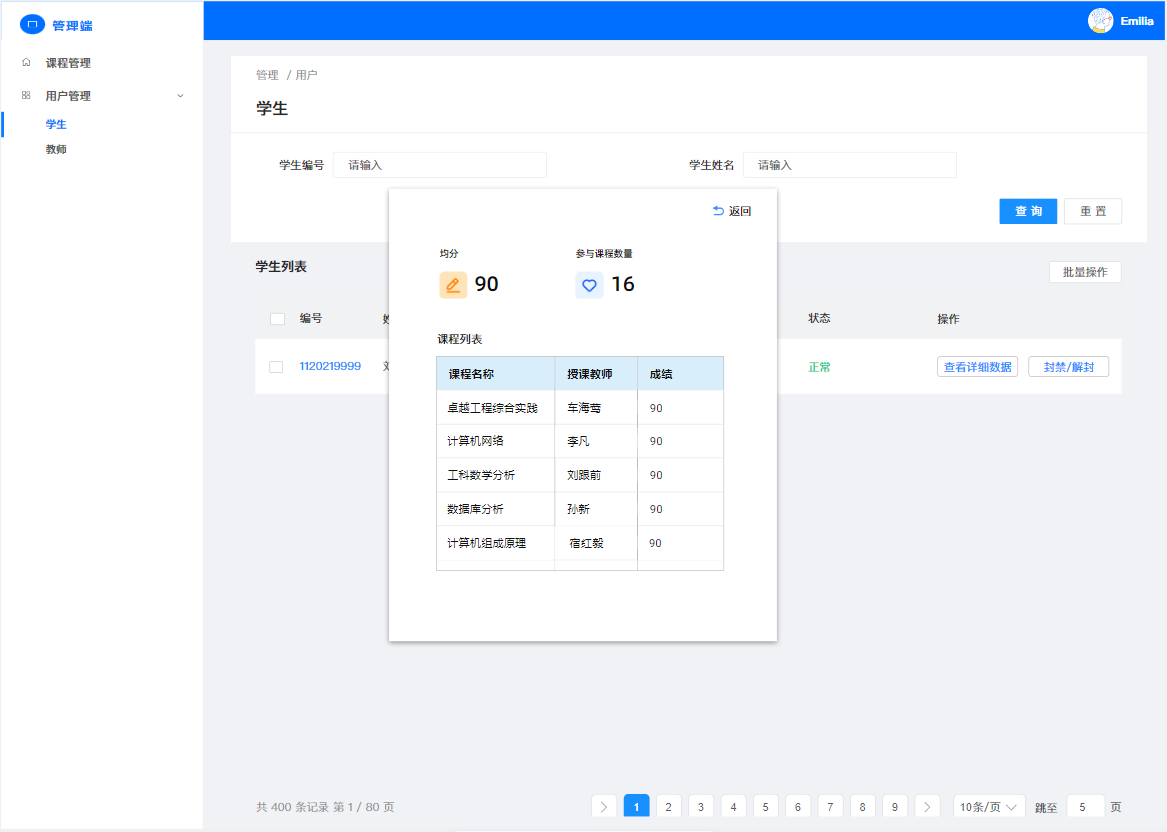
### 6.3.3封禁/解封课程



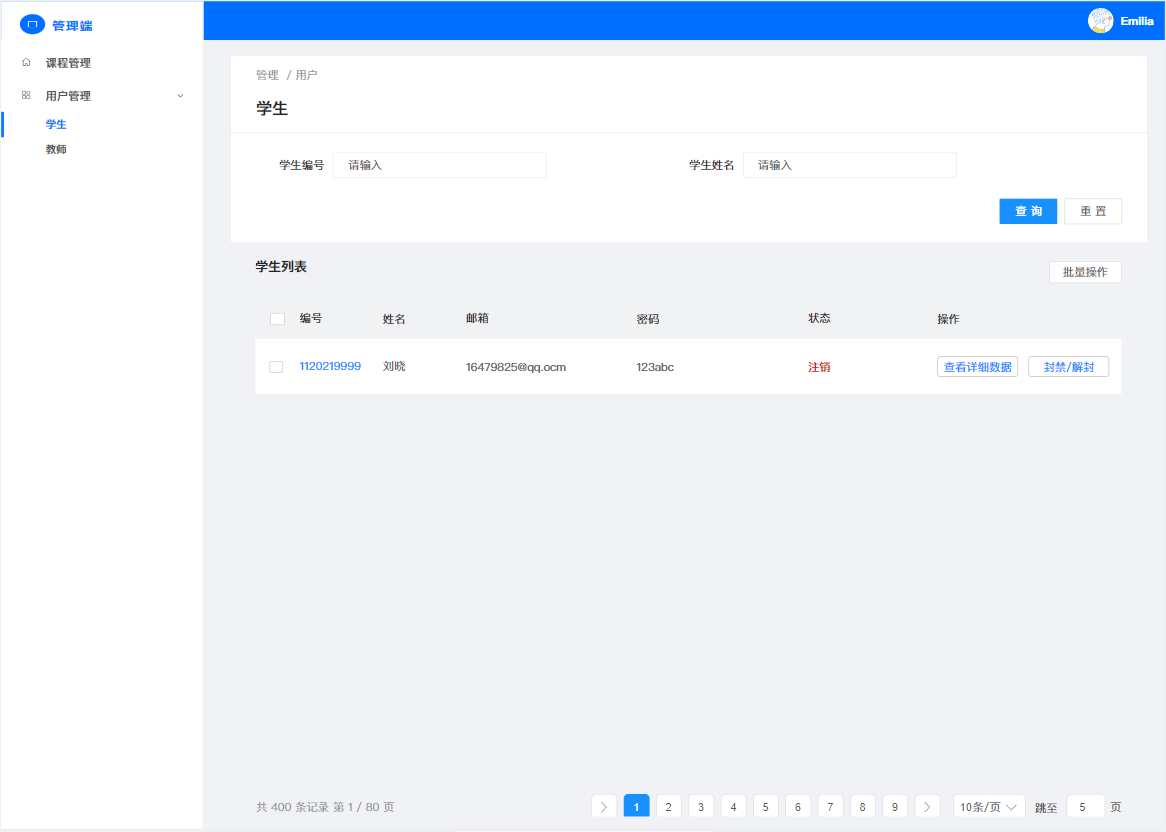
### 6.3.4学生管理



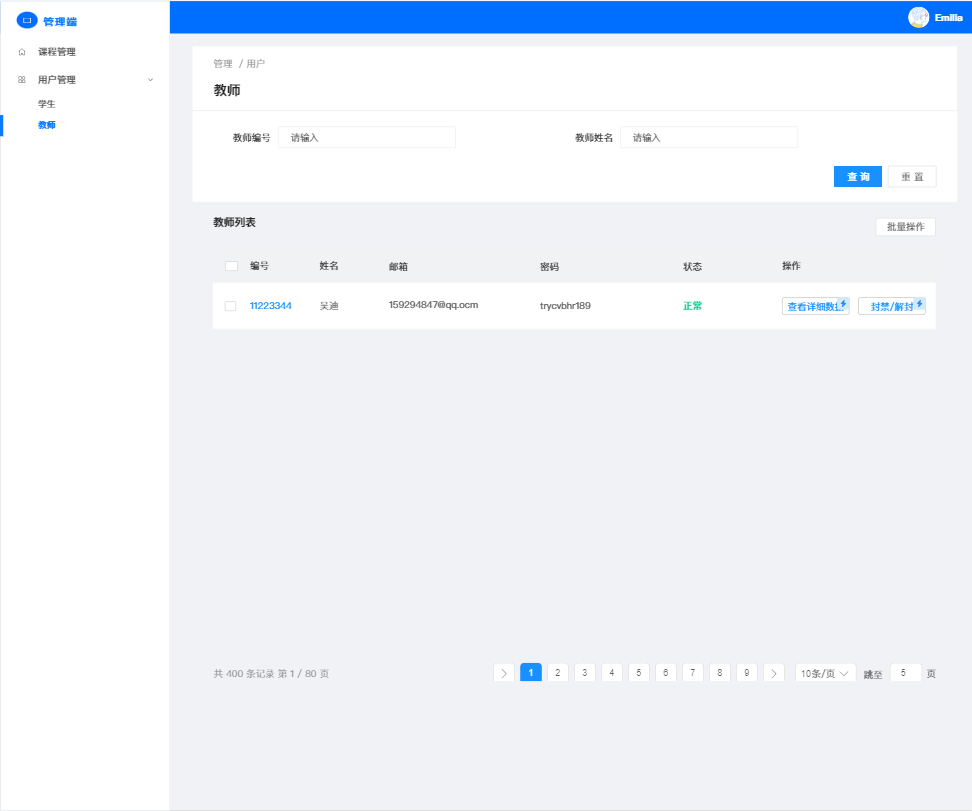
### 6.3.5查看学生信息



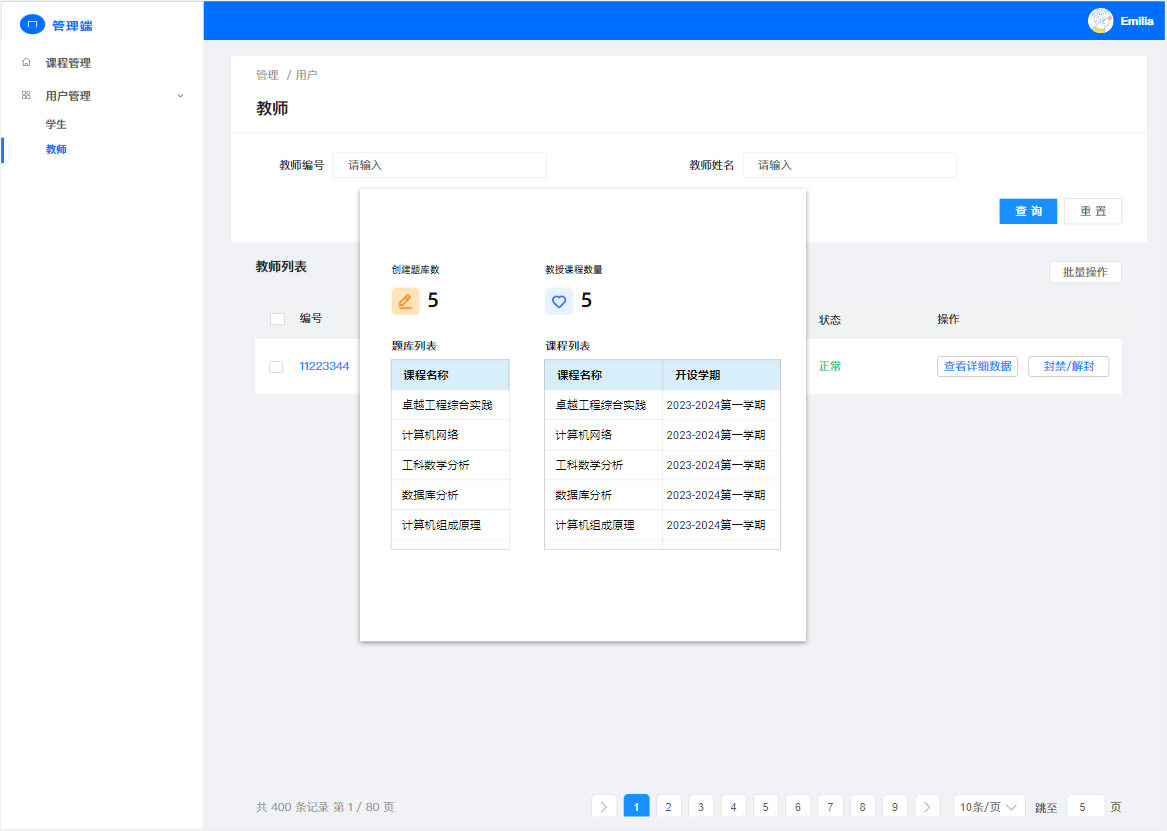
### 6.3.6封禁/解封学生



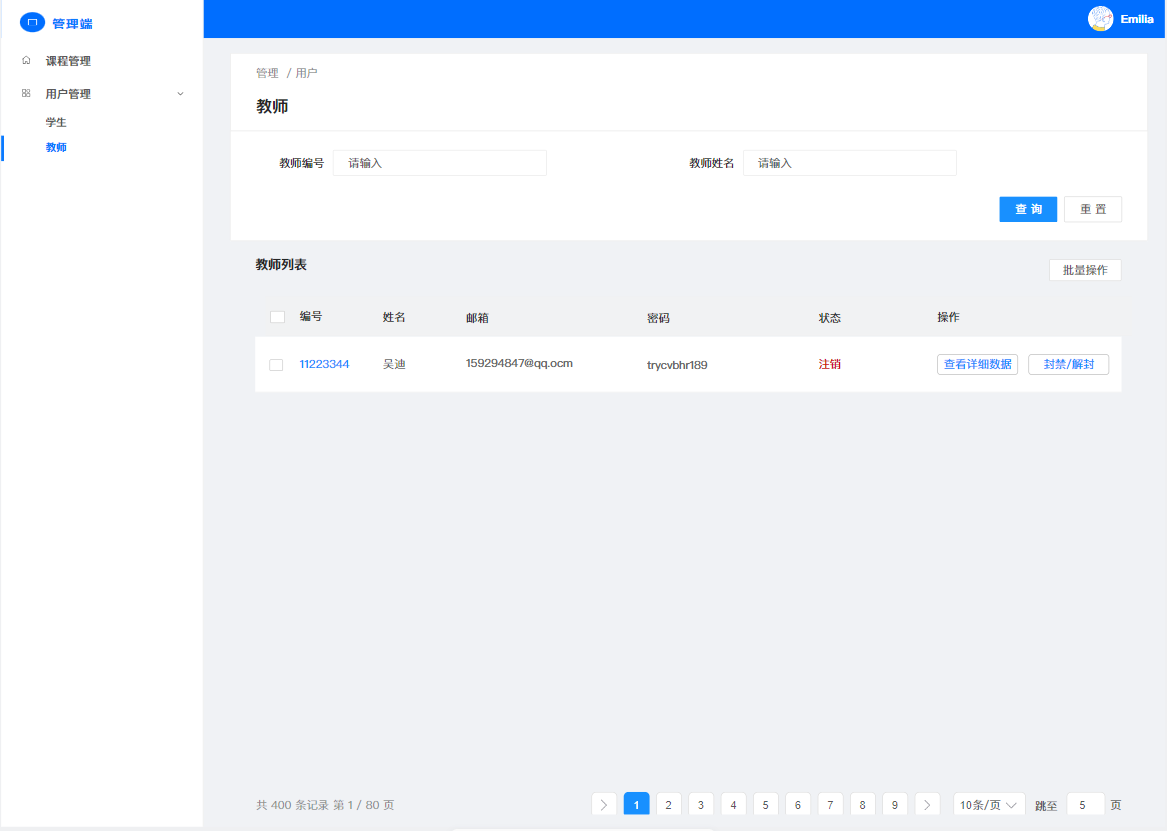
### 6.3.7教师管理



### 6.3.8查看教师信息



### 6.3.9封禁/解封教师



# 7. 开发环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | - 处理器：Intel i5 或同等性能  - 内存：8GB RAM | - 处理器：Intel i3 或同等性能  - 内存：4GB RAM |
| 软件 | - 操作系统：Windows 10/MacOS Mojave  - 开发工具：IntelliJ IDEA, Visual Studio Code  - 版本控制：Git | - 操作系统：Windows 7/MacOS High Sierra  - 开发工具：Eclipse, Visual Studio Code  - 版本控制：SVN |
| 网络通信 | - 带宽：100Mbps  - 延迟：小于100ms | - 带宽：50Mbps  - 延迟：小于200ms |
| 其它 | - 需要互联网连接以访问在线资源 | - 需要稳定的互联网连接 |

# 8. 运行环境的配置

***提示：****说明本系统应当在什么样的环境下运行，有什么强制要求和建议？*

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | - 处理器：Intel i5 或同等性能  - 内存：8GB RAM | - 处理器：Intel i3 或同等性能  - 内存：4GB RAM |
| 软件 | - 操作系统：Windows 10/MacOS Mojave  - 浏览器：最新版本的Chrome, Firefox或Safari | - 操作系统：Windows 7/MacOS High Sierra  - 浏览器：兼容HTML5的浏览器 |
| 网络通信 | - 带宽：100Mbps  - 延迟：小于100ms | - 带宽：50Mbps  - 延迟：小于200ms |
| 其它 | - 需要稳定的互联网连接以确保服务可用性 | - 需要互联网连接 |

# 9. 测试环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件 | - 处理器：Intel i5 或同等性能  - 内存：8GB RAM | - 处理器：Intel i3 或同等性能  - 内存：4GB RAM |
| 软件 | - 操作系统：Windows 10/MacOS Mojave  - 浏览器：最新版本的Chrome, Firefox或Safari | - 操作系统：Windows 7/MacOS High Sierra  - 浏览器：兼容HTML5的浏览器 |
| 网络通信 | - 带宽：100Mbps  - 延迟：小于100ms | - 带宽：50Mbps  - 延迟：小于200ms |
| 其它 | - 需要稳定的互联网连接以确保服务可用性 | - 需要互联网连接 |