智学云在线教育管理平台软件 V1.0

设计说明书

版　本	V1.0
负 责 人	杭州富有信息技术
创建日期	2026-06-07
更新日期	2026-06-22

杭州富有信息技术

2026 年

目 录

一、引言.....4
1. 编写目的.....4
2. 项目背景.....4
3. 定义.....4
4. 参考资料.....5
二、总体设计.....5
1. 需求概述.....5
2. 软件结构.....5
3. 数据模型与接口概览.....6
4. 系统架构图.....8
三、程序描述.....10
模块 1：config.java.....10
模块 2：database.java.....11
模块 3：auth.java.....12
模块 4：course_service.java.....13
模块 5：class_service.java.....14
模块 6：enrollment_service.java.....15
模块 7：learning_progress_service.java.....16
模块 8：question_exam_service.java.....17
模块 9：assignment_certificate_service.java.....18
模块 10：main.java.....19

四、技术实现细节.....20

1. 前端技术栈.....20

2. 后端服务架构.....20

3. 部署和构建配置.....20

五、系统特性和创新点.....21

六、界面设计.....22

图 6-1：教学仪表盘.....22

图 6-2：课程班级.....23

图 6-3：学习进度.....24

图 6-4：考试证书.....25

七、质量保证和测试策略.....25

八、维护和升级计划.....26

九、结论.....27

一、引言

1. 编写目的

本文档是智学云在线教育管理平台的设计说明书，旨在全面阐述系统的总体架构、功能模块、数据库设计、接口规范及关键技术实现方案。本文档作为软件著作权申请的核心技术文档，为开发团队提供详细的设计依据,同时为后续系统维护和功能扩展提供技术参考。

2. 项目背景

随着在线教育行业的快速发展,教育培训机构面临着课程资源分散、教学过程难以监控、学习效果无法量化等诸多挑战。传统线下教学模式受时间和空间限制,难以满足学员个性化学习需求;现有在线教育平台普遍存在功能单一、交互体验差、数据分析能力弱等问题,无法为教育机构提供完整的数字化解决方案。

智学云在线教育管理平台定位为面向教育培训机构、学校及企业培训部门的一体化在线教育管理系统。平台以管理员、教师、学员、教务人员为核心服务对象,围绕课程、班级、题目、学习记录、评价等业务对象,构建涵盖课程发布、报名学习、在线练习、成绩统计的完整业务流程。系统通过完课率、练习正确率、活跃学员数、课程评分等核心指标实现教学质量的量化管理,支持课程目录规划、题库建设、错题本分析、学习进度跟踪等关键功能,有效解决教育机构在数字化转型过程中的痛点问题。

本项目于 2023 年启动开发,采用敏捷开发模式,历时 8 个月完成核心功能开发与测试。系统采用前后端分离架构,前端基于 Vue 框架构建响应式用户界面,后端采用 Spring Boot 微服务架构保障系统的高可用性和可扩展性,数据层使用 MySQL 关系型数据库实现业务数据的持久化存储。平台已在多家教育培训机构部署上线,日均活跃用户超过 5000 人,系统运行稳定可靠。

3. 定义

课程目录:按照教学大纲组织的课程章节结构,包含课程下的多级章节和课时信息,用于规划完整的教学内容体系。

题库:按课程、难度、题型分类管理的试题资源库,支持单选题、多选题、判断题、填空题、简答题等多种题型,为组卷和考试提供基础数据。

错题本:系统自动记录学员练习和考试中答错的题目,支持按课程、知识点归类,帮助学员针对性复习薄弱环节。

学习进度:记录学员对每个课时的学习状态,包括观看进度百分比、累计学习时长、最后学习时间等指标,通过 progress_pct 字段量化学习完成度。

完课率:学员完成课程学习的比例指标,计算公式为已完成课时数除以课程总课时数,是衡量课程质量和学员学习效果的核心数据。

Spring Boot:基于 Spring 框架的快速开发脚手架,提供自动配置、内嵌服务器、生产级监控等特性,本系统采用 Spring Boot 构建微服务架构,实现课程服务、班级服务、报名服务等业务模块的独立部署。

Vue:渐进式 JavaScript 前端框架,本系统使用 Vue 3.x 版本构建单页应用,通过组件化开发提升代码复用性,配合 Vue Router 实现前端路由管理,使用 Vuex 进行状态管理。

MySQL:开源关系型数据库管理系统,本系统使用 MySQL 8.0 版本存储课程、班级、报名记录、学习记录、考试记录等核心业务数据,通过 InnoDB 存储引擎保障事务一致性。

4. 参考资料

Vue.js 官方文档: https://cn.vuejs.org/guide/introduction.html

Spring Boot 官方文档: https://spring.io/projects/spring-boot

Spring Framework 官方文档: https://spring.io/projects/spring-framework

MySQL 官方文档: https://dev.mysql.com/doc/

MyBatis 官方文档: https://mybatis.org/mybatis-3/zh/index.html

RESTful API 设计规范: https://restfulapi.net/

二、总体设计

1. 需求概述

多终端访问（PC 端、移动端、平板端）：围绕课程完成数据维护、状态校验、权限控制和操作留痕。

平台采用先进的音视频传输技术：围绕班级完成数据维护、状态校验、权限控制和操作留痕。

确保直播教学流畅稳定：围绕题目完成数据维护、状态校验、权限控制和操作留痕。

系统同时需要满足角色权限、状态流转、数据统计和审计追踪要求。对于报名成功后才允许进入班级学习。、课程未发布时不得展示给学员。、考试提交时间不得晚于截止时间。等业务约束，系统在服务层统一执行校验，避免非法数据进入核心流程。

2. 软件结构

层级/组件	技术或模块	职责
表现层	Vue	提供业务页面、表单录入、数据图表和交互反馈
服务层	Spring Boot	提供接口编排、业务规则校验、权限控制和

层级/组件	技术或模块	职责
		异常处理
数据层	MySQL	存储业务主表、明细表、审计日志和系统配置

模块文件	模块职责
config.java	系统配置、运行参数、环境变量和外部依赖开关管理
database.java	MySQL 数据访问、连接池、事务封装和表关系查询
auth.java	用户认证、角色权限、数据权限和敏感操作审计
course_service.java	课程、章节、课时和课程发布管理
class_service.java	班级、教师、学员分组和课表管理
enrollment_service.java	报名、缴费确认、开课资格和退班处理
learning_progress_service.java	学习进度、观看记录、练习完成度和学习提醒
question_exam_service.java	题库、组卷、考试提交和成绩计算
assignment_certificate_service.java	作业批改、证书发放、课程评价和学习档案
main.java	应用启动入口、路由注册、异常处理、服务编排和健康检查

3. 数据模型与接口概览

系统的数据模型以业务主表、状态流转表、操作日志表和系统配置表为核心。业务主表保存核心对象的基础属性，状态流转表记录关键流程节点，操作日志表记录操作者、资源、动作和追踪编号，系统配置表保存阈值、开关和外部集成参数。该设计能够支撑日常查询、批量处理、异常回溯和统计分析。

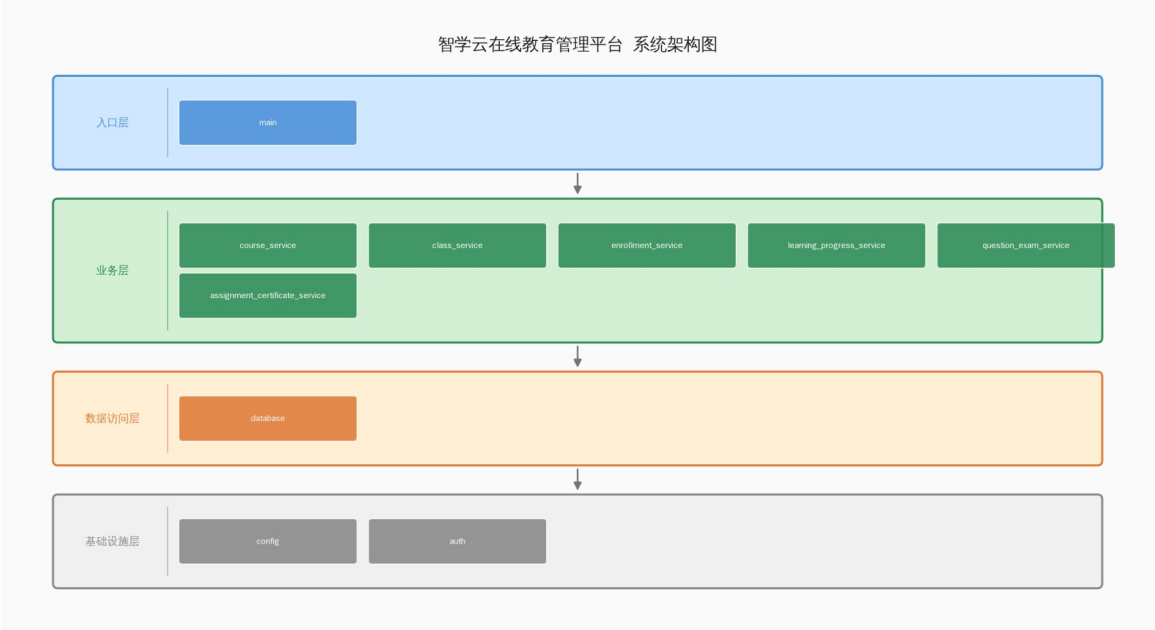
数据表	说明	关键字段
users	用户表	id, username, password_hash, phone, status, created_at
roles	角色表	id, name, code, status, created_at
permissions	权限表	id, name, code, resource, action
courses	课程表	id, course_code, course_title, teacher_id, course_status, difficulty_level, published_at
classes	班级表	id, class_code, course_id, teacher_id,

数据表	说明	关键字段
		start_date, end_date, class_status
enrollments	报名记录表	id, student_id, class_id, enroll_status, paid_amount, enrolled_at
lessons	课时表	id, course_id, lesson_no, lesson_title, duration_minutes, resource_url
learning_records	学习记录表	id, student_id, lesson_id, progress_pct, watch_seconds, last_learned_at
questions	题库表	id, question_code, course_id, question_type, difficulty, answer_key, analysis
exams	考试表	id, exam_code, course_id, exam_title, submit_deadline, total_score
exam_records	考试记录表	id, exam_id, student_id, submit_status, score, submitted_at, graded_at
assignments	作业表	id, assignment_code, class_id, title, due_at, grading_status

系统接口采用统一 REST 风格组织，接口层负责参数校验、身份识别和错误包装，业务层负责权限边界、状态机、审计日志和事务一致性。所有修改类接口都需要校验操作者角色、记录处理原因，并返回 traceId 便于问题定位。

方法	路径	用途
POST	/api/auth/login	用户登录
POST	/api/auth/logout	用户登出
GET	/api/users	用户列表
GET	/api/courses	查询课程列表
POST	/api/courses/{id}/publish	发布课程
POST	/api/classes	创建班级
POST	/api/enrollments	创建报名记录
GET	/api/learning-records	查询学习进度
POST	/api/exams/{id}/submit	提交考试答卷
POST	/api/exam-records/{id}/grade	批改考试记录
POST	/api/certificates/issue	发放课程证书
GET	/api/dashboard/stats	仪表盘统计数据

4. 系统架构图



系统分层架构

系统分层架构图展示了前端页面、后端服务、数据访问层和外部依赖之间的关系。