

# 软件工程案例实践课程管理平台

## 一 概要设计说明书

文档编号: CP-HLD-001

版本号: V1.0

状态: [√] 正式发布

密级: 内部公开

发布日期: 2025-12-23

### 文档变更历史

版本	日期	修改人	修改类型	修改描述
V0.1	2025/12/20	架构组	新建	基于需求分析完成初始架构草案
V0.2	2025/12/21	架构组	修改	细化数据库触发器逻辑与API路由规范
V0.3	2025/12/22	架构组	增加	补充前端界面交互设计与安全审计章节
V1.0	2025/12/23	项目组	发布	评审通过, 发布正式版

## 1. 引言

### 1.1 编写目的

本文档旨在从宏观角度对“软件工程案例实践课程管理平台”进行架构设计,明确系统的技术路线、功能模块划分、数据结构及核心业务流程。本文档是后续详细设计、编码实现及测试验收的直接依据。

### 1.2 项目背景

针对高校软件工程实践课程中存在的“组队管理混乱、项目过程监控难、评价维度单一(缺乏互评)”等痛点,本项目拟构建一个全流程管理系统。该系统需支持从学生自由组队、选题立项、分阶段开发(需求/设计/测试/验收)到成果提交与互评的完整闭环管理。

## 1.3 术语定义 ..

- **Stage (阶段)**: 指软件工程生命周期中的标准环节，如需求分析、概要设计等。
  - **Locked (锁定)**: 指团队在立项后成员不可变更的状态，或阶段未开启的状态。
  - **Archived (归档)**: 指课程结束后，班级进入的只读状态。
- 

## 2. 总体设计

### 2.1 运行环境 ..

系统采用容器化部署，运行于 Linux 服务器环境。

- 服务端 **Runtime**: Bun v1.x (高性能 JavaScript 运行时)
- 数据库: SQLite 3 (文件型数据库，适合教学场景，易备份迁移)
- 前端环境: 现代浏览器 (Chrome/Edge/Firefox/Safari)

### 2.2 系统架构设计 ..

系统采用 **B/S (Browser/Server)** 前后端分离架构。

1. 表现层 (Frontend):

- 采用 **Vue 3 (Composition API) + TypeScript** 构建单页应用 (SPA)。
- 使用 **Pinia** 进行状态管理 (用户会话、班级上下文)。
- UI 组件库选用 **Element Plus**。
- 架构特点: 模块化视图 (**views/student**, **views/teacher**), 通过 API Service 层与后端通信。

2. 服务层 (Backend):

- 基于 **ElysiaJS** 框架提供 RESTful API。
- 中间件层: 负责 JWT 鉴权、RBAC 角色校验、错误处理及 CSRF 防护。
- 业务逻辑层: 封装 Auth, Team, Project, Grading 等 Service, 实现 ABAC 细粒度权限控制。

3. 数据层 (Database):

- 使用 **Drizzle ORM** 管理 Schema。
- 核心机制: 利用 **SQLite Triggers** 将“归档即只读”、“立项即锁定”等核心业务规则下沉至数据库层, 确保数据绝对一致性。

3. 功能模块设计

系统依据用户角色划分为三个子系统, 共计 14 个核心功能模块。

3.1 学生子系统 (Student Subsystem) :

面向选课学生, 提供组队与项目实践环境。

模块名称	功能描述	关键逻辑
我的班级	查看已加入班级, 切换当前上下文	本地持久化 activeClassId
团队大厅	创建团队、申请入队、处理申请、转让队长	立项后团队自动锁定 (Locked)
项目工作台	提交立项草稿、查看阶段任务、提交作业	支持多版本提交 (V1, V2...)
互评中心	在规定窗口内对其他组/同组进行双盲评分	仅在 Window 状态为 Open 时可用
个人中心	修改密码、查看个人档案与成绩单	成绩包含教师评分与互评加权分

### 3.2 教师子系统 (Teacher Subsystem) ..

面向授课教师，提供教务管理与过程监控能力。

模块名称	功能描述	关键逻辑
教务管理	创建班级、导入/管理学生名单、班级归档	归档后触发全量写保护 (Archived)
审批中心	审核项目立项、处理特殊变更申请	立项通过触发 DB 触发器初始化 5 个阶段
进度看板	全班项目阶段状态可视化监控 (Passed/Open/Locked)	支持点击单元格快速批改作业
互评管理	开启/封存/发布互评窗口，裁决异常评分	封存时计算 Z-Score 系数
成绩管理	录入作业分数，导出总成绩单	支持自定义评分细则 (Rubric)

### 3.3 管理员子系统 (Admin Subsystem) ..

面向系统运维人员。

模块名称	功能描述
用户管理	教师账号的增删改查、密码重置
系统审计	查看不可篡改的审计日志 (audit_logs)，监控系统错误

## 4. 数据库设计

### 4.1 实体关系图 (ERD) 概念描述 ..

- Users (1) ----< (N) Profiles (分表存储学生/教师详情)
- Classes (1) ----< (N) Teams
- Classes (1) ----< (N) ClassStudents (选课关系)
- Teams (1) ---- (1) Projects (严格的一对一关系)
- Projects (1) ----< (5) ProjectStages (预设 5 个标准阶段)
- Assignments (1) ----< (N) Submissions
- Submissions (1) ----< (N) SubmissionFiles

## 4.2 关键数据表结构

### 1. `classes` (班级表)

- `status`: 枚举值 `active` / `archived`。一旦置为 `archived`，数据库触发器拦截所有关联表的写操作。
- `config_json`: 存储动态规则（如团队人数上下限）。

### 2. `teams` (团队表)

- `status`: 枚举值 `recruiting` / `locked`。
- `is_locked`: 布尔值。由触发器 `trg_projects_active_lock_team` 自动维护，立项通过即置为 1。

### 3. `submissions` (提交表)

- `version`: 整数。同一作业、同一团队下的提交版本号自动递增。
- `file_id`: 关联 `files` 表，物理文件名为 UUID。

### 4. `audit_logs` (审计日志表)

- 记录 `actor_id`, `action`, `target_id`, `before_json`, `after_json`。
- 安全约束：数据库层设置 `BEFORE UPDATE/DELETE` 触发器，抛出异常，确保日志不可篡改。

## 5. 接口与交互设计

### 5.1 接口设计规范

系统对外提供 RESTful API，统一前缀 `/api/v1`。

- 认证：Header 携带 `Authorization: Bearer <JWT>`。
- 错误码：统一 JSON 格式，如 `{ "error": { "code": "CLASS_ARCHIVED_READONLY", ... } }`。

## 5.2 核心界面交互设计

### 5.2.1 学生端 - 项目工作台 (Project Workspace)

参考《概要设计案例.pdf》图 6-21 风格。

- 布局：顶部为“阶段导航条” (Steps)，显示需求、设计、测试等阶段状态。
- 左侧：阶段列表菜单，使用图标区分 **Locked** (锁)、**Open** (进行中)、**Passed** (已通过)。
- 主内容区：
  - 草稿态：显示项目申报表单，底部有“保存”和“提交审核”按钮。
  - 运行态：显示当前阶段的任务描述、截止时间及提交历史列表。提交列表展示版本号 (V1, V2)、提交人、时间及下载链接。
  - 交互：点击“上传”按钮，弹窗选择文件。上传成功后，列表自动刷新，版本号 +1。





### 5.2.2 学生端 - 团队管理 (Team Management)

参考《概要设计案例.pdf》图 6-24 风格。

- 布局：采用 Tab 标签页切换结构。
  - **Tab 1: 基本信息**：队名、口号、Logo。队长可编辑。
  - **Tab 2: 成员管理**：列表展示成员。队长可移除成员（仅限 **Unlocked** 状态）。若团队已锁定，移除按钮置灰并提示“项目已立项，团队锁定”。
  - **Tab 3: 加入申请**：处理其他学生的入队请求（同意/拒绝）。

### 5.2.3 教师端 - 进度监控看板 (Progress Dashboard)

参考《概要设计案例.pdf》图 6-23 风格。

- 布局：二维矩阵表格。
  - 行：各学生团队。
  - 列：5 个标准开发阶段。
- 单元格状态：
  -  绿色圆点：阶段已通过 (Passed)。
  -  蓝色圆点：阶段进行中 (Open)，待评审。
  -  灰色锁头：阶段未解锁 (Locked)。
  -  红色感叹号：提交逾期或被驳回。
- 交互：点击蓝色圆点，右侧滑出抽屉，展示该团队该阶段的最新提交物及 **JsonDiff** 版本对比视图，教师可直接打分或驳回。

## 6. 安全与可靠性设计

### 6.1 归档保护 (Archiving Protection) ::

系统实施物理级的数据冻结策略。当教师在前端将班级状态设为“归档”时，后端不仅更新状态位，数据库层面的触发器 (`trg_deny_write_when_class_archived`) 立即生效，拦截任何针对该班级数据的 INSERT/UPDATE/DELETE 操作，防止API被恶意调用篡改历史数据。

### 6.2 密码安全 ::

所有用户密码不以明文存储，采用 **Argon2id** 算法进行哈希处理，并强制实施随机盐策略，有效抵御彩虹表攻击。

### 6.3 细粒度鉴权 (ABAC) ::

除基础的 RBAC 角色控制外，系统在 Service 层实施基于属性的访问控制 (ABAC)。例如：

- 学生只能修改自己团队的草稿项目。
- 学生只能在互评窗口状态为 `Open` 时提交互评数据。
- 所有涉及资源 ID 的操作均校验其 `class_id` 归属，防止越权访问其他班级数据。