

乒乓球培训管理系统开发计划

项目概述

本项目是为乒乓球培训机构开发的一套信息化管理解决方案，采用前后端分离架构，支持学员、教练、校区管理员和超级管理员四种角色。经过两次迭代开发，已完成基础架构搭建和核心功能实现。

迭代发展历程

第一次迭代（2025年9月初 - 2025年9月7日）

目标：建立项目基础架构，实现基本的前端页面和后端API

主要成果：

1. 技术架构搭建
 - 前端：Vue 3 + Vite项目初始化，集成Element Plus、Vue Router、Pinia
 - 后端：Kotlin + Ktor项目初始化，集成Exposed ORM框架，配置MySQL数据库
2. 核心功能实现
 - 用户注册登录系统
 - 多角色权限管理框架
 - 基础课表页面
 - 数据库基础表结构设计
3. 项目成果
 - 前端代码约28,000行，Vue组件50+个
 - 后端代码约2,000行，API端点10+个
 - Git提交85+次

第二次迭代（2025年9月8日 - 2025年9月13日）

目标：完善业务功能，建立设计系统，实现前后端深度集成

主要成果：

1. 设计系统建立
 - Glassmorphism设计语言
 - 15+种可复用组件（GlassCard、ModernButton等）
 - 42个CSS变量统一管理主题色彩
2. 核心业务功能完善
 - 消息系统（完整CRUD操作）
 - 充值系统（钱包余额查询和充值功能）
 - 校区管理系统
3. 前后端深度集成
 - 用户认证系统完全打通
 - 权限管理系统完善
 - 消息系统前后端对接
4. 项目成果

- 前端新增代码约8,000行
- 后端新增代码约3,000行
- 总Git提交达到93次

后续功能增强（2025年9月15日 - 2025年9月26日）

主要成果：

- 1. 比赛管理系统
 - 比赛创建与报名
 - 赛程安排与查看
- 2. 预约管理系统
 - 学生预约训练
 - 教练审核预约
- 3. 管理端功能完善
 - 超级管理员和校区管理员页面补全
 - 用户管理、校区管理、系统日志等功能
- 4. 系统优化
 - 表结构和底层数据重构
 - 权限缺陷修复
 - API充实

当前功能状态

通用功能

- 登录/注册系统（Session持久化）
- 基于角色的权限管理系统
- 消息中心（消息列表、未读计数、标记已读、删除）
- 日志与错误处理机制

角色功能

- 1. 学员端
 - 教练查找与详情查看
 - 训练预约与课程表管理
 - 钱包充值与余额查询
 - 比赛报名与赛程查看
- 2. 教练端
 - 预约审批管理
 - 学员反馈查看
 - 双选流程支持
- 3. 管理员端
 - 校区管理（创建/列表）
 - 用户管理

- 系统日志查看
- 球台资源管理

技术架构

前端技术栈

- Vue 3.5.18 + Vite 7.0.6
- Vue Router 4.5.1 + Pinia 3.0.3
- Element Plus 2.11.1
- 自研Glassmorphism设计系统

后端技术栈

- Kotlin 2.1.10 + Ktor 3.2.3
- Exposed 0.56.0 ORM框架
- MySQL 8.0 + HikariCP 5.1.0
- BCrypt密码加密

团队贡献

- 主要开发者：155次提交
- 协作开发者1：44次提交
- 协作开发者2：24次提交

当前存在的问题与风险

1. 自动化测试覆盖不足
2. 并发与资源冲突风险（预约/球台/赛程的时间段冲突）
3. 性能与容量评估欠缺
4. 移动端深度体验有待提升
5. 运维与可观测性基础薄弱

后续开发路线图

短期目标（可靠性与质量）

1. 建立核心功能回归测试用例
 - 登录/注册流程
 - 预约创建与审核
 - 比赛创建/报名/编排
 - 钱包充值流程
2. 引入自动化测试
 - 后端：JUnit + Testcontainers集成测试
 - 前端：关键组件单元测试（3-5个）
3. CI/CD流程建设
 - GitHub Actions接入
 - 代码检查、单元测试、构建自动化

中期目标（业务完善与治理）

- 1. 预约与赛事系统优化
 - 时间冲突校验与并发控制
 - 唯一键与业务锁机制
- 2. 支付系统完善
 - 分账户流水管理
 - 限额与风控规则
 - 实际支付网关对接预研
- 3. 消息系统增强
 - 批量操作优化
 - 分页查询服务端实现
 - 推送与未读策略优化
- 4. 比赛编排系统
 - 支持分组/淘汰赛制
 - 赛程导出与结果复核

长期目标（可观测性与运维）

- 1. 健康检查与监控
 - /health健康检查端点
 - /metrics指标端点（Micrometer/Prometheus）
 - 日志结构化处理
- 2. 审计与安全
 - 关键操作日志记录
 - 管理员行为留痕
- 3. 部署优化
 - Docker Compose环境参数化（dev/test/prod）
 - 线上部署文档完善

用户体验优化

- 1. 设计系统持续完善
 - 空状态、加载、错误状态规范
 - 可访问性支持
- 2. 移动端体验提升
 - 触控优化
 - 离线功能支持
 - 推送通知机制
- 3. 错误处理统一化
 - 空/错文案统一
 - 异常码与提示映射完善

总结

该项目已经完成了基础架构搭建和核心功能实现，建立了完整的前后端分离架构和现代化的设计系统。后续开发应重点关注质量保障、业务完善和用户体验提升，逐步构建一个稳定、可靠、易用的乒乓球培训管理系统。