

同濟大學

TONGJI UNIVERSITY

软件工程课程设计

| | |
|---------|-----------------|
| 课 题 名 称 | 面向高校体育教学的智慧课堂平台 |
| 学 院 | 计算机科学与技术学院 |
| 专 业 | 软件工程 |
| 项 目 成 员 | 2354100 郝哲逸 |
| | 2354090 王家睿 |
| | 2457054 齐震罡 |
| | 2350223 戴昊晟 |
| 指 导 教 师 | 黄杰 |
| 日 期 | 10 月 7 日 |

1 引言

1.1 项目介绍

本项目旨在规划与开发一个面向高校体育教学的“体育健身智慧课堂”平台。该平台构建了一个集在线课程学习、学生训练过程管理与教学数据分析于一体的数字化教学环境，以支持教师教学与学生自主学习的现代化改革。平台采用 B/S 架构，基于云计算技术，为高校体育教学提供全方位的智能化解决方案。

本项目通过引入计算机视觉技术和人工智能算法，实现对体育训练过程的智能化监测与指导，打破传统体育教学中时间与空间的限制。平台不仅提供基础的教学视频点播功能，更创新性地集成了动作识别、训练评估和个性化反馈等智能功能。

1.2 项目动机

当前高校体育教学普遍面临诸多挑战。首先，师生课后互动严重不足，学生在课堂外难以获得专业的训练指导。其次，学生课外训练缺乏有效监督与科学评估机制，训练效果难以保证。第三，教学过程数据采集困难，缺乏量化评估依据，教学决策往往基于主观经验。

本项目的动机在于利用现代信息技术手段，打通课堂内外的教学环节。通过构建数字化的体育教学平台，实现教学过程的全程可追溯、训练效果的可量化评估。平台的建设将有力推动体育教学的标准化、个性化与数据化发展，为高校体育教学改革提供技术支撑和实践案例。

特别是在后疫情时代，线上线下混合式教学已成为教育发展的重要趋势。体育教学作为实践性极强的学科，亟需探索适合自身特点的

数字化教学模式。本项目正是基于这一背景提出的创新性解决方案。

1.3 项目创新点

本平台的创新性并在于对成熟技术进行教育场景的深度整合与应用模式创新。具体体现在三个层面：

在流程数字化方面，平台将传统的“教师课堂示范-学生练习”单向教学模式，延伸至线上的“课程学习-自主训练-智能反馈-数据评估”的完整闭环。学生可以通过平台获得持续性的训练指导，教师可以基于数据驱动进行教学决策。

在管理集中化方面，为教师提供一个统一的数字化管理视图，实现课程内容管理、学生训练进度监控和教学效果评估的一体化操作。通过数据可视化技术，将复杂的教学数据转化为直观的图表展示。

在技术架构方面，创新性地采用 C# 与 Python 混合的技术栈方案。C# 负责核心业务逻辑和系统稳定性，Python 专注于 AI 算法服务，充分发挥各自的语言优势，实现系统性能与智能功能的平衡。

1.4 技术挑战与应对

项目面临的主要技术挑战体现在三个方面。首先是异构技术模块的集成挑战，平台需要整合 Web 管理系统、C# 后端服务、Python AI 服务等多个技术组件，确保系统整体的稳定性和性能。其次是 AI 模型在真实场景下的精度与性能优化问题，特别是在复杂环境下对人体动作的准确识别。第三是系统部署与运维的挑战，需要保证多服务协同工作的高可用性。

异构技术模块集成方面，通过定义标准化的 RESTful API 接口实

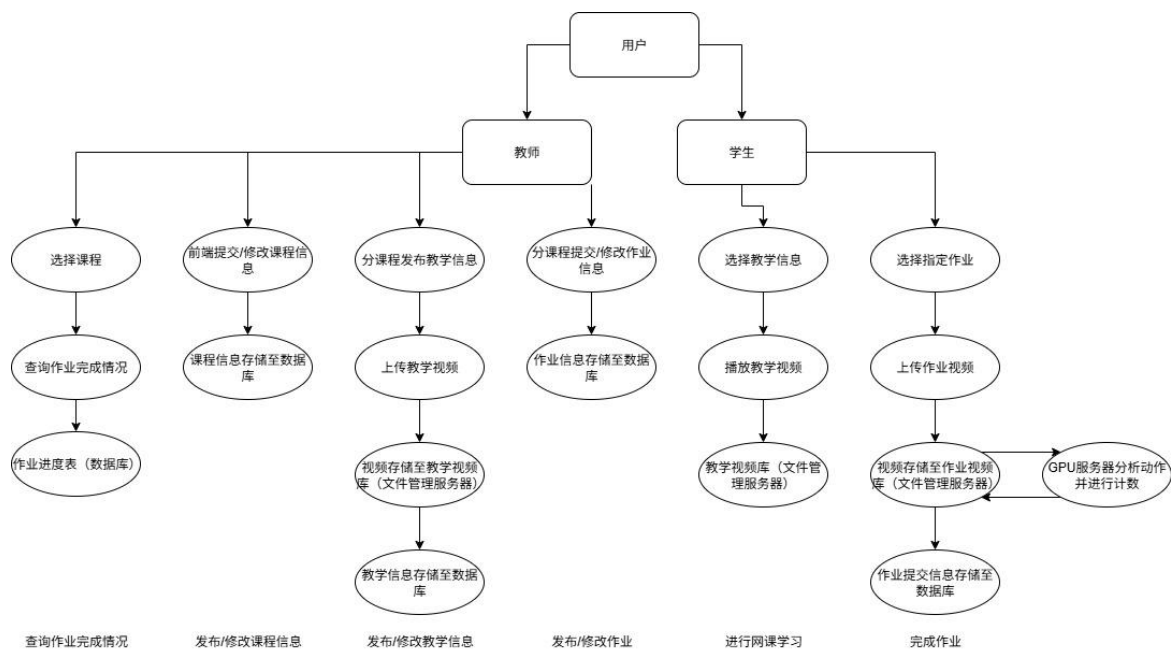
现 C#与 Python 服务的高效通信。采用 Docker 容器化部署，统一运行环境，降低集成复杂性。

AI 模型精度与性能方面，初期聚焦基础动作（如深蹲、引体向上）的识别，使用公开运动数据集预训练 YOLOv8 模型，逐步引入真实场景数据优化模型鲁棒性。

系统部署与运维方面，使用 Docker 管理多服务部署，配合健康检查和自动重启机制确保高可用性。借助学校提供的测试服务器，定期监控服务性能，优化资源分配。

2 客户需求与系统目标

下图是主要的客户与功能需求一览表：



2.1 主要客户群体

学生用户作为平台的核心服务对象，是系统设计的主要关注点。他们通过平台完成在线学习、自主训练、作业提交和进度查询等核心

活动。学生群体的需求特征表现为：期望获得界面友好、操作便捷的使用体验，能够随时随地访问课程内容，快速提交训练作业，并清晰查看个人的训练记录与智能反馈报告。

体育教师作为平台的关键使用者，承担着教学内容发布、学习过程监督和训练效果评估的重要职责。教师用户需要高效管理多个平行教学班级，便捷地发布与更新多样化教学资源，并直观掌握班级整体训练数据与个体学生进度差异。

次要利益相关者：

学校体育部门：需要平台提供教学效果的量化数据（如作业提交率、动作标准度统计），以支持教学改革评估和课程优化。

学校 IT 部门：负责平台与学校现有系统的对接（如用户认证系统），要求平台符合学校的数据安全规范和 API 标准，确保数据隐私和系统兼容性。

2.2 功能性需求

2.2.1 学生端核心功能需求

用户账户管理模块需要提供完整的注册、登录、密码修改和个人信息维护功能。学生通过身份验证后进入个人学习空间，系统需根据用户角色显示相应的功能入口和内容权限。

课程学习模块应支持学生通过课程码加入指定课程，观看教学视频和图文内容。系统需要记录学习进度，提供课程目录导航和学习记录查询功能。视频播放需支持多种清晰度选择和进度控制。

训练作业模块实现训练视频的上传、格式转换和智能分析功能。

学生能够查看作业要求、提交训练视频并获取动作识别结果和纠正建议。系统需提供作业提交状态跟踪和历史记录查询。

智能交互模块集成 AI 对话功能，学生可以基于个人训练情况提出问题，获得个性化的训练建议和问题解答。

2.2.2 教师端核心功能需求

课程管理模块提供课程的创建、编辑、发布和归档功能。教师可以设置课程基本信息、教学进度和参与学生范围。系统支持课程资源的批量上传和分类管理。

学生管理模块实现学生信息的导入导出、分班管理和学习进度监控。教师可以查看学生的课程参与情况、作业完成质量和训练效果趋势。

作业管理模块支持作业的发布、批阅和统计分析。教师可以设置作业要求、提交时限和评分标准，系统自动汇总提交情况并生成统计报表。

教学分析模块提供基于数据的学情分析功能，包括班级整体表现、个体进步趋势和常见问题识别，为教学改进提供依据。

2.3 用户故事定义

2.3.1 学生课程学习与作业提交

作为一名学生，我想要在线观看教学视频并提交我的训练视频，以便教师可以查看我的课后练习情况并获得专业的动作指导。

验收测试标准：给定学生已成功登录平台并选择目标课程，当他在课程页面完成指定视频的观看后，系统应准确记录学习进度；当学

生上传训练视频文件后，系统应在合理时间内完成处理并显示“提交成功”的确认信息。

2.3.2 教师班级管理 with 学情监控

作为一名教师，我想要查看我所负责班级所有学生提交的训练视频列表和基本信息，以便我进行作业批阅和学情统计。

验收测试标准：给定教师使用有效账户登录系统后台，当他进入“班级管理”页面选择目标班级时，系统应准确列出所有在册学生及其视频提交状态统计；当教师点击具体学生记录时，系统应展示该学生的详细训练数据和视频分析结果。

2.3.3 智能动作识别与反馈

作为一名学生，我希望系统能够自动分析我提交的训练视频，识别动作标准程度并提供改进建议，以便我能够及时纠正错误动作。

验收测试标准：给定学生已提交训练视频文件，当系统完成视频处理和分析后，应在学生端清晰展示动作识别结果，包括动作计数、标准度评分和具体的改进建议。

2.3.4 教学数据可视化

作为一名教师，我希望系统能够以图表形式展示班级学生的训练数据统计，以便我快速把握整体学情和个体差异。

验收测试标准：给定教师进入数据分析模块，当选择统计维度和时间范围后，系统应准确生成相应的数据图表，包括但不限于作业提交率分布图、动作标准度趋势图和个体进步对比图。

3 项目目标

3.1 核心问题定位

本项目主要致力于解决高校体育教学中的一个关键痛点：课后训练缺乏有效监督与及时指导。在传统体育教学模式下，学生在课堂外进行自主训练时，往往处于“盲目练习”状态——他们无法准确判断自己的动作是否标准，难以及时获得专业指导，教师也无法全面了解每位学生的课后训练情况。这种训练过程的“黑箱”状态，直接影响了学生的训练效果和技能提升速度。

具体来说，我们观察到三个突出的问题表现：首先，学生的错误动作得不到及时纠正，容易形成错误的肌肉记忆；其次，教师难以掌握学生的真实训练投入和进步情况；最后，教学反馈周期过长，往往要等到下次课才能发现问题，错过了最佳纠正时机。

3.2 用户价值收益

在不涉及具体技术实现的前提下，本系统将为不同用户群体提供以下核心价值：

对于学生用户，系统将提供个性化的训练陪伴与指导。学生能够获得对自己训练动作的即时反馈，清楚了解哪些动作做对了、哪些需要改进，以及如何改进。系统还会记录他们的训练历程，让他们能够清晰地看到自己的进步轨迹，获得持续训练的动力。

对于教师用户，系统提供可视化的教学管理支持。教师可以超越课堂时空限制，全面了解学生的训练情况和进步趋势。系统会自动识别共性问题和个别差异，帮助教师更有针对性地安排教学内容，让教

学决策建立在客观数据基础上。

3.3 体验支撑

这些用户收益将直接支撑客户所期望的“智能化、个性化、数据化”的整体体验：

当学生完成训练视频上传后，系统提供的动作分析和改进建议，让学生感受到“专属教练”般的个性化指导，大大提升了训练的有效性和成就感。

教师通过系统提供的图表和数据看板，能够直观地把控班级整体学情，这种“一切尽在掌握”的体验减轻了教学管理的盲目性，增强了教学决策的信心。

3.4 构想验证

我们已经向多个相关方验证了本项目构想的合理性。首先，我们访谈了 3 位高校体育教师，他们普遍认同课后训练指导不足的问题，并对平台的监督和反馈功能表示期待。特别是年轻教师，对利用技术手段提升教学效率表现出浓厚兴趣。

其次，我们与 20 余名大学生进行了交流，其中体育选修课的学生对动作纠错功能最为关注，他们表示需要这样的工具来帮助自己更快掌握运动技能。普通学生则更看重系统的便捷性和实用性。

我们还咨询了学校体育部的教学管理人员，他们从教学改革角度肯定了项目的方向，认为这与学校推动体育教学数字化的趋势相符。

3.5 收益达成判断

我们将通过多种方式判断客户是否获得了期望的收益：

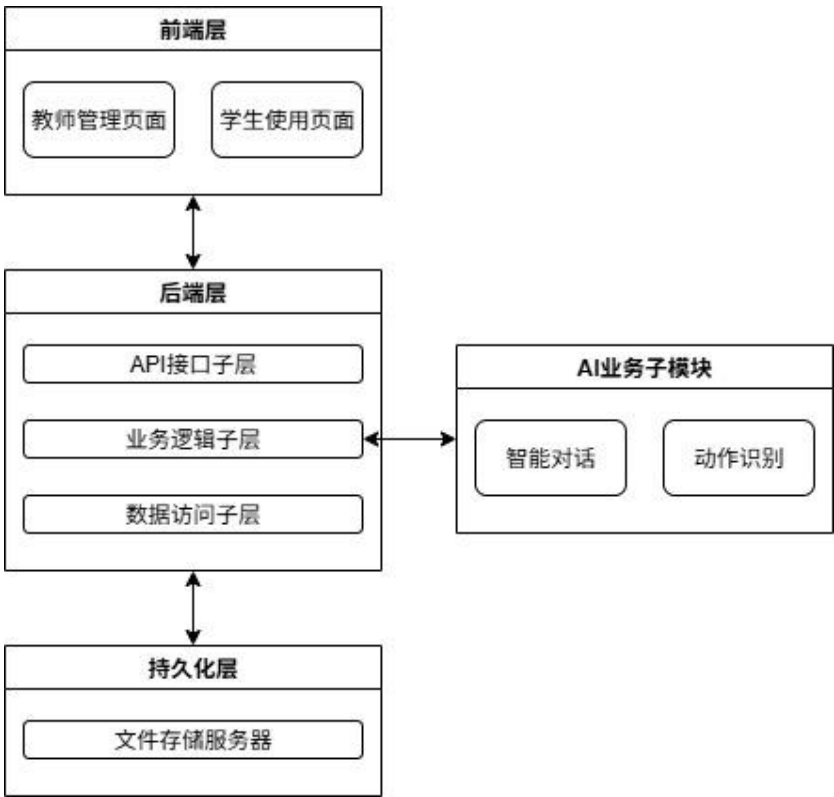
通过系统后台数据，我们可以客观地看到学生的训练频率、动作改进情况、成绩提升趋势等硬性指标。比如，学生上传训练视频的积极性、动作标准度的提升幅度等。

我们将定期收集用户的定性反馈，包括通过问卷了解学生对系统指导效果的满意度，通过访谈了解教师使用系统后教学效率的变化。

我们将观察用户的实际使用行为，比如学生是否根据系统建议调整训练方法，教师是否依据系统数据优化教学安排，这些行为改变最能体现系统的实际价值。

4 系统描述

4.1 系统整体架构图



4.2 系统架构介绍

基于上述架构图，我们的智慧课堂平台采用分层设计理念，确保系统的可维护性和扩展性。整个系统以前端 Web 界面作为用户交互入口，通过 C#构建的核心业务服务层处理主要业务逻辑，Python 专门负责 AI 相关服务，数据库层负责数据的持久化存储。

当学生在网页端提交训练视频时，请求首先到达 C#主业务服务层，进行身份验证和数据校验后，将视频文件存储到文件服务系统。同时，系统会异步调用 Python AI 服务中的动作识别模块，对视频进行分析处理。分析完成后，结果返回到 C#服务层并存入数据库，最终在前端界面展示给用户。

C#主服务作为系统的调度中心，负责协调各个模块的工作。当需要 AI 分析时，通过内部 API 调用 Python 服务；当需要数据持久化时，通过数据访问层操作 MySQL 数据库。这种设计确保了系统的松耦合性，各个模块可以独立开发和部署。

4.3 系统主要组成部分

4.3.1 前端展示层

学生功能模块包含课程学习界面、作业提交页面、个人训练记录查看等功能。学生可以在这里观看教学视频，上传自己的训练视频，查看 AI 分析结果和进步情况。

教师管理模块提供课程管理、学生管理、作业管理和学情分析等功能。教师可以创建和维护课程内容，发布作业任务，查看班级整体情况和个别学生进度。

通用功能模块包括用户登录注册、个人信息管理、消息通知等基础功能，确保用户能够顺畅使用系统。

4.3.2 业务逻辑层（C#服务）

API 接口层提供 RESTful 风格的 API 接口，负责接收前端请求并进行参数验证、权限检查等预处理工作。

核心业务模块包括：

1. 用户管理：处理用户注册、登录、权限控制
2. 课程管理：课程创建、维护、学生选课管理
3. 作业管理：作业发布、提交、批改流程控制
4. 数据统计：学习进度、训练效果等数据分析

数据访问层封装所有数据库操作，提供统一的数据访问接口，确保数据安全性和访问效率。

4.3.3 AI 服务层（Python）

动作识别服务基于 YOLO 模型开发，提供视频分析、动作标准度评估、错误动作识别等功能。该服务会分析学生上传的训练视频，生成详细的动作评估报告。

智能对话服务集成大语言模型，为学生提供训练建议、动作要领解答、个性化训练计划等智能问答服务。

5 解决方案与技术路径

5.1 混合技术栈战略

本项目采用 C#与 Python 相结合的混合技术栈方案，充分发挥两

种语言在不同场景下的技术优势。C#作为主业务逻辑的处理平台，负责构建稳定可靠的业务系统架构，提供用户管理、课程管理、作业管理等核心功能。Python 作为 AI 服务的执行环境，专注于动作识别算法和自然语言处理任务的实现。

两种技术栈通过定义清晰的 RESTful API 接口进行协同工作，确保系统各模块之间的松耦合和高内聚。这种架构设计既保证了业务系统的稳定性和性能，又为 AI 算法的迭代优化提供了充分的灵活性。接口设计采用标准化协议，支持未来的功能扩展和技术演进

5.2 前端技术实现

前端开发采用 Vue 3 组合式 API 配合 TypeScript，提供良好的类型安全和开发体验。组件化开发模式确保 UI 的一致性和可复用性，基于角色的动态路由方案实现精细化的访问控制。状态管理通过 Pinia 实现，支持复杂应用状态的可预测管理。

5.3 后端服务实现

基于 .NET 平台构建的后端主服务采用整洁架构理念，通过清晰的依赖关系确保核心业务逻辑的独立性和可测试性。领域驱动设计方法指导业务模型的精确定义和演化，领域事件机制实现业务组件之间的解耦通信。

API 设计遵循 RESTful 最佳实践，配合 Swagger/OpenAPI 提供完善的接口文档。请求验证和异常处理通过过滤器统一实现，确保接口的健壮性和一致性。性能优化方面，实施缓存策略、数据库查询优化和异步编程模式，提升系统吞吐能力。

Python AI 服务基于 FastAPI 框架构建，提供高性能的模型推理接口。动作识别服务集成 YOLOv8 模型，通过 OpenCV 进行视频帧处理和特征提取。

大语言模型服务基于微调的 ChatGLM 模型，通过 Prompt 工程优化教育场景的对话效果。

5.4 测试与质量保障

单元测试覆盖核心业务逻辑和算法组件，确保代码的正确性和可维护性。集成测试验证模块之间的协作和 API 契约的一致性。端到端测试模拟真实用户场景，验证关键业务流程的完整性。

性能测试评估系统在高并发场景下的表现，识别性能瓶颈和优化机会。安全测试检查系统的安全防护能力，预防常见的安全漏洞。兼容性测试确保系统在不同浏览器和设备上的正常运行。

6 项目管理

6.1 变更日志

下面是我们项目成员进行开发时使用的变更日志样例：

| 变更日期 | 变更描述 | 变更动机 | 变更影响 |
|-----------|-----------------------------------|----------------------------------------|--------------------------------|
| 2025.9.30 | 后端技术栈由纯 Python 调整为 C#+Python 混合架构 | 充分发挥 C#在业务系统稳定性与 Python 在 AI 算法方面的各自优势 | 需要建立跨语言服务通信机制，增加集成复杂度但提升系统整体性能 |
| 2025.10.1 | 前端统一为 Web 网页形式，取消微信小程序方案 | 简化开发维护成本，确保功能一致性 | 需要加强响应式设计，确保移动端用户体验 |
| 2025.10.3 | 动作识别模型由 | 提升识别准确率和推理速度 | 需要重新训练模型并更新相关接口， |

| | | | |
|--|----------------------|--|--------|
| | YOLOv5 升级至 YOLOv8 | | 改善用户体验 |
|--|----------------------|--|--------|

每次变更记录时间、内容、原因和影响，确保团队成员了解项目的最新情况。

6.2 开发流程

我们团队将以每两周为一个周期进行开发。每个周期开始时，团队会召开计划会议，分配任务并确定最小可交付成果。在过程中，团队会通过每日例会（15 分钟左右）更新进展与困难。每个周期结束时，进行演示和回顾，根据反馈进行改进。

第一期要完成的功能（最小可用系统）：

1. 用户账号管理

学生和老师可以注册账号，且不同身份的用户登录后看到不同的页面，用户可以修改自己的基本信息。

2. 课程管理基础功能

老师可以创建一门课程，系统会生成一个课程邀请码，学生通过输入课程码加入课程；老师可以上传教学视频，学生可以观看；

3. 作业提交基础功能

老师可以创建一门课程，系统会生成一个课程邀请码；学生可以上传自己的训练视频（这个版本先不进行智能分析）；可以看到作业是否提交成功。

4. 技术基础搭建

完成 C# 后端的基本框架，完成网页前端的基本框架，建立数据库的基本结构，配置好服务器运行环境。

6.3 功能愿望清单

1. 对于视频上的运动的完整的动作识别和纠错功能。
2. 教师具有同时管理多门课程的能力和一个便捷的批改作业和打分的系统。
3. 用图表展示学生的训练数据；集成 AI 智能对话助手。

6.4 团队协作

6.4.1 分工方式：

1. 前端开发：2 名成员，分别负责前端不同模块的开发
2. 后端开发：2 名成员，分别负责数据库接口部分和 python 动作识别部分的开发

6.4.2 会议安排：

1. 每周一次线上会议（检查任务进展，讨论问题）
2. 每两周一次迭代计划与回顾会议

6.4.3 会议目标：

1. 确认任务分配与进度
2. 解决遇到的技术或协作问题
3. 根据客户反馈调整功能优先级

7 团队与角色

7.1 团队背景

我们的团队由 4 名成员组成，大家在校期间都学习过数据结构、

算法分析、数据库原理等核心课程，对软件开发有系统的理论知识储备。

在具体技术方向上，团队成员各有侧重。有的同学对 Web 开发特别感兴趣，在课程项目中积累了不少前端开发经验；有的同学专注于后端技术，对服务器编程和数据库设计有深入理解；还有的同学对人工智能方向充满热情，学习过机器学习和计算机视觉相关课程。

7.2 项目经验情况

虽然团队成员之前没有开发过完全类似的体育教学平台项目，但大家都参与过不同类型的软件开发实践。比如有的同学做过在线商城的课程设计，有的同学开发过校园社交小程序，这些经历都为我们完成当前项目提供了宝贵的经验借鉴。

7.3 成员角色分工

7.3.1 C#后端开发（1 人）

负责设计和搭建 C#后端系统架构；开发用户管理、课程管理、作业管理等核心功能模块；设计和维护 MySQL 数据库，编写数据访问层代码；与前端和 Python 服务进行接口对接；保证后端系统的稳定性和性能。

7.3.2 Python 后端开发（1 人）

负责 YOLO 动作识别模型的集成和优化；开发和维护 AI 对话服务接口；处理视频分析任务，提供动作识别结果；与 C#主服务进行数据交互；优化 AI 服务的性能和准确率。

7.3.3 前端开发（2 人）

设计和开发 Web 用户界面；实现响应式布局，确保在不同设备上的显示效果；与后端 API 进行数据交互；优化用户体验和界面交互效果

同学 A：主要负责学生端界面开发，包括课程学习、作业提交、进度查看等功能

同学 B：主要负责教师端和管理端界面开发，包括课程管理、作业批阅、数据统计等功能

8 约束与风险

8.1 外部约束条件

在项目开发过程中，我们需要考虑到一些外部的约束条件。首先是政策法规方面，我们的平台会收集学生的训练视频和个人信息，这涉及到数据隐私保护问题。我们需要确保符合学校的相关规定，也要遵守基本的网络安全法规。

在伦理道德方面，AI 系统给出的动作评价和建议需要客观公正，避免出现有偏见或者歧视性的判断。特别是对运动能力不同的学生，系统应该给予鼓励性的反馈，而不是简单的批评。

学校的信息化管理规定也是我们需要考虑的。平台可能需要与学校的现有系统进行对接，比如用户认证系统，这就需要遵循学校的技术规范和数据标准。

8.2 资源获取情况

目前来看，项目所需的基本资源还是比较充足的。在数据方面，我们可以先使用公开的运动数据集来训练模型，后续再通过试点班级收集真实的训练数据。不过需要特别注意，收集学生数据时要获得充分的授权，并且要确保数据的安全存储。

在技术服务方面，我们计划使用开源的 YOLO 模型和一些现成的 AI 工具，这些都可以免费获取。云服务器方面，学校可能提供测试用的服务器资源，或者我们可以申请教育版的云服务，这些都能满足项目的基本需求。

但是如果要大规模推广使用，可能需要更多的服务器资源和网络带宽，这就会涉及到成本问题。好在目前只是原型开发阶段，现有的资源应该够用。

8.3 潜在风险分析

技术方面最大的风险是动作识别的准确率问题。体育动作比较复杂，光线、角度、服装等因素都可能影响识别效果。如果识别结果不够准确，给出的纠正建议就会不可靠，影响用户体验。我们的应对策略是先聚焦几个基础动作，做好做精，而不是贪多求全。

另一个风险是系统性能问题。视频处理比较耗费计算资源，如果多个学生同时上传视频，系统可能会响应缓慢。我们计划采用任务队列的方式，让视频分析在后台进行，不影响学生的其他操作。

用户接受度也是个需要考虑的风险。老师和学生可能不习惯这种新的教学方式，或者觉得操作太复杂。我们需要把界面做得尽量简单

易用，同时提供充分的使用指导。