

AI-EduMate 项目团队报告

1. 引言

本报告旨在总结 AI-EduMate 智能学习辅助系统项目团队在开发过程中的工作分工、成员贡献及各阶段的工作成果。AI-EduMate 是一款旨在通过整合人工智能技术，为用户提供高效学习管理、日程规划和知识整理服务的系统。项目团队紧密协作，遵循软件工程化实践，成功实现了系统的核心功能。

2. 团队成员与分工

本项目团队由以下成员组成，并明确了各自的职责：

- 陈品亮 (组长):** 负责项目的整体设计、架构规划、核心接口定义
- 李卓夫:** 负责接口设计与实现，与组长共同确保前后端数据交互的顺畅。
- 田正灏&谢汶余:** 负责前端开发，主要关注用户界面的实现和交互逻辑，共同构建用户友好的界面。
- 王一茗&林宏宇:** 负责后端开发，主要关注业务逻辑的实现和数据库交互。共同保障后端服务的稳定性和数据处理。

3. 各阶段工作分配与成员贡献

3.1 需求分析与系统建模阶段

工作内容: 深入理解项目背景与目标，进行详细的需求分析，并完成系统用例模型、静态模型（类图）、动态模型（序列图、状态图）以及数据库模型的设计。

成员贡献:

- 陈品亮 (组长):** 作为项目负责人，主导了整个需求分析过程，定义了系统核心功能和用户交互方式。在系统建模方面，负责整体用例总图的规划，并指导核心用例详情的梳理。在数据库模型设计中，负责核心表结构的设计与索引约束的定义，确保数据模型的合理性。
- 王一茗 & 林宏宇 (后端):** 参与了需求讨论，并根据后端实现的可行性，为数据库模型和实体类关系设计提供了输入，确保了 tbl_appointments、tbl_course_schedules、tbl_notes 等核心表结构的合理性。

- **田正灏 & 谢汶余 (前端):** 参与了用户界面与交互相关的需求讨论，为前端模块的用例设计提供了反馈，例如 login.html、appointment.html、schedule.html 和 note.html 等页面的功能需求。

3.2 架构设计与接口定义阶段

工作内容: 确立 MCP (Model-Controller-Persistence) 分层架构，定义前后端交互 API 接口规范，并进行技术选型。

成员贡献:

- **陈品亮 (组长):** 负责确立 MCP 架构，并主导了所有核心接口的定义，包括用户认证、日程管理和笔记管理等模块的 API 规范。他在这一阶段的工作为前后端并行开发奠定了坚实基础。
- **李卓夫 (接口):** 紧密配合组长，参与了接口的详细设计与文档编写，确保接口的清晰性和可实现性。在 JWT 鉴权访问、日程新建/获取/删除、笔记新建/编辑/删除等接口的设计上贡献突出。
- **王一茗 & 林宏宇 (后端):** 根据接口定义，负责后端 Spring Boot 服务的搭建，并实现了各模块的控制器 (Controller) 和业务逻辑层，确保接口能够正确响应前端请求。

3.3 后端开发阶段

工作内容: 实现用户认证、日程管理、课程表管理、笔记管理等核心业务逻辑，并与 MySQL 数据库进行交互。

成员贡献:

- **王一茗 (后端):** 负责用户认证模块的后端实现，包括用户注册、登录、JWT 令牌生成与验证，以及用户信息管理。同时，他主导了日程管理模块的后端逻辑，包括日程的增删改查。
- **林宏宇 (后端):** 负责笔记管理模块的后端实现，包括笔记的新建、编辑、删除和查询列表功能。此外，他协助王一茗完成了课程表管理等模块的后端开发，确保了数据库操作的正确性和效率。
- **陈品亮 (组长):** 持续对后端开发进行指导和代码审查，确保代码质量和架构符合整体设计。

3.4 前端开发阶段

工作内容： 根据 UI/UX 设计，实现用户界面，包括聊天界面、登录界面、课程表、预约行程和笔记页面，并与后端 API 进行集成。

成员贡献：

- **田正灏 (前端):** 主要负责聊天界面的开发，包括消息的发送与接收、Markdown 渲染、代码高亮以及“正在输入”指示器的实现。他确保了聊天功能的流畅性和用户体验。
- **谢汶余 (前端):** 主要负责用户管理相关的前端页面（如登录界面），以及课程表、预约行程和笔记等学习工具模块的界面实现。她确保了各功能模块的 UI 一致性和交互逻辑。
- **陈品亮 (组长):** 对前端开发提供了技术指导，特别是在前后端分离架构下的数据交互和状态管理方面，确保前端能够正确消费后端接口。

3.5 测试与质量保证阶段

工作内容： 制定测试计划，设计测试用例，执行单元测试、集成测试、系统测试、验收测试和性能测试，并进行缺陷跟踪与管理。

成员贡献：

- **陈品亮 (组长):** 负责测试目标的制定和测试类型的规划。他关注整体测试进度，并推动缺陷管理流程的执行，确保项目质量。
- **王一茗 & 林宏宇 (后端):** 负责后端 Java 业务方法和 DAO 层方法的单元测试，并参与了后端接口与数据库的集成测试。他们积极修复测试中发现的缺陷，例如“并发添加日程时偶发时间戳冲突”等问题。
- **田正灏 & 谢汶余 (前端):** 参与了前后端交互的集成验证和系统测试，确保前端功能与后端数据的一致性。他们也积极修复前端相关的 UI 不一致或交互问题。
- **李卓夫 (接口):** 在接口测试阶段发挥了关键作用，确保 API 的稳定性和数据一致性。

3.6 工程化与持续改进阶段

工作内容： 实施自动化构建、代码生成、测试自动化、配置管理自动化和部署自动化等工程化手段，并规划持续改进。

成员贡献：

- **陈品亮 (组长):** 规划了 Maven 多模块构建系统、Git 分布式版本控制策略、CI/CD 流水线等自动化和协作化手段。他对项目工程化效果评估和持续改进计划的制定起到了核心作用。
- **王一茗 & 林宏宇 (后端):** 积极实践后端自动化构建 (Spring Boot Maven Plugin)、代码生成 (MyBatis 代码生成器)，并配置了 JUnit 和 Spring Boot Test 进行单元测试自动化。他们还参与了 Spring Boot 自动配置和资源文件处理自动化。
- **李卓夫 (接口):** 在接口测试和 API 文档生成方面，为工程化提供了支持，确保接口定义的规范性。
- **田正灏 & 谢汶余 (前端):** 虽然文档中前端工程化内容较少，但他们积极适应了团队的协作流程，包括 Git 分支管理、代码提交规范和代码审查流程，确保前端代码与整体工程化体系的融合。

4. 总结

AI-EduMate 项目团队在组长陈品亮的带领下，通过明确的分工和紧密的协作，成功地完成了项目的各个阶段任务。陈品亮在整体设计、接口定义和项目管理方面发挥了核心作用，确保了项目的顺利推进和高质量交付。李卓夫在接口层面的贡献保障了前后端通信的效率和准确性。田正灏和谢汶余作为前端开发的核心力量，为用户带来了直观、流畅的交互体验。王一茗和林宏宇在后端开发中，构建了稳定可靠的业务逻辑和数据服务。整个团队通过实施全面的软件工程化手段，包括自动化构建、测试、部署以及规范的版本控制和代码审查流程，显著提升了开发效率和代码质量，为 AI-EduMate 项目的成功奠定了坚实的基础，并展现了良好的团队协作精神。