**迭代计划**

　　　　　　　　　　　　　制定日期：2024.3.27

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 7 | 项目名称 | 课表小助手 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 3.27-5.14 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 选择前端技术框架（React Native或Flutter） | 3.27-4.3 | 张倍宜，陈瑞涵 | | 2 | 学习Jaccount接口 | 3.27-5.14 | 顾一帆 | | 3 | 设计架构视图 | 4.8-4.11 | 全体 | | 4 | 选定编程规范 | 4.3-4.4 | 张倍宜 | | 5 | 开发前端技术原型（代码） | 4.4-5.10 | 张倍宜 | | 6 | 构建课表数据库 | 4.4-5.10 | 陈瑞涵 | | 7 | 测试前端技术原型 | 5.11-5.12 | 全体 | | 8 | 编写《迭代评估报告》 | 5.13-5.14 | 张倍宜 | | | | |
| 预期成果：   1. 前端技术原型（代码） 2. 迭代评估报告 | | | |
| 主要的风险和应对方案：   1. 技术风险：利用技术栈将界面原型和用例等建模实现为具体软件（包括前端子系统或用例），需要共同完成更大量级的代码编写，考验团队组员面对未知技术的学习能力和实践能力；同时也需要团队组员的密切沟通和配合。   应对方案：鼓励组员间的知识共享和经验交流，建立技术分享机制，促进彼此学习；合理分配任务；预留一定的缓冲时间，以应对技术学习和掌握的不确定性。   1. 需求风险：在界面原型中设计的建模尚为抽象阶段，实际实现的过程中可能会遇到诸多设计上、体验上、技术上等方面的难题，或者是需要返工和修改需求的问题，需要组员共同努力化解。   应对方案：前期做好需求调研和分析，尽量完善和细化需求，减少后期大的改动；采用敏捷开发模式，尽早开发可用的原型并收集反馈，及时调整需求；合理安排时间,在进度计划中预留应对需求变更的时间缓冲；注重模块化和松耦合的设计,提高系统的可修改性和可维护性。  3. 进度风险：组员们不具备较为成熟的开发经验，未来的学习成本和工作量是未知数，如果组员的学习压力较大，可能会对工作的推进造成阻碍。  应对方案：制定详细的学习计划，分阶段有针对性地开展技术学习；注意平衡任务分配，避免个别组员学习压力过大。 | | | |