**迭代计划**

　　　　　　　　　　　　　制定日期：2024年3月22日

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 第四组 | 项目名称 | “交集”评价交流平台 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2024.3.22至2024.5.14 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 选择架构风格，确定语言、框架、工具 | 3.22至3.24 | 李亚辉、易中富 | | 2 | 根据第一次评审结果改进前端界面原型 | 3.25至3.27 | 陈昊宇、胡文杰 | | 3 | 设计多个架构视图 | 3.28至3.30 | 吴先道、李亚辉 | | 4 | 选定并撰写编程规范 | 3.31至4.1 | 前端：胡文杰，后端易中富 | | 5 | 初步搭建后端开发环境 | 4.2至4.7 | 全组成员 | | 6 | 搭建软件架构，按编程规范实现典型的2－3个用例 | 4.8至4.10 | 易中富、吴先道 | | 7 | 设计关键算法 | 4.11至4.14 | 全组成员 | | 8 | 设计物理架构 | 4.15至4.20 | 全组成员 | | 9 | 设计前端逻辑架构 | 4.21至4.27 | 全组成员 | | 10 | 设计后端逻辑架构 | 4.28至5.2 | 全组成员 | | 11 | 搭建数据库，完成页面视图 | 5.3至5.8 | 全组成员 | | 12 | 测试是否达到需求中预定的技术要求，并根据测试结果进行改进 | 5.9至5.10 | 李亚辉、易中富 | | 13 | 撰写、评审软件架构文档 | 5.11至5.12 | 胡文杰、陈昊宇 | | 14 | 编写《迭代评估报告》，评审记录和测试记录、开发总结 | 5.13至5.14 | 全组成员 | | | | |
| 预期成果：   1. 迭代计划 2. Vision文档 3. 软件架构文档 4. UML模型（含用例模型、分析模型和设计模型） 5. 本迭代的迭代评估报告 6. 确定前端与后端开发所用的语言、框架、工具 7. 完成编程规范的制定和实施 8. 完成概念建模 9. 完成数据库搭建 10. 完成前端、后端软件架构的设计、构建 11. 完成关键算法的设计与实现 12. 完成技术原型测试 | | | |
| 主要的风险和应对方案：  项目风险：  1.进度风险：本次迭代要完成的任务较多，在第一次迭代中，我们的UI不是很完善，会再次升级，且本次迭代正好与ICS的期中考试时间有重合，进度压力大  2.技术风险：后端使用Go语言进行编码，这是我们之前未曾接触过的语言，对于Go语言的Beego框架我们也不熟悉；且代码可能存在难以解决的bug；  3.需求风险：在第一次迭代中没有考虑技术相关问题，提出的一些需求可能无法实现，对于需求理解分析不正确的情况会导致需求变更。  缓解和应急措施：  1. 合理制定迭代计划以提升效率；增加组内沟通交流，减少返工；  2. 尽早开始学习新技术；同时可以借鉴网上的技术博客或者向大佬寻求帮助  3. 对于需求变更，我们要在设计软件架构的时候就考虑到，软件架构应该力求高度解耦，从而能够更加灵活地应对需求变更 | | | |