**迭代评估报告**

　　　　　　　　　　　　　　　　　　评估日期：2024/5/13

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 第四组 | 项目名称 | “交集” |
| 迭代名称 | 技术原型迭代 | 实际起止日期 | 2024/3/20至2024/5/14 |
| 任务达成情况：   1. 确定了本次迭代的迭代计划，将完成各项任务的时间粒度细分到2~3天，确定了各项任务的负责人。 2. 根据第一次评审结果改进前端界面原型。确定了后端开发所用的spring框架，对项目已存在的代码风格进行了统一，提升了团队开发的效率，减少了错误出现的概率。本任务从3月22日开始，到3月27日如期完成，完成质量良好。 3. 选定并撰写了编程规范，搭建了软件架构，按编程规范实现一些基础用例，同时初步搭建起后端开发环境。本任务从3月28日开始，到4月10日如期完成，完成质量良好。 4. 设计了物理架构、前后端逻辑架构等，并以此为基础完成了前后端功能的基础实现，同时根据项目需求搭建起了数据库，完成了页面视图，明确了各逻辑视图的内部结构以及各个模块之间的关联。本任务从4月11日开始，到5月2日如期完成，完成质量良好。 5. 实现了本项目的技术原型，完成了本项目数据库、后端服务以及前端的整体架构的实际搭建工作。初步完成了后端的实现代码以及前端大部分页面的实现代码。编写了设计模式和本次项目的迭代评估报告。本任务从5月3日开始，到5月13日如期完成，完成质量良好。 | | | |
| 评审/测试的结果：   1. 对后端接口的实现进行了一系列测试，保证了接口实现的正确性。 2. 小组内讨论了前后端代码的风格和实现方式，使代码更加统一，增加了代码的可读性。 | | | |
| 问题、变更和返工：   1. 在第一次迭代中数据都在前端固定，且由于团队内没有严格设计数据库的实现，导致前端组件需要的信息无法与从后端获取的数据有效匹配，不同组员需要从其他组员设计的表中获取的信息也不尽相同，对开发造成了一系列的困扰。在团队内部多次交流后，对数据库的做出了基本的设计，能够满足各位组员在开发时的基本需求。 2. 第一次迭代中，前端没有很好地完成组件化设计，使得后期在组件复用时会产生许多问题。在优化了组件代码后，代码的复用性得到了很大的提升，提高了开发效率，不同组员负责的部分也能更好的联系起来。 | | | |
| 经验和教训：   1. 无论是前端的组件实现还是后端API、数据库等的实现都要考虑到后续的迭代，不能为了快速实现功能而导致在后续团队交流或者功能变更时产生更多问题。 2. 团队内部要确定好API的实现和功能，提升团队开发项目的效率，避免无效的时间和精力付出。 3. 使用新技术的时应该提早开始对于新技术的学习，尽早熟悉掌握需要用到的技术，以有效规避技术风险。 4. 写代码要符合编码规范，统一代码风格，提升代码可读性，减少因代码冲突产生的问题，不然很可能会降低团队开发的效率。 | | | |