**迭代评估报告**

　　　　　　　　　　　　　　　　　　评估日期：2024.5.13

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 7 | 项目名称 | 课表小助手 |
| 迭代名称 | 技术原型迭代 | 实际起止日期 | 3.27-5.14 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）   1. 选择前端技术框架：选定flutter作为前端技术框架 2. 学习jAccount接口：申请jAccount接口使用权，学习接口使用、单点登录接入，目前可以获取个人信息，但课程信息获取仍在调试中。 3. 完成编程规范、架构文档。 4. 开发课程表前端：实现课程表展示、切换周数、添加课程、删除课程、设置等功能。 5. 构建课表数据库：后端完成了flutter常用数据库sqflite的学习与搭建，能够对课程数据进行多种增删查操作，后续会继续丰富接口的实现，以满足更便捷地调用；还要进行数据安全性的优化，防止异常行为导致的数据破坏。 6. 测试技术原型 | | | |
| 评审/测试的结果：（执行了哪些评审和测试？评审和测试的结果如何？）  对代码质量、合理性以及项目结构的合理性进行了评审，以及进行了关键数据结构与算法和数据库工作情况的正确性测试。评审结果认为项目结构清晰，各文件意义明确，且不同职责的代码包之间分工明确，能够满足目前技术迭代的需求和后续的完善开发；代码主要分为三部分，即前端代码、后端数据库代码以及与jAccount的对接实现代码，由于小组均首次接触dart语言以及flutter框架，因此代码的编写可能不够规范和熟练，但通过学习和利用AI生成代码，整体代码质量能够符合要求。测试结果主要为利用flutter自带的test功能，通过多次编写单元测试代码，已经确认了重要部分的正确性和对部分边界情况的正常处理，但可能涵盖得不够全面，需要后续在开发的过程中继续滚动测试。 | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）   1. 测试jAccount接口时，按照开发者文档上的示例，会报错，暂时无法获取个人课表信息 2. 对移动端开发、flutter框架的机制以及组件生命周期等知识不够熟悉，编写代码过程较为坎坷，但经过学习目前已经有了较大改善。 | | | |
| 经验和教训：   1. 使用git的分支功能，前后端同步开发，提高效率 2. 在本次迭代的代码实现过程中前后端的协同开发遇到了很多困难，这提醒我们应当让前后端各自提供的接口更加清晰，需要让其他开发人员足够了解自己的部分实现了哪些接口、需要对接哪些接口。 3. 接触不熟悉的技术栈时，一定要尽早主动学习，以防在编写代码时遇到各种问题；同时，应当编写或从网上找到几个易上手的demo，组织组员进行学习。 | | | |