**迭代评估报告（简化版）**

　　　　　　　　　　　　　　　　　　评估日期：11.17

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 10 | 项目名称 | 一心·时间管理APP |
| 迭代名称 | 技术原型迭代 | 实际起止日期 | 10.11-11.16 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）   1. 在用例建模的基础上，进行面向对象分析，识别出了概念类，建立了概念类图。 2. 建立了闹钟用例的用例实现，以及用例实现与用例间的跟踪图，识别出了实体类、控制类和边界类，画出了时序图、生成了通信图和完成了VOPC类图。 3. 对大作业进行了架构设计，采用了微服务架构，画出了物理架构、逻辑架构、数据视图、页面视图，选择了本大作业关键的质量属性的设计战术，完成了《软件架构文档》。 4. 运用了GOF设计模式对大作业进行了详细设计，写出了两种设计模式在系统中的使用并画出了相应的UML类图。 5. 及时更新了迭代计划，并按照计划基本完成了这次迭代的任务。 6. 前端完成了闹钟功能（完成的小任务包括小游戏、指定物品拍照、摇晃手机、算术题）、学习功能（开始学习之后如果切换到别的应用时会有提醒）、完善了个人中心的列表、提出了公共自习室和小组学习室的功能。 7. 后端完成了客户端的搭建，并与前端成功连接。 | | | |
| 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？）   1. 对图像识别任务的测试：会出现闪退的问题（已解决）、图像识别的时间过长 2. 对小游戏的测试：通过测试 3. 对摇晃手机的测试：通过测试 4. 对用例分析建模的评审：修改了部分细节 5. 对架构设计的评审：修改了部分细节 | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）  问题：   1. 在使用flutter与安卓原生代码交互时产生的问题，如安卓底层权限获取问题、安卓原生与flutter通信以及资源共享问题、网络请求与线程分配问题等。 2. 调用百度云图像识别的时候，一开始无法正确读取到我们所储存的图像信息，导致无法识别成功。在使用了图像识别功能之后，安装之后的app会出现闪退的情况。 3. 在微服务架构设计时，主流的都是一库一服务，在数据库层的耦合，提高自治性，但是这种设计不同服务之间的数据调用会带来问题。   变更：   1. 完善修改《迭代计划》，添加了其他优先级较高的技术风险，根据实际实施情况，更为具体合理地指定了任务的成员安排，细化了任务的时间安排。 2. 完善架构设计，增加了部署视图中web端对应的内容，调整了《软件架构文档》的格式。 3. 完善了概念类图，细化了对类的划分粒度。   4. 在微服务架构中，从一库一服务转变成了一库多服务。  返工：   1. 在配置文件中添加了网络权限，解决了图像识别的闪退问题。 2. 使用methodChannel进行通信，并通过查阅资料了解了android应用的结构，解决了通信问题。 3. 使用handler与thread解决网络请求与主线程冲突以及线程阻塞的问题。 | | | |
| 经验和教训：   1. 在迭代的计划上，应该经常对迭代计划的任务安排进行重新评估，结合项目实际的开发进度和需求，调整、细化时间安排，让计划能够更好地指导开发工作的进行。 2. 在架构设计时，要考虑到自身项目的实际情况，选择适合自己的架构设计模式。 | | | |
|  | | | |