**项目总结报告**

日期：2021/1/5

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 20 | 项目名称 | 时间管理App |
| 编程语言 | kotlin | 开发平台和框架 | Android Studio + IntelliJ IDEA + spring boot + maven + MYSQL + Gradle + Volley + junit+kubenetes+jenkins |

|  |  |
| --- | --- |
| **软件需求特性** | |
| 1.是否实现了项目立项时的所有需求？列出实现的新增需求和未实现的需求。  没有实现所有需求，根据需求优先级实现了高优先级功能，部分中低优先级功能  新增需求：  a.增加用户交互的自定义任务，如自定义算术题  b.增加个人信息展示页面，好友间展示更多信息  未实现的需求：  a.短信验证码登录注册，  b.分享报告至qq、微信等第三方平台。  2.采用哪种架构风格？哪些设计模式？  本系统使用分层架构和Client/Server架构。  采用了原型模式、适配器模式、抽象工厂模式等。  3.技术方案有哪些亮点？  a.用kotlin作为开发语言，既能更好的支持安卓原生平台，也能带来更少的空指针异常，更少的代码量，更快的开发速度，更一致的开发体验。  b.用开源网络通信库Volley负责网络通信模块，既提供了高性能高可靠性的网络通信，又提高了开发效率  c.采用了本地数据库和服务端数据库结合方式，减少服务器端负担  d.用开源框架jenkins做CICD，实现了自动化的集成和部署，提高效率  e.用docker作为容器化代码运行平台，隔绝环境影响，保证了服务的可迁移性  f.用kubernetes作为容器编排管理工具，提供了弹性扩容、分布式容错、热更新等机制的支持  4.是否做了单元测试？是否做了系统功能测试？是否做了性能测试？是否做了兼容性等其他非功能测试？  完成了单元测试  完成了系统功能测试  完成了性能测试  完成了压力测试  完成了安装测试  完成了界面测试  完成了部署测试 | |
| **项目组成员对项目的贡献度（%）** | |
| 肖一：20%  赵阳：20%  张明煜：20%  王禹：20%  陈治西：20% | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 12.9k |
| 复用他人代码行数： | 0.3k~0.5k |
| 类的个数： | 80~100 |

|  |
| --- |
| **经验、教训和建议** |
| 经验：   1. 涉及网络通信层这类很多模块都需要交互的设计时，应当和其他对网络层有需求的开发者讨论统一使用同一种网络层通讯策略，优化代码结构减少冗余； 2. 进度风险高时适当删减掉低优先级的需求，集中精力先实现核心需求； 3. 在动手写代码前应当做好很多前置工作，比如明确数据库结构、明确对接接口、明确公共变量等，工程目录要整理到能够区分各模块，不然目录混乱修改、维护代码会极其困难。   教训：   1. 涉及到第三方平台接口的工作应该早做准备，例如短信验证，走第三方云平台就需要通过一系列的认证、信息审核，需要一段时间才能准备妥当，在时间上应该早做准备，以避免进度风险； 2. 界面UI设计需要专门的思考，用户对系统的第一印象好坏取决于UI，所以UI设计要经过细致的打磨，要充分考虑用户的体验，这也是我们项目的欠缺之处；   建议：   1. 要详细制定迭代计划并且严格按照计划执行 2. 团队成员之间要做好沟通，定期一起开会讨论 3. 有规律投入开发时间，保证软件质量 4. 注重开发流程规范，严格的code review机制 |