**项目总结报告**

日期：2023.06.18

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 8 | 项目名称 | LifeLogger |
| 编程语言 | Javascript/Typescript | 开发平台和框架 | React-native |

|  |
| --- |
| **项目工作小结** |
| 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。  立项时的需求均已实现；  额外实现功能：统计分享事件所在的城市，并以地图形式展现，地图上颜色的深浅代表在这一城市的分享频率  2.采用哪种架构风格？哪些设计模式？  架构风格：采用分层架构，后端分为数据访问层、数据持久化层、服务层和控制层，前端分为应用层、接口层、视图层、本土化层（i18n）  设计模式：前端采用组合（Composite）模式，将组件封装起来后组合为新的组件和页面；后端采用代理（Proxy）模式，由数据访问层提供接口间接地访问数据库。  3.技术方案有哪些亮点？  1）使用Expo开发React-Native，支持跨平台，在ios和安卓系统上都能正常运行  2）使用i18n框架，支持多语言切换  3）支持推荐同城相似喜好的好友  4）事件记录功能在退出APP后仍然会保持计时状态  5）设置多级缓存（内存缓存， Redis缓存和关系型数据库缓存），并采用分页加载的策略，防止静态资源过多导致的性能问题  6）统计页面采用了内嵌WebView的方式实现，解决了React Native中无法调用图表API的问题  7）使用JWT单点登录减少用户登录的次数  8）后端和数据库部署在多服务器上，提高性能  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？  单元测试：是  系统功能测试：是  性能测试：是，在1000用户并发时，平均响应时间为1.8s |
| **项目组成员对项目的贡献度（%）** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 姓名 | 需求 | 设计 | 编码 | 测试 | 项目管理\* | 小计 | | 刘洋 | 25% | 25% | 30% | 0 | 15% | 25% | | 闫景升 | 25% | 25% | 40% | 30% | 15% | 26% | | 杨征 | 25% | 25% | 15% | 0 | 45% | 24.5% | | 张馨匀 | 25% | 25% | 15% | 70% | 25% | 24.5% |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 8938 |
| 复用第三方代码行数： | 11345 |
| 类的个数： | 17 |

|  |
| --- |
| **经验、教训和建议** |
| 1. 前端样式应该统一存储在样式层中，方便后期修改样式或添加新的样式 2. 前端样式应该先进行原型设计再进行编码，保证风格的统一 3. 在项目开始时，应该先确定好数据格式和前后端通信的各项请求格式，方便前后端并行开发，提高开发效率 4. 较大的模块应该进行细化，做到一个模块尽量由一个人开发，一个模块对其他模块有良好的封装性，方便后期维护 |

项目组各成员签字：