**项目总结报告**

日期：2016/9/9

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 5140379050 | 姓名 | 周天琪 |
| 学号 | 5140379051 | 姓名 | 李轶 |
| 学号 | 5140379068 | 姓名 | 许泽资 |
| 项目名称 | Memory | | |
| 编程语言 | Java | 开发平台和框架 | Spring Boot/ Android |

|  |  |
| --- | --- |
| **软件需求特性** | |
| [列出所实现软件的需求特性，即已实现的概要的功能和非功能需求。]  **基本功能：**   * 注册登录   用户可以进行注册登录的行为，进而获得下一步的功能权限   * 添加物品   用户可以添加图片、音乐、书籍及对应的内容，选择开放权限。   * 管理物品感想   用户通过扫码，或者在自己的管理空间打开物品对应的内容。用户可以选择新增感想、阅读感想。   * 管理好友   用户可以添加、删除好友，浏览某一用户的部分信息。   * 浏览大厅及好友物品   用户可以在大厅中浏览公开的有意义的物品，也可获取好友发布的公开的物品   * 管理数据及内容   具有管理员权限的用户可在管理员界面中管理用户、音乐内容、管理书籍、管理标签   * 评论他人公开的回忆项   用户可以在他人的公开资源下对他的内容进行评论   * 在大厅添加感兴趣的人为好友   用户可以在看到感兴趣的内容时申请添加主人为好友，对方将收到验证消息，待对方审核通过即可成为好友  **进阶功能：**   * 心情录入   用户在上传回忆时可以记录自己当时的心情   * 获取好友匹配（基础实现）   用户可以获取系统推荐的好友列表   * 物品推荐 （基础实现）   根据用户上传的图片，与服务器端的图片库进行比较，可以获取资源的相似度从而进行推荐   * 利用Edison进行存储的本地服务器（基础实现）   系统与Edison开发板有良好的互动   * 背景音乐   用户除了上传音乐相关的回忆之外，还可以给每个回忆添加背景音乐，阅读时可进行播放，提升用户体验 | |
| **软件设计和技术特点** | |
| [列出所实现软件的架构风格、所采用的设计模式、所具有的技术特点等。]   1. 采用了Restful风格的API设计 2. 采用了Spring Boot框架开发后端 3. 采用C/B/S混搭，包括安卓端与web端（管理） 4. 在MVC基础上不断解耦，每一个组件可以方便复用；DAO,Service,ViewController等之间基本采用Spring IoC进行注入管理，维护方便快捷。 5. 统一使用的构建开发为工具包，方便日志、连接等行为。 6. 采用MongoDB,与MySQL同时使用的方式，利用各自优点，提升储存性能。 7. html页面采用动态模板渲染的技术，与开发人员更友好；与后端更加解耦。 8. 基于android特性进行开发：主体是android应用的关键组件activity，使用Intent进行activity间的交互；使用sqlite建立本地数据库；界面布局根据需要选择使用了relativelayout，linearlayout，drawer，floatingActionButton等框架/组件 9. ui上根据需求、易用性、安卓UI设计流行趋势、移动端操作特性选择并自定义android控件，例如listview，cardview，fragment等等 10. 调用系统工具例如相机相册与播放器，以及现有的音乐、书籍API进行更高效的开发 11. 使用BitmapRegionDecoder，imageDownloader进行高清图片加载，实现了图片异步刷新 12. 系统与Edison开发板有良好的互动。 | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 13k |
| 复用他人代码行数： | 6000+ |
| 类的个数： | 160+ |
| 是否全部完成软件开发和测试： | 没有 |
| 工作量： | 120人天 |

|  |
| --- |
| **经验、教训和建议**   1. 开发过程交流不足。开发的过程当中，有许多地方发生了返工、重复等现象，原因是交流不足，这较为严重的影响了整个开发流程。有些地方只有细致入微的交流才能做好，我们做的不够。大家的行为习惯不太一致，加上没有有经验的管理等等因素，团队协作之间互相差半拍。所以，交流沟通好，是重中之重。 2. 经验不足。我们在一些模块的选择上浪费了较多时间，在简单问题的解决方案上也浪费了一些时间。面对从未接触的新问题、新技术上手很慢，没有系统的概念，边写边尝试学习理解的开发方式严重影响了进度与质量。惨痛的教训告诉我们，应该多问，应该先着手去做，而不是去等待，不只是去学习，而是动手。 3. 风险低估。特别是技术风险低估的较为严重。包括安卓端，web端在内，有许多地方理解不够好，在加上经验不足，浪费了大把时间，造成后期进度赶不上；一部分放在假期的任务严重拖沓，测试最终也没能全部完成。预计要用的新特性，包括图片识别匹配也没能如期正常上线。 4. github平台使用效率不高。第一次使用代码管理工具，组员并不太习惯于使用，一度出现了忘记的现象，好在补救及时。而且平台上issue的使用较少，管理与督促的作用没能起来，让团队的开发效率偏低。此后应该熟练使用issue驱动的开发，来提升效率。 5. 编写过程与最初设计有偏离。没有牢牢抓住设计文档、需求规约一步步制定计划、评估、总结，使得开发的节奏比较混乱，应该将在软件工程课程中学到的开发素养运用到实际中来，脚踏实地一丝不苟的将每一小步走好。 |