**项目总结报告**

日期：

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 18 | 项目名称 | 有课-交大课程分享平台 |
| 编程语言 | Java，JavaScript | 开发平台和框架 | ReactNative, Spring |

|  |  |
| --- | --- |
| **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。   实现了项目立项时的所有需求，无新增需求或未实现需求。   1. 采用哪种架构风格？有哪些设计模式？   前后端采用了传统的c/s架构风格。后端采用了负载均衡来调和多台服务器。  后端的敏感词过滤采用了单例模式。  前端的一些编辑界面采用了模版模式。   1. 技术方案有哪些亮点？   前端导入高德地图的api，将课表表可视化，用户可以看到当日所有课程的所处位置以及周边建筑，用户能更便捷的查看当天的上课情况。  使用协同过滤算法来做推荐系统，随着用户的增加，推荐的效果会越来越人性化。  使用负载均衡技术来协同多台服务器，使得单台服务器的压力减小，提高了整个系统的抗压性。  基于Oauth2.0的协议与Jaccount对接，用户无需多注册账号，也更好的保障了用户的数据隐私性。  所有的课程数据都是通过爬虫在教学信息网上得到，具有完整真实的课程库。  加入了敏感词过滤系统，用户在发布关于暴力，政治敏感，色情等不文明语言时会被屏蔽，给用户营造了一个绿色文明的平台环境。  mysql和MongoDB结合使用使得用户评测具有可扩展性，也保持了可关联性  除app外，还开发了管理员管理的web界面。因为给用户提供了反馈投诉的渠道，系统管理员在管理界面能够看到用户们的反馈，从而能够更快地解决app中潜在的bug，提高app的可持续发展性。系统管理员在管理界面也有发布公告的权利，当发布公告后用户们可以在首页查看公告，提高了用户与管理员（程序开发员）之间的交互性。  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，系统功能测试以及性能测试，详情请见测试文件夹。对于兼容性测试，打包发布的apk在主流的安卓平台上都能正常运行。 | |
| **项目组成员对项目的贡献度（%）** | |
| 柳正威：25%  赵旭阳：25%  郭志：25%  朱朝阳：25% | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 21000行 |
| 复用他人代码行数： | 1000行 |
| 类的个数： | 100左右 |

|  |
| --- |
| **经验、教训和建议** |
| 1.在每个迭代开始前，应该仔细详尽地考虑好当次迭代需要完成的任务数及分配情况，在第一次迭代时没有完整地考虑好任务量，导致迭代中频繁地改变需求量任务量，也导致第一次迭代开发时有些许混乱，总体效率不高，在之后的迭代中慢慢纠正了这个问题，使得任务更加量化，每个人的开发效率更高。  2.在前端开发时，要特别注意异步请求特别是网络请求的异常情况，在项目开发前期没有太注意异步请求的异常情况，导致真机调试时如果有网络之类的问题容易出现各种异常。  3.保持每日例会可以有效地提高项目开发效率，保持互相督促和工作的互相了解  4.开发时组中各个人分配的任务应该相对独立，不互相干扰。在git+github的协同环境下，若工作任务有交叉，则容易导致大量的文件冲突与不同commit版本的不兼容，降低了开发效率  5.在开发后立刻完成测试，或者在开发前就定好测试，既可以让开发更有目的性，也有助于代码更加健壮，减少返工与bug  6.开发时的规划不仅要考虑到任务难度与时间，任务是基础还是进阶。还要注意有些任务对整个开发工作有着能够带来很大效率。这样的任务应该优先安排。比如刚开发时没有部署到服务器上，就导致前端开发人员还要装数据库，跑后端代码，影响了整个开发的进度。  7、后端的功能性检测应当具备全，足的特性。即需要照顾到可能出现的所有情况，并且在此基础上，进行足量的数据测试。以避免在项目进行的过程中，因部分功能耦合严重，且未实现而造成的问题（包括但不限于级联问题） |