**项目总结报告**

日期：2021-01-08

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 7 | 项目名称 | 大学生问答社区 |
| 编程语言 | Java, JavaScript | 开发平台和框架 | React，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。  实现的需求：  用户注册时需填写相关信息（包括用户名、密码、姓名、性别、邮件地址）  对用户的手机号码和电子邮箱地址的格式有效性和唯一性进行校验。  用户的密码需包含大小写字母以及数字  用户注册后，向用户注册邮箱发送激活邮件，邮件中包含激活地址，点击该地址完成激活  只有完成激活的用户才能登录，否则提示用户必须先激活  提问。  个人信息维护（包括用户名、密码、姓名、性别、邮件地址，个人头像）  用户可以提出问题，问题在提出问题24小时内可以由自己进行修改，提问包含标题、内容和标签。  回答。用户可以回答其他人提问，用户回答问题的编辑器支持markdown和富文本两种格式。用户可以修改和删除自己的回答  评论。任何用户都可以对于回答进行点赞与评论，以及回复评论  用户可以通过关键字从问题，回答，用户三个维度进行搜索  用户可以关注其他用户，也可以取消关注  用户可以对感兴趣的问题进行关注，也可以取消关注  用户的问题、回答文字审核  用户头像等图片审核  对用户的提问和回答进行管理  我们对一部分需求进行了扩展实现：  我们使用了分词算法来事先更准确的搜索。  根据用户浏览记录，分析与其共同爱好相似的用户（根据浏览的问题及tag等），根据用户社交关注，推荐可能感兴趣的问题和用户  实现了等级系统  未实现的需求：  回答中可以包含图片和视频。  搜索回答  2.采用哪种架构风格？哪些设计模式？  采用B/S 浏览器-服务器架构。用户使用外部浏览器访问网页，通过互联网连接到web服务器  。Web服务器部署在谷歌云端，面向用户时，服务器提供支持的服务，如提问，回答等。面向后端数据库时，服务器负责进行对数据的增删改查。数据库服务器MySQL和MongoDB，分别负责存储关系型数据和非关系型数据，为web服务器提供必要的数据信息。  采用了Template等设计模式，后端使用了MVC架构。  3.技术方案有哪些亮点？  我们应用了有效的分词算法：基于前缀字典实现高效的词图扫描，采用动态规划查找最大概率路径，找出基于词频的最大切分组合。对于未登录词，采用基于汉字成词能力的HMM模型。  在项目中实现了推荐系统：根据用户浏览记录，分析与其共同爱好相似的用户（根据浏览的问题及tag等）。根据用户社交关注，推荐可能感兴趣的问题和用户。  我们使用持续集成对代码进行持续的集成、测试，并使用持续部署保持对新功能与bug修复的落地。  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？  我们进行了单元测试。代码覆盖率达到了约90%，充分保证了代码各个模块的正确性。  系统功能测试按照系统测试用例进行了测试。  客户端的兼容性由babel的browser list进行保证。 | |
| **项目组成员对项目的贡献度（%）** | |
| 胡昊源 25%  施超 25%  王兴宇 25%  熊天磊25% | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 12056 |
| 复用他人代码行数： | / |
| 类的个数： | 约70 |

|  |
| --- |
| **经验、教训和建议** |
| 本次项目最大的风险在于进度风险，由于学期末时间紧需求重，导致部份需求没有完成，但是由于前期根据要求对项目需求做了优先级划分，因此未完成的部份都是优先级较低的需求（如对回答的搜搜），对整体系统的功能展示并没有太大影响。  **人**是软件开发的核心，小组合作要实现1+1>2的成果，在开始阶段我们分工明确，前后端通信需求表达清晰，但是到了项目后期，由于进度原因，我们的沟通没有以前多，因此出现了沟通交流出现分歧，导致我们后期出现不少返工。  通过此次项目模拟实战，我们充分学习到了软件工程的诸多思想与方法，例如设计模型，面向对象等思路，帮助我们极大的提高了工作效率。 |