**项目总结报告**

日期：2023年1月13日

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 5 | 项目名称 | 《白云社区》 |
| 编程语言 | HTML，JavaScript，Java，Python | 开发平台和框架 | 平台：Intellij IDEA，VSCode，Navicat  框架：  √前端：Vue3, Element-Plus, axios  √后端：Spring Boot, Apache Shiro, Spring Data JPA, Flask  √数据库：MySQL |

|  |
| --- |
| **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。  已经实现立项时的所有基本需求，四个进阶需求。  新增需求：管理员权限，用户私聊，拦截器控制访问  未实现需求：匿名发言，邀请回答  2.采用哪种架构风格？哪些设计模式？  采用了 3 Tiers 架构风格。  √创建型模式：工厂模式，建造者模式。  √结构型模式：组合模式，过滤器模式，装饰器模式，外观模式，代理模式。  √行为型模式：责任链模式，命令模式，迭代器模式，策略模式，观察者模式。  3.技术方案有哪些亮点？  √前后端分离，后端服务器中，语义搜索功能与推荐功能也各自分离出独立的服务器，很好地实现了各模块的解耦，方便开发与测试，还可以提高安全性，限制错误的传播。  √前端采用 Vue 框架，基于单文件组件构建网页，大大提高了代码的可复用性。  √后端采用 Spring Boot 框架，将对象之间的依赖关系交由 Spring 控制，避免硬编码所造成的过度程序耦合，使得开发时能够将更多的精力放在上层的业务逻辑上。  √使用Shiro安全框架进行身份验证和权限控制，可以灵活而严格地控制权限，保护数据，避免安全问题。  √语义搜索中，使用句向量的相似性来衡量文本语义之间的相似程度，使用sentence-transformers库生成句向量，再使用Elasticsearch基于向量的搜索，实现查找语义相似的文本。  √内容检测技术：   * 图片内容检测采取小规模模型结构的神经网络架构，既能出色满足识别某一类图像的需求，也方便移植、对设备要求无太高要求。 * 文本内容检测与过滤使用DFA算法进行敏感词查找，并能够对查找出来的敏感词进行替换。该算法并不需要额外搜集大量的数据以及花费较多时间进行训练，但也可以高效地识别敏感词并加以替换。   √推荐算法：推荐系统采用双层强化学习推荐智能体的结构，对于本社区较为多元化的数据类型与数据层次能够更好地向用户进行推荐。考虑到设备配置不高、模型运行较慢的问题，又额外采用基于用户的协同过滤 (user-based CF) 来提高在线推荐的速度，即大模型仅离线运行，生成的数据提供给user-based CF进行实时的在线推荐，既充分运用了双层模型推荐结果的优秀，也通过user-based CF快速的推荐速度避免了后端获取推荐列表时的较大延迟。  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？  √完成了对后端各类的单元测试，语句覆盖率超过90%  √完成了系统功能性测试，86用例数，83通过数，通过率96.51%  √完成了性能测试，3用例数，2通过数，通过率66.67%  √完成了兼容性测试，3用例数，2通过数，通过率66.67%  √完成了易用性测试，1用例数，1通过数，通过率100.00%  √完成了界面测试，6用例数，6通过数，通过率66.67%  √完成了压力测试，2用例数，1通过数，通过率50.00% |
| **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 | | 吴基洋 | 20% | 20% | 20% | 20% | 20% | 20% | | 朱楷文 | 20% | 20% | 20% | 20% | 20% | 20% | | 陈梓钊 | 20% | 20% | 20% | 20% | 20% | 20% | | 龚子钦 | 20% | 20% | 20% | 20% | 20% | 20% | | 程凯文 | 20% | 20% | 20% | 20% | 20% | 20% |   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）： | 15722 |
| 复用第三方代码行数： | 604 |
| 类的个数： | 103 |

|  |
| --- |
| **经验、教训和建议** |
| √合理评估任务的难度，避免产生进度风险：对内容检测、推荐系统中遇到的困难估计不足，接口对接后出现问题等。  √尽量采用低耦合的编程方式：低耦合可以提高组员之间合作的效率，排查错误、作出修改和调整也更为方便。  √善于搜索和阅读资料：网上的资料信息比较繁杂，除了要学会筛选有用的信息以外，很多时候要学着耐心地看官方文档。  √加强交流和沟通：组员之间加强沟通，对于当前遇到的问题及时沟通，共同解决问题，避免信息差造成的返工。  √技术学习：技术的学习建议分工合作，组员之间相互帮助学习技术，不可闭门造车 |

项目组各成员签字：**吴基洋，朱楷文，陈梓钊，龚子钦，程凯文**