**项目总结报告**

日期：2024.6.20

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 16 | 项目名称 | 代码管理平台Code Cube |
| 编程语言 | JavaScript,Java,Python | 开发平台和框架 | IDEA,React,Spring boot,Flask,mySQL |

|  |
| --- |
| **项目工作小结** |
| 1.采用哪种架构风格？哪些设计模式？  我们采用了3Tiers的三层式架构风格，主要分为表示层、业务逻辑层和数据层。  在表示层中，我们基于react框架进行前端的实现，而客户端浏览器通过基于axios的http请求与业务逻辑层进行连接。在业务逻辑层中，我们基于java的springboot框架和python的pytorch和flask框架进行开发。在数据层中，我们主要分为数据访问层和数据存储层。数据访问层采用springboot jpa进行数据库的操作。使用关系型数据库mysql进行数据的存储。  我们采用了命令模式，组合模式等设计模式。在设计智能助手时使用到命令模式，将智能助手的行为抽象为命令对象，实现智能助手与命令的松耦合。在设计仓库文件结构时运用到组合模式，对文件夹这一抽象构建类，既可以充当叶子又可以充当容器。  3.技术方案有哪些亮点？  使用了大模型，增加了新的人机交互方式，采用对话实现部分业务逻辑，方便用户更好了解系统的功能。采用双后端，java后端负责主要业务逻辑，python后端适配pytorch框架。在兼顾性能的同时能够融合大模型的应用。使用适合用户的前端设计，采用三维的文件结构，在登陆界面设计循环动画介绍业务范围。支持使用双语切换，提高系统易用性。  3.是否做了单元测试？是否做了系统的功能测试、性能测试、兼容性测试和易用性测试等？  我们利用mock模拟方法和junit等框架，对项目中的各组件及各层代码进行了单元测试，实现了分支覆盖、条件覆盖和语句覆盖，语句覆盖率达到100%，并且结果正确。我们还对整个系统做了较为详尽的功能测试和部分的兼容性测试与易用性测试。在功能性测试中我们对整个项目的核心应用与较重要的模块进行了着重的测试，未检测出严重错误，并对一般错误进行了改正。兼容性测试中测试了网站在不同浏览器、硬件设施上的运行情况，均无异常。我们还对潜在用户群体进行了易用性测试，并将得到的反馈实现在新版本的代码中。  4.是否采用大模型来辅助开发？对开发生产率约有百分之几的提升？  否。 |
| **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 | | 王晨丞 | 1/3 | 45 | 50 | 1/3 | 1/3 | 39% | | 项百川 | 1/3 | 35 | 40 | 1/3 | 1/3 | 35% | | 龚畅玮 | 1/3 | 20 | 10 | 1/3 | 1/3 | 26% |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件规模** | |
| 前端的代码行数（不包括注解行、空行和复用代码）： | 3835 |
| 后端的代码行数（不包括注解行、空行和复用代码）： | 2763 |

|  |
| --- |
| **经验、教训和建议** |
| 1. 版本管理实行较好，没有出现代码版本冲突，需要回溯的情况。 2. 编程规范执行较好，文件结构清晰，代码书写规范，变量命名易懂，组件使用一致。 3. 对代码的注释缺少，成员较难读懂其他人的代码，需要询问后得知某段代码的位置与用途。 4. 小组成员人数较少，虽然方便组员间统筹规划、分配任务、验收任务，但是人均任务量重，难以支持完善所有需求。 |

项目组各成员签字：