**项目总结报告**

日期：2023/9/9

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 5 | 项目名称 | 代码在线评测平台 |
| 编程语言 | Java，JavaScript | 开发平台和框架 | Intellij IDEA |

|  |  |
| --- | --- |
| **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。  完成了基本需求；  进阶需求中未完成：支持代码提交历史记录的查看和比对，方便用户进行代码版本管理；  支持题目讨论区，用户可以在讨论区中与其他用户交流问题和经验  2.采用哪种架构风格？哪些设计模式？  微服务架构；策略模式、观察者模式、命令模式、适配器模式  3.技术方案有哪些亮点？  使用容器化技术，如Docker，将评测任务隔离在独立的容器中，以确保环境隔离和一致性。  提供对多种编程语言的支持，使用户可以选择使用他们熟悉的语言进行编程。  使用实时通信技术，以实现实时代码执行结果的反馈，包括编译错误、运行时错误和输出信息。  实施严格的安全措施，包括代码审查、漏洞扫描和权限控制，以确保用户数据的安全性。  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？  是  是  是  是 | |
| **项目组成员对项目的贡献度（%）** | |
| 注：总计100%  吴昶霖：25%  王瑜彬：25%  方奕雄：25%  常朋宇：25% | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 36320 |
| 复用他人代码行数： | 1972 |
| 类的个数： | 24 |

|  |
| --- |
| **经验、教训和建议** |
| * **清晰的需求分析：** 在项目开始之前，确保对需求进行详尽的分析，并与利益相关者进行充分的沟通，以确保对系统功能和性能的期望明确。 * **模块化设计：** 采用模块化和微服务架构，将系统划分为小的、独立的模块，以提高可维护性和可扩展性。 * **选择合适的技术栈**： 选择合适的编程语言、框架和工具，以满足系统的需求。 * **团队协作**： 确保团队成员之间的协作和沟通顺畅，使用版本控制系统进行代码管理。 * **迭代总结**：在一次迭代之后要进行及时的总结和团队讨论，及时提出并解决当前出现的问题 * **版本控制：** 使用版本控制工具（如Git）和持续集成/持续部署流程，以确保代码的质量和稳定性。 |

项目组各成员签字：