**项目总结报告**

**项目总结报告**

日期：2023.9.9

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 2 | 项目名称 | “Testland”——代码在线测评平台 |
| 编程语言 | Python, Java, Javascript | 开发平台和框架 | Spring boot, React |

|  |  |
| --- | --- |
| **软件需求特性** |  |
| 1. 需求实现情况   实现的新增需求：   * 用户代码历史版本管理 * 深色/浅色模式UI   未实现的需求：   * 数据的备份和恢复 * 对于不同窗口比例的兼容性  1. 架构风格和设计模式   采用REST架构，前端React工程和后端Springboot工程采用REST api相互通信，后端与测评机之间通过消息队列相互通信。设计中采用了单例、模板和工厂等设计模式。   1. 技术方案亮点  * WebSocket：在评测完成后，后端通过WebSocket向前端发送信息，实时向用户提供代码评估和反馈，无需他们手动刷新页面。 * YARA：在评测用户代码前，参考开源的常见恶意代码规则组，使用YARA对其进行扫描，防止恶意代码的运行对   服务器造成破坏。   * Docker：通过使用docker cotainer进行评测，我们有效地将每次评测任务独立地与服务器主机分隔，增加了安全性和并行性。 * RabbitMQ：通过使用消息队列，我们将用户提交代码后不会由于等待评测造成数据丢失。多线程对队列中的评测信息进行处理，提高响应速度  1. 测试情况  * 单元测试：对后端代码进行了函数、语句100%覆盖率的单元测试。 * 系统功能测试：对代码测评平台的各项功能进行验证： * 用户管理和权限控制：测试用户注册、登录、查看代码项目等功能，并验证权限系统的正确性。 * 题目发布和代码评估：测试管理员发布题目、学生提交代码并进行评估的流程，以及评估结果的准确性。 * 代码编辑和执行：测试在在线集成开发环境（IDE）中编写和执行各种编程语言的代码，并验证输出结果的正确性。 * 性能测试：通过模拟大量用户同时提交代码的场景，测试代码测评平台的性能和稳定性： * 并发性能：测试平台在高负载情况下的响应速度和并发处理能力。 * 安全测试：检查代码测评平台的安全性： * 数据隐私保护：验证用户的代码和数据在传输和存储过程中是否得到合理的保护，防止未经授权访问。 |  |
| **项目组成员对项目的贡献度（%）** |  |
|  |  |
| **软件度量** |  |
| 软件代码行数（不包括注解行、空行和复用代码）： | 439161（包含所用库代码）35390（不包含所用库代码） |
| 复用他人代码行数： | 332 |
| 类的个数： | 前端：5 后端：13 |

|  |
| --- |
| **经验、教训和建议** |
| 1. 区分各项功能的粒度和优先级   在提出需求和安排进度时，没有按照功能的粒度和优先级排序，把大量时间花在解决一些细节问题上面，一定程度上拖慢了进度。未来会更从大局出发，合理安排每一项待办地顺序。   1. 及时通过测试找出代码中的漏洞   在最后阶段花费大量时间修改前面代码留下的一部分bug，应尽早通过完善的单元测试来找出代码的漏洞并在尽可能靠近编码阶段时解决，防止影响其他功能。 |

项目组各成员签字：