**项目总结报告**

日期：2024/8/9

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 17 | 项目名称 | 任易帮 |
| 编程语言 | Javascript + Java | 开发平台和框架 | React + Spring Boot + Spring Cloud |

|  |
| --- |
| **项目工作小结** |
| 1.采用哪种架构风格？哪些设计模式？  架构风格采用了微服务架构。  设计模式采用了代理模式、责任链模式与命令模式。  2.技术方案有哪些亮点？   1. 采用了Spring Cloud微服务架构； 2. 使用Nginx实现反向代理与负载均衡； 3. 使用Nacos实现服务注册、发现和负载均衡； 4. 使用API网关集中管理请求路由、安全认证等功能； 5. 使用sentinel实现流量监控管理、微服务熔断降级等微服务保护功能； 6. 使用关系型数据库MySQL与非关系型数据库MongoDB； 7. 使用WebSocket技术实现实时聊天。 8. 微服务模块实现了多实例部署； 9. 使用华为云的Api Gateway和Function Graph，实现后端实例崩溃监视和重启； 10. 使用Prometheus实现后端实例的健康监视。   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？  (1) 本项目进行了全面的单元测试，包括前端的Jest测试与后端的Junit测试；  (2) 本项目制定了详细的系统测试用例，并进行了详细的功能测试、性能测试(Jmeter)、压力测试(Jmeter)、兼容性测试(测试了在5种不同浏览器上的兼容性)和易用性测试。  4.采用大模型来辅助了开发的哪些阶段（需求/架构/详细设计/编码与调试/测试/计划与报告）？对开发生产率约有百分之几的提升？  采用ChatGPT等大模型辅助开发了架构、编码与调试、测试阶段，对开发生产率约有16.27%的提升。 |
| **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 | | 徐培公 | 20% | 23% | 22% | 20% | 18% | 103% | | 于明睿 | 17% | 23% | 22% | 20% | 18% | 100% | | 王熠笑 | 21% | 20% | 22% | 20% | 18% | 101% | | 吴坤臻 | 22% | 17% | 22% | 20% | 18% | 99% | | 周泓宇 | 20% | 17% | 12% | 20% | 28% | 97% | |

|  |  |
| --- | --- |
| **软件规模** | |
| 前端的代码行数（不包括注解行、空行和复用代码）： | 3996 |
| 后端的代码行数（不包括注解行、空行和复用代码）： | 13542 |

|  |
| --- |
| **经验、教训和建议** |
| 1. 项目技术应当尽早确定，避免项目已经做了一些内容时换架构等等的问题，也便于开发成员尽早学习、掌握以及熟练使用各种技术。 2. 本次项目的部署方式比较传统，在配环境上以及启动上耗费了不少的经历，可以使用docker和kubernetes进行部署。 3. 迭代安排需更加合理地分配，应当均衡地安排每次迭代的持续时间和任务量。 4. 居家远程工作的时候，开发成员需保持不断联系，避免出现某一开发成员长时间处于“失联”的状态。 |

项目组各成员签字：    