**项目总结报告**

日期：2024.8.8

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 1 | 项目名称 | UniGPT - v2.0 |
| 编程语言 | Java, JavaScript | 开发平台和框架 | VS Code, React框架, Spring boot框架 |

|  |
| --- |
| **项目工作小结** |
| 1.采用哪种架构风格？哪些设计模式？  采用了3 Tiers架构和微服务架构的架构风格；单例模式、模板方法模式、门面模式的设计模式。  2.技术方案有哪些亮点？   1. 云服务器：利用华为云平台，云端部署网页应用和数据库。 2. Jaccount登录：与上海交通大学的单点登录系统进行集成，避免繁琐的注册与验证流程。 3. WebSocket协议：更强的实时性、更短的通讯响应时延，极大提高对话体验。 4. 多种大模型支持：使用ChatGPT、Claude、Llama3、KimiAI四种大模型的API调用，用户制作机器人有更丰富的base model选择。 5. Langchain4j框架：深度封装多种大模型接口，支持RAG（知识库）与工具调用。 6. 工具调用：Docker容器化运行用户函数进行自定义函数的预测试，结合华为云FunctionGraph无服务接口实现调用用户自定义函数。 7. 多种类数据库：除使用Mysql外，还使用了PostgreSQL存储向量数据。 8. Kubernetes容器编排平台：自动化应用程序的部署、扩展和管理，实现负载均衡、自动重启、日志监测等功能。 9. Nginx：前端统一访问Nginx代理的端口，微服务仅接受由Nginx转发的请求。在Nginx上进行鉴权操作。 10. Redis缓存：减少数据库查询次数，提升性能。 11. Kafka消息队列：实现微服务之间的高效通信。 12. 测试：前端测试采用Jest框架，后端测试采用Junit，性能测试采用Jmeter，端到端测试采用Cypress。全面测试确保系统可用性。 13. MUI组件库：灵活度更高、更美观的前端组件库，满足各种设计需求。 14. 多语言切换：充分考虑不同用户的使用偏好。   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？  单元测试、系统的功能测试、性能测试、兼容性测试和易用性测试全部进行。测试结果符合预期。  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Copilot、ChatGPT等大模型，辅助了开发的架构、编码与调试、测试、计划与报告阶段。大模型能起到拓展解决问题的思路、缩短编码时间的作用，对开发生产率约有20%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 | | 陈启炜 | 5% | 4% | 5% | 4% | 7% | **25%** | | 黄峻涛 | 5% | 3% | 6% | 5% | 6% | **25%** | | 韦东良 | 5% | 4% | 5% | 6% | 5% | **25%** | | 叶懿芯 | 5% | 9% | 4% | 5% | 2% | **25%** | |

|  |  |
| --- | --- |
| **软件规模** | |
| 前端的代码行数（不包括注解行、空行和复用代码）： | 11084 |
| 后端的代码行数（不包括注解行、空行和复用代码）： | 18614 |

|  |
| --- |
| **经验、教训和建议** |
| 1. 遭遇验收提前的进度风险，及时进行调整。舍弃部分优先级较低的需求，优先确保高优先级任务。 2. 进行多种形式的测试，确保产品的稳定性与性能。 3. 参考已有的设计和功能、利用现有的组件库，可以减少重复开发工作，提高效率。 4. 后端开发过程中问题和变更是正常现象，需要灵活应对并及时迭代返工，以确保后端应用符合预期功能和效果。 5. 遵循良好的Git分支管理流程，确保团队协作顺畅，代码版本控制清晰可控。 6. 必须及时进行代码评审，采用自动化代码质量检查工具，提高代码的可读性和可维护性。 7. 组员之间的协调沟通至关重要，定期开会交流进展、解决问题，建立团队合作氛围。 |

项目组各成员签字：陈启炜，黄峻涛，韦东良，叶懿芯