**迭代评估报告（简化版）**

　　　　　　　　　　　　　　　　　　评估日期：2024.07.03

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 1 | 项目名称 | UniGPT - v2.0 |
| 迭代名称 | 界面原型迭代 I | 实际起止日期 | 2024.06.24 – 2024.07.03 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 实际起止日期 | 质量 | | 1 | 调研与分析需求，编写需求文档 | 6.24-6.25 | 优秀 | | 2 | 编写《软件项目计划》 | 6.25 | 优秀 | | 3 | 前端bug修复与代码重构 | 6.26-6.30 | 优秀 | | 4 | 后端bug修复与代码重构 | 6.26-6.30 | 优秀 | | 5 | 前端新增界面设计稿 | 6.26 | 优秀 | | 6 | 华为云CodeArt探索，代码仓库迁移 | 6.26 | 优秀 | | 7 | 华为云服务器配置 | 6.27-6.29 | 优秀 | | 8 | 学习langchain4j 框架，编写学习文档 | 6.29-7.3 | 良好 | | 9 | 编写《软件架构设计》文档 | 7.1 | 良好 | | 10 | 前端测试框架探索 | 7.1-7.2 | 良好 | | 11 | 对现有项目进行简单的测试 | 7.3 | 优秀 | | 12 | 代码与文档评审，根据评审结果进行改进，编写迭代总结报告 | 7.2-7.3 | 优秀 | | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）   1. 问题：前端错误处理格式混乱，不同小组成员的代码有不同的处理方法   返工：由重构前端的成员统一规范错误处理格式，并形成规范文档   1. 问题：v1.0版本后端数据库设计较差，不满足范式，外键关联有误   返工：由重构后端的成员重新设计数据库   1. 问题：langchain4j 框架的大多数学习资料针对的是Python版本，对Java版本仅有官方的英文文档可供学习   变更：采用Java版本，学习框架时进行文档的翻译，遇到难题及时通过github issue向框架作者提出询问 | | | |
| 经验和教训：   1. 作为一个基于已有项目继续开发2.0版本的项目，代码重构非常重要，能提高后续开发的效率，避免积重难返。 2. 跟进学习langchain4j这样的前沿技术并应用于项目中，使项目更具技术亮点。 3. 参考已有的设计和功能，可以提高开发效率和减少开发的复杂度。 4. 问题和变更的发生是正常的，需要及时进行返工，确保应用达到预期的功能和效果。 5. 在开发过程中，及时进行代码评审，能够提高代码的可读性和可维护性。 | | | |