**项目总结报告**

日期：2024.06.14

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 8 | 项目名称 | NSD课表小助手 |
| 编程语言 | Java,JavaScript | 开发平台和框架 | React Native,Spring Boot |

|  |
| --- |
| **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：分层架构  设计模式：适配器设计模式、模板设计模式、迭代器设计模式、状态设计模式  2.技术方案有哪些亮点？  jaccount单点登录、事件的多视图展示（日周月视图的集成）、调休功能、工作表个性化、自然语言操作指令识别、教务处课表导入、笔记模块、导出与分享、日程提醒功能、时间冲突提醒功能、集成课程和日程的管理、首次使用帮助提示  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均完成。  单元测试：对后端的进行了测试，语句覆盖率达到99.2%。  功能测试：  完成了功能正确性测试、准确性测试。  功能正确性测试：需求文档中基本所有的需求功能都进行了实现，并根据测试规约文档对它们设计了单个功能使用的测试、功能间交互、场景法测试，从而对其正确性进行了测试。最终功能测试全部通过，有一定强壮性，符合预期  功能准确性测试：使用AI语言模型对于用户的自然语言输入进行处理，设计了大量语言测试用例，对AI识别准确度进行测试，得到准确度达90%以上。  易用性测试：  请来非本组同学进行试用，不给予真人的帮助和提示让他们自行使用该软件。测试结构发现，app设计的用户帮助很大程度上帮助了同学们快速理解app的使用方式。且app的功能基本能够满足同学对于日程管理软件的需求。    兼容性测试：  在Redmi K50 Pro，Huawei Mate 20 Pro，Nova 11，Xiaomi 12四台手机上分别运行功能测试中的场景法测试，观察其app界面显示是否出现异常，功能是否均可以正常使用。得到app界面基本没有出现异常情况，功能均可正常使用，只有部分手机因为屏幕过窄或是曲面屏分别导致周视图课表显示拥挤、屏幕边缘部分显示略不美观，基本符合预期。  性能测试：  性能测试均使用postman进行，并行压力测试方面仅起到一定参考作用，对项目的并行能力进行简单测试。  时间特性测试：对系统的核心功能函数loadDayVision，changeEventInfo等进行效率测试，串行发送请求，发送请求间隔2ms，发送200个请求，得平均响应时间均小于200ms，最大响应时间小于500ms，出错率0%，符合预期。  压力测试：设计用户操作执行脚本，100个并行用户同时发送请求，请求平均响应时间小于800ms，最大响应时间小于3s，出错率0%，符合预期。  4.是否采用大模型来辅助开发？对开发生产率约有百分之几的提升？  否 |
| **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 | | 林承亮 | 20 | 18 | 27 | 25 | 25 | 23 | | 丁牧云 | 20 | 26 | 26 | 18 | 20 | 22 | | 刘安源 | 20 | 20 | 22 | 22 | 21 | 21 | | 张佳豪 | 20 | 18 | 15 | 18 | 19 | 18 | | 赵异升 | 20 | 18 | 10 | 17 | 15 | 16 |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件规模** | |
| 前端的代码行数（不包括注解行、空行和复用代码）： | 8342 |
| 后端的代码行数（不包括注解行、空行和复用代码）： | 5915 |

|  |
| --- |
| **经验、教训和建议** |
| 1．git版本管理问题  各成员应分工明确、负责独立板块，防止merge冲突和版本更新不及时问题。我们选择使用版本控制系统（如Git）进行代码管理，建立适当的分支策略和代码合并流程。实施持续集成和持续交付，确保代码质量和稳定性。  2．组员迭代分工DDL制定问题  较大模块的开发是多位组员合作进行的，如果出现DDL制定不明确，会导致项目整体进度被拖累  3. 代码规范问题  前期没有制定完善的代码规范，导致调试和测试遇到一点问题，要提前指定完善的代码书写、命名规范。可以建立定期的代码审查制度，通过团队成员之间的互相审查和技术分享，提升代码质量和团队整体水平。  4. 需求和目标不明确问题  项目有时因为需求更改而导致返工，所以要，确保前后端开发团队对需求和目标有清晰的共识。这包括功能需求、用户体验、性能指标等方面。  5. 沟通和协作问题：  后端模块之间、前端模块之间以及前后端之间，开发需要保持频繁和有效的沟通，否者会出现接口、代码规范等等问题，我们最后选择使用沟通工具（如微信，钉钉）和协作平台（如github，apifox）来跟踪进度、讨论问题，并及时解决交叉依赖和接口定义问题，这样使得团队保持频繁和有效的沟通。  6. 测试和质量保证：  前端和后端开发过程中，测试的覆盖率和质量保证是关键因素。在上一次迭代过程中，由于测试未到位，在最后校验时多次返工。 |

项目组各成员签字：林承亮，丁牧云，刘安源，张佳豪，赵异升