**迭代评估报告**

　　　　　　　　　　　　　　　　　　评估日期：2024.5.14

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 8 | 项目名称 | NSD课表小助手 |
| 迭代名称 | 技术原型迭代 | 实际起止日期 | 2024.3.27-2024.5.14 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）   |  |  |  |  | | --- | --- | --- | --- | | No | 完成的任务 | 实现功能 | 进度 | | 1. | 确定技术栈 | 前端react native，后端采用Spring Boot框架，数据库使用MySQL实现 | 完成 | | 2. | 配置各个软件的环境 | 以react为基础，了解react native框架，并配置完毕Android Studio环境；使用IDEA完成了Spring Boot的环境配置；下载以及安装配置好了MySQL server 和MySQL workbench，并建立连接 | 完成 | | 3. | 组员分工 | 明确每个组员的工作部分：前端：丁牧云；后端：林承亮，赵异升；前后端交互：刘安源；数据库：张佳豪； | 完成 | | 4. | 撰写软件架构文档 | 完成了软件架构文档的撰写，评审，修改，确定NSD课表助手的整体架构设计 | 完成 | | 5. | 确定代码和注释规范 | 完成了代码规范的文档，详细见代码规范文档 | 完成 | | 6. | component设计分类 | 将页面分离成各种component，明确各个component的设计与功能 | 完成 | | 7. | 绘制主要界面 | 按照第一阶段Figma设计，使用Android studio 完成登陆界面，主界面，以及按钮的绘制 | 完成 | | 8. | 绘制完整的界面 | 绘制了各项功能界面：调休界面，增加事件界面，新建课表界面，笔记界面，论坛界面；并且添加了按钮的点击响应跳转功能 | 笔记，论坛界面尚未实现，其余完成。 | | 9. | 后端Spring Boot，数据库MysQL的学习 | 学习Spring Boot和MySQL的相关知识 | 完成 | | 10. | 设计数据库 | 使用MySQL workbench 设计出数据库中的Table 以及每张Table中的Column元素，以及主键，外键的设计 | 完成 | | 11. | 后端与数据库连接 | 将数据库和建立好的后端项目连接起来，并且学习了通过使用IDEA对数据进行增删改查 | 完成 | | 12. | 明确后端的接口 | 开会确定了后端需要留出的接口，需要使用的api、人工智能模型等等 | 完成 | | 13. | 前后端交互 | 前端设计根据后端一些需求进行一些修改，并且可以实现前端发出请求后端进行响应，返回信息 | 完成 | | 14. | 撰写迭代评估报告 | 对迭代评估报告初稿进行评审，评估当前软件开发进度，根据开发情况对于迭代计划、分工、软件功能实现做调整、优化 | 完成 | | 15. | 修改vision文档和软件架构设计文档 | 根据调整、优化的结果修改第一阶段迭代的vision文档、第二阶段迭代的软件架构文档 | 完成 | | 16. | 测试实现 | 制定测试计划，选定测试工具与测试单元，使用测试工具、测试单元对于软件的功能、性能、安全性进行测试 | 完成 | | 17. | 编写迭代评估报告 | 编写迭代评估报告 | 完成 | | 18. | 答辩PPT制作 | 答辩PPT制作 | 完成 | | 19. | 对于下一阶段进行展望、提前制定迭代计划 | 对于下一阶段进行展望、提前制定迭代计划 | 完成 | | | | |
| 评审/测试的结果：（执行了哪些评审和测试？评审和测试的结果如何？）  **代码规范评审：**  代码规范文档已经详细制定，确保团队成员在开发过程中遵循一致的代码风格和注释规范，提高了代码的可读性和可维护性  **组件设计评审：**  页面已经成功分离成各种component，并明确了各个component的设计与功能，确保了前端开发的模块化和可复用性。  **界面设计评审：**  主要界面已经绘制完成，包括登陆界面、主界面和各项功能界面，并且实现了按钮的响应跳转功能，为用户界面的基本实现奠定了基础。  **后端与数据库评审**  成功学习了Spring Boot和MySQL相关知识，设计了数据库结构，并且完成了后端与数据库的连接，确保了数据的持久化和安全性。  **后端接口评审**  团队开会确定了后端需要留出的接口，并明确了需要使用的API（如jaccount接口）和人工智能模型，为前后端交互提供了明确的接口定义。jaccount接口和人工智能模型不会在本次迭代中落地。  **前后端接口测试：**  使用apifox和postman进行前后端接口测试，后端开发时使用postman进行接口测试，前端开发时使用apifox从云端获取虚拟数据并进行运行，实现前后端的接口测试。  **前后端交互评审**：  小组成员线下开会，将分别测试完毕的前后端进行交互并成功运行。前后端对于一些对于文档的误解而分别进行了微调，最终确保了前端发出请求后能够得到后端的响应，并正确处理返回信息，完成NSD课表小助手的基本功能。  **迭代评估报告编写**  对迭代初稿进行了评审，根据当前软件开发进度对迭代计划、分工和软件功能进行了调整和优化，确保了项目的顺利推进。 | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）  **问题：**   1. 在环境配置过程中，可能会遇到一些与操作系统或软件版本相关的兼容性问题，例如React Native在某些特定平台上的配置可能不够顺利，需要额外的解决方案。 2. 小组项目没有真机的同学只能在Android Studio上进行运行，有安卓真机组员少，而as配置react native环境问题极多，导致小组项目在部署之前只能在少部分组员那里完整运行。 3. 在界面设计和组件分类过程中，团队成员可能会对于某些功能的界面设计产生分歧，需要及时沟通和调整。 4. 在前后端交互过程中，可能会出现请求参数或响应数据格式不一致的问题，需要进行统一规范和调整。   **变更：**   1. ****在软件架构设计阶段，可能会根据实际开发情况进行一些微调和优化，例如调整系统模块之间的交互方式或数据流向。 2. 在界面设计过程中，根据用户反馈或团队讨论，可能会对某些界面进行修改或增加新的功能，以提升用户体验和功能完整性。 3. 在后端接口设计阶段，可能会根据实际需求增加或修改一些接口，以适应业务需求的变化或新功能的添加。   **返工：**   1. 在前后端交互过程中，如果出现了数据传输或接口调用方面的问题，可能需要对相关代码进行返工，以确保前后端能够正常通信和数据交互。 2. 在界面设计和组件分类过程中，如果发现某些组件功能设计不合理或界面效果不佳，可能需要进行返工，重新设计或优化相应的组件。 | | | |
| 经验和教训：  **经验：**   1. **加强沟通与协作**：在项目初期就建立良好的沟通机制，确保团队成员之间能够及时交流、协调和解决问题，可以减少因沟通不畅而导致的误解和延误。 2. **详细规划与准确评估：**在项目启动阶段充分规划和评估，包括环境配置、软件架构设计、任务分工等方面，确保项目目标清晰、可行性高，避免后期因需求变更或误差而带来的额外工作量。   3. **持续学习与知识分享：**团队成员应持续学习和积累相关技术知识，及时分享新的技术进展和解决方案，提高整个团队的技术水平和应对问题的能力。  **教训：**   1. **灵活应对变更与风险：**我们在项目开发初期，对于前后端的接口问题没有商量好，导致有所返工，所以我们之后要做好风险预测，提前预知可能出现的问题，及时应对。   2. **严格质量控制与测试：**项目初期时，大家代码风格都不相同，导致后期在协同开发时，大家对于彼此代码难以看懂，所以在开发过程中注重代码质量、界面设计和功能实现的质量，同时建立完善的测试机制，确保软件的稳定性、性能和安全性。 | | | |