**迭代计划**

　　　　　　　　　　　　　制定日期：2024.3.27

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 8 | 项目名称 | NSD课表小助手 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2024.3.27-2024.5.14 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 参考第一次迭代时助教的意见，修改figma的页面显示，同时将修改方向记录到文档中 | 2024.3.21-2024.3.31 | 丁牧云，林承亮 | | 2 | 定期了解市场需求变更情况，在市场需求出现较大变更时及时告知组员。若出现类似情况，开会商议是否需要对app功能进行修改、新增与删除 | 2024.3.21-2024.5.7 | 赵异升 | | 3 | 以react为基础，了解react native框架，并配置完毕Android Studio环境 | 2024.3.25-2024.3.31 | 林承亮，丁牧云，刘安源 | | 4 | 开会确定第二阶段的具体分工，选择架构风格，确定语言、框架、工具；明确各阶段ddl，确定功能实现顺序，初步描绘出多个架构视图雏形 | 2024.3.28 | 全体组员 | | 5 | 将多个架构视图进行绘制 | 2024.3.28-2024.4.7 | 丁牧云 | | 6 | 撰写软件架构文档 | 2024.4.7-2024.4.8 | 赵异升 | | 7 | 评审软件架构文档 | 2024.4.8 | 林承亮，刘安源，张佳豪 | | 8 | 根据评审结果对架构视图进行优化，文档进行修改 | 2024.4.8-2024.4.11 | 丁牧云，赵异升 | | 9 | 设计代码书写和注释规范（可能直接采用google编码规范） | 2024.4.1 | 张佳豪，刘安源 | | 10 | 完成前端主要页面的绘制，在github上共享component供组员使用 | 2024.4.1-2024.4.14 | 林承亮，丁牧云，刘安源 | | 11 | 进行所有页面的绘制完善，并完成页面之间的跳转等操作，添加必要的功能模块，和后端进行简单交互，基本完成软件架构 | 2024.4.15-2024.4.22 | 林承亮，丁牧云，刘安源 | | 12 | 进行后端+数据库即spring+MySQL的环境配置与初步学习、熟悉。 | 2024.3.25-2024.4.1 | 张佳豪，赵异升 | | 13 | 开会确定要后端需要留出的接口，需要使用的api、人工智能模型等等 | 2024.4.4 | 全体组员 | | 14 | 书写基础的后端框架、熟悉数据库使用，能够实现简单功能 | 2024.4.1-2024.4.14 | 张佳豪，赵异升 | | 15 | 先接触后端的组员帮助其他组员快速上手后端和数据库，加入开发 | 2024.4.14-2024.4.22 | 全体组员（由张佳豪、赵异升带头） | | 16 | 对于当前的程序进行测试，找出当前app的优缺点，对应vision文档查看技术要求的完成情况 | 2024.4.18-2024 | 赵异升，刘安源 | | 17 | 撰写迭代评估报告初稿 | 2024.4.14-2024.4.15 | 林承亮，赵异升 | | 18 | 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 | 2024.4.17 | 全体组员 | | 19 | 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 | 2024.4.17-2024.4.18 | 林承亮，刘安源 | | 20 | 根据调整、优化分工后的分工继续工作（属于大迭代阶段中的第二分阶段，详细分工会在下面列出） | 2024.4.17-2024.5.10 | 全体组员 | | 21 | 前端页面根据后端、数据库设计以及app功能需求等进行微调 | 2024.4.17-2024.5.7 | 林承亮，丁牧云 | | 22 | 根据模块化划分以及后端接口的需求，分工进行后端和数据库的书写 | 2024.4.17-2024.5.7 | 全体小组成员 | | 23 | 制定测试计划，选定测试工具与测试单元 | 2024.4.22-2024.4.28 | 张佳豪，丁牧云 | | 24 | 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 | 2024.4.28-2024.5.1 | 张佳豪，丁牧云 | | 25 | 撰写测试情况报告 | 2024.4.29-2024.5.1 | 张佳豪 | | 26 | 根据测试情况对软件代码进行改进 | 2024.5.1-2024.5.10 | 全体组员 | | 27 | 编写迭代评估报告 | 2024.5.2-2024.5.12 | 全体组员 | | 28 | 答辩PPT制作 | 2024.5.7-2024.5.13 | 丁牧云，张佳豪，林承亮 | | 29 | 对于下一阶段进行展望、提前制定迭代计划 | 2024.5.10-2024.5.14 | 全体组员 | | | | |
| 预期成果：   1. 架构视图 2. 软件架构文档 3. 小组编程规范 4. 关键算法逻辑与实现说明（如果需要） 5. 软件架构搭建源代码 6. 软件在安卓端手机安装的安装包 7. 软件使用说明文档、视频 8. 技术原型测试计划与测试情况报告文档 9. 迭代评估报告 | | | |
| 主要的风险和应对方案：  1、技术选型风险：  风险描述：在选择架构风格、语言、框架和工具时，可能存在选择的工具、架构不当导致后续开发困难或性能问题的风险。  应对方案：在做出技术选型决策之前，进行充分的技术调研和评估，考虑项目需求、团队技术能力等因素。可以提前咨询软工学长学姐、助教、老师，了解往年安卓app开发的常用技术选型，最终结合小组的app特性，对于技术选型进行决策。  2、缺乏足够的技术储备风险：  风险描述：团队成员因为基本都是从零开始学习react native框架，ant design组件使用，spring后端结构，mySQL数据库搭建，java等语言，缺乏足够的技术储备，可能因此无法有效地完成技术原型的搭建和实现。  应对方案：提前进行技术培训和知识共享，让部分组员提前对后端和数据库开发进行学习，而不是所有人都同步进行类似的任务。让部分先学的人带动其他成员快速上手确保团队成员具备必要的技术能力。必要时也可以向学长学姐、助教老师等高技术人员求助，帮助团队成员提升技术水平。  3、需求变更风险：  风险描述：在迭代过程中，需求可能会发生变化，导致技术原型的设计和实现需要进行调整。  应对方案：建立灵活的变更管理机制，及时响应需求变更。让一位小组成员始终关注当下的市场需求，在出现较大变更时与组员沟通，确保小组能够及时了解需求变更的情况，并评估其对技术原型的影响，更快速地适应需求变化，不在无用的功能块上浪费太多精力，减少返工量。  4、技术集成风险：  风险描述：在搭建软件架构和实现功能模块时，可能存在技术集成方面的困难，导致系统组件无法有效地协同工作。  应对方案：采用模块化设计和接口规范化的方式，全组成员开会确定对于模块的严格划分，接口的功能和使用规范化，降低技术集成的复杂度。同时，进行适当的技术预演和原型验证，在分开开发的同时，也保证能够一定时间后将各个组件组合起来确定能够使用，保证各个组件之间的兼容性和稳定性，减少最后阶段大规模返工的可能性。并在迭代中期和迭代末期的两次集成阶段进行全面的测试，及时发现和解决集成问题。  5、测试覆盖不足风险：  风险描述：测试工作可能没有覆盖到所有的功能和技术要求，导致无法发现隐藏的缺陷或性能问题。  应对方案：制定全面的测试计划，包括功能测试、性能测试、安全测试等，确保覆盖到所有重要的功能和技术要求。同时，采用自动化测试工具，提高测试效率和覆盖率。 | | | |