**迭代计划**

　　　　　　　　　　　　　制定日期：2023.2.26

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 8 | 项目名称 | 生活记录仪 |
| 迭代名称 | 界面原型迭代 | 计划起止日期 | 2.27-3.14 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 学习React Native框架，确定使用的UI库 | 2.27-3.1 | 全体小组成员 | | 2. | 制作前端界面的基本模板 | 3.1-3.3 | 杨征，闫景升 | | 3. | 发布调查问卷，调查用户需求 | 3.1-3.3 | 刘洋，张馨匀 | | 4. | 小组讨论分析需求结果，根据结果确定软件具体功能 | 3.3-3.6 | 全体小组成员 | | 5. | 根据功能设计UI界面 | 3.6-3.9 | 刘洋，张馨匀 | | 6 | 代码实现UI界面 | 3.9-3.14 | 杨征，闫景升 | | | | |
| 预期成果：  [列出本迭代计划交付的文档、模型、源代码、安装包等]  **文档：**   1. 提交Vision文档，明确产品定位和受众情况，结合市场情况进行相关分析，同时根据相关需求确定产品总体效果和相关特性。 2. 产品相关资料：如前期问卷调研数据和有关文献的收集。 3. 部分用户文档：结合需求和项目进展情况完成部分用户文档的编写。 4. 迭代记录和评估报告：根据每次迭代的计划和实际完成情况，以及迭代目标的完成情况完成迭代评估报告，并进行相关的总结反思。   **模型：**   1. 前端界面草图设计 2. 项目整体架构思维导图   **源代码：**  在界面原型迭代中，我们计划在编码层面完成项目界面的构建，让程序能在前端响应一定的事件，并确定界面风格。源代码的更新会在GitHub上同步。 | | | |
| 主要的风险和应对方案：  [分析当前项目风险，列出最大的3~5个风险，自高到低排列，并列出每个风险缓解和应急的措施。注：随着项目的进行，风险会不断变化，故每个迭代要重新对风险评估和控制。]  1. 需求风险  在项目开始之初，没有完全了解项目的需求，导致项目已经在实施过程中，需求仍然在不断的变化。并没有详细说明具体需求细节，需求含混不清。在项目进行过程中，不断添加额外需求。  解决方法：在项目开始之初，就尽量完全了解需求，清晰的了解需求细节，对重要的部分做好评估。  项目进行时，模块化项目，以便在有额外需求需要实现时，可以将其以模块的形式实现，尽快添加至主项目中。  添加富裕时间来解决额外的需求，尽量不把项目计划完全定在截止日期前后。  2. 人员风险  组员与组长之间缺乏沟通，导致组长无法获取足够信息进行决策，影响全局计划。组员之间缺乏沟通，导致项目模块重复实现，影响效率，或者部分模块未实现，影响最终项目呈现。  解决方法：每周组织讨论会，讨论各自的项目进度。并在讨论会上积极进行项目沟通，组员与组长，组员与组员之间多交流当前进度以及短期计划目标，统筹安排项目具体实现细节。  3. 技术风险  可能会使用到未经测试过的组件代码，陈旧的、即将被淘汰的开发技术和工具，某些库之间并不兼容，导致程序无法运行。库缺少使用说明文档，难以使用。  解决方法：尽量使用主流的、稳定的和说明文档齐全开发工具和开源代码库。  4. 过程风险  进行大量重复性的无用工作，导致项目进展比预期的慢。开发审查不严格，项目存在大量风险与错误。  解决方法：每次代码提交都要进行一定的测试。组长要做好统筹安排规划，避免大量重复性工作。 | | | |