**迭代计划**

　　　　　　　　　　　　　制定日期：2022/3/16

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 8 | 项目名称 | 牙科医院管理系统 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2022/3/15- 2022/4/27 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 技术原型迭代计划编写 | 2022/3/15-2022/3/15 | 李昱翰，并由全组成员共同审核并修改 | | 2 | 通过小组讨论初步建立本项目的概念模型，进行项目的概念建模 | 2022/3/16-2022/3/17 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 3 | 确定项目将需要的基本框架（如前端的React框架）以及语言类型（如java等），并选择合适的开发工具以及编程规范（如Google编程规范等） | 2022/3/17-2022/3/18 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 4 | 通过小组讨论，确定本项目在本次迭代过程中要实现的各个关键功能的具体实现细节，并通过集中讨论确定其对应的算法实现思路。 | 2022/3/19-2022/3/21 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 5 | 进行软件架构的搭建 | 2022/3/22-2022/3/23 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 6 | 开始进行具体技术原型的实现，并主要先实现优先级较高的几个功能对应的关键算法逻辑。 | 2022/3/24-2022/4/14 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰  （之后还会对每个组员的具体任务进行分工） | | 7 | 对已经完成的技术原型部分进行原型测试，并对其中出现的技术性问题进行改进 | 2022/4/14-2022/4/21 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 8 | 撰写并评审软件架构文档，并根据上一阶段的测试结果来撰写《迭代评估报告》 | 2022/4/22-2022/4/23 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 9 | 结合本次技术迭代的进展等情况，对之前的界面原型迭代的结果进行进一步的改进以及优化。 | 2022/4/24-2022/4/27 | 全组成员：  宋金骋  时宪  谷金龙  李昱翰 | | 10 | 将相关的代码以及框架上传至github中 | 2022/4/27-2022/4/27 | 时宪 | | | | |
| 预期成果：  需交付的文档:《技术原型迭代计划》、软件架构文档、《迭代评估报告》  需交付的模型、代码等：本次技术迭代完成的关键功能以及属性的实现算法以及对应程序，同时也应包括进行模型测试时所用到的代码。  安装包：暂无 | | | |
| 主要的风险和应对方案：   1. 风险：关键功能算法实现面临的技术风险（技术风险）   应对方案：首先通过小组的仔细讨论已确定最合适的实现算法类型，并将不同的关键功能的实现分配给不同的组员以减少个人开发压力；应频繁的进行性能测试，在算法实现的每个阶段，都应通过组内自编测试的方式来测试每一个新实现好的功能的完备性与正确性。   1. 风险：进行技术迭代时的时间风险（进度风险）   应对方案：为应对迭代的时间风险，有必要在原型实现阶段进行详细的时间分配。通过每周两次讨论或线上会议的方式及时的汇报个人实现进度，并根据每个组员的不同实现进度进行任务的再次划分以最大限度的提升在实现阶段的团队工作效率。另外，小组成员在开发过程中应该统一实行已经选定的编程规范，以保证代码质量的统一性，这也保证了小组成员在进行开发时不会出现由于规范不同致使代码之间的兼容性以及可读性差而导致浪费不必要的时间的情况。   1. 风险：性能不满足要求导致的风险（性能风险）   应对方案：在项目的实现过程中，可能会出现由于算法虽然实现了功能但是时间复杂度不好而导致不满足性能要求的问题。为应对此类型的问题，应注重：  ①前期的性能规划：在小组内部进行算法实现相关的讨论时，应额外注重性能问题，如并发数要求等问题。在进行规划时应在基本模型之外添加一些额外的性能优化模型，（如延迟加载等），并且最好不要在实现过程中中途修改性能优化方案，防止浪费不必要的时间。  ②进行性能测试：在进行性能测试时应尽可能的搜集资源去模拟实际的使用环境，并应在实现的每个阶段都进行相应的性能压力测试（如对并发数的测试等）。 | | | |