**迭代计划V2.3**

　　　　　　　　　　　　　制定日期：10.17

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 17 | 项目名称 | 云作业平台 |
| 迭代名称 | 迭代二：技术原型搭建 | 计划起止日期 | 10.13--11.16 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1.1 | 课程后端开发 | 10.13-10.14 | 杨亘 | | 1.2 | 确定后端整体架构 | 10.13-10.14 | 所有人 | | 1.3 | 登录注册页面对接后端 | 10.13-10.14 | 杨亘 | | 1.4 | 通知页面对接后端 | 10.13-10.14 | 太靖文 | | 1.5 | 包装邮件服务接口 | 10.13-10.14 | 周峰 | | 1.6 | 作业主页对接后端 | 10.13-10.14 | 秦梁 | | 1.7 | 概念图编写 | 10.13-10.14 | 刘兆翰 | | 2.1 | 服务器修复，环境配置 | 10.15-10.16 | 杨亘 | | 2.2 | 课程主页，课程详情页对接后端 | 10.15-10.16 | 杨亘 | | 2.3 | 作业主页对接后端 | 10.15-10.16 | 太靖文 | | 2.4 | 包装发短信服务接口 | 10.15-10.16 | 周峰 | | 2.5 | 作业详情页对接后端 | 10.15-10.16 | 秦梁 | | 2.6 | 迭代计划编写 | 10.15-10.16 | 杨亘，刘兆翰 | | 3.1 | 公告页面，课程作业页面前后端对接 | 10.17-10.18 | 杨亘 | | 3.2 | 用户信息等页面前后端对接 | 10.17-10.18 | 太靖文，秦梁 | | 3.3 | 增加用户头像， | 10.17-10.18 | 周峰 | | 3.4 | 概念图和迭代计划更新修改 | 10.17-10.18 | 刘兆翰 | | **3.5** | 优化接口，增加数据分页功能 | 10.17-10.18 | **杨亘** | | **4.1** | 完成提交情况前后端配合 | 10.19-10.20 | **秦梁** | | **4.2** | 完成一个完整的用例分析 | 10.19-10.20 | **刘兆瀚** | | **4.3** | 完成excel导入学生的功能 | 10.19-10.20 | **太靖文** | | **4.4** | 添加储存拿取用户头像的后端 | 10.19-10.20 | **周峰** | | **4.5** | 完善批改页面 | 10.19-10.20 | **杨亘** | | **5.1** | 完善excel导入学生的功能 | 10.21-10.22 | **太靖文** | | **5.2** | 完善课程统计信息页面 | 10.21-10.22 | **杨亘** | | **5.3** | 参与用例分析的编写 | 10.21-10.22 | **杨亘，刘兆瀚** | | **5.4** | 完成课程管理学生的界面 | 10.21-10.22 | **周峰** | | **5.5** | 用例图的编写与修改 | 10.21-10.22 | **刘兆瀚** | | **5.6** | 完善提交情况和前后端整合 | 10.21-10.22 | **秦梁** | | Later | 前后端及数据库完全对接 | 10.13-11.10 | 所有人 | | Later | 架构视图设计 | 10.19-10.24 | 所有人 | | Later | 确定并实施编程规范 | 10.19-11.5 | 所有人 | | Later | 设计关键算法并 | 10.25-11.1 | 所有人 | | Later | 编写软件架构文档 | 11.2-11.15 | 所有人 | | Later | 测试技术原型 | 11.5-11.10 | 所有人 | | Later | 迭代报告编写 | 11.10-11.15 | 所有人 | | | | |
| 预期成果：  文档：多版本迭代计划、软件架构文档、编程规范文档、迭代评估报告  模型：use case模型图、uml概念类图、VOPC用例实现图、多个软件架构视图  源代码：前端所有页面的源代码，后端三个微服务的源代码、前后端性能测试的代码 | | | |
| 主要的风险和应对方案：  **1.与人工智能相关的技术运用不熟练，可能会在图像、语言识别技术上遇到瓶颈**  解决方案：加紧学习人工智能相关的知识和技术（如本学期的人工智能课），或在github上寻找可用的轮子。  2**.高并发低时延的项目要求。当课程，作业，用户等数据量急剧上升后，高并发下可能不能达到规定的响应时间。**  解决方案：首先先在多个角度尝试优化，优化后端数据库的schema；减少sql操作的成本。优化前端的请求调用，减少琐碎的请求；使用k8s等架构管理微服务，提升速度；采用分页pageable等技术，降低时延。若预算允许，还可以购买配置更高级的服务器。  **3.服务器使用时可能因为各种各样的原因出现崩溃、无法连接、内存爆满、CPU占用率过高等状况，导致所有工作无法继续测试运行**  解决方案：在安排计划时预留出足够的时间，把修复服务器问题花费的时间计算在内；保持本地环境配置准确无误再到服务器上进行部署；利用安全组放行指定ip，使用强密码，更换默认端口等措施防止服务器被木马侵入（之前出现过被来自荷兰的挖矿病毒侵入的惨痛经历）。记下环境配置的步骤，积累经验，实在出现无法解决的问题时快速进行服务器重装。  **4.建模方面，由于软件架构搭建不熟练，可能无法设计出完美的软件架构图**  在遇到新问题时多查阅书本，查看技术网站和求助助教，小组成员合作对于设计的软件架构进行相互评审  **5.时间风险：项目功能性需求多，页面元素复杂，需要大量时间和精力投入。风险在于小组成员是否能够齐心协力，投入在项目中，稳定的推进云作业平台项目。**  解决方案：组长以2-3天为单位分发布置任务，并实施督促。每周选定一天在学校内的小组自习室讨论，同步进度并进行互相评审。  **6.人性化和可用性的项目要求，需要确保第一阶段所设计界面与功能是用户友好的，否则潜在用户（中小学教师、学生）可能会在使用时感到棘手**  解决方案：在第一阶段设计的界面的基础上做进一步改进，细化功能页面的分类，增加导航栏说明文字，对用户界面做一定的美化，如增加立绘等等。 | | | |