**迭代计划**

　　　　　　　　　　　　　制定日期：2023/11/14

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 2 | 项目名称 | 学生选课系统 |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2023/11/15~~2023/12/20 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 技术方案设计   1. 选择架构风格，确定语言、框架、工具 2. 设计多个架构视图 3. 撰写和评审软件架构文档 | 11/15 – 11/20  11/15 – 11/16  11/17 – 11/18  11/19 – 11/20 | 董俊良、张子桓、唐乐轩 | | 2 | 实现技术原型   1. 搭建软件架构 2. 实现用户注册和登录的程序 3. 实现学生查询、选退课程和评教的程序 4. 实现教师创建、查询课程和给分的程序 5. 实现管理员开启和关闭选课的程序 6. 实现管理员删除课程、结课、设置课程时间和地点的程序 | 11/21 – 12/10  11/21 – 11/22  11/23 – 11/26  11/27 – 11/30  12/01 – 12/04  12/05 – 12/08  12/09 – 12/12 | 董俊良、张子桓、唐乐轩 | | 3 | 测试技术原型  测试是否达到需求中预定的技术要求，并根据测试结果进行改进 | 12/13 – 12/18 | 张子桓、唐乐轩 | | 4 | 编写《迭代评估报告》  评审记录和测试记录、开发总结 | 12/19 – 12/20 | 董俊良、唐乐轩 | | | | |
| 预期成果：  项目版本控制与更新日志完善，最终版实现所有目标功能。  完成Django与python搭建下的项目后端控制，并实现HTML+CSS的前端版面  完成软件的整体框架，初步实现学生、教师和管理员的各项功能。  程序内部算法实现没有纰漏，并严格封装，项目测试后应表现出较好的性能。 | | | |
| 主要的风险和应对方案：   1. 程序内部新功能的增加可能会导致内部算法变得冗余难读，以致难以修改更迭。   解决方案：加强项目版本管理与搭建思路的设计，在项目结构设计阶段尽可能完善的预估未来功能，以便于减少功能新增的次数。   1. HTML+CSS可能学习效果不佳，导致界面模型简陋难以调控，或无法实现需要的前端功能。   解决方案：在前端修改方面，以绝对的接口功能实现为主要目标，其次再对界面窗口样式进行修正。 | | | |