**组号：26**

**成员：**

**2021211183 郭泽远**

**2021211184 郭晨旭**

**2021211174 叶沛鑫**

|  |  |  |
| --- | --- | --- |
| **验收内容** | | **完成情况** |
| **系统架构（采用的编程语言、架构采用BS/CS/Android）** | | 整体BS架构  前端html+vue+js；  后端java语言，采用Spring boot+Spring Cloud微服务架构 |
| **课程日程管理** | **完成的基本功能描述** | 完成课程和考试的增删查改，所有操作均有权限与数据校验，添加和修改时有时间的冲突检测。  通过合理设计内存与数据库管理，保证所有同学课程信息的同步性，并且实现了管理员对组内同学的批量添加修改课程；  另外还完成了在课程开始前一个小时进行提醒，以及提醒第二天课程的功能。 |
| **核心算法描述、算法优缺点及性能** | 自己实现了泛型的hashmap，查询课程信息，及根据用户id查找对应课程的时间复杂度均为O（1）；  检测冲突时查询该时间段内所有课程，进而判断有误冲突，复杂度为O（1），日程检测提醒功能类似。具体复杂度及源代码见ppt  排序算法实现了随机枢轴的快速排序，时间复杂度为O（nlogn）。 |
| **课外活动日程管理** | **完成的基本功能描述** | 完成课程和考试的增删查改，所有操作均有权限与数据校验，添加和修改时有时间的冲突检测。  通过合理设计内存与数据库管理，保证所有同学课程信息的同步性，并且实现了管理员对组内同学的批量添加修改课程；  用户可以自定义活动类型，并且根据活动类型或者时间查询当天活动，并在返回时根据起始时间排序；完成了每天晚上提醒第二天活动的功能；  完成了添加活动时冲突检测的功能，并且在发生冲突是提供三个可选的时间。 |
| **核心算法描述、算法优缺点及性能** | 查询、检测冲突的方法同上面课程管理的部分。  个人活动查询可用时间是对线段树查询6：00-22：59的所有日程，如果找到空闲时间则记录选取，时间复杂度为O（n）（n为该用户的日程数量）；  集体活动是查询集体内每个用户的空闲时间，使用map记录并排序，选取三个空闲人数最多的时间段，时间复杂度为O（n）（n为同组用户的日程数量）。 |
| **临时事务日程管理** | **完成的基本功能描述** | 用户可以自定义临时事务类型，并且根据这个类型或者时间查询当天活动，并在返回时根据起始时间排序；  完成了每天晚上提醒第二天临时事务的功能；完成冲突检测功能，不会与课程和活动冲突，并且能够让多个临时事务同时发生。 |
| **核心算法描述、算法优缺点及性能** | 同课程管理部分，为hash+线段树，并有快排等公共算法 |
| **日程导航** | **完成的基本功能描述** | 完成了点到点导航  完成了途径多点导航  完成了根据模拟时间过程中，下一个小时/下一天的课程及活动，自动规划导航路线  完成了导航的可视化，使用真实校园地图，标点丰富，道路丰富，并有动画演示 |
| **核心算法描述、算法优缺点及性能** | 点到点导航，采用dijstra+缓存加速  途径多点导航，根据需要途径点的数量，分情况使用了状压dp求哈密顿回路（精确解），与bfs求近似解两种方法。  算法及其复杂度分析详见ppt |
| **用户数和日程时长、模拟系统时钟** | | 用户数任意(测试用户及日程已大量创建，运行无误)，可多用户并发访问。购买了服务器及域名并部署程序，已经可投入实际使用  日程时长（系统最小时间单位）以分钟为单位，检测冲突时提供的备选时长单位为小时，模拟系统时间可以精确到毫秒级  模拟系统时钟可以由用户自由操作，每个用户一个单独的时钟。可调节时间倍率、暂停恢复模拟、设置模拟时间，以及倒转时间模拟等，功能丰富 |
| **日志功能** | | 采用slf4j+log4j，记录了用户状态变化及其操作(谁，什么时候，做了什么，得到/造成了什么)。并且可自动根据日期进行打包存档。 |
| **选做功能（功能简介、核心算法描述）** | | 完成了导航功能的图形界面、以及课程表图形界面，采用vue+js（具体可以直接看演示效果） |

注：

1. 闹钟为独立的功能，不与课外活动和临时事务绑定，用户可以自己设定指定时间的闹钟，也是使用map和线段树存储，增删改查时间复杂度均为O（1）
2. 考试也为独立日程，不与课程绑定，管理员可以自定义考试。
3. 以上算法的优点均为时间复杂度低，通过结合项目实际情况来控制数据范围，得到了接近常数的时间复杂度；即便少数的遍历算法（如字典树的查找）仍然可以限制操作数在十来次
4. 快排使用了随机枢轴，避免了分配不均导致时间复杂度退化到O（n^2）的情况发生，大大提高了算法效率