# QG工作室暑期实习生周记

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名： 李浩 | 组别：后端 | 年级：2021级 | 周次：第六周 |

|  |  |
| --- | --- |
| **第六周周记** | |
| **生活随记** | 这是最后一周了，临近暑期训练营结束，相应地也变忙了，我要顾着两个项目。好想放假。 |
| **学习**  **开发**  **比赛**  **情况** | * 项目管控平台   2022.8.15  1.继续寻找redis缓存失效的原因。最后发现是在http执行完后，有一个代码是删除redis的key的，所以导致redis缓存找不到以token的key值  2.对接前端移动端，细化功能。  3.用户被冻结后，会在user表中记录，但当冻结时间截止，应立即更新表单。利用定时器，自动更新表单。功能实现  4.强制用户登出功能，通过用户名与redis缓存中的信息匹配，若用户名相同，则删除该key值  5.查看项目对应的监控者，并撤销其监控身份  6.与后台进行接口联系  2022.8.16  1.冻结时间更新bug修复  2.发布项目、审核项目、用户强制下线、冻结用户  3.项目审核功能  4.项目冻结功能  2022.8.17  1.邀请成功发布者、申请监控  2.服务端日志信息  3.完成Js错误的分类别查询  4.完成url的js错误查询  2022.8.18  1.完成消息功能  2.完成资源错误的路径显示  3.完成资源错误的数量统计  4.完成用户消息处理（查看自己发布的申请信息），申请消息的同意拒绝，前端js信息展示  2022.8.19  1.完成资源错误统计  2.进行多线程优化  3.完成性能分析  4.完成各种Error接口  5.解决了客户端依赖在已有全局异常处理的情况下，无法截取异常的问题  2022.8.20  1.最终版本  2.完成管理员日志功能   * 算法可视化平台   2022-8-14  1.与人工智能组沟通协商好数据表的结构  2.建立好项目框架，数据库  2022-8-14  1.完成算法列表接口  2.完成获取点集、点间关系接口  3.完成部署  2022-8-17  1。针对已有接口只能处理小数据量的问题进行优化  2.提供上传文件接口  3.与人工智能组提供的接口对接  2022-8-19  使用多线程查询，进一步优化查询速度  1. 通过算法名、数据样例、数据密度进行数据表命名，主要分为两类：点集和邻接矩阵集。均采用null进行时间段的分割。  2. 查询：先查询null记录获取分割点数组，后进行分页查询，以时间段为粒度封装数据。查询慢时考虑使用多线程分配任务查询。  封装：将数据分装成前端要求的格式，如添加标识、将邻接矩阵分割成点间关系数组等。  接口：提供了三种接口（查询全部、查询指定时间段、指定时间段数量平均查询），其中第三个接口是为了在尽量不影响算法过程展示的前提下，解决数据量过大，响应时间过长的问题；采用了抽帧算法，从所有时间段中平均取出指定数量的时间段，即所谓抽帧。  抽帧：  1. 先从总数中排除首尾两个时间段的数据；  2. 总数/目标，并向下取整得到步长  3. 根据步长平均取出指定时间段数量的数据  4. 从取到的数据，随机剔除超出目标值的部分  5. 加回首尾两个时间段的数据 |
| **一周总结** | 关于算法可视化平台，一开始拿到这个任务还以为很简单，只需要CRUD，结果却耗费了不少时间，因为有各种繁琐的需求。我认为主要是我们后台和前端一开始不太懂这些算法、算法的目的还有如何可视化出算法的过程，导致了我们改了很多次代码。  好在几天后我们确定了一份稳定的方案。功能完成后，我也开始不断优化代码，一开始这个系统甚至只能处理小数据量，数据多一点就需要极长的时间，甚至过多数据导致内存不足而崩溃。到最后，我使用了抽帧算法和多线程查询，前端也在保持视觉效果的基础上舍去了一些数据，这些问题都解决了。  这么一个流程下来，我还是有不少收获，也因为优化性能而体验到成就感。  关于项目管控平台，本周处理了客户端的依赖的一些问题，并部署在别的项目进行测试，与服务端对接。还有服务器nginx的处理，相较于中期，这次多了lua配置和监控信息的通信，麻烦了一些，而且我自己的服务器内存不足以支撑多个服务及程序，只好部署在工作室的服务器上。 |
| **存在问题**  **未来规划** | 相关功能还需要花时间，文档也需优化。 |
| **导师评价** |  |