#### 系统安全设计

1. 教师和助教不提供注册功能，教师账号需在数据库中手动添加，助教账号可以由教师通过新增功能添加，也可以直接在数据库中添加。
2. 数据库可以自动定期备份，每天0点进行一次备份，数据出现丢失或其他异常后可以通过备份恢复，当然也可以进行事务回滚，在操作发生异常后回滚数据库操作，保证数据的安全性和完整性。
3. 对用户的密码使用MD5或SHA加密，可以避免密码被直接截获破解。
4. **使用正则表达式过滤传入的参数，**防止SQL注入
5. 后端添加了拦截器后设定拦截规则，获取请求主机IP地址,如果通过代理进来，则透过防火墙获取真实IP地址，在获取用户真实IP地址后，如果某ip的访问太过频繁，则用拦截器拦截掉IP并将其加入黑名单。
6. 通过RBAC进行权限管理，所有用户不能越权操作不能访问权限以外的数据。
7. 前端进行数据校验后，后端对前端参数进行校验，前端验证减少了对服务器的不必要的不合法的请求，后端校验保证了对于一些敏感数据比如用户登录数据进行了双重验证，另外可以用许多工具来绕过前端，修改前端的方式避开js的处理验证，所以后端的验证可以保证安全性

系统权限设计

安卓端

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 权限 | 登陆 | 注册 | 查看分数 | 组间评分 | 个人中心 | 组内互评 | 查看成员信息 | 寻找结对作业队友 |
| 学生 | √ | √ | √ | √ | √ | √ | √ | √ |
| 老师 |  |  |  |  |  |  |  |  |
| 助教 |  |  |  |  |  |  |  |  |

Web端

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 权限 | 登录 | 分数查询（本人） | 作业提交 | 助教管理 | 评分细则管理 | 答辩评分 | 博客评分 | 学生成绩管理（增删改查） | 学生管理 | 公共统计信息 |
| 学生 | √ | √ | √ |  |  |  |  |  |  |  |
| 老师 | √ |  |  | √ | √ | √ | √ | √ | √ | √ |
| 助教 | √ |  |  |  | √ | √ | √ | √ | √ | √ |
| 权限 | 团队管理 | 博客作业列表管理 | 助教学生总览 | 助教学生新增 | 细则表管理 | 细则信息搜索 | 细则详情管理 | 学生总成绩管理 |  |  |
| 学生 |  |  |  |  |  |  |  |  |  |  |
| 老师 | √ | √ | √ | √ |  | √ | √ | √ |  |  |
| 助教 | √ | √ |  |  |  | √ | √ | √ |  |  |

在系统权限管理中，我们采用基于角色的访问控制（RBAC）对进行权限管理，RBAC中，对系统操作的各种权限不是直接授予具体的用户，而是在用户集合与权限集合之间建立一个角色集合。每一种角色对应一组相应的权限。一旦用户被分配了适当的角色后，该用户就拥有此角色的所有操作权限。这样做的好处是，不必在每次创建用户时都进行分配权限的操作，只要分配用户相应的角色即可，而且角色的权限变更比用户的权限变更要少得多，这样可以简化用户的权限管理，减少系统的开销。

在所采用的RBAC0中，我们对应有三种角色：教师、助教、学生对应三种角色拥有不同的权限（如上表所示），依据每个用户的角色不同，其拥有的角色权限的限制下，我们可以划定每个用户所访问的资源边界。

在整个系统中，教师拥有的权限是最高的，考虑到助教是使用评分系统操作最多的管理员角色，故将大部分管理员权限赋予了助教，教师继承拥有助教的所有权限，还可以对助教进行新增、删除、修改、查看，这种角色相当于树形结构中的根。

助教拥有的权限包括所有对学生的操作，如打分、学生成绩的增删改查、评分细则的管理，助教分担了教师的压力，助教在一定程度上可以限制学生这一角色的活动，避免非法用户的使用。

学生是系统的主要使用对象，他们的主要使用安卓端进行操作，在web端中仅有分数查询和作业提交等少数权限对其开放。