

RBAC Manual

医院信息系统——基于角色的访问控制（RBAC）说明书。

Author: zyc

医院信息系统需要对用户权限进行管理，以保证数据安全性。根据需求与实现的复杂度，HIS使用基于URL和资源对象的RBAC权限管理系统。由于Django仅支持数据库层面的大粒度权限与权限组，因此基于Django.contrib.auth模块开发HIS.rbac模块。

基本说明

HIS.rbac包含两个权限系统，分别控制URL访问和数据库特定数据行的权限。

对象资源权限 Object Permission

包括三个方面：数据库对象、行级资源对象和操作类型。即可以对特定数据库的某个数据行做出何种操作，例如能够对内科1号患者数据进行访问（患者数据库 + row: 1 + 访问）。

URL 访问权限 URL Permission

控制系统用户能够访问的页面。即能否访问特定的URL。

用户 User

包括医院职工、大型设备与机器等。由于C端用户（患者）不应该深入使用医院内部的信息系统，因此患者使用另外一套及其简单的权限管理方式。

角色 Role

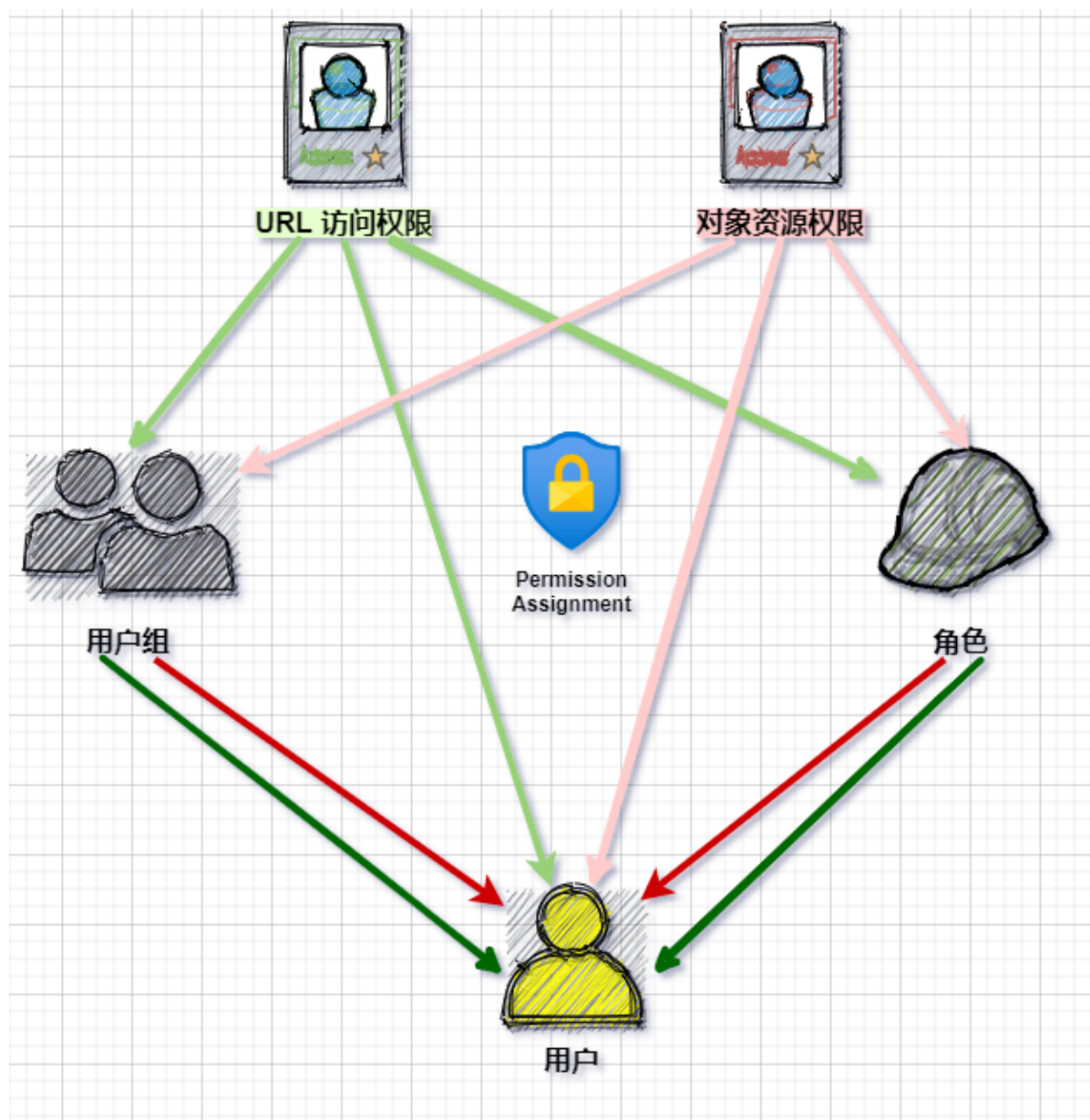
角色作为多个相似权限的集合，用于简化向用户分配权限的过程。一个角色可以拥有多个权限，一个用户可以拥有多个角色。向用户授权时，关联相关角色即可。

用户组 UserGroup

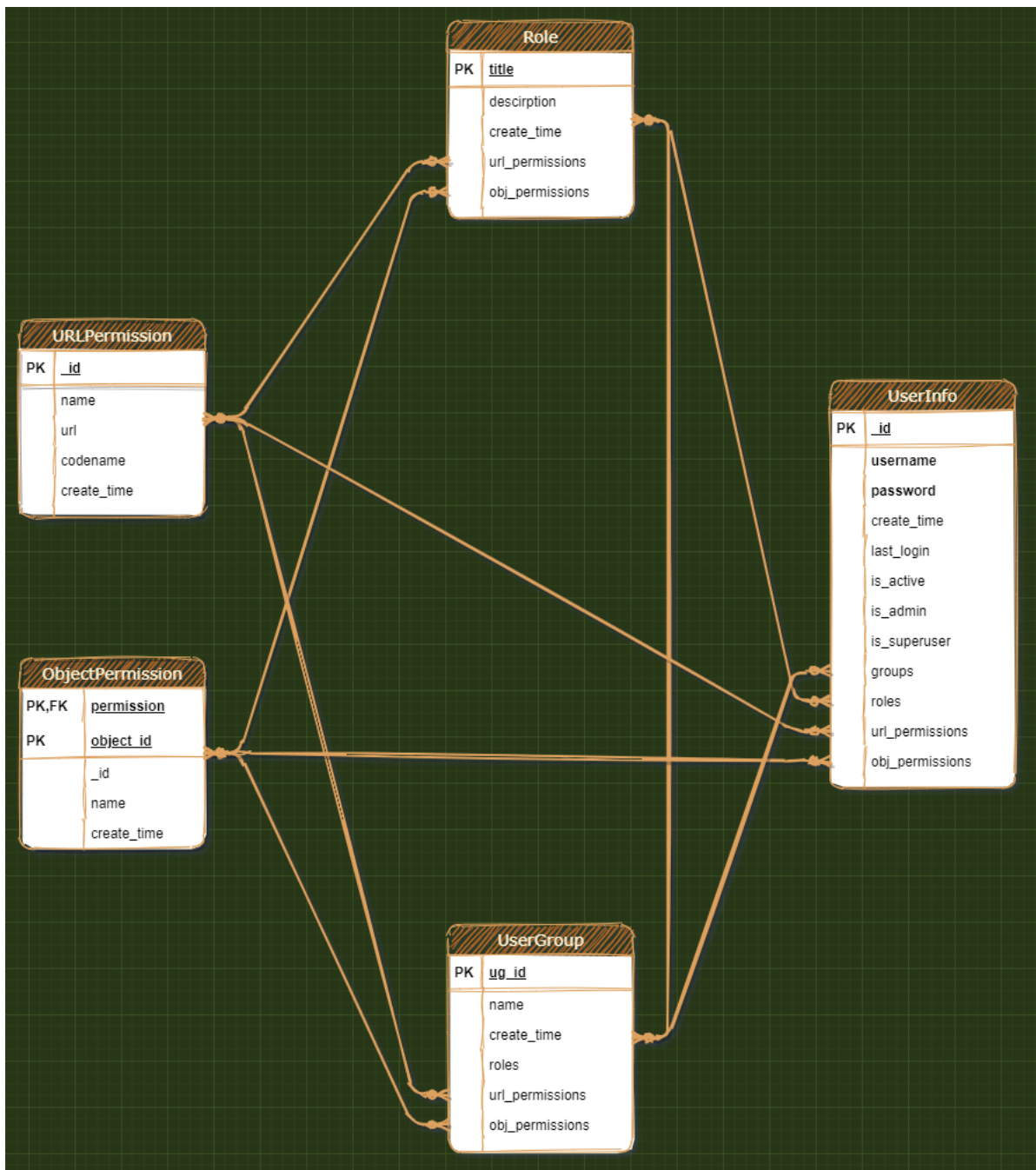
在用户基数很大时，通过将相关角色关联到一个个用户来进行授权会变得十分麻烦。这时使用用户组来汇集具有相同权限和角色特点的用户。

权限分配与数据库设计

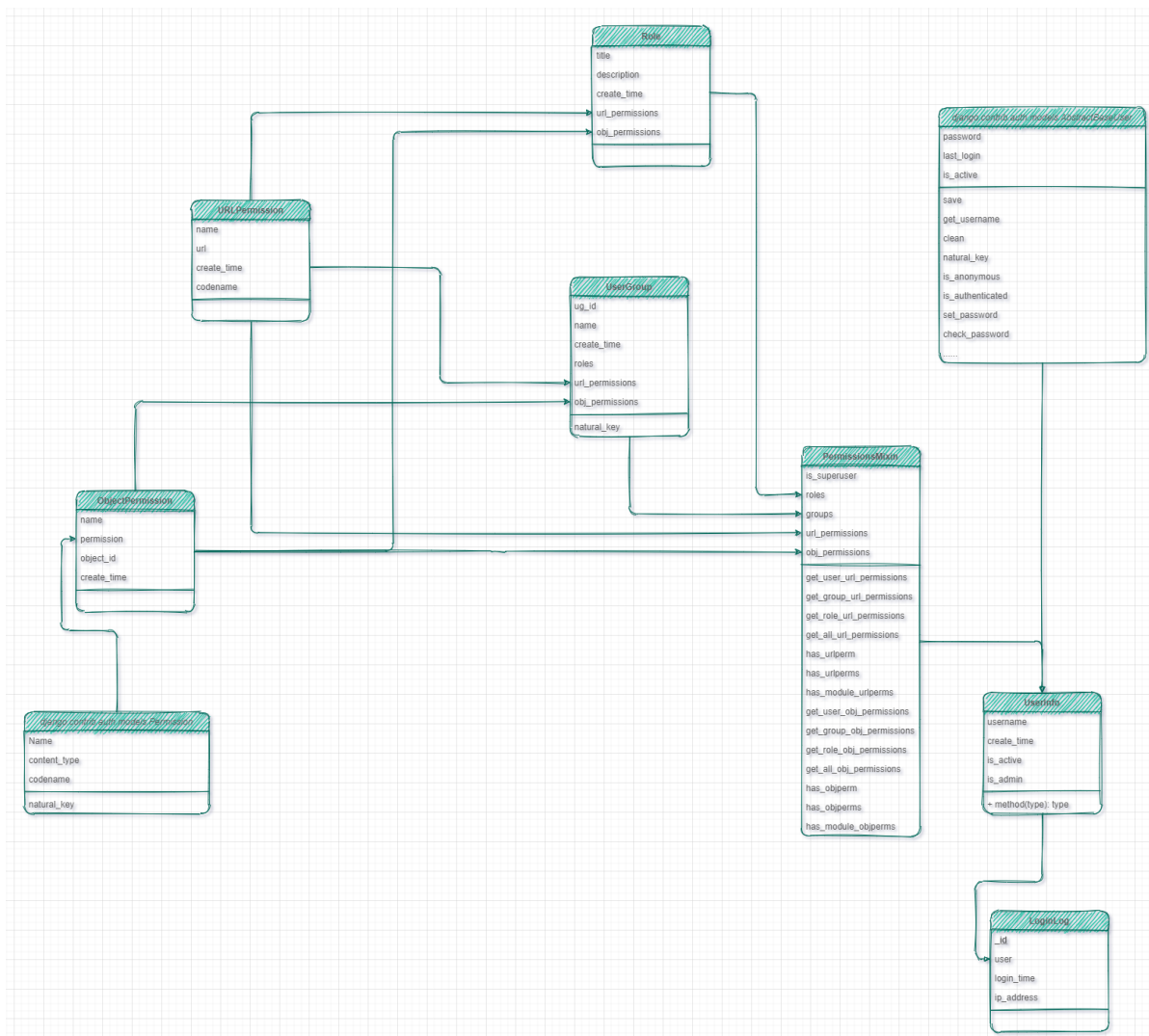
用户权限来源图



ER图



UML 类图



代码实现

测试用例

权限预定义与预分配

1. URL 访问权限：

| <input type="checkbox"/> | 权限名 | URL | 2 ▲ | 权限代码 | 1 |
|--------------------------|-------------|------|-----|---------|---|
| <input type="checkbox"/> | 测试URL访问权限1 | /1/ | | test_1 | |
| <input type="checkbox"/> | 测试URL访问权限10 | /10/ | | test_10 | |
| <input type="checkbox"/> | 测试URL访问权限2 | /2/ | | test_2 | |
| <input type="checkbox"/> | 测试URL访问权限3 | /3/ | | test_3 | |
| <input type="checkbox"/> | 测试URL访问权限4 | /4/ | | test_4 | |
| <input type="checkbox"/> | 测试URL访问权限5 | /5/ | | test_5 | |
| <input type="checkbox"/> | 测试URL访问权限6 | /6/ | | test_6 | |
| <input type="checkbox"/> | 测试URL访问权限7 | /7/ | | test_7 | |
| <input type="checkbox"/> | 测试URL访问权限8 | /8/ | | test_8 | |
| <input type="checkbox"/> | 测试URL访问权限9 | /9/ | | test_9 | |

10 URL访问权限

2. 对象访问权限:

| <input type="checkbox"/> | 对象权限名称 | 权限 | 对象ID | ▲ |
|--------------------------|---------|------------------------------|------|---|
| <input type="checkbox"/> | 测试对象权限1 | his 科室部门 Can change 科室部门 | 1 | |
| <input type="checkbox"/> | 测试对象权限2 | his 科室部门 Can delete 科室部门 | 2 | |
| <input type="checkbox"/> | 测试对象权限3 | his 科室部门 Can view 科室部门 | 3 | |
| <input type="checkbox"/> | 测试对象权限4 | his 职工 Can change 职工 | 4 | |
| <input type="checkbox"/> | 测试对象权限5 | his 职工 Can delete 职工 | 5 | |
| <input type="checkbox"/> | 测试对象权限6 | his 职工 Can view 职工 | 6 | |
| <input type="checkbox"/> | 测试对象权限7 | his 职工 Can change 职工 | 7 | |
| <input type="checkbox"/> | 测试对象权限8 | his 职工 Can delete 职工 | 8 | |
| <input type="checkbox"/> | 测试对象权限9 | his 职工 Can view 职工 | 9 | |

9 对象权限

3. 角色:

| ID | Role | URL Perm | OBJ Perm |
|----|-----------|------------|-----------|
| 1 | HIS 开发人员 | All (1-10) | All (1-9) |
| 2 | HIS 测试工具人 | 1, 2 | 1, 2 |
| 3 | 内科住院部医生 | 3, 4 | 3, 4 |
| 4 | 外科门诊医生 | 5, 6 | 5, 6 |

4. 用户组：

| ID | UserGroup | URL Perm | OBJ Perm |
|----|-----------|------------|-----------|
| 1 | HIS开发部 | All (1-10) | All (1-9) |
| 2 | HIS测试部 | 9, 10 | 8, 9 |
| 3 | 内科 | 7, 8 | 6, 7 |
| 4 | 外科 | 5, 6 | 4, 5 |

用户权限分配

用户：

| User | Total URL Perms | Total OBJ Perms | Roles | UserGroups | URL Perms | OBJ Perms |
|--------|-------------------|------------------|-------|------------|-----------|-----------|
| zyc | All (1-10) | All (1-9) | 1 | 1 | | |
| 000000 | 1, 2, 9, 10 | 1, 2, 8, 9 | 2 | 2 | | |
| 000001 | 1, 2, 7, 8 | 1, 2, 6, 7 | 2 | 3 | | |
| 000002 | 1, 2, 5, 6 | 1, 2, 4, 5 | 2 | 4 | | |
| 000003 | 1, 2, 7, 8, 9, 10 | 1, 2, 6, 7, 8, 9 | 2 | 2, 3 | | |
| 000004 | All (1-10) | All (1-9) | 2, 3 | 2, 3, 4 | | |
| 000005 | 1, 2, 5, 9, 10 | 1, 2, 5, 8, 9 | 2 | 2 | 5 | 5 |
| 000006 | 1, 2, 7, 8, 9 | 1, 2, 3, 6, 7 | 2 | 3 | 7, 8, 9 | 1, 2, 3 |
| 000007 | 1, 2, 3, 4 | 2, 3, 4 | 3 | | 1, 2 | 2, 3 |
| 000008 | 4, 5, 6, 7 | 3, 4, 5, 6 | | 4 | 4, 7 | 3, 6 |
| 000009 | 1, 5, 10 | 2, 6, 9 | | | 1, 5, 10 | 2, 6, 9 |

结果

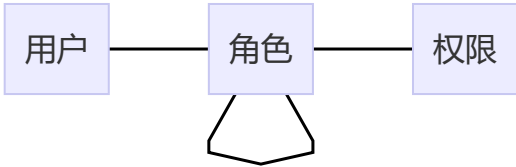
测试用例全部通过。

未来开发方向

HIS.rbac 只完成了 RBAC0 级别的权限管理系统，如下图所示：

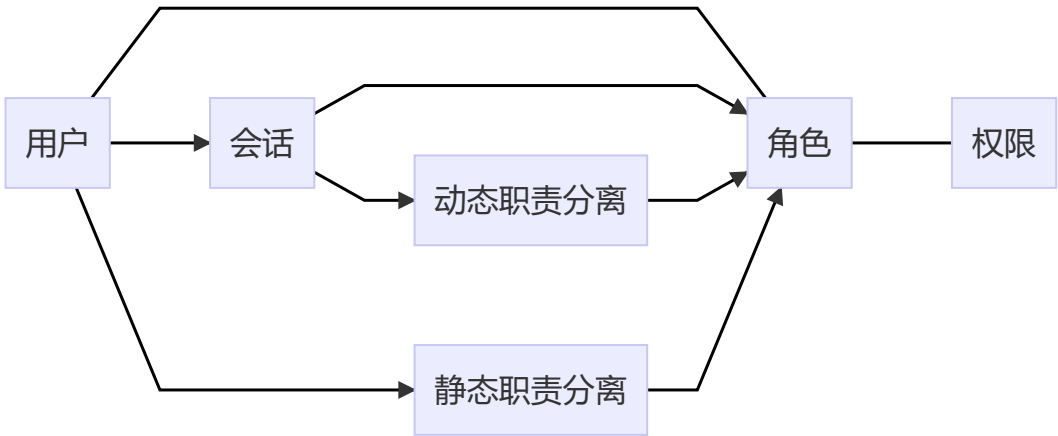


未来可以进一步完善 RBAC 系统，加入角色的继承机制，达到 RBAC1：



或者加入授权时应遵循的强制性规则，完成动态职责分离（DSD）、静态职责分离（SSD）：

例如，一个用户不应该同时具有出纳和会计两个角色的权限。



或者，使系统同时具有 RBAC1 和 RBAC2 的特点，达到 RBAC3：

