

## 1. 数据库详细设计

表 Admin，用于存储管理员的信息

|   | Field         | Schema       | Table | Type    |
|---|---------------|--------------|-------|---------|
| 1 | adminid       | houserenting | admin | INT     |
| 2 | adminname     | houserenting | admin | VARCHAR |
| 3 | adminphone    | houserenting | admin | VARCHAR |
| 4 | adminsex      | houserenting | admin | CHAR    |
| 5 | adminnickname | houserenting | admin | VARCHAR |
| 6 | adminpassword | houserenting | admin | VARCHAR |
| 7 | adminnumber   | houserenting | admin | VARCHAR |

adminid:管理员编号，主键

adminnickname:管理员昵称

adminname:管理员姓名

adminpassword:管理员密码

adminnumber:管理员身份证号码

adminsex:管理员性别

adminphone:管理员电话号码，主键

表 Buyinfo，用于存储买房信息

|   | Field     | Schema       | Table   | Type    |
|---|-----------|--------------|---------|---------|
| 1 | userid    | houserenting | buyinfo | INT     |
| 2 | username  | houserenting | buyinfo | VARCHAR |
| 3 | userphone | houserenting | buyinfo | VARCHAR |
| 4 | usersex   | houserenting | buyinfo | CHAR    |
| 5 | houseid   | houserenting | buyinfo | INT     |
| 6 | buystate  | houserenting | buyinfo | TINYINT |

userid:用户编码，主键

username:用户名称

userphone:用户电话，主键

usersex:用户性别

houseid:房屋编号

buystate:出售状态

表 houseinfo，存储房屋信息

|    |                  |              |           |         |
|----|------------------|--------------|-----------|---------|
| 1  | houseid          | houserenting | houseinfo | INT     |
| 2  | username         | houserenting | houseinfo | VARCHAR |
| 3  | userphone        | houserenting | houseinfo | VARCHAR |
| 4  | houseaddress     | houserenting | houseinfo | VARCHAR |
| 5  | rentnotice       | houserenting | houseinfo | VARCHAR |
| 6  | houseelevator    | houserenting | houseinfo | TINYINT |
| 7  | rentstate        | houserenting | houseinfo | TINYINT |
| 8  | sellstate        | houserenting | houseinfo | TINYINT |
| 9  | housepicture     | houserenting | houseinfo | VARCHAR |
| 10 | houserent        | houserenting | houseinfo | DOUBLE  |
| 11 | houseprice       | houserenting | houseinfo | DOUBLE  |
| 12 | housetype        | houserenting | houseinfo | VARCHAR |
| 13 | houseorientation | houserenting | houseinfo | VARCHAR |
| 14 | houseage         | houserenting | houseinfo | INT     |

|    |               |              |           |         |
|----|---------------|--------------|-----------|---------|
| 15 | housearea     | houserenting | houseinfo | DOUBLE  |
| 16 | housechecked  | houserenting | houseinfo | TINYINT |
| 17 | housejoinrent | houserenting | houseinfo | TINYINT |
| 18 | houseproperty | houserenting | houseinfo | VARCHAR |

houseid:房屋编号，主键

username:用户名称

userphone:用户电话，主键

houseaddress:房屋地址

rentnotice:租赁注意事项

houseelevator:有无电梯

rentstate:出租状态

sellstate:出售状态

housepicture:房屋图片

houserent:租金

houseprice:售价

housetype:户型

houseorientation:房屋朝向

houseage:房龄

housearea:房屋面积

housechecked:审核状态

housejoinrent:是否合租

houseproperty:房屋产权

表 Post，存储贴子信息

|   |              |              |      |         |
|---|--------------|--------------|------|---------|
| 1 | textid       | houserenting | post | INT     |
| 2 | usernickname | houserenting | post | VARCHAR |
| 3 | userphone    | houserenting | post | VARCHAR |
| 4 | text         | houserenting | post | VARCHAR |

textid:贴子编号，主键

usernickname:用户昵称

userphone:用户电话

text:贴子内容

表 Rentinfo，存储租房信息

|   |           |              |          |         |
|---|-----------|--------------|----------|---------|
| 1 | userid    | houserenting | rentinfo | INT     |
| 2 | username  | houserenting | rentinfo | VARCHAR |
| 3 | userphone | houserenting | rentinfo | VARCHAR |
| 4 | usersex   | houserenting | rentinfo | CHAR    |
| 5 | houseid   | houserenting | rentinfo | INT     |
| 6 | joinrent  | houserenting | rentinfo | TINYINT |
| 7 | rentstate | houserenting | rentinfo | TINYINT |

userid:用户编号，主键

username:用户真实姓名

usersex:用户性别

userphone:用户电话号码

houseid:房屋编号

joinrent:是否合租

rentstate:租房状态

表 user，存储用户信息

|   |              |              |      |         |
|---|--------------|--------------|------|---------|
| 1 | userid       | houserenting | user | INT     |
| 2 | username     | houserenting | user | VARCHAR |
| 3 | userphone    | houserenting | user | VARCHAR |
| 4 | usersex      | houserenting | user | CHAR    |
| 5 | usernickname | houserenting | user | VARCHAR |
| 6 | useraddress  | houserenting | user | VARCHAR |
| 7 | userpassword | houserenting | user | VARCHAR |
| 8 | usernumber   | houserenting | user | VARCHAR |

userid:用户编号，主键

usernickname:用户昵称

userpassword:用户密码

username:用户真实姓名

usernumber:用户身份证号码

usersex:用户性别

userphone:用户电话号码，主键

useraddress:用户地址

## 2. 代码分层与接口

前端：Vue 框架和微信小程序

后端：采用 Spring Boot 框架，分为 Controller,DAO,Service,Entity 四层。其中 controller 是后端向前端开放的接口层，DAO 是后端使用数据库进行持久化的接口层，业务逻辑主要在 Service 层中完成，Entity 是映射数据库中数据表的实体类。

前后端相连接口：

### 1、 获取所有管理员信息

请求 URL:     /api/admin/resources

请求方式:     GET

### 2、 通过 Adminid 查找某个管理员信息

请求 URL:     /api/admin/resources/{aid}

请求方式:     GET

### 3、 管理员信息登陆

请求 URL:     /api/admin/login

请求方式:     POST

### 4、 管理员注册

请求 URL:     /api/admin/register  
请求方式:     POST

#### 5、 删除管理员

请求 URL:     /api/admin/unsubscribe/{aid}  
请求方式:     DELETE

#### 6、 通过房屋状态查找已审核或未审核房屋列表

请求 URL:     /api/house/resources/list/{state}  
请求方式:     GET

#### 7、 通过 hid 查找已审核或未审核的某个房屋信息

请求 URL:     /api/house/resources/{hid}/{state}  
请求方式:     GET

#### 8、 发布房屋信息

请求 URL:     /api/house/resource  
请求方式:     POST

#### 9、 修改房屋审核状态

请求 URL:     /api/house/resources/{hid}  
请求方式:     PUT

#### 10、 删除房屋信息

请求 URL:     /api/house/resources/{hid}  
请求方式:     PUT

#### 11、 获取所有贴子信息

请求 URL:     /api/message/list  
请求方式:     GET

#### 12、 发贴

请求 URL:     /api/message/resource  
请求方式:     POST

#### 13、 获取贴子信息

请求 URL:     /api/message/list/{tid}  
请求方式:     GET

#### 14、 删除贴子

请求 URL:     /api/message/resources/{tid}  
请求方式:     DELETE

#### 15、 查询租房信息

请求 URL:     /api/house/resources/tenement/{hid}  
请求方式:     GET

#### 16、 查询租房状态

请求 URL:     /api/house/resources/tenement/{hid}/state  
请求方式:     GET

#### 17、 获取用户列表

请求 URL:     /api/user/resources  
请求方式:     GET

#### 18、 获取用户信息

请求 URL:     /api/user/resources/{uid}  
请求方式:     GET

#### 19、 用户登陆

请求 URL:     /api/user/login  
请求方式:     POST

#### 20、 用户注册

请求 URL:     /api/user/register  
请求方式:     POST

#### 21、 删除用户

请求 URL:     /api/user/unsubscribe/{uid}  
请求方式:     DELETE