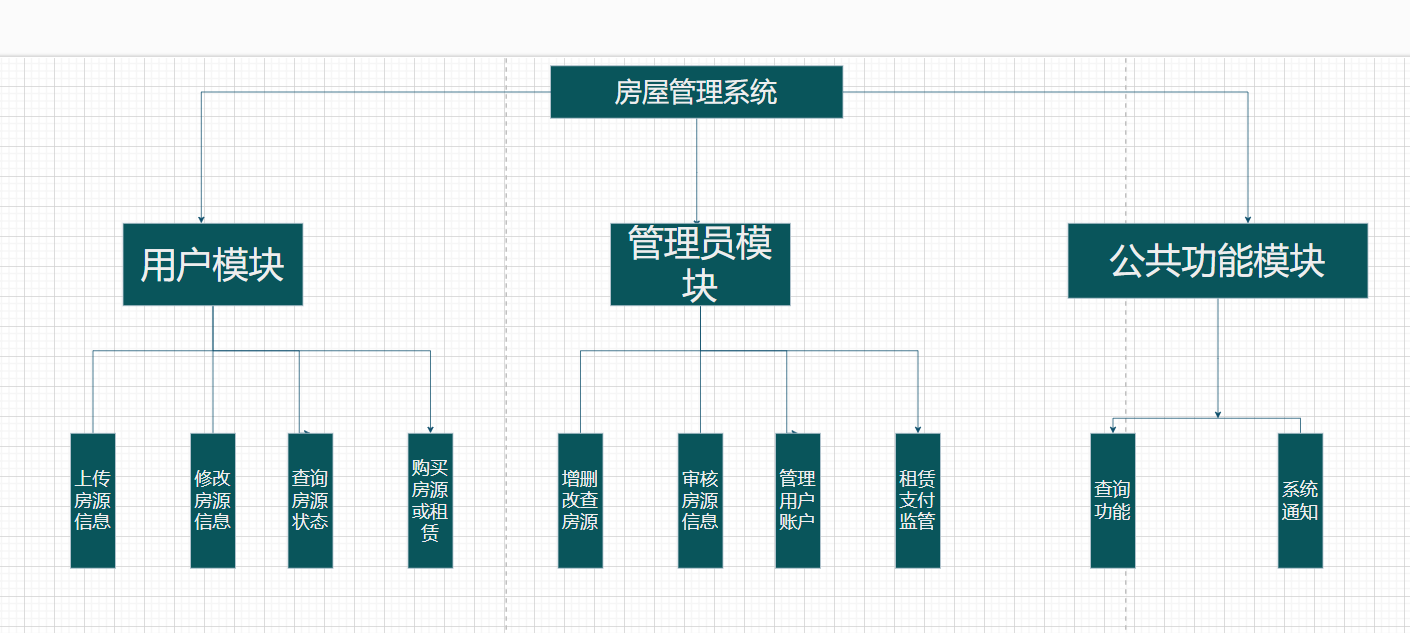
**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于jQuery的房屋管理系统的分析与设计 | | |
| 课 程 | 软件工程 | | |
| 学 院 | 信息工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2023级 |
| 学生姓名 | 周春洁 | 学号 | 012322100050 |

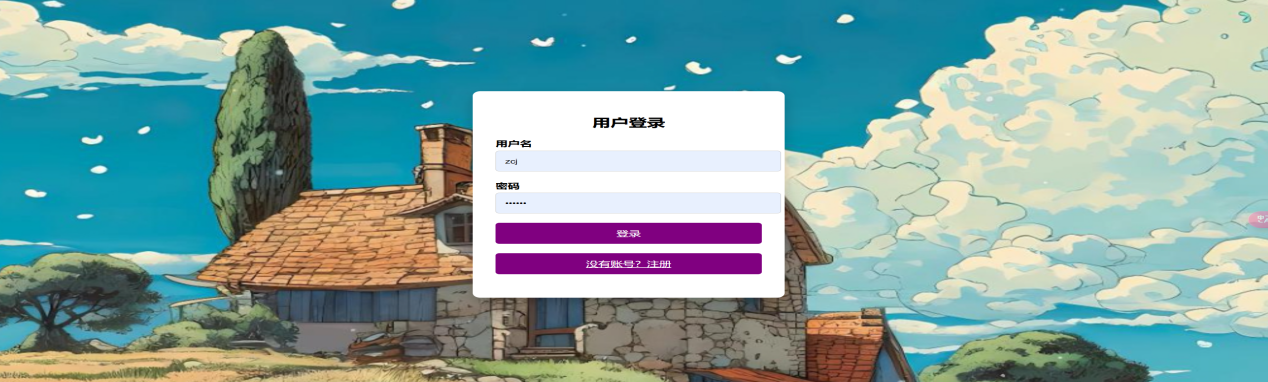
|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方，内容: 可行性分析、需求分析、概要设计、详细设计。 | 20 |  |
| 用例图及规约 | 用例图完整，准确，能够完全体现需求 | 10 |  |
| 顺序图与活动图 | 顺序图能够完全描述用例的设计思路和业务流程，活动图能够描述部分算法的流程。 | 10 |  |
| 类图 | 类图能够完整，准确反映业务的需数据的组织结构 | 10 |  |
| 数据库 | 数据库表的设计来自于类图，合理 | 10 |  |
| 界面设计 | 界面设计美观，清晰，合理，能够完全反映用例图的设计思路 | 10 |  |
| API接口设计 | 为每个界面设计合理的API接口，以便进行前后端分离开发 | 15 |  |
| 内容一致性 | 用例图，类图、数据库、界面相互印证，相互依赖，环环相扣 | 15 |  |
| **得分合计** | | |  |

2024 年 12 月 28 日

# 1. 系统总体结构



**2.登录模块用例**



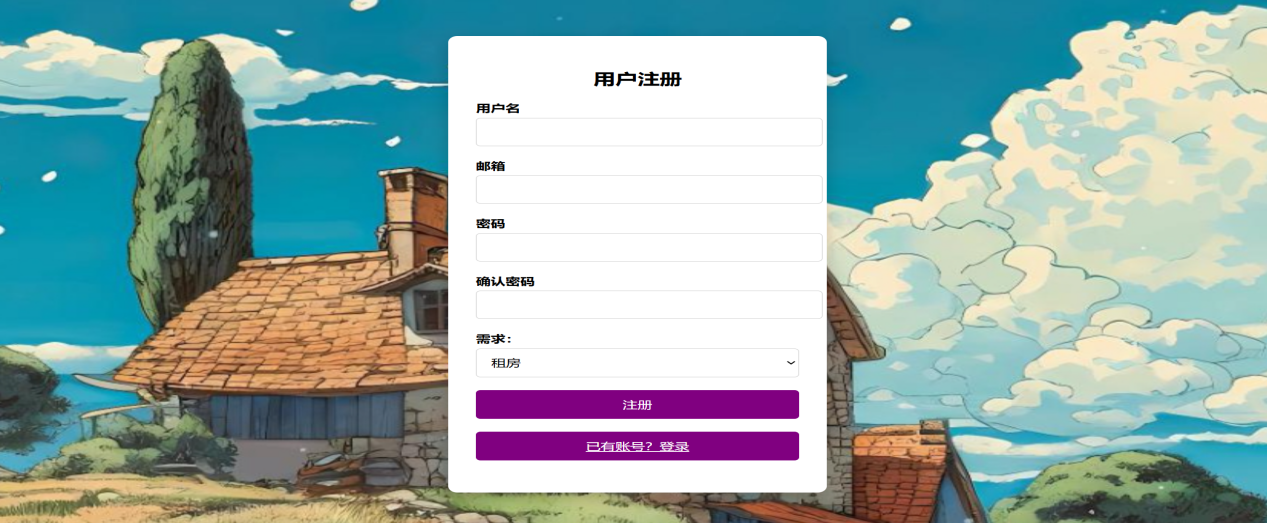
接口：POST /login

#### 登录与注册接口

**对应前端功能模块**：login.html

**功能**：用户进行账号登录和注册功能。

##### 后端接口设计 接口 ：LoginServlet



###### 首页展示





#### 控制台接口

**对应前端功能模块**：Console.html

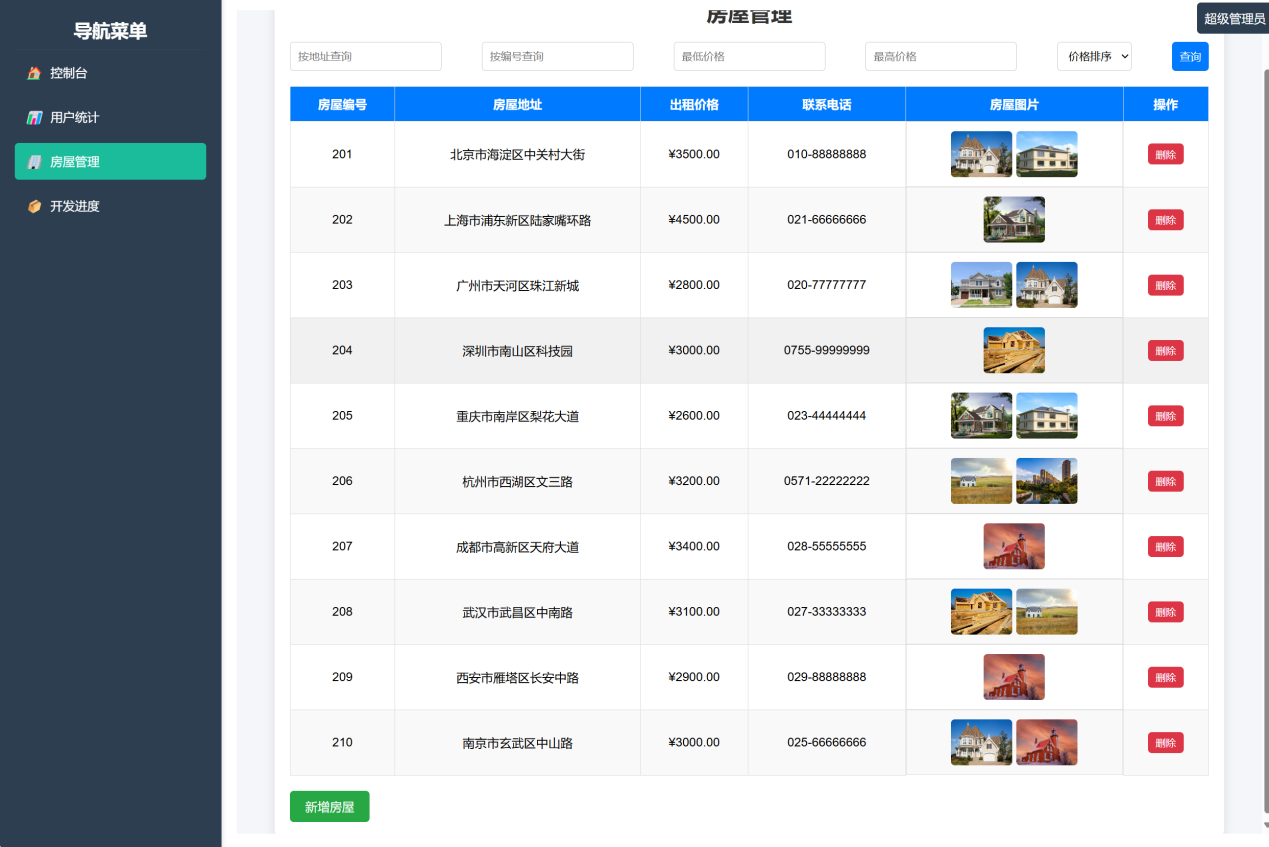
**功能**：展示现有房屋的数量（可跟实际房屋数量变化），和用户访问量。

##### 后端接口设计 接口 ：StatsServlet

4.用户统计



5.房屋管理





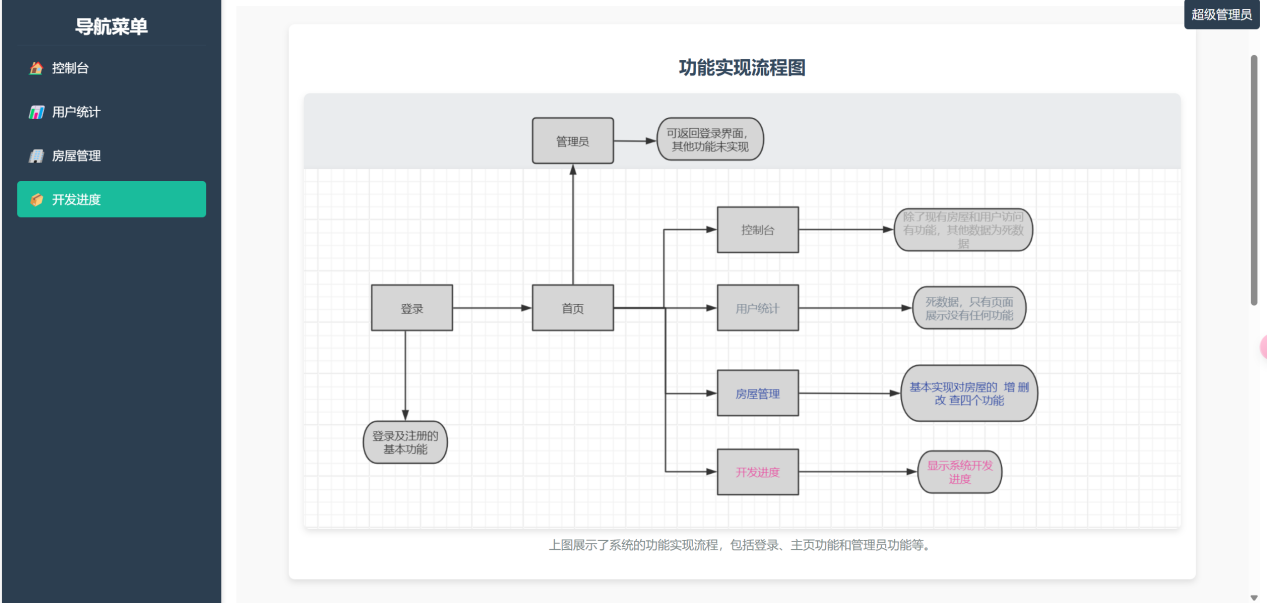
#### 房屋管理接口

**对应前端功能模块**：Housing management.html

**功能**：对现有房屋进行增 删改 和不同条件的筛选和排序。

##### 后端接口设计 接口 ：StatsServlet

1. 现有进度展示

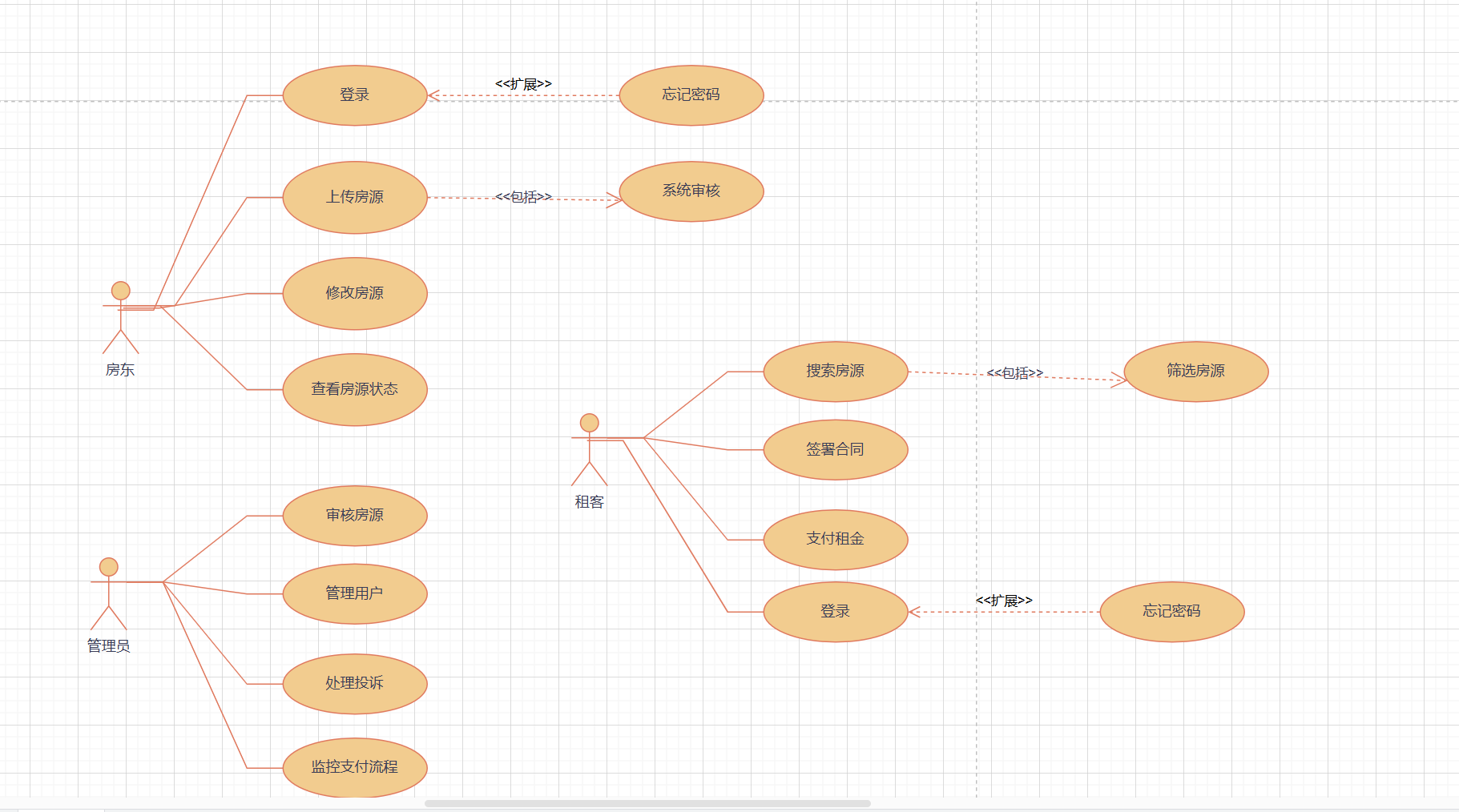




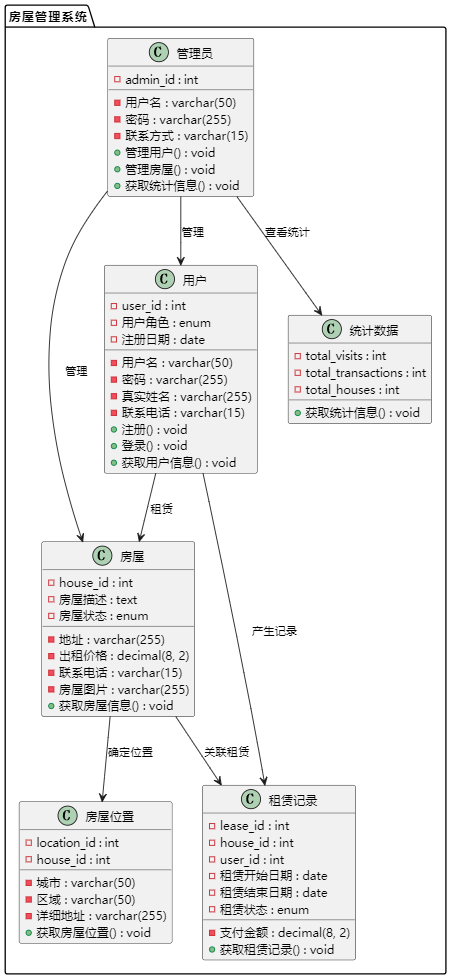
功能：展示目前系统项目开发进度和一些前后端文件的梳理

二．

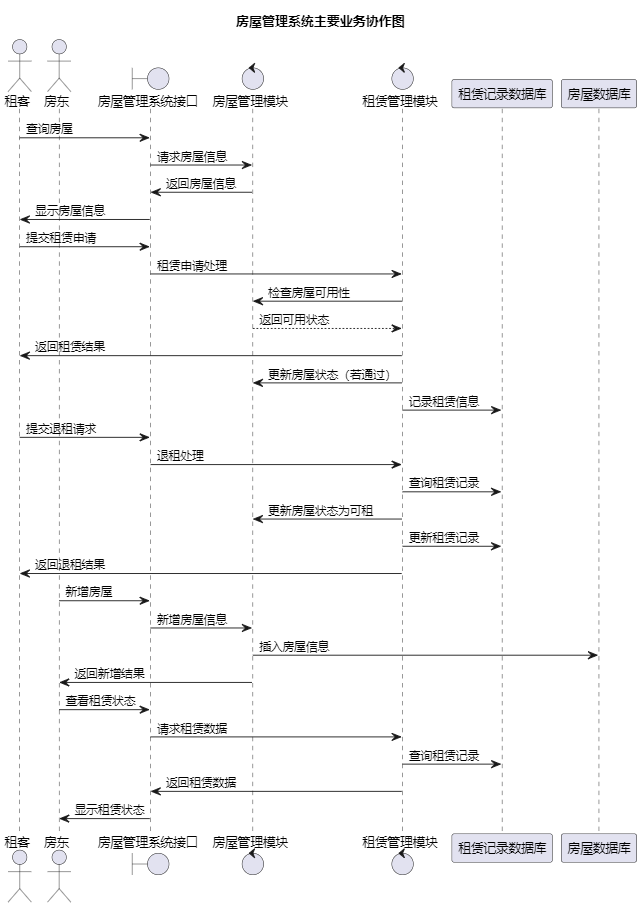
# 用例图设计



# 类图设计



# 5. 协作图设计



# 5. 数据库设计

## HOUSES表（房源表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 说明 |
| HOUSE\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  | 房源的ID |
| ADDRESS | VARCHAR2(200 BYTE) |  | 否 |  | 房源的地址 |
| AREA | NUMBER(6,2) |  | 否 |  | 房源面积（㎡） |
| RENT | NUMBER(10,2) |  | 否 |  | 租金（¥） |
| STATUS | VARCHAR2(20 BYTE) |  | 否 | 空置 | 房源状态（空置、已出租、维修中） |
| CREATED\_AT | DATE |  | 否 | 当前时间 | 房源录入时间 |

## TENANTS表（租客表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 说明 |
| TENANT\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  | 租客的ID |
| NAME | VARCHAR2(50 BYTE) |  | 否 |  | 租客的姓名 |
| PHONE | VARCHAR2(15 BYTE) |  | 否 |  | 租客的联系电话 |
| ID\_NUMBER | VARCHAR2(18 BYTE) | 唯一 | 否 |  | 租客身份证号 |
| EMAIL | VARCHAR2(50 BYTE) |  | 是 | 空 | 租客的邮箱地址 |
| CREATED\_AT | DATE |  | 否 | 当前时间 | 租客注册时间 |

## CONTRACTS表（租赁合同表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 说明 |
| CONTRACT\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  | 合同的ID |
| HOUSE\_ID | VARCHAR2(50 BYTE) | 外键 | 否 |  | 房源的ID |
| TENANT\_ID | VARCHAR2(50 BYTE) | 外键 | 否 |  | 租客的ID |
| START\_DATE | DATE |  | 否 |  | 租赁开始日期 |
| END\_DATE | DATE |  | 否 |  | 租赁结束日期 |
| RENT\_AMOUNT | NUMBER(10,2) |  | 否 |  | 租金金额 |

## HOUSE\_SELECTION表（选房表）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 主键，外键 | 可以为空 | 默认值 | 说明 |
| SELECTION\_ID | VARCHAR2(50 BYTE) | 主键 | 否 |  | 选房记录的ID |
| TENANT\_ID | VARCHAR2(50 BYTE) | 外键 | 否 |  | 租客的ID |
| HOUSE\_ID | VARCHAR2(50 BYTE) | 外键 | 否 |  | 房源的ID |
| SELECTION\_DATE | DATE |  | 否 | 当前时间 | 选房日期 |

1. 房源表（Houses）：

- house\_id：房源编号（主键）

- address：房源地址

- area：面积

- rent：租金

- status：状态（空置、已出租等）

- created\_at：创建时间

2. 租客表（Tenants）：

- tenant\_id：租客编号（主键）

- name：租客姓名

- phone：联系电话

- id\_number：身份证号

3. 合同表（Contracts）：

- contract\_id：合同编号（主键）

- house\_id：房源编号（外键）

- tenant\_id：租客编号（外键）

- start\_date：租赁开始日期

- end\_date：租赁结束日期

- rent\_amount：租金总额

### 5.2 表结构

CREATE TABLE houses (

house\_id INT AUTO\_INCREMENT PRIMARY KEY,

address VARCHAR(255) NOT NULL,

area FLOAT NOT NULL,

rent FLOAT NOT NULL,

status ENUM("空置", "已出租", "维修中") DEFAULT "空置",

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE tenants (

tenant\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

phone VARCHAR(20) NOT NULL,

id\_number VARCHAR(18) UNIQUE NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE contracts (

contract\_id INT AUTO\_INCREMENT PRIMARY KEY,

house\_id INT NOT NULL,

tenant\_id INT NOT NULL,

start\_date DATE NOT NULL,

end\_date DATE,

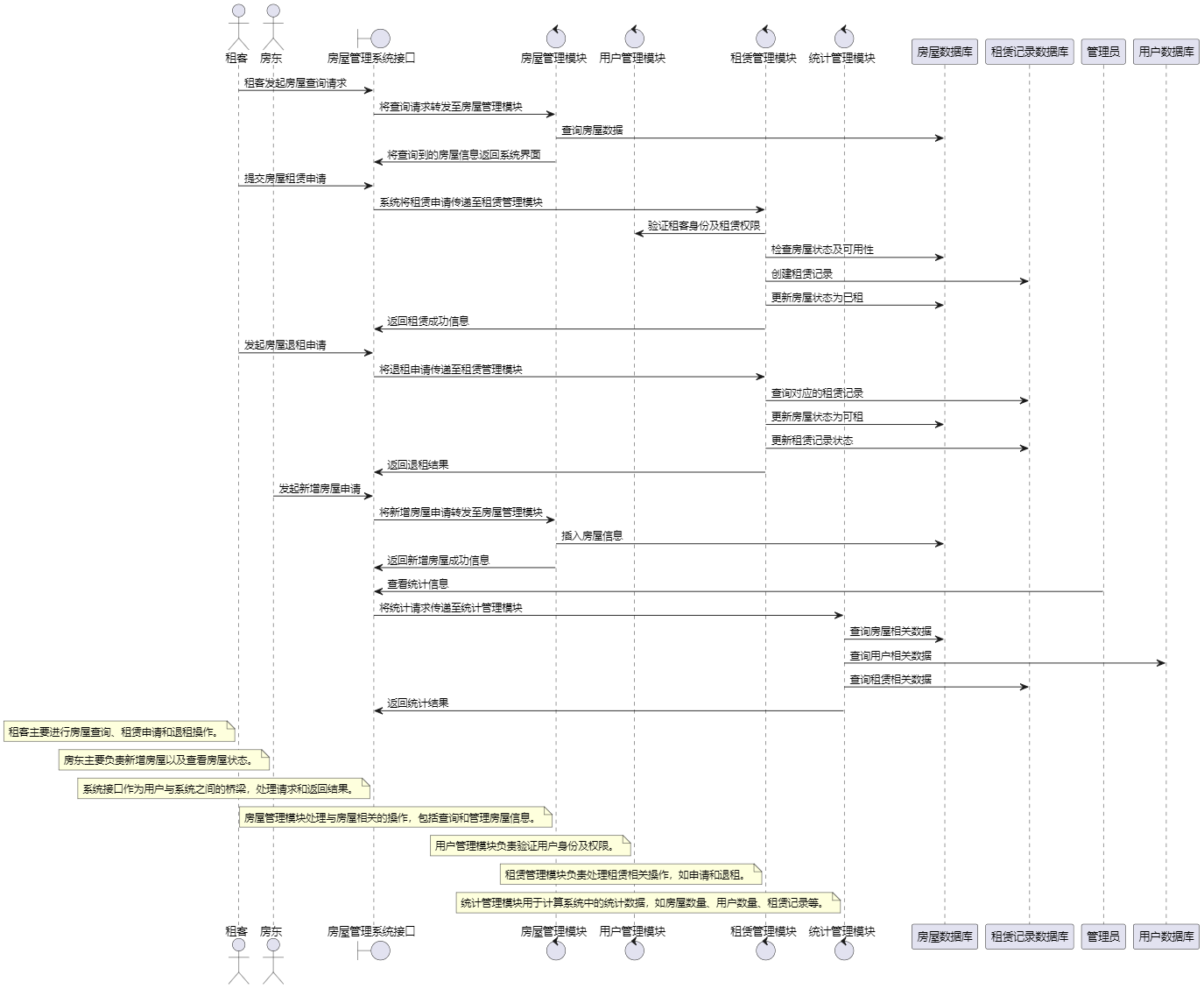
rent\_amount FLOAT NOT NULL,

FOREIGN KEY (house\_id) REFERENCES houses(house\_id),

FOREIGN KEY (tenant\_id) REFERENCES tenants(tenant\_id)

);

### 6.2业务流程（顺序图）



### 6.3界面设计



### 6.4接口设计

## 2. 接口清单

以下是房屋管理系统的接口列表：

1. 获取房屋列表接口（GET /get-houses）

2. 新增房屋接口（POST /get-houses）

3. 删除房屋接口（DELETE /get-houses）

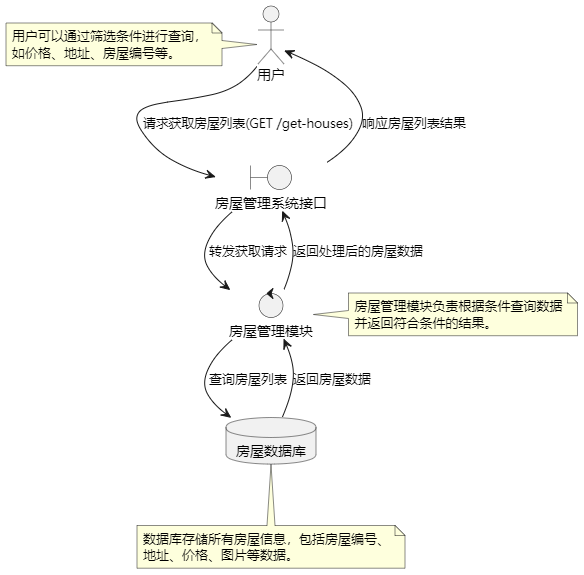
4. 登录接口（POST /login）

5. 获取统计数据接口（GET /get-stats）

## 3. 接口设计详情

### 3.1 获取房屋列表接口

流程图



功能：查询符合条件的房屋信息列表。

API 请求地址：/get-houses

请求方式：GET

#### 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| address | String | 否 | 按地址模糊搜索 |
| id | Integer | 否 | 按房屋编号精确搜索 |
| minPrice | Double | 否 | 最低价格 |
| maxPrice | Double | 否 | 最高价格 |
| priceOrder | String | 否 | 价格排序方式，取值为 "asc" 或 "desc" |

#### 返回参数说明：

返回值为 JSON 格式的房屋列表，包含以下字段：

[

{

"id": 201,

"address": "北京市海淀区中关村大街",

"price": 3500.0,

"contact": "010-88888888",

"imageUrls": ["/path/1.jpg", "/path/2.jpg"]

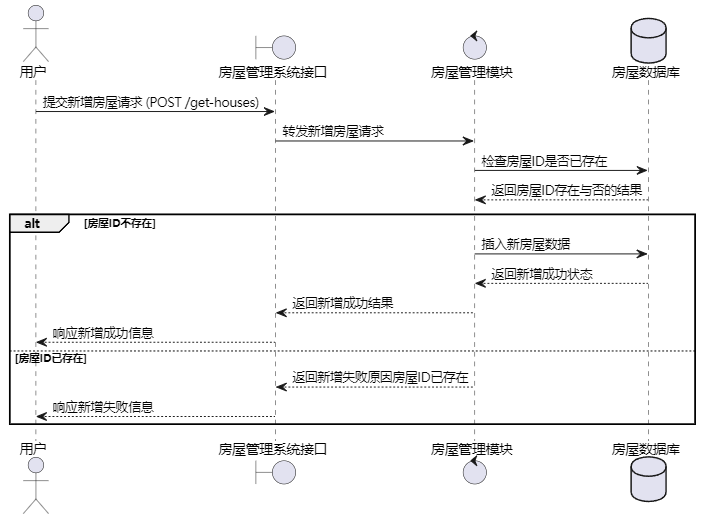
},

...

]



### 3.2 新增房屋接口



功能：添加新的房屋信息到系统中。

API 请求地址：/get-houses

请求方式：POST

#### 请求参数：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 类型 | 是否必填 | 说明 |
| id | Integer | 是 | 房屋编号，需唯一 |
| address | String | 是 | 房屋地址 |
| price | Double | 是 | 出租价格 |
| contact | String | 是 | 联系电话 |
| imageUrls | Array<String> | 是 | 房屋图片 URL 列表 |

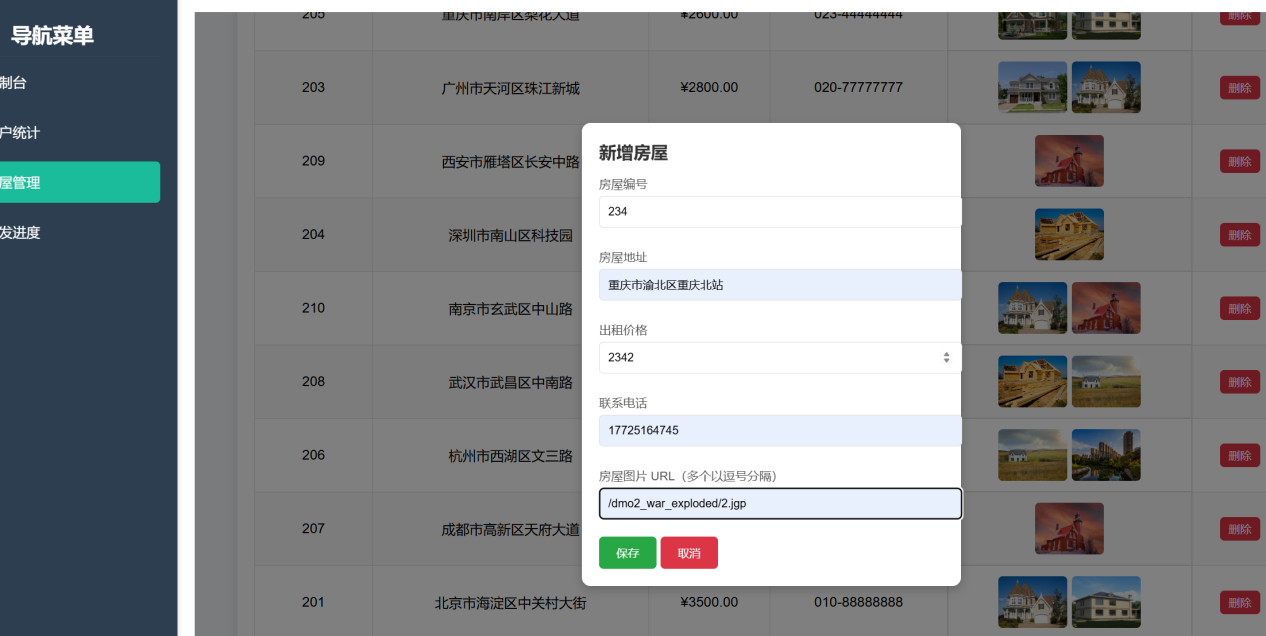
#### 返回参数说明：

{

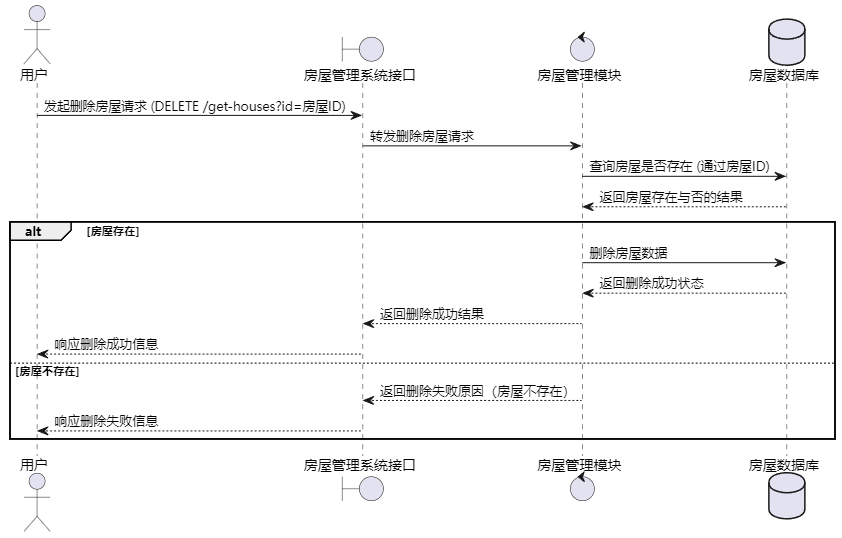
"status": "success" | "failure",

"message": "相关提示信息"

}



#### ****接口3：删除房屋接口****



* **功能**：删除指定房屋信息。
* **权限**：用户必须登录后才可以删除房屋。
* **API请求地址**：/get-houses
* **请求方式**：DELETE

##### ****请求实例****：

json

{

"id": 101}

##### ****请求参数说明****：

| **参数名称** | **类型** | **是否必填** | **说明** |
| --- | --- | --- | --- |
| id | int | 是 | 要删除的房屋ID。 |

##### ****返回实例****：

**成功**：

json

{

"status": "success"}

**失败**：

json

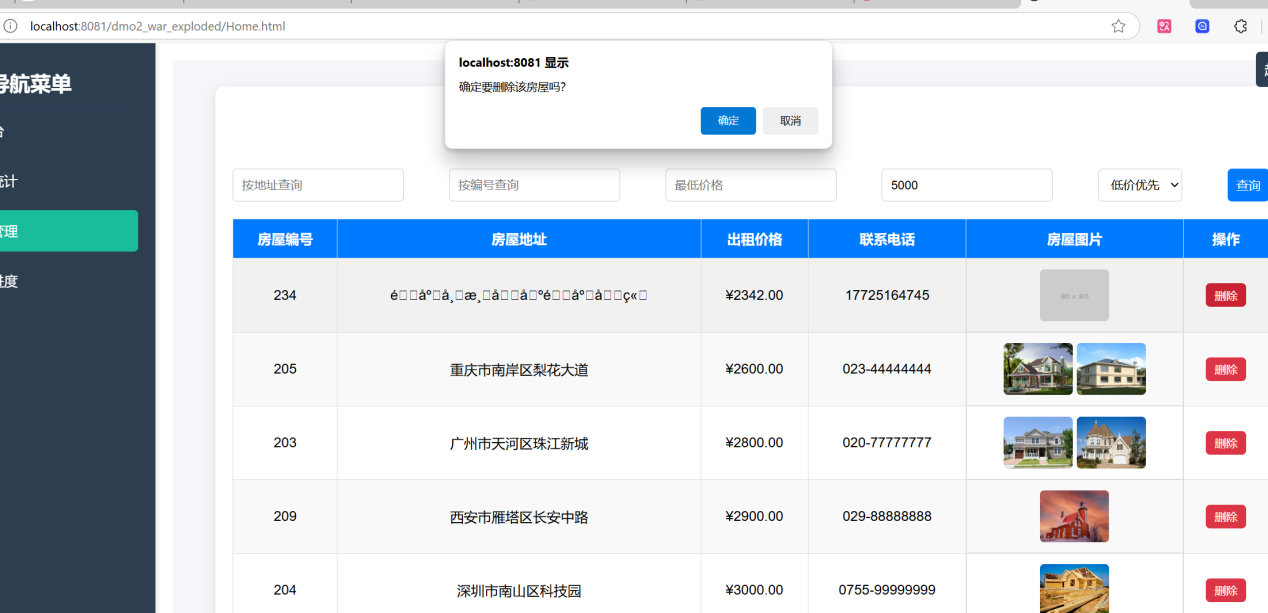
{

"status": "failure",

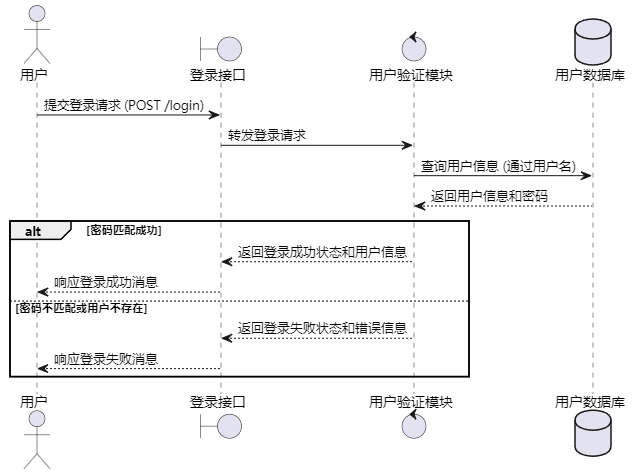
"message": "房屋ID不存在"}

##### ****返回参数说明****：

| **参数名称** | **类型** | **说明** |
| --- | --- | --- |
| status | string | success表示删除成功，failure表示删除失败。 |
| message | string | 返回失败时的具体原因。 |



#### ****接口4：登录接口****



* **功能**：实现用户的登录验证。
* **权限**：所有用户均可使用。
* **API请求地址**：/login
* **请求方式**：POST

##### ****请求实例****：

json

{

"username": "test\_user",

"password": "test\_password"}

##### ****请求参数说明****：

| **参数名称** | **类型** | **是否必填** | **说明** |
| --- | --- | --- | --- |
| username | string | 是 | 用户的用户名。 |
| password | string | 是 | 用户的登录密码。 |

##### ****返回实例****：

**成功**：

json

{

"success": true,

"message": "登录成功！"}

**失败**：

json

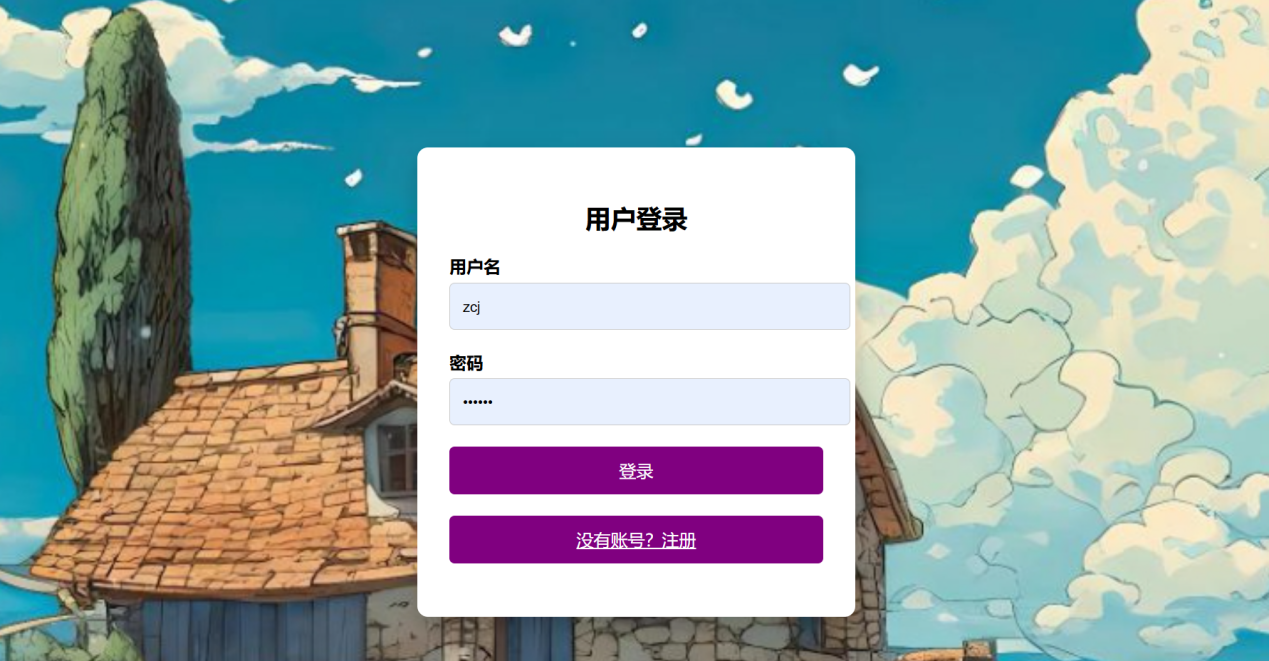
{

"success": false,

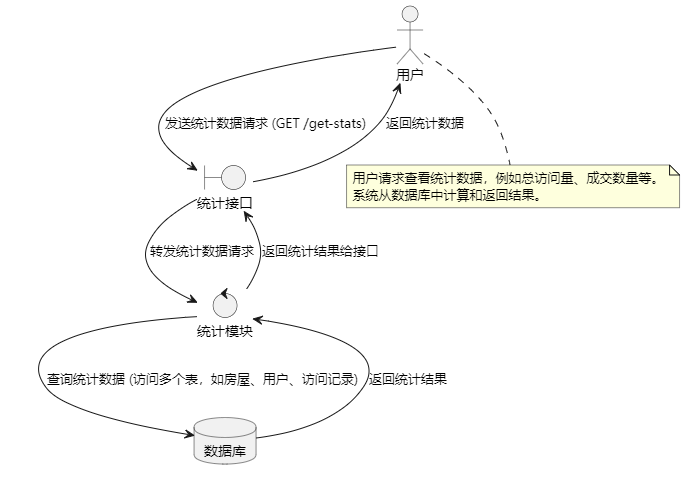
"message": "密码错误，请重试。"}

##### ****返回参数说明****：

| **参数名称** | **类型** | **说明** |
| --- | --- | --- |
| success | bool | true表示登录成功，false表示登录失败。 |
| message | string | 提示用户的登录结果信息。 |



#### ****接口5：********获取统计数据接口****



* **功能**：返回系统的统计数据，包括成交数量、访问量、房屋总数等。
* **权限**：无需登录即可调用。
* **API请求地址**：/get-stats
* **请求方式**：GET

##### ****请求实例****：

无请求参数。

##### ****返回实例****：

json

{

"totalTransactions": 45,

"totalVisits": 1200,

"totalHouses": 89,

"newUserCount": 10}

##### ****返回参数说明****：

| **参数名称** | **类型** | **说明** |
| --- | --- | --- |
| totalTransactions | int | 成交房屋的总数量。 |
| totalVisits | int | 系统访问量。 |
| totalHouses | int | 当前系统中房屋的总数量。 |
| newUserCount | int | 新增注册用户的数量。 |

