1. **类图**

@startuml

package "房屋管理系统" {

class "房屋" {

- house\_id : int

- 地址 : varchar(255)

- 出租价格 : decimal(8, 2)

- 联系电话 : varchar(15)

- 房屋图片 : varchar(255)

- 房屋描述 : text

- 房屋状态 : enum

+ 获取房屋信息() : void

}

class "用户" {

- user\_id : int

- 用户名 : varchar(50)

- 密码 : varchar(255)

- 真实姓名 : varchar(255)

- 联系电话 : varchar(15)

- 用户角色 : enum

- 注册日期 : date

+ 注册() : void

+ 登录() : void

+ 获取用户信息() : void

}

class "租赁记录" {

- lease\_id : int

- house\_id : int

- user\_id : int

- 租赁开始日期 : date

- 租赁结束日期 : date

- 支付金额 : decimal(8, 2)

- 租赁状态 : enum

+ 获取租赁记录() : void

}

class "房屋位置" {

- location\_id : int

- house\_id : int

- 城市 : varchar(50)

- 区域 : varchar(50)

- 详细地址 : varchar(255)

+ 获取房屋位置() : void

}

class "统计数据" {

- total\_visits : int

- total\_transactions : int

- total\_houses : int

+ 获取统计信息() : void

}

class "管理员" {

- admin\_id : int

- 用户名 : varchar(50)

- 密码 : varchar(255)

- 联系方式 : varchar(15)

+ 管理用户() : void

+ 管理房屋() : void

+ 获取统计信息() : void

}

"用户" --> "房屋": 租赁

"用户" --> "租赁记录": 产生记录

"房屋" --> "租赁记录": 关联租赁

"房屋" --> "房屋位置": 确定位置

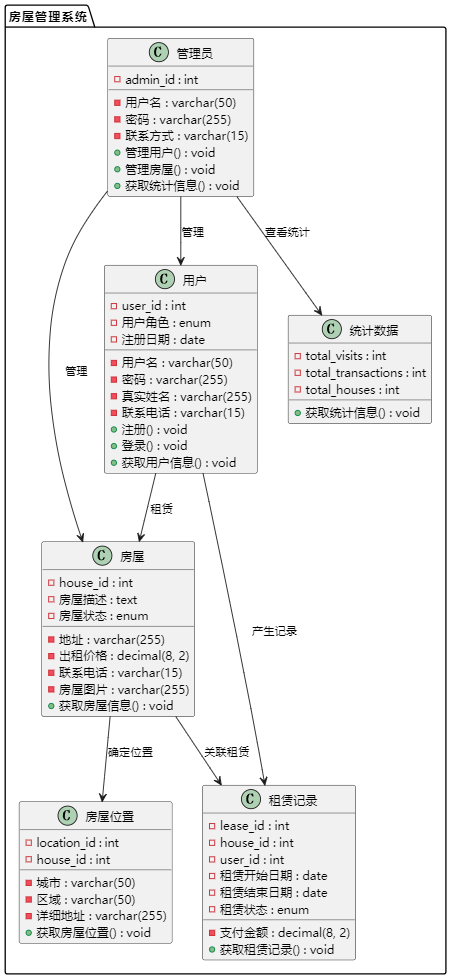
"管理员" --> "统计数据": 查看统计

"管理员" --> "用户": 管理

"管理员" --> "房屋": 管理

}

@enduml



2.

@startuml

actor 租客

actor 房东

boundary 房屋管理系统接口

control 房屋管理模块

control 用户管理模块

control 租赁管理模块

control 统计管理模块

租客 -> 房屋管理系统接口 : 租客发起房屋查询请求

房屋管理系统接口 -> 房屋管理模块 : 将查询请求转发至房屋管理模块

房屋管理模块 -> 房屋数据库 : 查询房屋数据

房屋管理模块 -> 房屋管理系统接口 : 将查询到的房屋信息返回系统界面

租客 -> 房屋管理系统接口 : 提交房屋租赁申请

房屋管理系统接口 -> 租赁管理模块 : 系统将租赁申请传递至租赁管理模块

租赁管理模块 -> 用户管理模块 : 验证租客身份及租赁权限

租赁管理模块 -> 房屋数据库 : 检查房屋状态及可用性

租赁管理模块 -> 租赁记录数据库 : 创建租赁记录

租赁管理模块 -> 房屋数据库 : 更新房屋状态为已租

租赁管理模块 -> 房屋管理系统接口 : 返回租赁成功信息

租客 -> 房屋管理系统接口 : 发起房屋退租申请

房屋管理系统接口 -> 租赁管理模块 : 将退租申请传递至租赁管理模块

租赁管理模块 -> 租赁记录数据库 : 查询对应的租赁记录

租赁管理模块 -> 房屋数据库 : 更新房屋状态为可租

租赁管理模块 -> 租赁记录数据库 : 更新租赁记录状态

租赁管理模块 -> 房屋管理系统接口 : 返回退租结果

房东 -> 房屋管理系统接口 : 发起新增房屋申请

房屋管理系统接口 -> 房屋管理模块 : 将新增房屋申请转发至房屋管理模块

房屋管理模块 -> 房屋数据库 : 插入房屋信息

房屋管理模块 -> 房屋管理系统接口 : 返回新增房屋成功信息

管理员 -> 房屋管理系统接口 : 查看统计信息

房屋管理系统接口 -> 统计管理模块 : 将统计请求传递至统计管理模块

统计管理模块 -> 房屋数据库 : 查询房屋相关数据

统计管理模块 -> 用户数据库 : 查询用户相关数据

统计管理模块 -> 租赁记录数据库 : 查询租赁相关数据

统计管理模块 -> 房屋管理系统接口 : 返回统计结果

note left of 租客

租客主要进行房屋查询、租赁申请和退租操作。

end note

note left of 房东

房东主要负责新增房屋以及查看房屋状态。

end note

note left of 房屋管理系统接口

系统接口作为用户与系统之间的桥梁，处理请求和返回结果。

end note

note left of 房屋管理模块

房屋管理模块处理与房屋相关的操作，包括查询和管理房屋信息。

end note

note left of 用户管理模块

用户管理模块负责验证用户身份及权限。

end note

note left of 租赁管理模块

租赁管理模块负责处理租赁相关操作，如申请和退租。

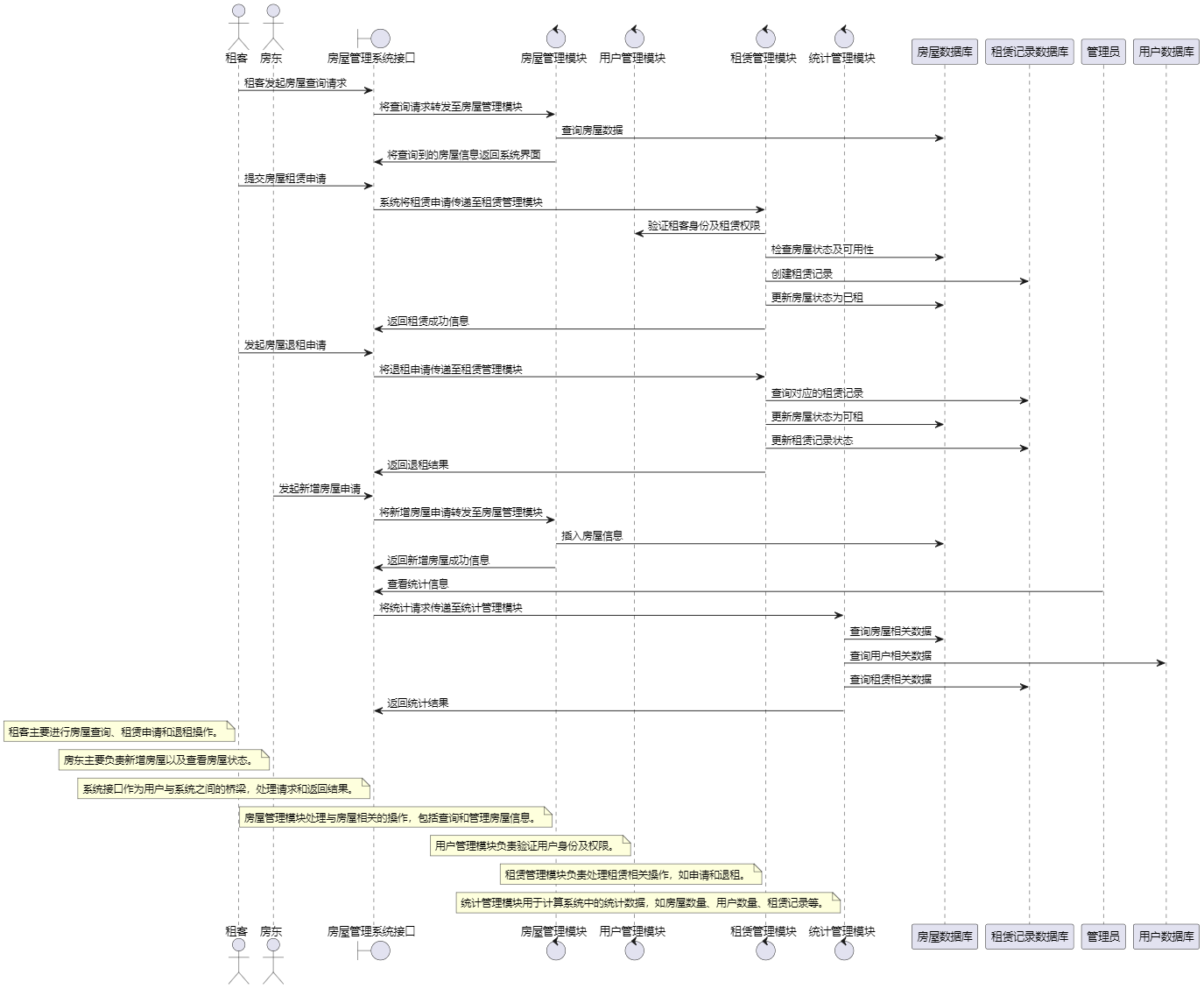
end note

note left of 统计管理模块

统计管理模块用于计算系统中的统计数据，如房屋数量、用户数量、租赁记录等。

end note

@enduml



3.时序图

@startuml

title 房屋管理系统主要业务协作图

actor 租客 as Renter

actor 房东 as Landlord

boundary 房屋管理系统接口 as Interface

control 房屋管理模块 as HouseModule

control 租赁管理模块 as LeaseModule

Renter -> Interface: 查询房屋

Interface -> HouseModule: 请求房屋信息

HouseModule -> Interface: 返回房屋信息

Interface -> Renter: 显示房屋信息

Renter -> Interface: 提交租赁申请

Interface -> LeaseModule: 租赁申请处理

LeaseModule -> HouseModule: 检查房屋可用性

HouseModule --> LeaseModule: 返回可用状态

LeaseModule -> Renter: 返回租赁结果

LeaseModule -> HouseModule: 更新房屋状态（若通过）

LeaseModule -> 租赁记录数据库: 记录租赁信息

Renter -> Interface: 提交退租请求

Interface -> LeaseModule: 退租处理

LeaseModule -> 租赁记录数据库: 查询租赁记录

LeaseModule -> HouseModule: 更新房屋状态为可租

LeaseModule -> 租赁记录数据库: 更新租赁记录

LeaseModule -> Renter: 返回退租结果

Landlord -> Interface: 新增房屋

Interface -> HouseModule: 新增房屋信息

HouseModule -> 房屋数据库: 插入房屋信息

HouseModule -> Landlord: 返回新增结果

Landlord -> Interface: 查看租赁状态

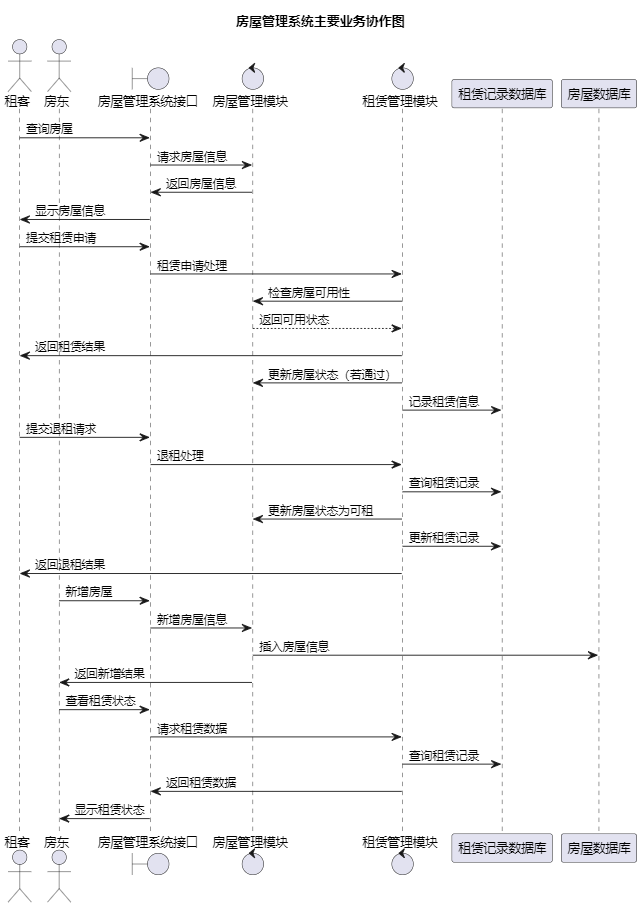
Interface -> LeaseModule: 请求租赁数据

LeaseModule -> 租赁记录数据库: 查询租赁记录

LeaseModule -> Interface: 返回租赁数据

Interface -> Landlord: 显示租赁状态

@enduml



4.状态图：

@startuml

title 房屋状态管理图

[\*] -> 空置 : 初始状态为房屋空置

空置 -> 已出租 : 租客签订租赁合同

已出租 -> 空置 : 租客退租并归还房屋

已出租 -> 欠租 : 租客未按时支付租金

欠租 -> 已出租 : 租客补缴租金

欠租 -> 空置 : 强制退租，房屋归还

note left of 空置

在空置状态下，

房屋可以被查看和租赁。

end note

note left of 已出租

表示房屋已被租赁，

租客正在使用中。

end note

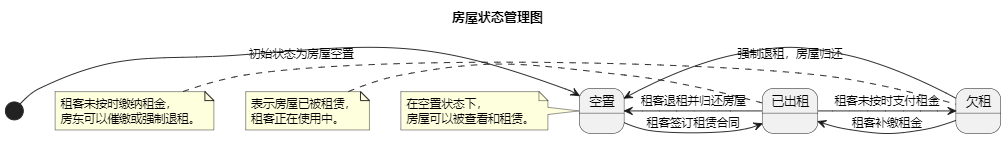
note left of 欠租

租客未按时缴纳租金，

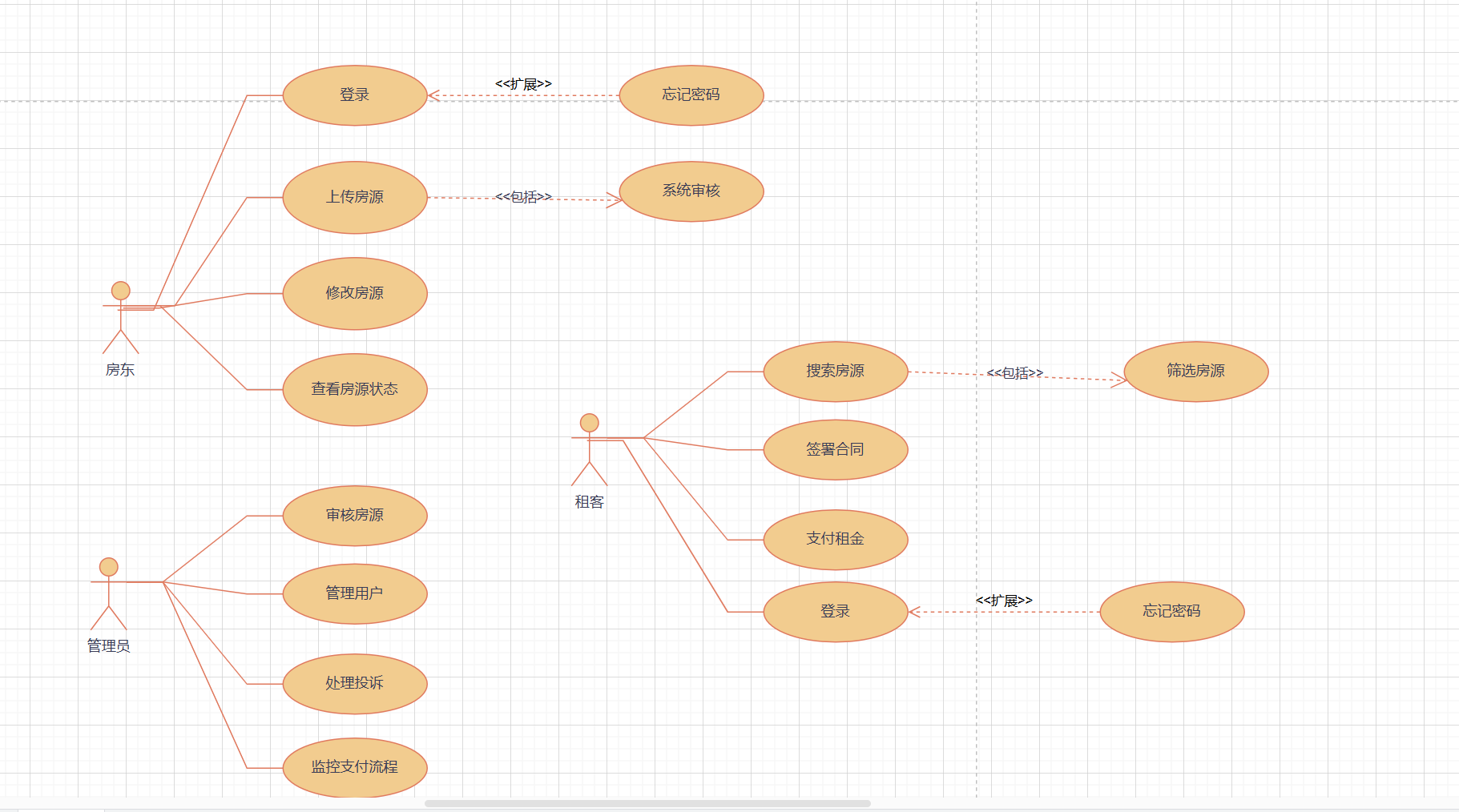
房东可以催缴或强制退租。

end note

@enduml



1. 用例图



1. **活动图**

@startuml

start

: 用户登录房屋管理系统;

if (是租客?) then (是)

if (是否查询房源?) then (是)

: 租客输入查询条件;

: 系统根据条件查询房源信息;

: 将查询到的房源信息显示给租客;

else if (是否提交租房申请?) then (是)

: 租客选择要租赁的房屋;

: 系统检查房屋状态及租客信息;

if (房屋是否可租?) then (是)

: 系统记录租赁信息;

: 更新房屋状态为已租;

: 提示租客租赁成功;

else (否)

: 提示租客房屋不可用原因;

endif

else if (是否取消租房?) then (是)

: 租客提交取消租赁请求;

: 系统检查租赁记录;

: 更新房屋状态为可租;

: 提示租客取消成功;

endif

else if (是房东?) then (是)

if (是否新增房源?) then (是)

: 房东输入新增房源信息;

: 系统保存房源信息到数据库;

: 提示房东房源发布成功;

else if (是否修改房源信息?) then (是)

: 房东选择需要修改的房源;

: 修改房源信息;

: 系统保存修改后的信息;

: 提示房东修改成功;

else if (是否删除房源?) then (是)

: 房东选择需要删除的房源;

: 系统检查房源状态;

if (房屋是否已租?) then (否)

: 系统删除房源信息;

: 提示房东删除成功;

else (是)

: 提示房东无法删除原因;

endif

endif

else if (是管理员?) then (是)

if (是否管理用户信息?) then (是)

: 管理员选择用户管理操作;

: 系统执行用户管理功能;

: 提示管理员操作结果;

else if (是否查看系统统计?) then (是)

: 管理员请求查看统计数据;

: 系统计算统计数据;

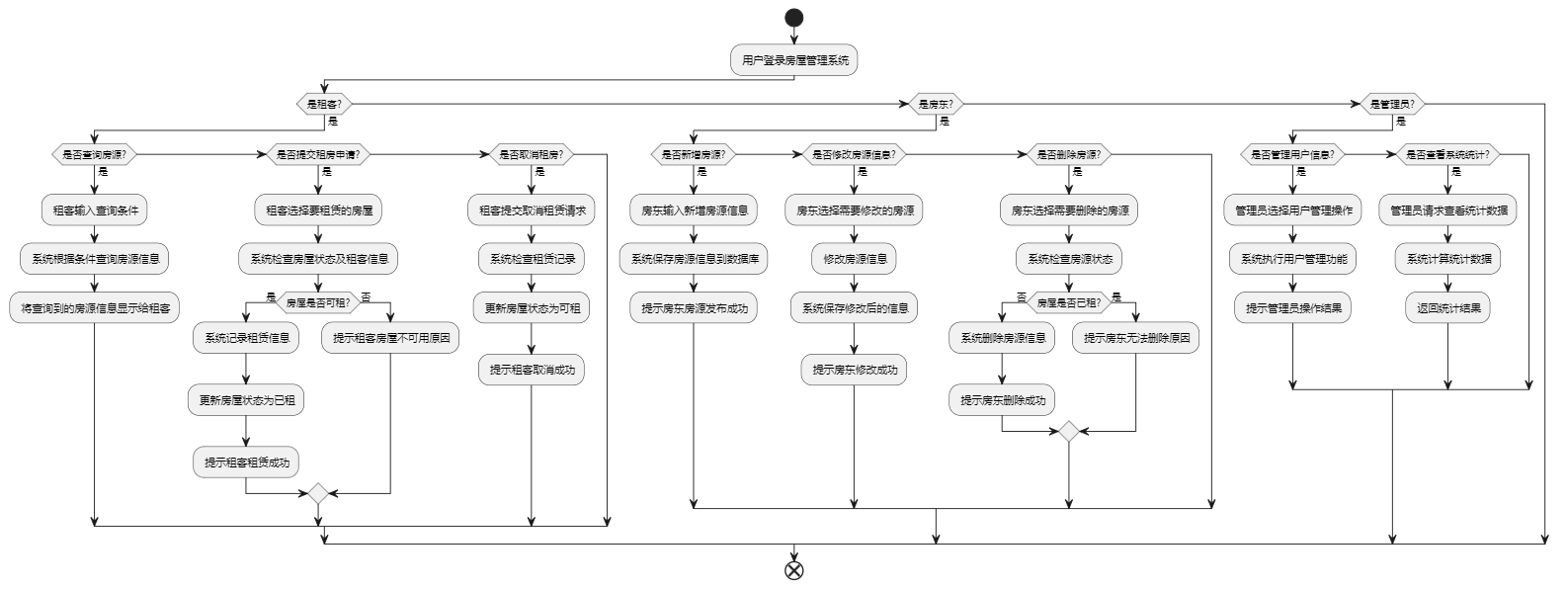
: 返回统计结果;

endif

endif

end

@enduml



7.组件图

@startuml

package "房屋管理系统" {

component "租客端界面" as TenantUI {

interface "查询房源接口"

interface "租赁接口"

interface "取消租赁接口"

}

component "房东端界面" as LandlordUI {

interface "新增房源接口"

interface "修改房源接口"

interface "删除房源接口"

}

component "管理员端界面" as AdminUI {

interface "用户管理接口"

interface "统计查询接口"

}

component "房源管理模块" as HouseManager {

interface "房源数据查询服务"

interface "房源数据更新服务"

}

component "租赁管理模块" as LeaseManager {

interface "租赁处理服务"

interface "取消租赁服务"

interface "库存检查服务"

}

component "用户管理模块" as UserManager {

interface "用户信息管理服务"

}

component "系统统计模块" as StatsManager {

interface "统计数据生成服务"

}

component "房源数据库" as HouseDB {

interface "存储房源数据"

}

component "用户数据库" as UserDB {

interface "存储用户数据"

}

component "租赁记录数据库" as LeaseRecordDB {

interface "存储租赁记录数据"

}

TenantUI --> HouseManager : 使用房源数据查询服务

TenantUI --> LeaseManager : 使用租赁处理服务、取消租赁服务、库存检查服务

LandlordUI --> HouseManager : 使用房源数据更新服务

AdminUI --> UserManager : 使用用户信息管理服务

AdminUI --> StatsManager : 使用统计数据生成服务

HouseManager --> HouseDB : 读写房源数据

LeaseManager --> HouseDB : 更新库存数据

LeaseManager --> LeaseRecordDB : 读写租赁记录数据

LeaseManager --> UserManager

