**社区服务平台详细**

**（版本v5.0.0）**

Java实训小组

2022年9月7日

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 成文信息 | | | | | | | |
| 主题词： | | 详细设计 | | | | | |
| 作 者： | | Java实训小组 | | 文档类别： | | | 技术方案、设计文档 |
| 审 核： | |  | |
| 批 准： | |  | | 文档性质： | | | 正式稿 |
| 主 送： | |  | | 存档日期： | | | 2022.9.7 |
| 抄 送： | |  | | 发布日期： | | | 未知 |
| 签收信息 | | | | | | | |
| 发送方Java小组 | | | 接收方：实训指导老师 | 接收方： | | 接收方： | |
| 代表人：张哲维 | | | 代表人： | 代表人： | | 代表人： | |
| 日期：2022.9.7 | | | 日期：2022.9.8 | 日期： | | 日期： | |
| 变更信息 | | | | | | | |
| 版本5.0 | 原因 | | | | 作者 | 日期2022.9.7 | |

目录

[第一章 引言 1](#_Toc433639044)

[1.1 编写目的 1](#_Toc433639045)

[1.2 背景 1](#_Toc433639046)

[1.3 定义 1](#_Toc433639047)

[1.4 列出有关的参考资料 1](#_Toc433639048)

[第二章 程序系统结构 2](#_Toc433639049)

[2.1 体系架构 2](#_Toc433639050)

[2.2 程序结构设计 3](#_Toc433639051)

[2.2.1 名称设计 3](#_Toc433639052)

[2.2.2 目录规范 3](#_Toc433639053)

[2.2.3 配置文件规划 5](#_Toc433639054)

[第三章 数据库模块设计说明 6](#_Toc433639055)

[3.1.1 模块描述 6](#_Toc433639056)

[3.1.2 功能 6](#_Toc433639057)

[3.1.3 性能 7](#_Toc433639058)

[3.1.4 类设计 7](#_Toc433639059)

[3.1.5 算法 10](#_Toc433639060)

[第四章 外地人员模块设计说明 10](#_Toc433639061)

[4.1.1 模块描述 10](#_Toc433639062)

[4.1.2 功能 11](#_Toc433639063)

[4.1.3 人机交互 11](#_Toc433639064)

[4.1.4 类设计 14](#_Toc433639065)

[第五章 通知模块设计说明 15](#_Toc433639067)

[5.1.1 模块描述 15](#_Toc433639068)

[5.1.2 功能 16](#_Toc433639069)

[5.1.3 人机交互 16](#_Toc433639070)

[5.1.4 类设计 19](#_Toc433639071)

[第六章 社区工作人员模块设计说明 20](#_Toc433639073)

[6.1.1 模块描述 20](#_Toc433639074)

[6.1.2 功能 21](#_Toc433639075)

[6.1.3 人机交互 21](#_Toc433639076)

[6.1.4 类设计 24](#_Toc433639077)

[第七章 社区居民模块设计说明 25](#_Toc433639079)

[7.1.1 模块描述 25](#_Toc433639080)

[7.1.2 功能 26](#_Toc433639081)

[7.1.3 人机交互 26](#_Toc433639082)

[7.1.4 类设计 28](#_Toc433639083)

[第八章 办事申请模块设计说明 28](#_Toc433639084)

[8.1.1 模块描述 28](#_Toc433639085)

[8.1.2 功能 29](#_Toc433639086)

[8.1.3 人机交互 29](#_Toc433639087)

[8.1.4 类设计 31](#_Toc433639088)

[8.1.5 算法 31](#_Toc433639089)

# 引言

## 编写目的

编写本详细设计的目的是：在社区服务平台需求细化分析的基础上，完成系统的结构设计，划分系统模块并明确模块之间的关系，同时给出相关的用户界面设计。

本文的预期读者为： 社区普通居民，社区服务工作者。社区服务平台的设计、开发人员。软件的验收、评审人员。

## 背景

待开发的软件为社区服务平台。 此项目由组长张哲维提出，组员设计和开发。提升餐厅整体形象。本软件的目的是方便社区居民和社区服务工作者的联系与工作效率。

## 定义

1、系统：若未特别指出，统指社区服务系统。

2、平台：若未特别指出，特指本系统集成运行平台。

3、SQL：Structured Query Language（结构化查询语言）。

## 列出有关的参考资料

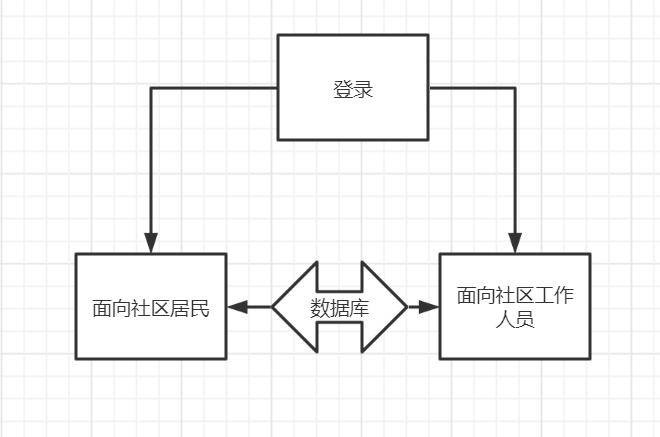
1、《社区服务平台需求规格说明书》

2、《社区服务系统概要设计说明书》

# 程序系统结构

## 体系架构

从功能构成的角度出发划分系统的职能层次，按照从底层到顶层，服务到应用的划分方法，社区服务系统基本上可以划分为四部分：登录系统，工作人员发布与管理系统，社区居民接收通知与申请系统，数据的储存系统。其基本组成图如下：



（一）登录

本部分包括工作人员与社区工作人员的登录。同时该部分有邮箱验证功能，通过邮箱与验证码登录，方便了不愿意记住密码的人。

（二）工作人员发布与管理系统

工作人员通过本部分可以实现发布通知，管理社区居民的办事申请。管理所注册的账号。管理外来人员的疫情管理。

（三）社区居民接收通知与申请系统

社区居民通过该程序可以接受到社区的最新消息和最新的防疫政策。同时居民还可以向社区发出申请。社区工作人员可以对居民的申请进行审批。

（四）数据库存储系统

该程序完全依赖于对于数据库的增删查改。这是本程序重要的一部分。对于账号的管理，对于通知的管理和都是基于对数据库的操作。

## 程序结构设计

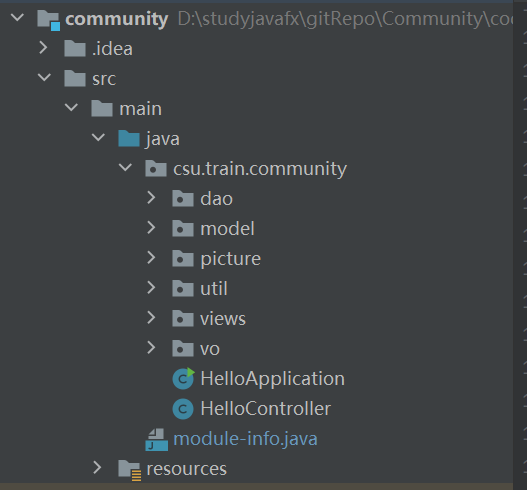
### 名称设计

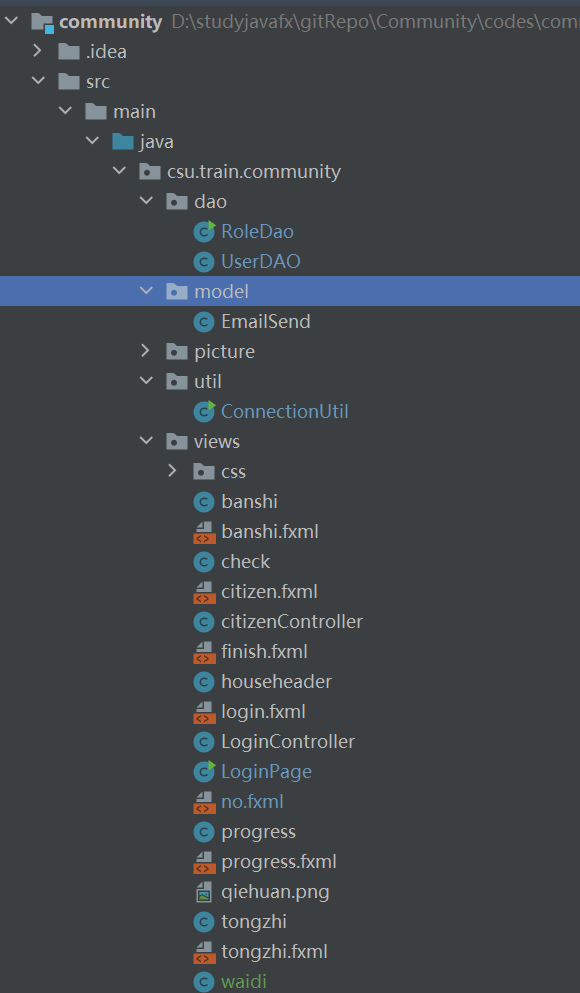
名称的设计尽量使用英文，遵循驼峰命名法，不同的单词之间可以使用下划线。简洁是最重要的。

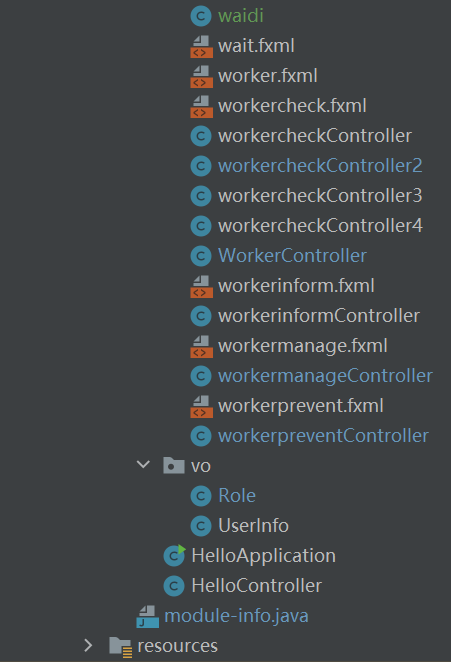
### 目录规范

为了社区的部署程序和工作数据，避免因系统各组件目录重复导致的错误，故需要统一规划应用软件相关目录结构。

本系统各组件采用以下目录结构存储应用程序及相关数据。







### 配置文件规划

系统中存在的配置文件规定如下：

**JDBC数据库参数配置文件：**

配置文件路径：

**CLASSPATH**

|--**jdbc.xml**

配置文件类型：XML文件

配置文件格式：

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!-- 根节点 -->

<jdbc>

<!-- 驱动名 -->

<drivername>com.mysql.jdbc.Driver</drivername>

<!-- 连接字符串 -->

<connectionstring>jdbc:mysql://localhost:3306/ordersys

</connectionstring>

<!-- 数据库用户名 -->

<dbmsusername>root</dbmsusername>

<!-- 数据库密码 -->

<dbmspassword>root</dbmspassword>

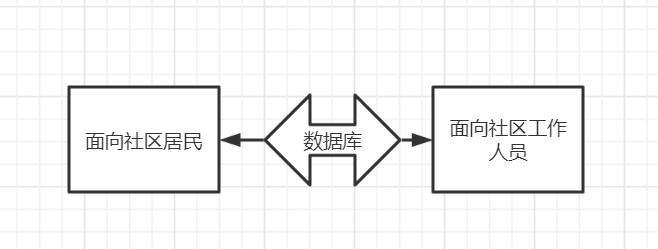
</jdbc>

# 数据库模块设计说明

### 模块描述

本模块是系统功能组件支撑组件，用于储存账号，储存通知信息，储存外来人员信息。

#### 模块图：



### 性能

#### 精度

必须保证通知的精确发布，同时社区居民的申请可以精确的到达工作人员的手中。

#### 灵活性

可以发送各种的通知，同时进行各种的申请。

#### 时间特性

对于数据库的增删查改，是需要耗费一些时间的，但是不可以让使用者等待过长的时间。

#### 输入项

工作人员的一段可以实现对通知的输入与发布。社区居民端，可以实现输入申请。

#### 输出项

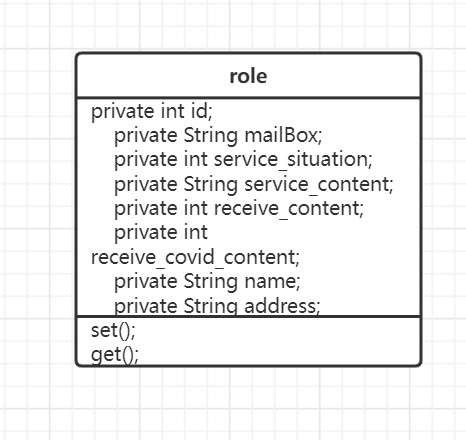
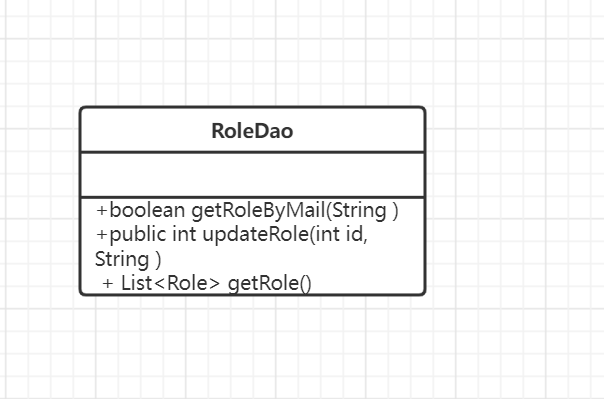
　　工作人员端可以输出各种的申请，以及外地人员的情况，注册居民的管理。居民端可以输出通知，可以查看申请的办事进度。

### 类设计

给出本模块的类设计，**包括类图和类说明**

在本程序中，最重要的是Role这一类，这一类就相当于居民的角色，拥有居民的各种属性与方法。

#### 类图：

#### 类说明

Role类是一个所有人的抽象类，有所人的属性与方法。

RoleDao是一个操作数据库的类。

### 算法

该程序不断的增删查改数据库来实现各种的功能。

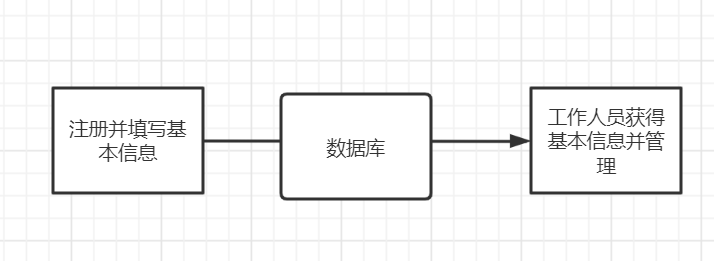
# 外地人员模块设计说明

### 模块描述

社区管理者可以通过该程序对外地人进行一定的管理，同时可以防范疫情的传播。

### 功能

#### 模块图：



### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 工作人员管理外来人员 | 外来人员输入基本信息 | 工作人员获得信息并用于疫情 |
|  |  |  |
|  |  |  |
|  |  |  |

#### 输入项

　　外地人员注册之后可以上传自己的疫情相关的数据，来配合社区工作人员进行疫情的传播防范。

#### 输出项

　　外地人员可以获得最新的疫情防控政策，同时关注核酸检测的时间。

### 类设计

#### 类说明

外地人员这一个类与Role相似，只是在原本Role的属性上加上外地人员的属性，比如从哪里来，做了几次核酸。

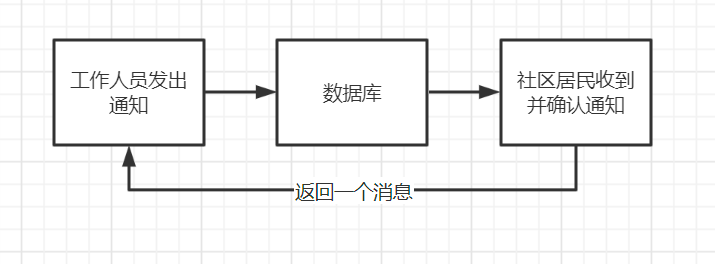
# 通知模块设计说明

### 模块描述

工作人员可以发布通知，居民可以接受通知，并确认收到通知。

### 功能

#### 模块图：



### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 工作人员发布通知 | 通知内容 | 无 |
| 社区居民接受通知 | 无 | 已接受的消息 |
|  |  |  |
|  |  |  |

#### 输入项

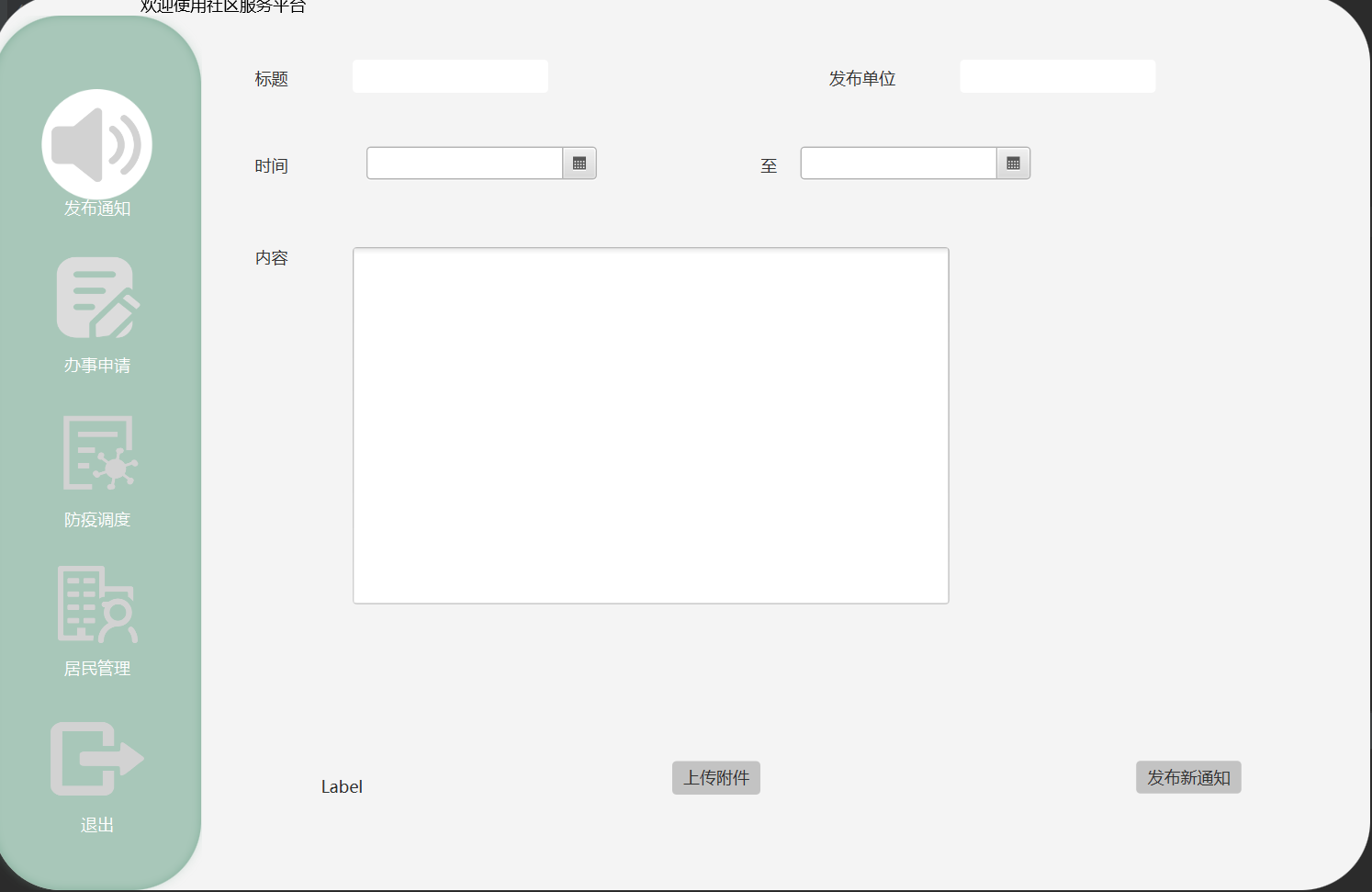
　　工作人员输入通知的具体信息。

#### 输出项

　　居民获得通知的内容，并确认。

#### 界面设计

1. 界面一：发布通知



1. 界面二：接受通知



### 类设计

#### 类说明

通知并没用单独成类，只是在Role和数据库上进行操作。

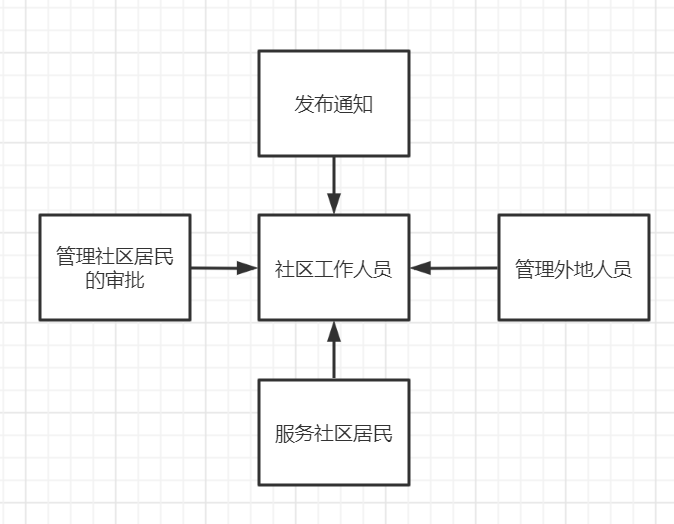
# 社区工作者模块设计说明

### 模块描述

社区工作人员以来于此程序，可以实现发布通知，对于居民的申请进行审批，管理外地人员，管理社区的全体居民。

### 功能

#### 模块图：



### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 发布通知 | 通知内容 | 无 |
| 管理外来人员 | 无 | 无 |
| 管理社区居民的审批 | 审批结果 | 无 |
| 管理居民 | 无 | 无 |

#### 输入项

发布通知，发布审批结果。

#### 输出项

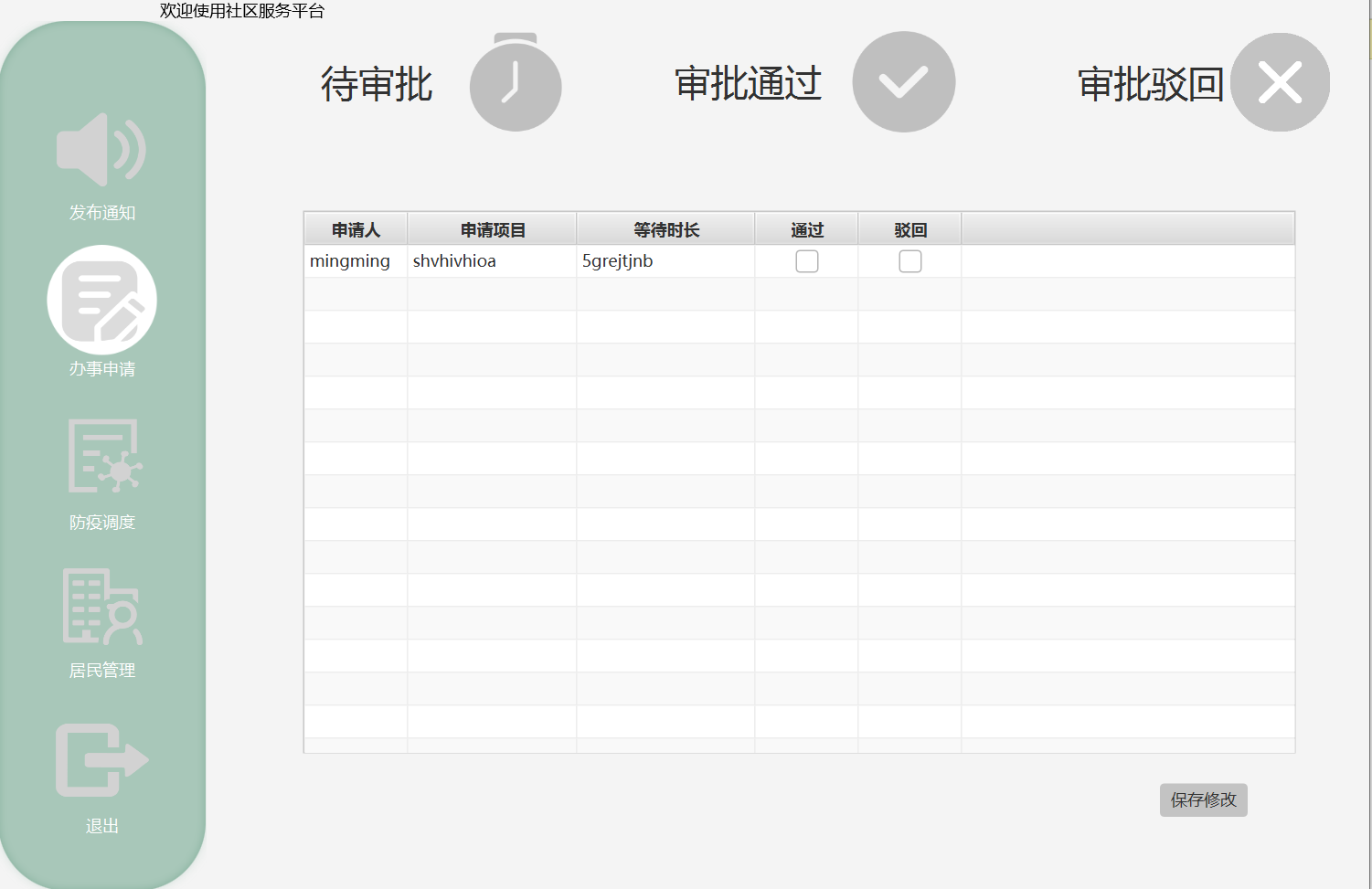
　　所用居民的信息都可以展示在工作人员的界面上。

#### 界面设计

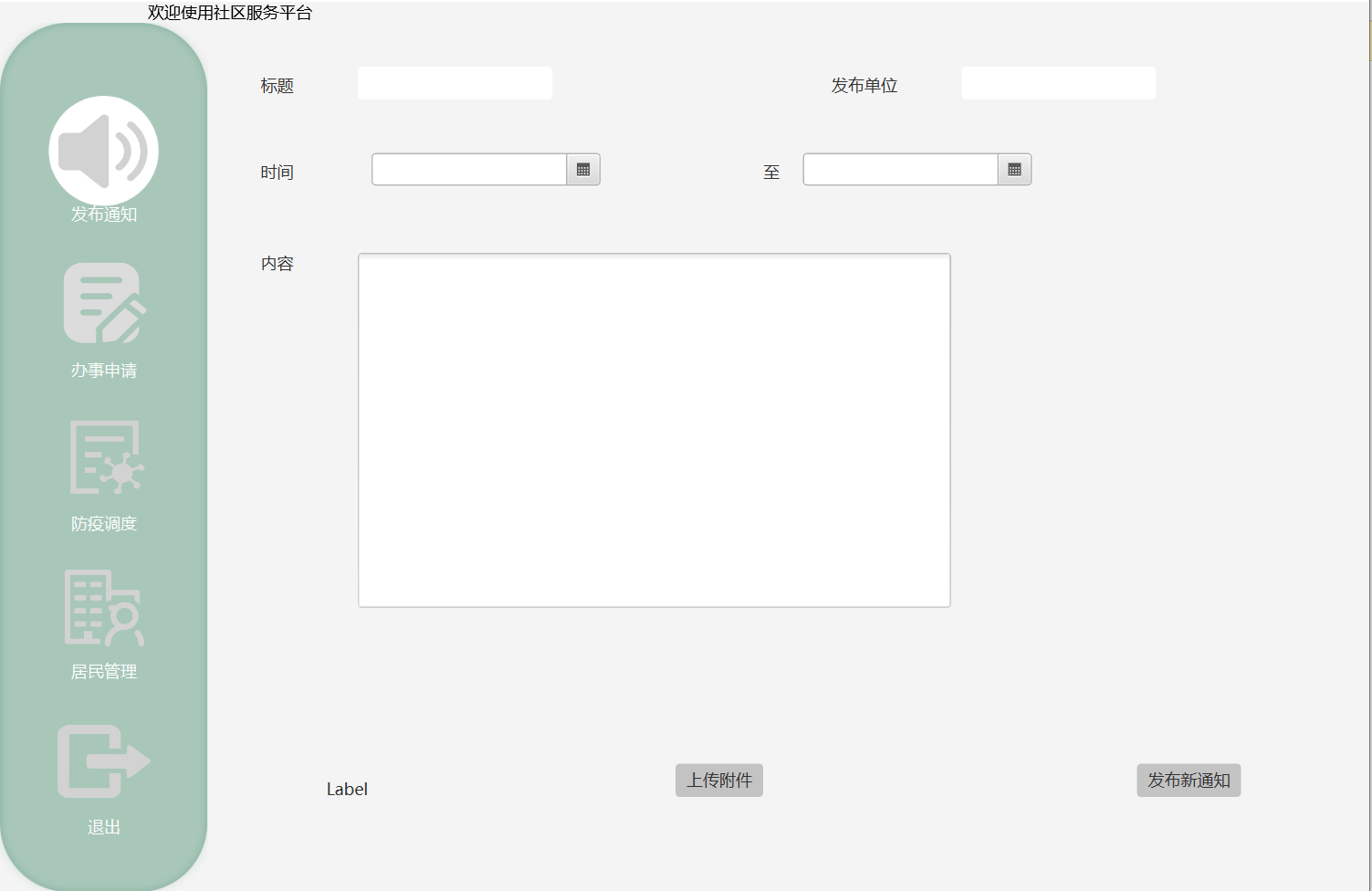
1. 界面一：登录界面



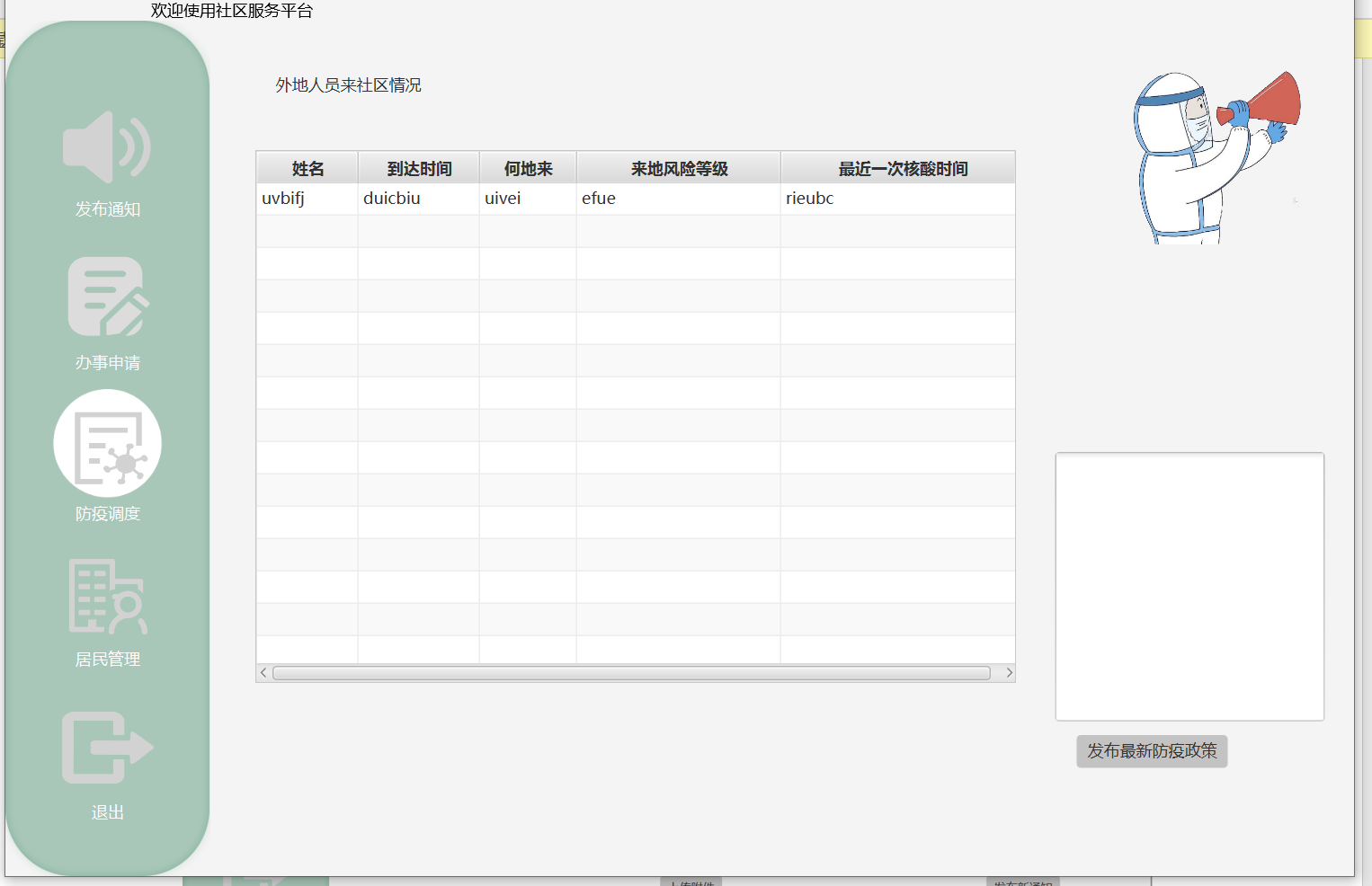
1. 界面二：审批界面



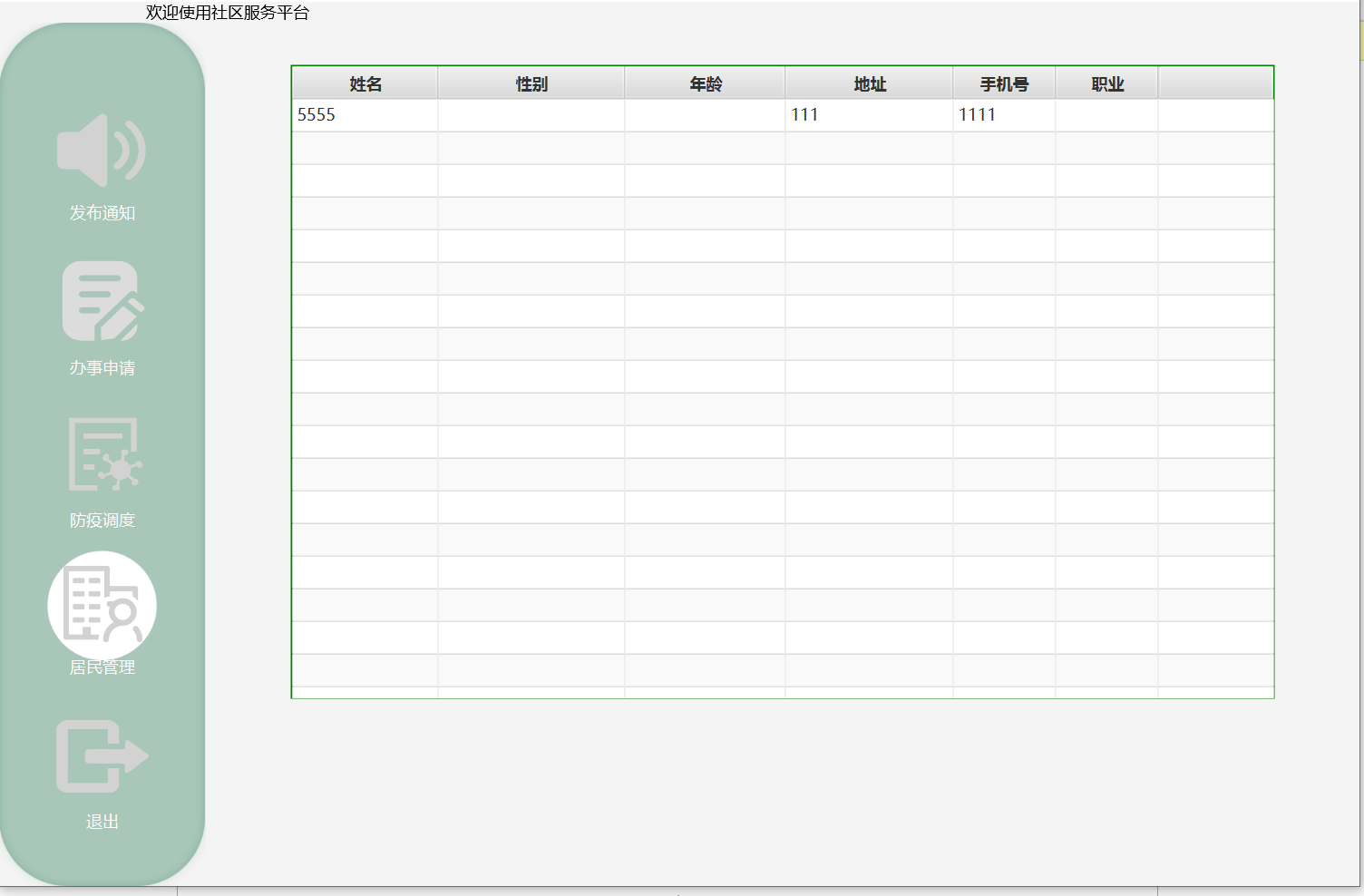
1. 界面三：发布通知



1. 界面四：防疫调度



1. 界面五：居民管理



### 类设计

#### 类说明：

在对表格进行操作的过程中，我在表格中添加CheckBox，为了将CheckBox与数据进行连接，我新建了Checkbox类，使之被勾选时，可以与数据建立连接。

public class Checkbox  
{  
    CheckBox checkbox=new CheckBox();  
    public ObservableValue<CheckBox> getCheckBox()  
    {  
        return new  ObservableValue<CheckBox>() {  
            @Override  
            public void addListener(ChangeListener<? super CheckBox> listener) {  
   
            }  
   
            @Override  
            public void removeListener(ChangeListener<? super CheckBox> listener) {  
   
            }  
   
            @Override  
            public CheckBox getValue() {  
                return checkbox;  
            }  
   
            @Override  
            public void addListener(InvalidationListener listener) {  
   
            }  
   
            @Override  
            public void removeListener(InvalidationListener listener) {  
   
            }  
        };  
    }  
    public Boolean isSelected()  
    {  
        return checkbox.isSelected();  
    }  
}

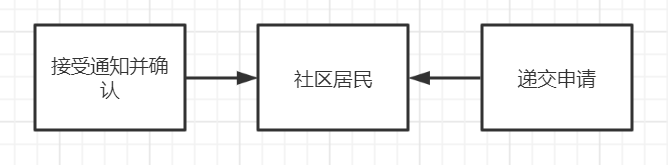
# 社区居民模块设计说明

### 模块描述

社区居民可以通过该程序查看并确认通知，进行办事申请，并查看进度。

### 功能

#### 模块图：



### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 接受通知 | 无 | 已接受的消息 |
| 递交申请 | 申请的项目 | 无 |

#### 输出项

　　输出通知和审批的进度和结果。

#### 界面设计

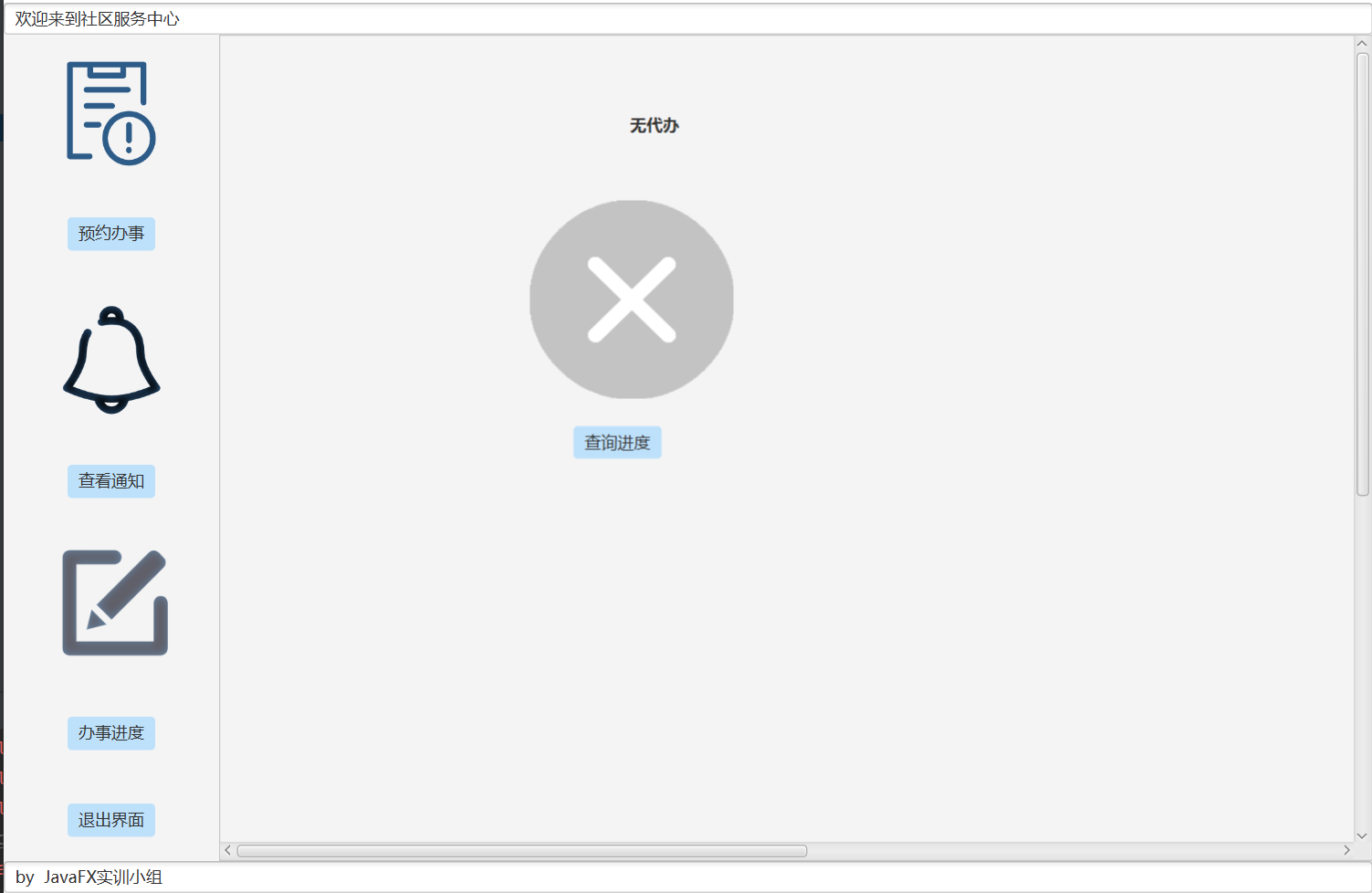
1. 界面一：申请办事



1. 界面二：查看通知



1. 界面三：办事进度



### 类设计

#### 类说明：

在这一部分中，仍然是基于对Role和数据库的操作进行的。

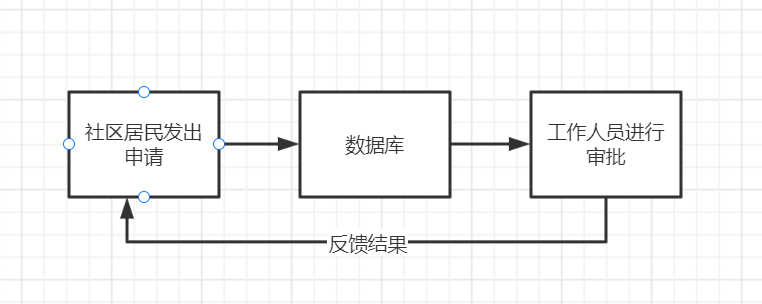
# 办事申请模块设计说明

### 模块描述

办事申请模块分布在两个界面上，社区居民可以递交申请，工作人员对申请进行审批于驳回。

### 功能

#### 模块图：



### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 居民递交申请 | 申请项目 | 无 |
| 工作人员审批 | 无 | 审批结果 |
| 反馈结果 | 审批 | 审批结果 |

#### 输入项

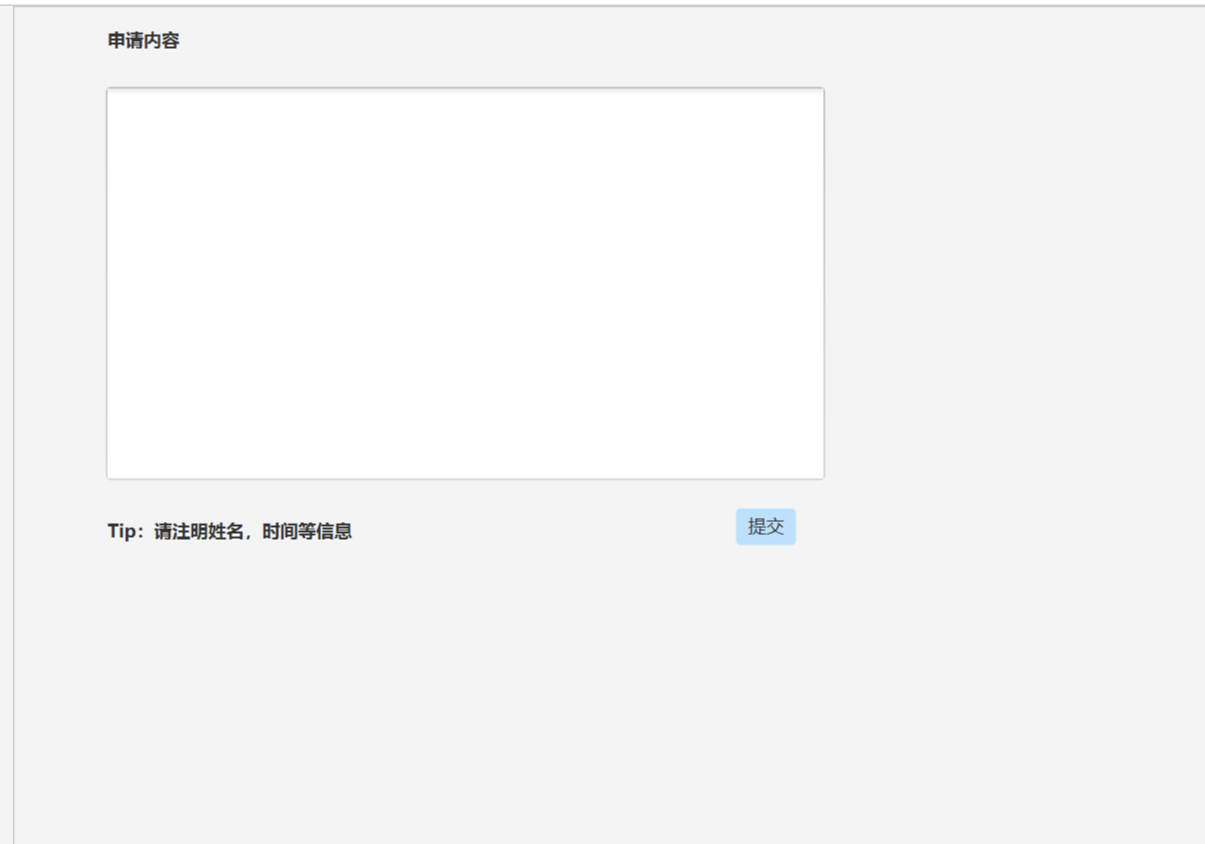
社区居民递交申请。

#### 输出项

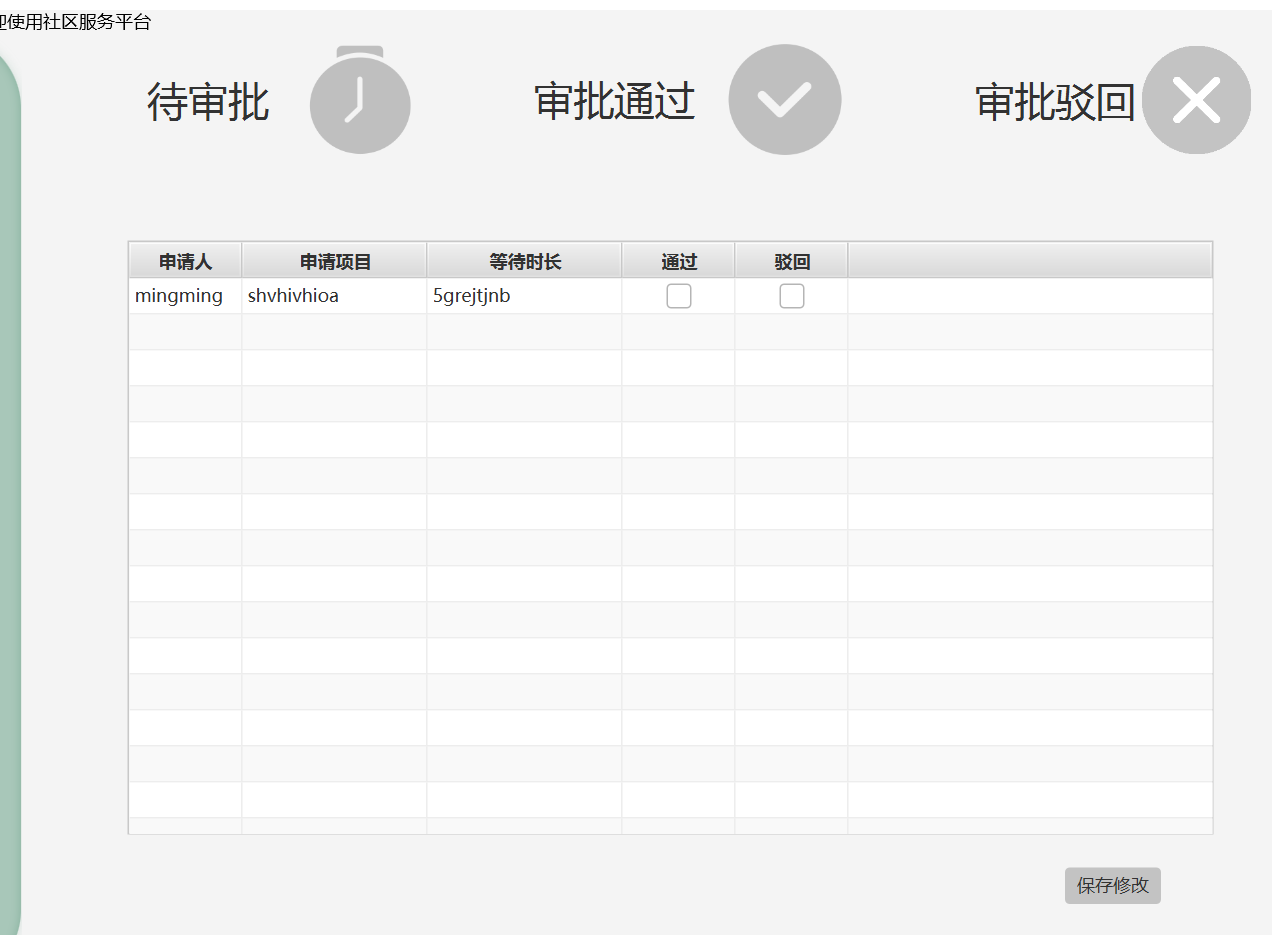
　　工作人员审批之后，返回一个结果。

#### 界面设计

1. 界面一：申请界面



1. 界面二：审批界面



### 类设计

#### 类说明：

办事申请中有用到Checkbox类，通过是否被勾选来对数据库的数据进行相应的操作。Checkbox类使得勾选框与数据的连接十分方便，避免了复杂的追踪。

### 算法

**工作人员审批之后向社区居民反馈结果。**