

进销存管理子系统

设计报告

组长 朱凯鑫

组员 濮昊昀、裘世超、丁立洲、

刘邵琛、尤博林、王昶昊

指导老师 陈中育

浙江师范大学软件工程专业

2018-12-12

目录

[一、引言 4](#_Toc4933113)

[1.1编写目的 4](#_Toc4933114)

[1.2子系统项目概述 4](#_Toc4933115)

[二、一般性描述 4](#_Toc4933116)

[2.1 系统用户特征 4](#_Toc4933117)

[三、子系统功能描述 4](#_Toc4933118)

[3.1 功能需求概述 4](#_Toc4933119)

[3.2 系统总体流程图 5](#_Toc4933120)

[3.3 各系统间关系图 6](#_Toc4933121)

[四、界面层设计 7](#_Toc4933122)

[4.1 登录界面 7](#_Toc4933123)

[4.1.1 登录界面设计草图 7](#_Toc4933124)

[4.1.2 界面事件 7](#_Toc4933125)

[4.2 主界面 8](#_Toc4933126)

[4.2.1 界面设计草图 8](#_Toc4933127)

[4.2.2 界面事件 8](#_Toc4933128)

[4.3 概览功能界面 9](#_Toc4933129)

[4.3.1 界面设计草图 9](#_Toc4933130)

[五.逻辑层设计(DFD) 10](#_Toc4933131)

[5.1.显示所有线索 10](#_Toc4933132)

[5.2.显示地区线索 11](#_Toc4933133)

[5.3.显示时间线索 12](#_Toc4933134)

[六.数据层设计(Data Base Design) 13](#_Toc4933135)

[6.1 数据表设计 13](#_Toc4933136)

[6.2 代码实现 14](#_Toc4933137)

[6.2.1数据库基本操作的封装方便调用 14](#_Toc4933138)

[6.2.2主要的业务实现类 17](#_Toc4933139)

[六、结果分析 19](#_Toc4933140)

[七、运行环境 21](#_Toc4933141)

# 一、引言

## 1.1编写目的

本系统主要实现对投诉信息进行管理。可以通过举报反映地址和处理时间进行查询。

## 1.2子系统项目概述

投诉线索筛查系统是药品公司的主要系统之一，直接对公司政府管理人员提供数据支持，政府各职能部门也以此为依据作出相应的服务管理，投诉线索筛查系统通过对群众来电的举报反映的统计和调用来提供数据分析与查询，有利于政府人员更快定位到所需要联系的地区与部门。

# 二、一般性描述

## 2.1 系统用户特征

本系统面对所有需要反映线索的职能部门机构。

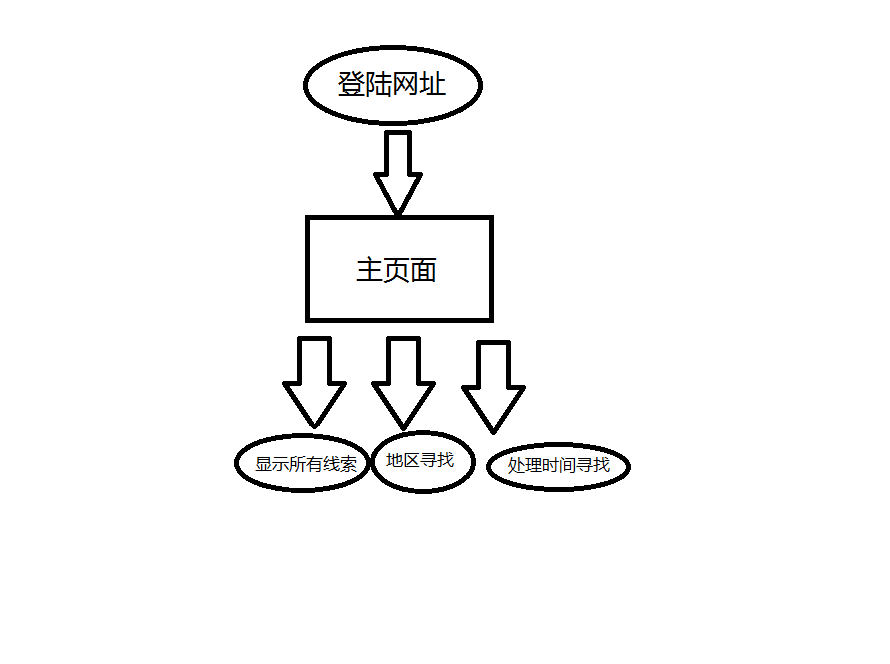
# 三、子系统功能描述

## 3.1 功能需求概述

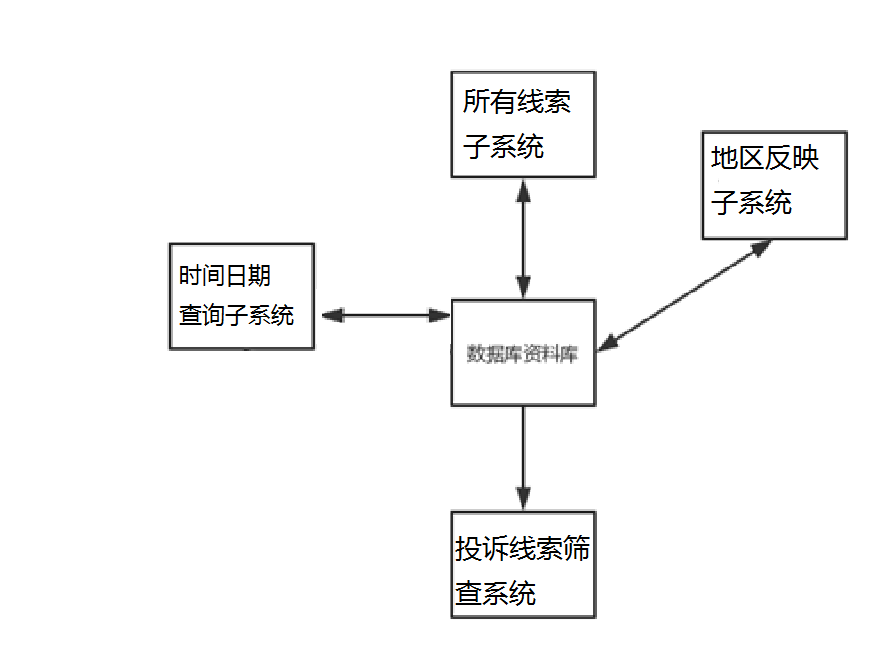
各个菜单选项功能的描述：首先进入登录界面，用户无需输入密码进入操作界面，界面有显示所有线索，通过线索查找功能，其中操作有：

1. Show All:展示所有问题及反映的线索。
2. Show Address:展示该地区集中反应的线索
3. Show Reaction:展示所有被处理时间节点的线索。

## 3.2 系统总体流程图



## 3.3 各系统间关系图



# 四、界面层设计

## 4.1 登录界面

### 4.1.1 登录界面设计草图



### 4.1.2 界面事件

|  |  |  |
| --- | --- | --- |
| 事件名称 | 事件激发 | 事件内容 |
| 查询所有信息 | go | 单机submit按钮显示所有内容 |
| 选择查询地区 | goAddress | 在option中选择需要查询的地区 |
| 通过时间节点来查询 | goTime | 点击此按钮，出发事件：  把text中内容提交进行查询 |

## 4.2 主界面

### 4.2.1 界面设计草图



### 4.2.2 界面事件

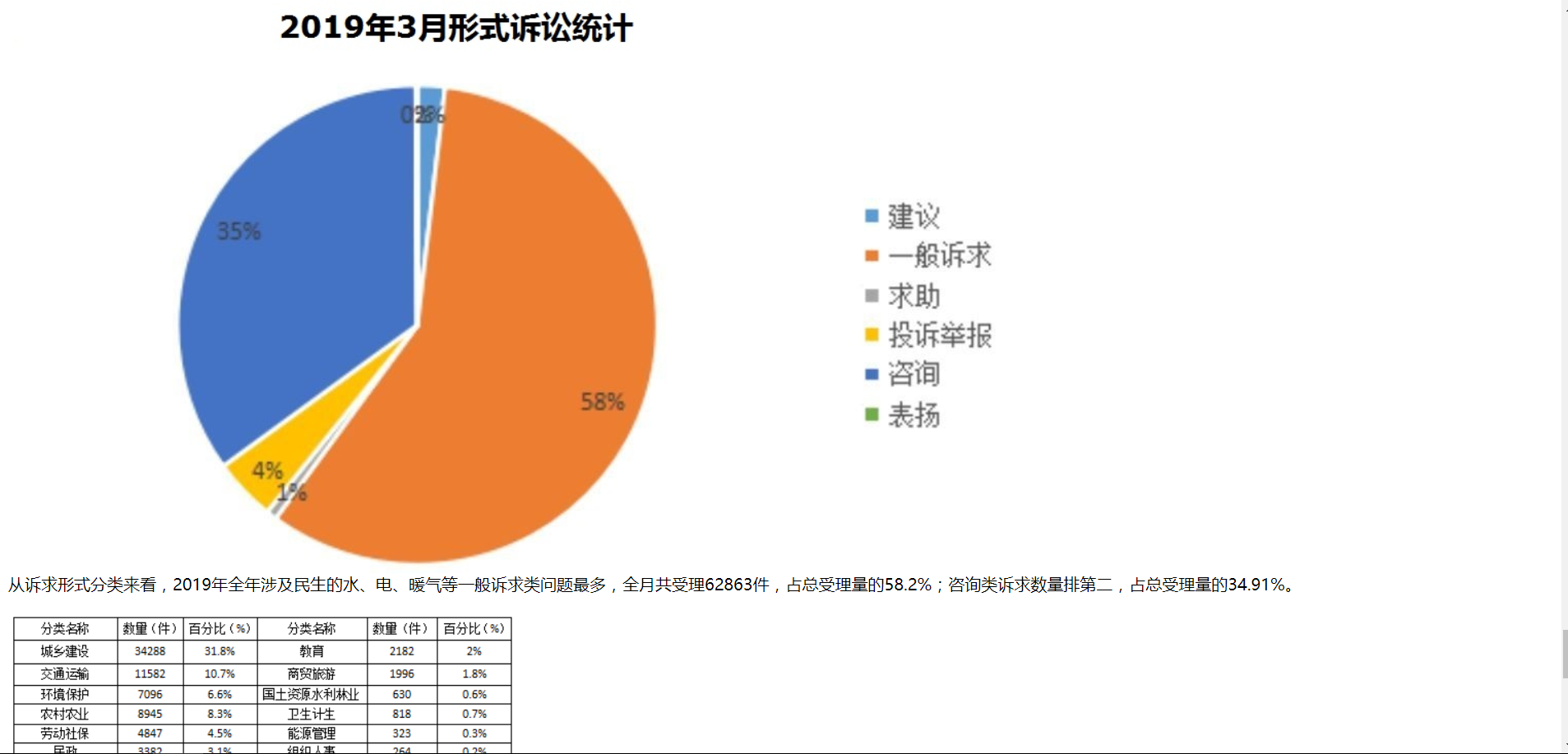
|  |  |  |
| --- | --- | --- |
| 事件名称 | 事件激发 | 事件内容 |
| 概览 | Button1 | 点击此按钮，触发事件，进入数据库所有线索视图页面 |
| 地区管理 | Button2 | 点击此按钮，触发事件，进入反映地区页面 |
| 时间管理 | Button3 | 点击此按钮，触发事件，进入处理时间查看 |

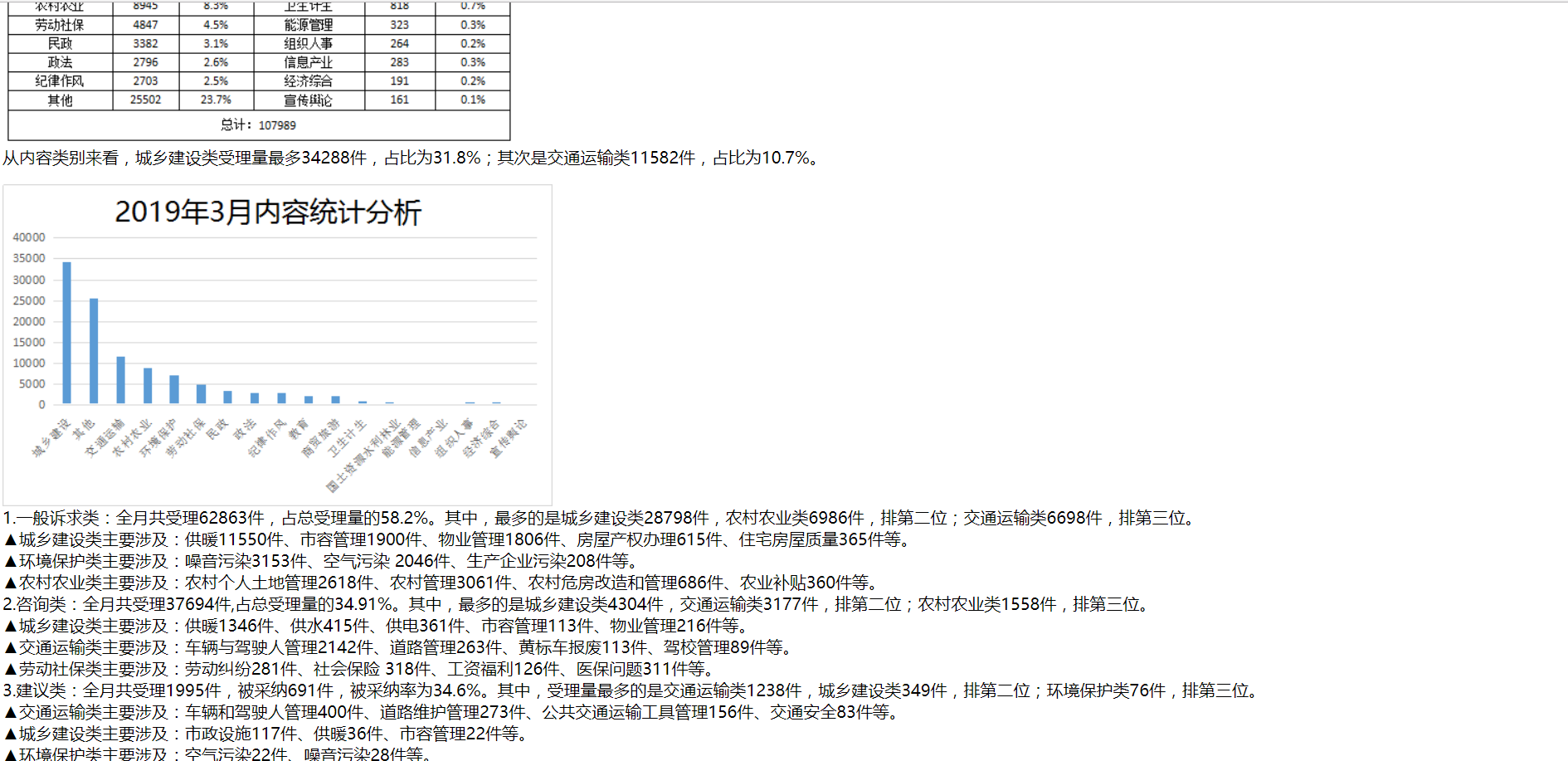
## 4.3 概览功能界面

### 4.3.1 界面设计草图









# 五.逻辑层设计(DFD)

## 5.1.显示所有线索

@Override

**public** List<InfoData> findByAll() {

List<InfoData> all=**new** ArrayList<InfoData>();

String sql = "select \* from infodata LIMIT 50";

**try** {

ps=con.prepareStatement(sql);

ResultSet rs=ps.executeQuery();

**while**(rs.next()){

InfoData id=**new** InfoData();

id.setCity(rs.getString(1));

id.setCounty(rs.getString(2));

id.setName(rs.getString(3));

id.setHandletimes(rs.getString(4));

id.setState(rs.getString(5));

id.setText(rs.getString(6));

id.setReflecttimes(rs.getString(7));

id.setSource(rs.getString(8));

id.setCompany(rs.getString(9));

id.setReplytimes(rs.getString(10));

id.setReplytext(rs.getString(11));

all.add(id);

}

rs.close();

ps.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** all;

}

## 5.2.显示地区线索

@Override

**public** List<InfoData> findByAddress(String address) {

List<InfoData> all=**new** ArrayList<InfoData>();

String sql = "select \* from infodata where county = ?";

**try** {

ps=con.prepareStatement(sql);

ps.setString(1, address);

ResultSet rs = ps.executeQuery();

**while**(rs.next()){

InfoData id=**new** InfoData();

id.setCity(rs.getString(1));

id.setCounty(rs.getString(2));

id.setName(rs.getString(3));

id.setHandletimes(rs.getString(4));

id.setState(rs.getString(5));

id.setText(rs.getString(6));

id.setReflecttimes(rs.getString(7));

id.setSource(rs.getString(8));

id.setCompany(rs.getString(9));

id.setReplytimes(rs.getString(10));

id.setReplytext(rs.getString(11));

all.add(id);

}

rs.close();

ps.close();

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** all;

}

## 5.3.显示时间线索

@Override

**public** List<InfoData> findByReaction(String reaction) {

List<InfoData> all=**new** ArrayList<InfoData>();

String sql = "select \* from infodata where reflecttimes =?";

**try** {

ps=con.prepareStatement(sql);

ps.setString(1, reaction);

ResultSet rs = ps.executeQuery();

**while**(rs.next()){

InfoData id=**new** InfoData();

id.setCity(rs.getString(1));

id.setCounty(rs.getString(2));

id.setName(rs.getString(3));

id.setHandletimes(rs.getString(4));

id.setState(rs.getString(5));

id.setText(rs.getString(6));

id.setReflecttimes(rs.getString(7));

id.setSource(rs.getString(8));

id.setCompany(rs.getString(9));

id.setReplytimes(rs.getString(10));

id.setReplytext(rs.getString(11));

all.add(id);

}

rs.close();

ps.close();

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

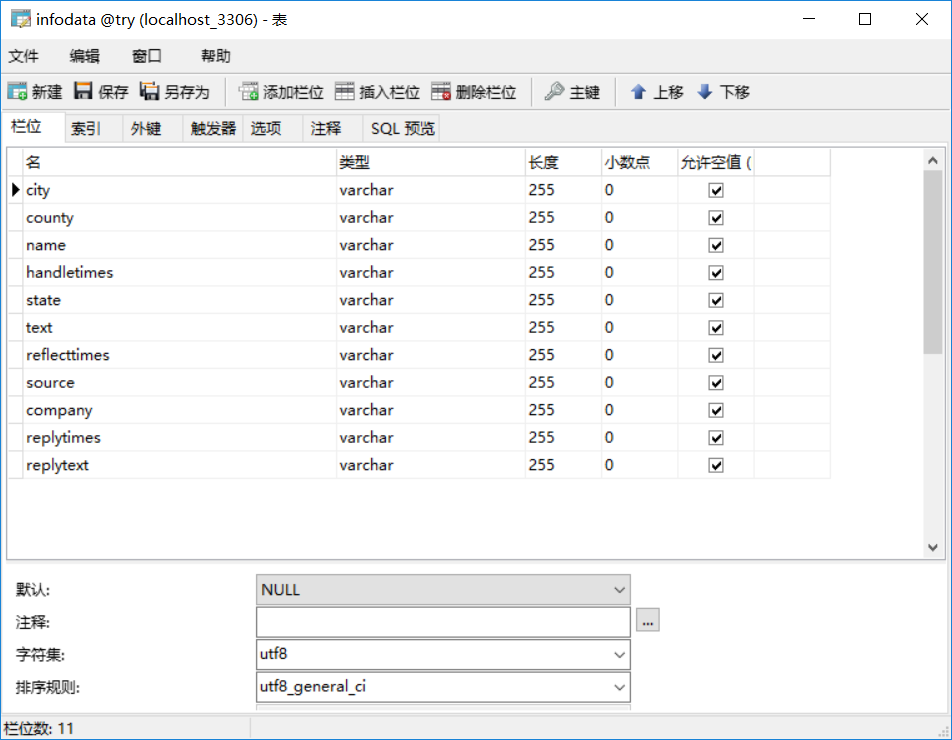
}

**return** all;

}

# 六.数据层设计(Data Base Design)

## 6.1 数据表设计



列表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 字段名 | 类型 | 宽度 | 是否非空 | 主键 |
| 1 | city | varchar | 255 | 是 | 是 |
| 2 | county | varchar | 255 | 是 | 否 |
| 3 | name | varchar | 255 | 是 | 否 |
| 4 | handletimes | varchar | 255 | 是 | 否 |
| 5 | state | varchar | 255 | 是 | 否 |
| 6 | text | varchar | 255 | 是 | 否 |
| 7 | reflecttimes | varchar | 255 | 是 | 否 |
| 8 | source | varchar | 255 | 是 | 否 |
| 9 | company | varchar | 255 | 是 | 否 |
| 10 | replytimes | varchar | 255 | 是 | 否 |
| 11 | replytext | varchar | 255 | 是 | 否 |

在系统中，用户可以按照自己的意愿对群众线索数据进行添加、删除、用户查找操作，提高可操作性和安全性。

## 6.2 代码实现

**InfoData.java**

### 6.2.1数据库基本操作的封装方便调用

**package** com.demo.vo;

**public** **class** InfoData {

**private** String city;

**private** String county;

**private** String name;

**private** String handletimes;

**private** String state;

**private** String text;

**private** String reflecttimes;

**private** String source;

**private** String company;

**private** String replytimes;

**private** String replytext;

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getCounty() {

**return** county;

}

**public** **void** setCounty(String county) {

**this**.county = county;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getHandletimes() {

**return** handletimes;

}

**public** **void** setHandletimes(String handletimes) {

**this**.handletimes = handletimes;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** String getText() {

**return** text;

}

**public** **void** setText(String text) {

**this**.text = text;

}

**public** String getReflecttimes() {

**return** reflecttimes;

}

**public** **void** setReflecttimes(String reflecttimes) {

**this**.reflecttimes = reflecttimes;

}

**public** String getSource() {

**return** source;

}

**public** **void** setSource(String source) {

**this**.source = source;

}

**public** String getCompany() {

**return** company;

}

**public** **void** setCompany(String company) {

**this**.company = company;

}

**public** String getReplytimes() {

**return** replytimes;

}

**public** **void** setReplytimes(String replytimes) {

**this**.replytimes = replytimes;

}

**public** String getReplytext() {

**return** replytext;

}

**public** **void** setReplytext(String replytext) {

**this**.replytext = replytext;

}

@Override

**public** String toString() {

**return** "InfoData [city=" + city + ", county=" + county + ", name="

+ name + ", handletimes=" + handletimes + ", state=" + state

+ ", text=" + text + ", reflecttimes=" + reflecttimes

+ ", source=" + source + ", company=" + company

+ ", replytimes=" + replytimes + ", replytext=" + replytext

+ "]";

}

}

**public** **interface** IM\_InfoData {

//查询所有线索

**public** List<InfoData> findByAll();

//查询地址线索

**public** List<InfoData> findByAddress(String address);

//查询时间线索

**public** List<InfoData> findByReaction(String reaction);

}

### 6.2.2主要的业务实现类

**InfoData\_ImpleMent.java**

**public** **class** InfoData\_ImpleMent **implements** IM\_InfoData{

**private** Connection con;

**private** PreparedStatement ps;

**public** InfoData\_ImpleMent(Connection coon){

**this**.con=coon;

}

@Override

**public** List<InfoData> findByAll() {

List<InfoData> all=**new** ArrayList<InfoData>();

String sql = "select \* from infodata LIMIT 50";

**try** {

ps=con.prepareStatement(sql);

ResultSet rs=ps.executeQuery();

**while**(rs.next()){

InfoData id=**new** InfoData();

id.setCity(rs.getString(1));

id.setCounty(rs.getString(2));

id.setName(rs.getString(3));

id.setHandletimes(rs.getString(4));

id.setState(rs.getString(5));

id.setText(rs.getString(6));

id.setReflecttimes(rs.getString(7));

id.setSource(rs.getString(8));

id.setCompany(rs.getString(9));

id.setReplytimes(rs.getString(10));

id.setReplytext(rs.getString(11));

all.add(id);

}

rs.close();

ps.close();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** all;

}

@Override

**public** List<InfoData> findByAddress(String address) {

List<InfoData> all=**new** ArrayList<InfoData>();

String sql = "select \* from infodata where county = ?";

**try** {

ps=con.prepareStatement(sql);

ps.setString(1, address);

ResultSet rs = ps.executeQuery();

**while**(rs.next()){

InfoData id=**new** InfoData();

id.setCity(rs.getString(1));

id.setCounty(rs.getString(2));

id.setName(rs.getString(3));

id.setHandletimes(rs.getString(4));

id.setState(rs.getString(5));

id.setText(rs.getString(6));

id.setReflecttimes(rs.getString(7));

id.setSource(rs.getString(8));

id.setCompany(rs.getString(9));

id.setReplytimes(rs.getString(10));

id.setReplytext(rs.getString(11));

all.add(id);

}

rs.close();

ps.close();

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** all;

}

@Override

**public** List<InfoData> findByReaction(String reaction) {

List<InfoData> all=**new** ArrayList<InfoData>();

String sql = "select \* from infodata where reflecttimes =?";

**try** {

ps=con.prepareStatement(sql);

ps.setString(1, reaction);

ResultSet rs = ps.executeQuery();

**while**(rs.next()){

InfoData id=**new** InfoData();

id.setCity(rs.getString(1));

id.setCounty(rs.getString(2));

id.setName(rs.getString(3));

id.setHandletimes(rs.getString(4));

id.setState(rs.getString(5));

id.setText(rs.getString(6));

id.setReflecttimes(rs.getString(7));

id.setSource(rs.getString(8));

id.setCompany(rs.getString(9));

id.setReplytimes(rs.getString(10));

id.setReplytext(rs.getString(11));

all.add(id);

}

rs.close();

ps.close();

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** all;

}

}

# 六、结果分析

受理群众反映的问题集中在道路交通、村务管理、供水供电、城市管理、环境保护、房产物业、暖气、社会治安、劳动保障等方面。和上季度相比，除道路交通、暖气、燃气和劳动保障问题有较大增幅外，其他事项都有不同程度的下降，其中，暖气问题从三季度的71件上升到四季度的1118件，增幅达到15.7倍。初步分析，暖气问题大幅增加是因为冬天是集中供暖的时间，此类事项的季节性较强，容易集中暴发；燃气问题增加的主要原因是今冬天然气供应不足，民用天然气气压低，车用天然气价格涨幅较大，引发群众投诉；道路交通问题的增加主要是因为冬季雨雪天气较多，引发的道路交通问题相对增多。村务管理、供水供电、城市管理、环境保护、房产物业、社会治安等问题的下降，主要是我市开展的精准扶贫、全国文明城市创建、大气污染防治、平安济源建设等方面政策的推行，有效化解了部分矛盾。 数字化城市管理监督指挥中心受理的城市管理问题集中在乱堆物料、无照经营游商、暴露垃圾、非法小广告、路灯缺亮、雨水箅子脏污或堵塞、垃圾箱损坏、行道树缺株或折枝等方面，较上季度均有不同程度下降，其中行道树缺株或折枝、垃圾箱损坏等问题降幅较大，分别达到72.8%和49.32%。初步分析，由于三季度是我市迎接全国文明城市复检的关键时期，数字城管中心联合市创建办共同开展了有奖参与“文明城市，共建共享”活动，群众通过微信公众号反映了大量身边的城市管理问题。数字城管中心受理后，积极协调相关责任单位处理，绝大多数城市管理问题得到彻底有效解决，四季度这些问题已不存在。行道树缺株或折枝降幅大是由于四季度天气平稳，因强对流天气导致的树枝折断现象明显减少；垃圾箱损坏降幅大是由于天气转冷，晚上因酒后失控而人为破坏垃圾箱的现象整体减少。

# 七、运行环境

系统环境：windows 10

数据库系统：MYSQL

开发环境：MyEclipse+Tomcat