

Map2Text数据收集系统

--生产实习课程设计报告

**组长： 计科1802高宝琪**

**组员：计科1802徐正阳、计科1802曹贤鹏**

**指导教师： 郭旦怀、葛世寅**

目录

[摘 要 4](#_Toc77862367)

[一、项目背景及意义 4](#_Toc77862368)

[1.1 项目背景： 4](#_Toc77862369)

[1.2 本课题主要研究内容 5](#_Toc77862370)

[二、项目实施方法设计 5](#_Toc77862371)

[2.1技术路线 5](#_Toc77862372)

[2.2任务一介绍：地图数据处理 6](#_Toc77862373)

[2.3任务二介绍：图形化操作 6](#_Toc77862374)

[2.4任务三介绍：后台数据库管理(进阶) 6](#_Toc77862375)

[2.5任务四介绍：收集更多的地图文本描述(进阶) 6](#_Toc77862376)

[2.6任务五介绍：实现文本评价预筛选机制(进阶) 7](#_Toc77862377)

[2.7任务六介绍：做出一个较完整的地图描述提交和管理系统(进阶) 7](#_Toc77862378)

[三、项目实现与测试 8](#_Toc77862379)

[3.1 项目源代码分析 8](#_Toc77862380)

[3.1.1 总体代码架构分析 8](#_Toc77862381)

[3.1.2 核心代码介绍 10](#_Toc77862382)

[3.2 项目实现效果 23](#_Toc77862383)

[3.2.1 任务一(地图数据处理)实现效果 23](#_Toc77862384)

[3.2.3 任务三(后台数据库管理(进阶))实现效果 26](#_Toc77862385)

[3.2.4 任务四(收集更多的地图文本描述(进阶))实现效果 32](#_Toc77862386)

[3.2.5 任务五(实现文本评价预筛选机制(进阶))实现效果 33](#_Toc77862387)

[3.2.6 任务六(做出一个较完整的地图描述提交和管理系统(进阶))实现效果 34](#_Toc77862388)

[3.3 项目难点分析 35](#_Toc77862389)

[3.3.1 难点一 如何实现对一个地点可以添加多条描述，且能区分出每条描述是那个用户所添加。 35](#_Toc77862390)

[3.3.2 难点二 学会Vue特殊的函数调用方法。 36](#_Toc77862391)

[四、结论与展望 37](#_Toc77862392)

[五、参考资源 38](#_Toc77862393)

[六、组员课题完成情况详细说明 39](#_Toc77862394)

# 摘 要

近年来，地理信息与大数据的结合使地理信息系统向更善于处理地理大数据、更能满足决策者各种需求的智能系统转化。因此，对于大规模高质量的地理数据的获取和储备就显得十分重要。

本课题旨在为Map2Text任务的训练阶段收集数量多、质量高、形式多样的地图描述文本，实现一个能够存储大量文本数据、能够供给大量用户使用的、可以通过管理员进行管理的、前后端分离的交互友好型系统来对地图描述文本进行收集和存储。从而为计算机自动完成地图到文本的转换提供训练数据，最终定义一种新概念的地图。

**关键词**：地图到文本的转换、管理系统、Java、SpringBoot、Vue

（GitHub链接：<https://github.com/cby6702/BUCT_1802/>）

# 一、项目背景及意义

## 1.1 项目背景：

地理信息人工智能作为学科交叉领域一大热门研究方向，已经受到学界高度重视，而地理大数据作为其中非常重要的一个环节，近年来也得到了飞速发展，地理信息与大数据的结合催生地理信息系统从一个简单的、基于地理空间的信息管理、统计、分析系统向高级的、善于处理地理大数据并能满足决策者各种需求的智能系统转化。因此对于大规模高质量的地理数据的准备和储备也显得十分重要。地图和自然语言的相互转换，是如今地理人工智能研究的一个方向，想要计算机自动完成地图到文本的转换则需要大量训练数据。

## 1.2 本课题主要研究内容

本课题旨在为Map2Text任务的训练阶段收集数量多、质量高、形式多样的地图描述文本，通过设计并实现一个能够存储大量文本数据、能够供给大量用户使用的、可以通过管理员进行管理的、前后端分离的交互友好型系统来对地图描述文本进行收集和存储。主要研究内容包括前端图形化页面的设计，数据库的建立，前端与后端数据库的连接。

# 二、项目实施方法设计

## 2.1技术路线

**Java后台**：SpringBoot 2.1.0 + Mybatis-plus+ hutool工具包 + Apache poi

**前台页面**：Vue2.0 + ElementUI + Jquery + tinymce（富文本插件）

**数据库**：Mysql 5.7+

**JDK**: 1.8版本

前后端分离，页面可单独部署，默认放在项目的static文件夹，随后端工程一起访问。

## 2.2任务一介绍：地图数据处理

调用百度地图api对经度116.2-116.5，纬度39.8-40内Zoom等级为18地图数据按步长0.02的地图界面展示在页面并对其进行数据处理，后采用mysql数据库对处理后的数据进行存储。

## 2.3任务二介绍：图形化操作

用户可点击随机地图中心点按钮来获取中心点随机的步长为0.02的地图界面，且地图页面无法被拖动。点击获取当前中心点按钮可得到当前地图页面的中心点坐标，点击查看该地点已有描述后可获得对当前地图的所有描述，在描述框中添加描述后点击提交，即可将数据存入数据库。

## 2.4任务三介绍：后台数据库管理(进阶)

设计后台数据库，用户对地图添加的描述后，地点信息管理数据库查询用户描述的中心点坐标，若不存在则新建数据来存储用户描述地图的中心点、西南角以及东北角的经纬度坐标，并以该数据的id为主键在描述信息管理数据库中存储用户描述和用户id。超级管理员可对后台管理数据库中的数据进行修改和删除。

## 2.5任务四介绍：收集更多的地图文本描述(进阶)

对不同的地点，收集高质量的地图文本描述。

## 2.6任务五介绍：实现文本评价预筛选机制(进阶)

对用户添加的文本描述进行预筛选，筛出掉无效信息和灌水信息。我们打算预筛查掉连续重复5个及以上字符的描述。

## 2.7任务六介绍：做出一个较完整的地图描述提交和管理系统(进阶)

将地图描述和后台管理进行整合，实现一个较完整的地图描述提交和管理系统

# 三、项目实现与测试

## 3.1 项目源代码分析

### 3.1.1 总体代码架构分析

#### 后端框架

**entity:**实体层，用于存放我们的实体类，与数据库中的属性值基本保持一致，实现set和get的方法。

**mapper:**对数据库进行数据持久化操作，他的方法语句是直接针对数据库操作的，主要实现一些增删改查操作，在mybatis中方法主要与与xxx.xml内相互一一映射。

**service:** 业务service层，给controller层的类提供接口进行调用。

**controller:** 控制层，负责具体模块的业务流程控制，需要调用service逻辑设计层的接口来控制业务流程。

#### 前端界面

### 3.1.2 核心代码介绍

##### index.html(主页代码)

**（****1）对地图的操作(随机地图中心点、获取当前中心点):**

<div class="info">

<button id="change-btn" onclick="setNewCenter()">随机地图中心点</button>

<button id="change-btn" onclick="getMapCenter()">获取当前中心点</button>

</div>

<div id="container">

</div>

...

var map = new BMapGL.Map('container', {

minZoom:18,

maxZoom:18,

enableMapClick:false

});

var precision = 5;

map.centerAndZoom(new BMapGL.Point(116.2, 39.8), 18)

map.enableScrollWheelZoom(false)

map.disableDragging();

var scaleCtrl = new BMapGL.ScaleControl(); // 添加比例尺控件

map.addControl(scaleCtrl);

var zoomCtrl = new BMapGL.ZoomControl(); // 添加比例尺控件

map.addControl(zoomCtrl);

function setNewCenter(){

var lng = 116.2 + Math.floor(Math.random() \* 15) \* 0.02;

var lat = 39.8 + Math.floor(Math.random() \* 10) \* 0.02;

var point = new BMapGL.Point(lng, lat);

map.setCenter(point); // 设置地图中心点

}

function getMapCenter(){

var cen = map.getCenter();

alert('当前地图中心点: (' + cen.lng.toFixed(precision) + ', ' + cen.lat.toFixed(precision) + ')');

}

**（2）描述提交功能的实现:**

<el-input v-model="input" style="width: 40%;" placeholder="请输入对该地点的描述"

minlength="1" type="textarea"></el-input>

<el-button @click="submit" type="primary" style="margin: 10px 0px">提交</el-button>

<el-input v-model="pid\_str" style="width: 20%;" suffix-icon="el-icon-search"

@keyup.enter.native="locate"></el-input>

...

submit(){ // 点击提交描述的按钮后要进行的操作;

// 获取中心点坐标, 西南角坐标, 东北角坐标等信息;

var cen = map.getCenter();

this.lng = cen.lng.toFixed(precision);

this.lat = cen.lat.toFixed(precision);

var sw\_lng = map.getBounds().sw.lng.toFixed(precision);

var sw\_lat = map.getBounds().sw.lat.toFixed(precision);

var ne\_lng = map.getBounds().ne.lng.toFixed(precision);

var ne\_lat = map.getBounds().ne.lat.toFixed(precision);

// 调用PositionController提供的功能, 查找position表中是否已有当前地点;

$.get("/api/position/" + "/center?pageNum=" + this.pageNum + "&pageSize=" + this.pageSize

+ "&lng=" + this.lng + "&lat=" + this.lat).then(res => {

if(res.data.total==0){ // 如果没找到当前地点, 则新增到position表中;

this.entity = {centerLng:this.lng, centerLat:this.lat,

swLng:sw\_lng, swLat:sw\_lat, neLng:ne\_lng, neLat:ne\_lat};

this.savePosition();

this.submitDescription();// 大概率第二次成功, 故提交两次;

this.submitDescription();

}

else{ // 如果找到了的话, 就显示"该地点已存在"的提示信息

this.$message({

message: "该地点已存在",

type: "success"

})

this.submitDescription(); // 大概率第一次成功, 故提交一次;

}

})

},

submitDescription() {// 调用PositionController提供的功能, 查找刚刚添加的地点的id;

// 对用户提交的描述进行预筛查:

var cnt=0;

for (var i=0;i<this.input.length;i++)

{

cnt=Math.max(cnt,(this.input.split(this.input[i])).length-1);

if(cnt>5) {

this.$message({

message: "描述中有超过4个字的重复字符, 无法添加",

type: "error"

})

return;

}

}

$.get("/api/position/" + "/center?pageNum=" + this.pageNum + "&pageSize=" + this.pageSize

+ "&lng=" + this.lng + "&lat=" + this.lat).then(res => {

if(res.data.total==1){

this.pid = res.data.records[0].id;

this.entity = {uid:this.user.id, pid:this.pid, name:this.input};

this.saveDescription();

}

})

},

**（3）通过pid定位到地图上的地点**

<el-input v-model="pid\_str" style="width: 20%;" suffix-icon="el-icon-search"

@keyup.enter.native="locate"></el-input>

…

locate() {

$.get("/api/position/" + "/idstr?id\_str=" + this.pid\_str).then(res => {

if(res.data.total==1){// 如果找到的话

var lng = res.data.records[0].centerLng;

var lat = res.data.records[0].centerLat;

var point = new BMapGL.Point(lng, lat);

map.setCenter(point); // 设置地图中心点

}

else{

this.$message({

message: "数据库中无该id地点",

type: "error"

})

}

})

}

**（4）按钮"查看该地点已有描述"功能的实现**

<el-popover

placement="right"

width="350"

trigger="click">

<el-table :data="descriptionData">

<el-table-column width="80" property="uid" label="用户id"></el-table-column>

<el-table-column width="250" property="name" label="描述"></el-table-column>

</el-table>

<el-button slot="reference" @click="loadDescription()" style="margin: 10px 30px">

查看该地点已有描述

</el-button>

</el-popover>

...

loadDescription() {

var cen = map.getCenter(); // 获取地图中心点

// 在position中查找中心点

$.get("/api/position/" + "/center?pageNum=" + this.pageNum + "&pageSize=" + this.pageSize

+ "&lng=" + cen.lng.toFixed(precision) + "&lat=" + cen.lat.toFixed(precision)).then(res => {

if(res.data.total==0){// 若未找到, 则该地点没有描述

this.descriptionData = [];

this.total = 0;

}

else{// 若找到, 则获取该地点的id, 用这个id作为pid去查description表, 将对这个地点的

// 描述存入this.descriptionData中;

this.pid = res.data.records[0].id;

$.get("/api/description/" + "/page/pid?pageNum=" + this.pageNum

+ "&pageSize=" + this.pageSize + "&pid=" + this.pid).then(res => {

this.descriptionData = res.data.records;

this.total = res.data.total;

})

}

})

$.get("/api/").then(res => {

this.options = res.data;

})

},

##### PositionController.java

loadDescription() {

var cen = map.getCenter(); // 获取地图中心点

// 在position中查找中心点

$.get("/api/position/" + "/center?pageNum=" + this.pageNum + "&pageSize=" + this.pageSize

+ "&lng=" + cen.lng.toFixed(precision) + "&lat=" + cen.lat.toFixed(precision)).then(res => {

if(res.data.total==0){// 若未找到, 则该地点没有描述

this.descriptionData = [];

this.total = 0;

}

else{// 若找到, 则获取该地点的id, 用这个id作为pid去查description表, 将对这个地点的

// 描述存入this.descriptionData中;

this.pid = res.data.records[0].id;

$.get("/api/description/" + "/page/pid?pageNum=" + this.pageNum

+ "&pageSize=" + this.pageSize + "&pid=" + this.pid).then(res => {

this.descriptionData = res.data.records;

this.total = res.data.total;

})

}

})

$.get("/api/").then(res => {

this.options = res.data;

})

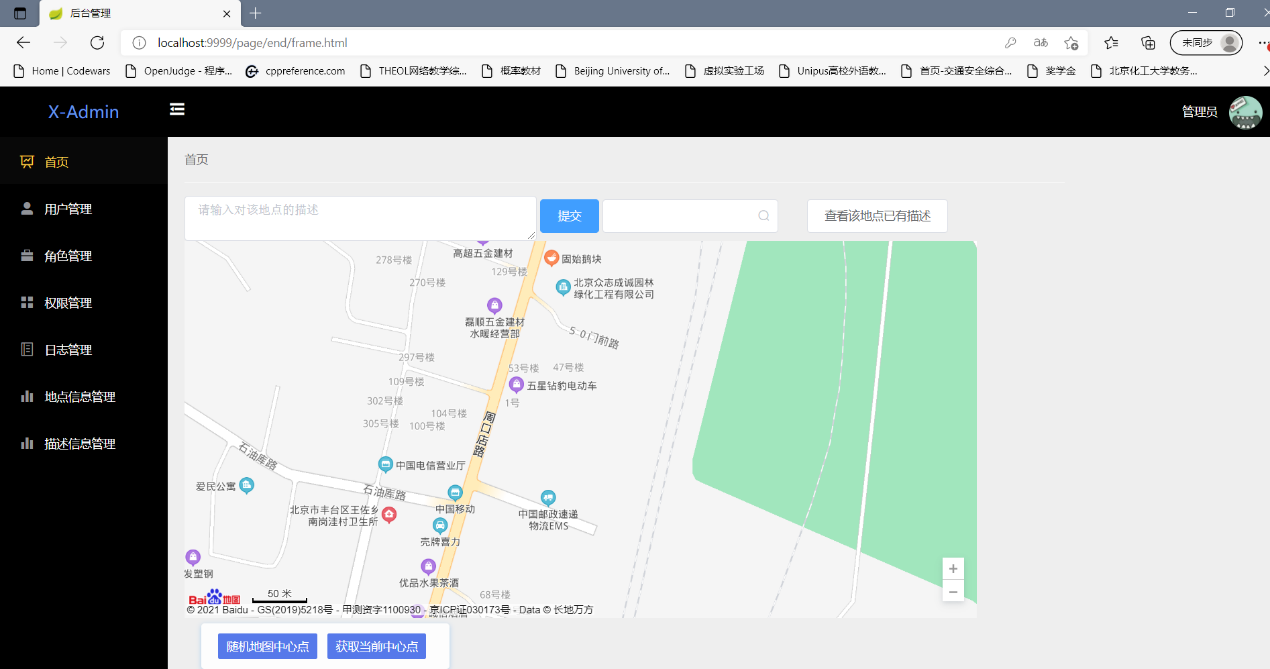
},

##### DescriptionController.java

// 分页查找  
@GetMapping("/page")  
public Result<?> findPageByName(@RequestParam(required = false, defaultValue = "") String name,  
 @RequestParam(required = false, defaultValue = "1") Integer pageNum,  
 @RequestParam(required = false, defaultValue = "10") Integer pageSize) {  
 LambdaQueryWrapper<Description> query = Wrappers.<Description>*lambdaQuery*().orderByDesc(Description::getId);  
 if (StrUtil.*isNotBlank*(name)) {  
 query.like(Description::getName, name);  
 }  
 return Result.*success*(descriptionService.page(new Page<>(pageNum, pageSize), query));  
}  
  
// 通过pid分页查找  
@GetMapping("/page/pid")  
public Result<?> findPageByPid(@RequestParam Long pid,  
 @RequestParam(required = false, defaultValue = "1") Integer pageNum,  
 @RequestParam(required = false, defaultValue = "10") Integer pageSize) {  
 LambdaQueryWrapper<Description> query = Wrappers.<Description>*lambdaQuery*().orderByDesc(Description::getId);  
 query.eq(Description::getPid, pid);  
 return Result.*success*(descriptionService.page(new Page<>(pageNum, pageSize), query));  
}  
  
// 通过uid分页查找  
@GetMapping("/page/uid")  
public Result<?> findPageByUid(@RequestParam Long uid,  
 @RequestParam(required = false, defaultValue = "1") Integer pageNum,  
 @RequestParam(required = false, defaultValue = "10") Integer pageSize) {  
 LambdaQueryWrapper<Description> query = Wrappers.<Description>*lambdaQuery*().orderByDesc(Description::getId);  
 query.eq(Description::getUid, uid);  
 return Result.*success*(descriptionService.page(new Page<>(pageNum, pageSize), query));  
}

## 3.2 项目实现效果

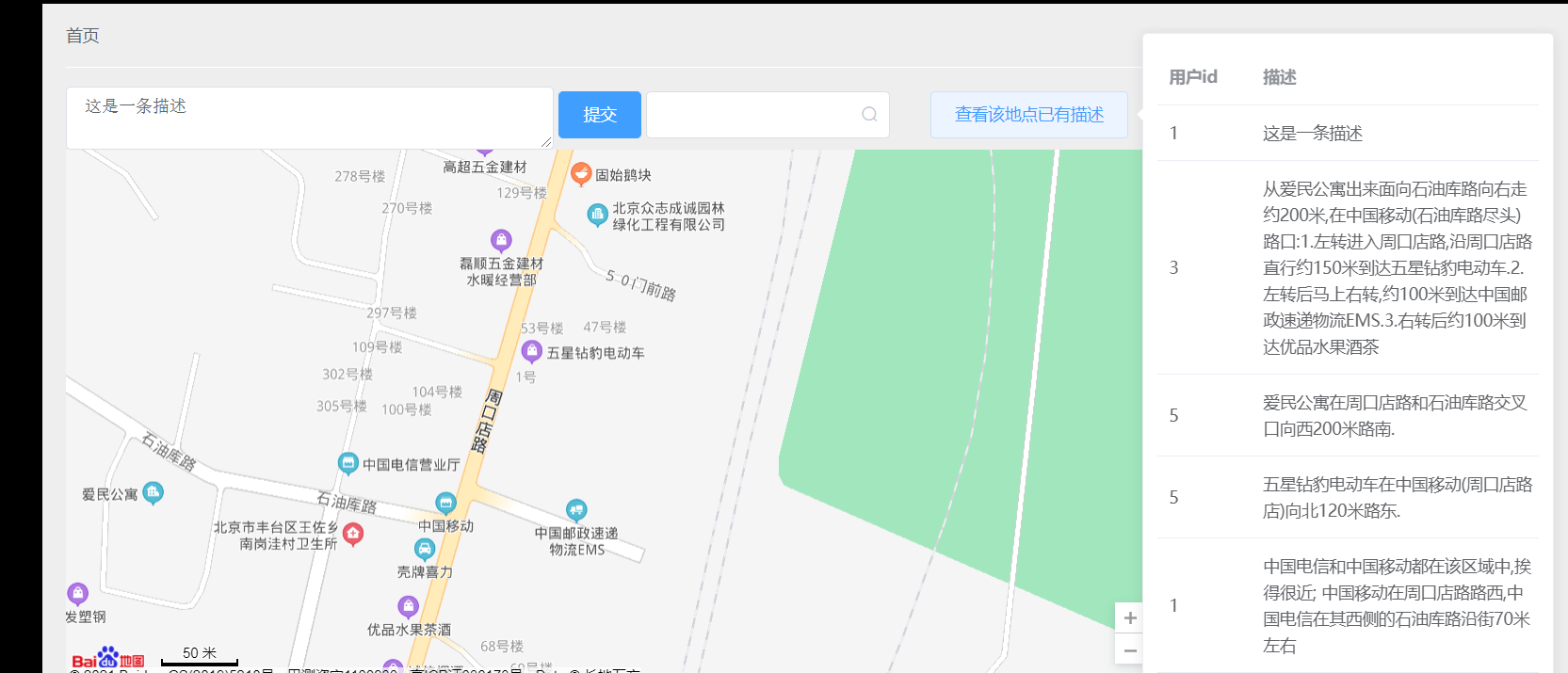
### 3.2.1 任务一(地图数据处理)实现效果

**3.2.2 任务二(图形化操作)实现效果**

#### 提交描述:



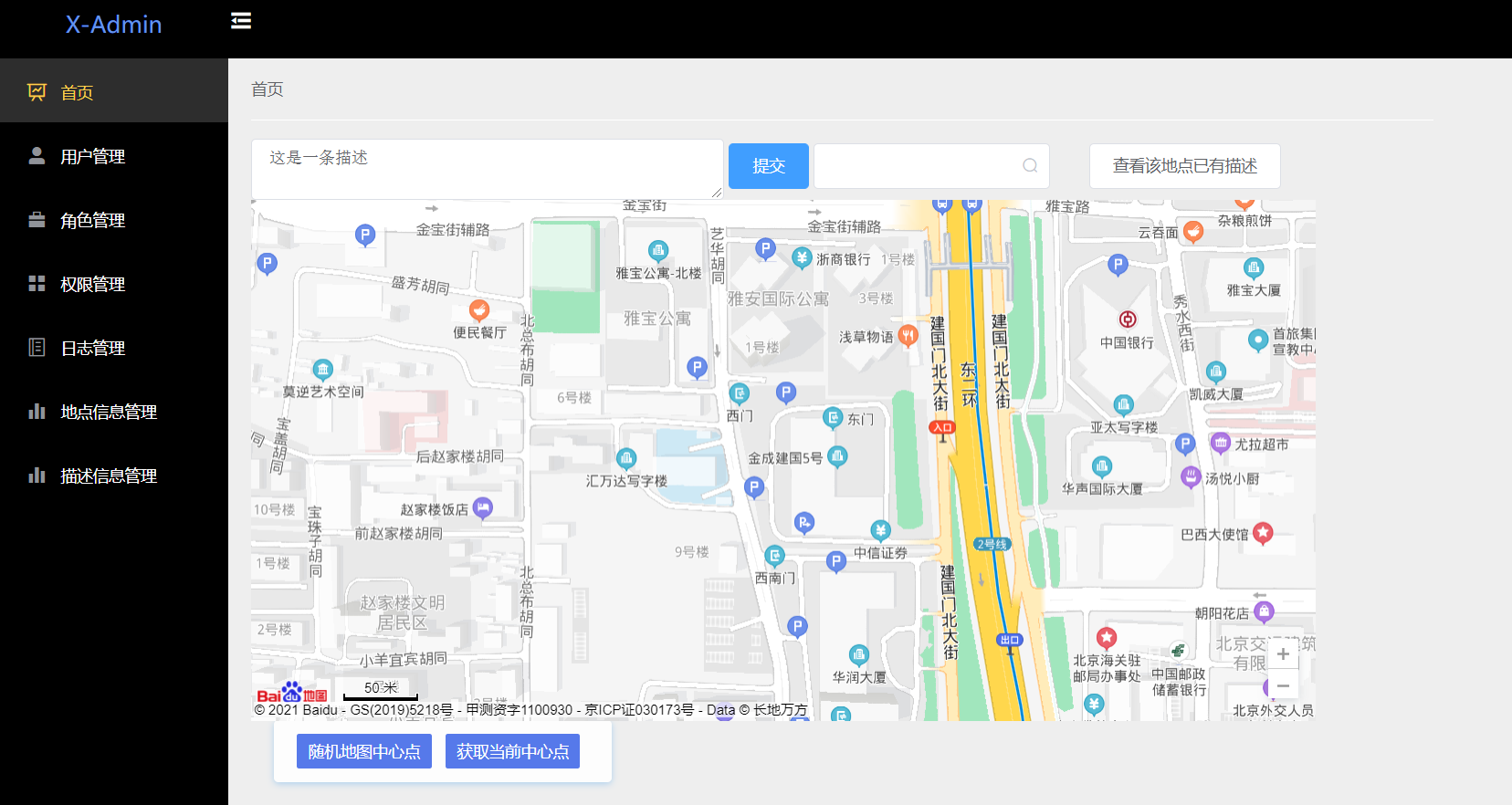
#### 查看该地点已有描述:



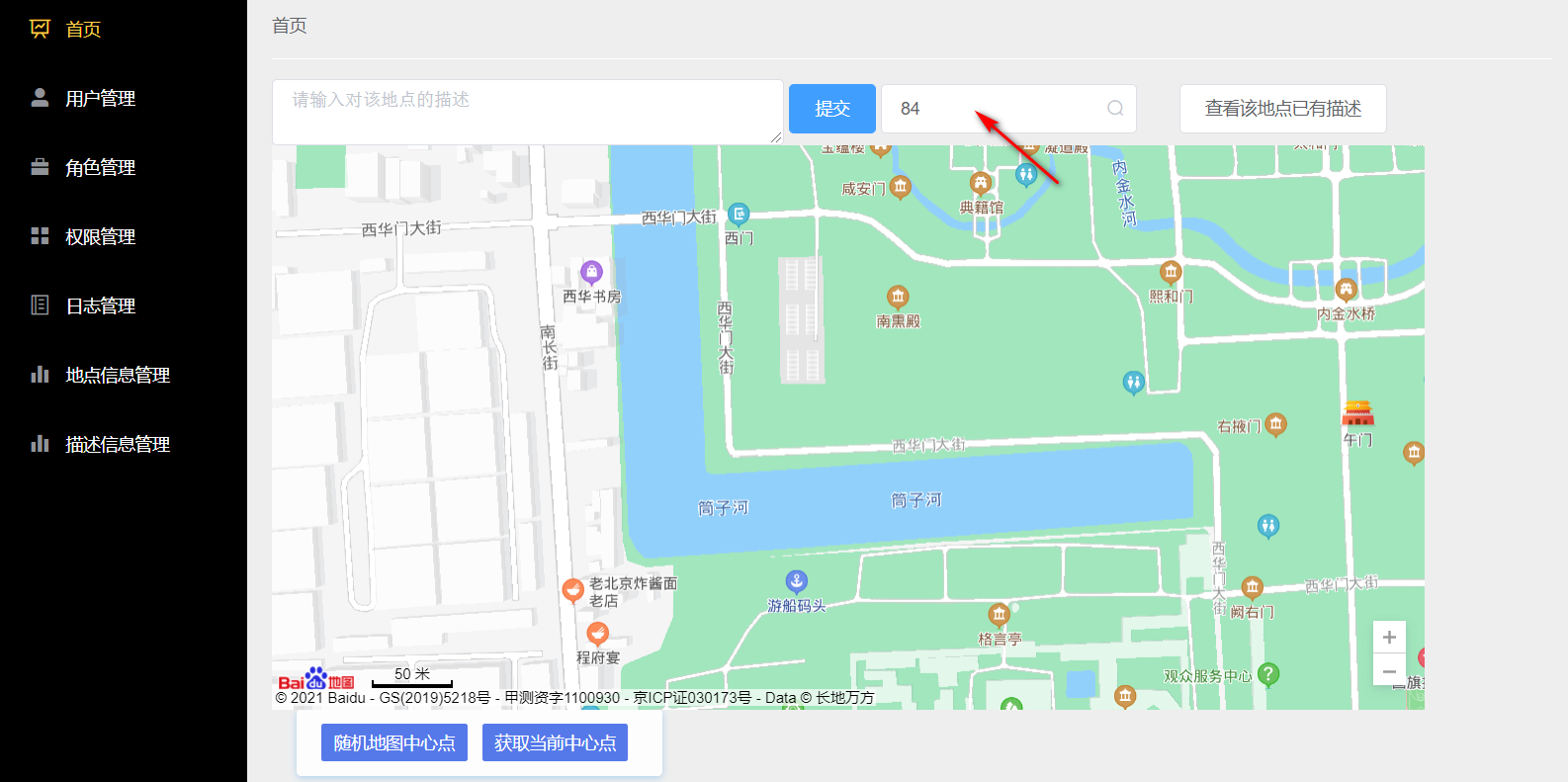
#### 描述预筛查



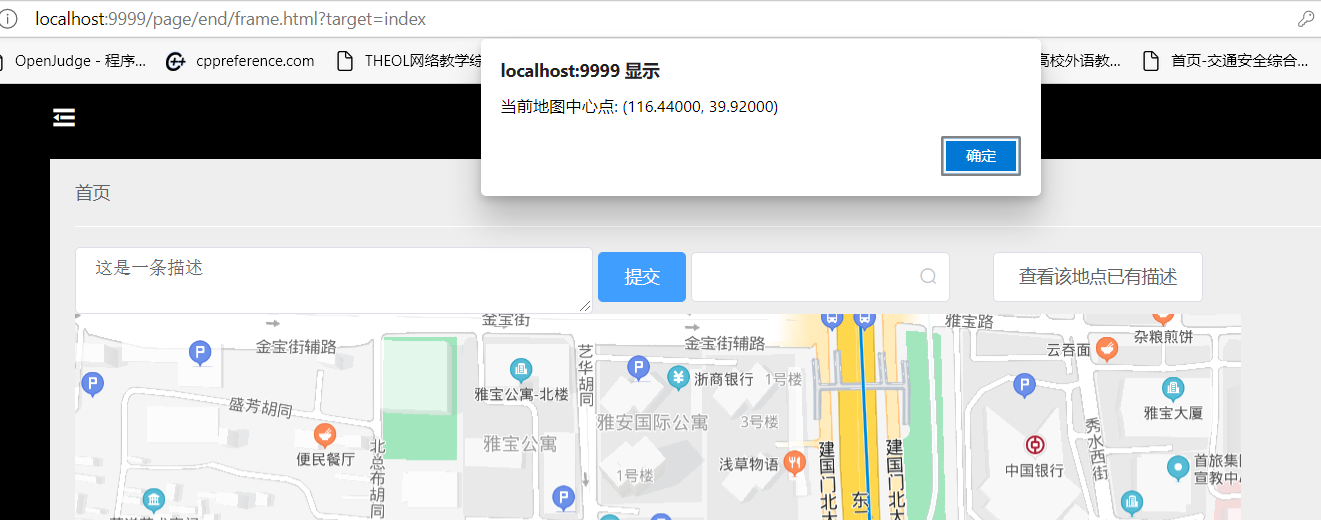
#### 随机地图中心点:



#### 通过地点id转到相应地点



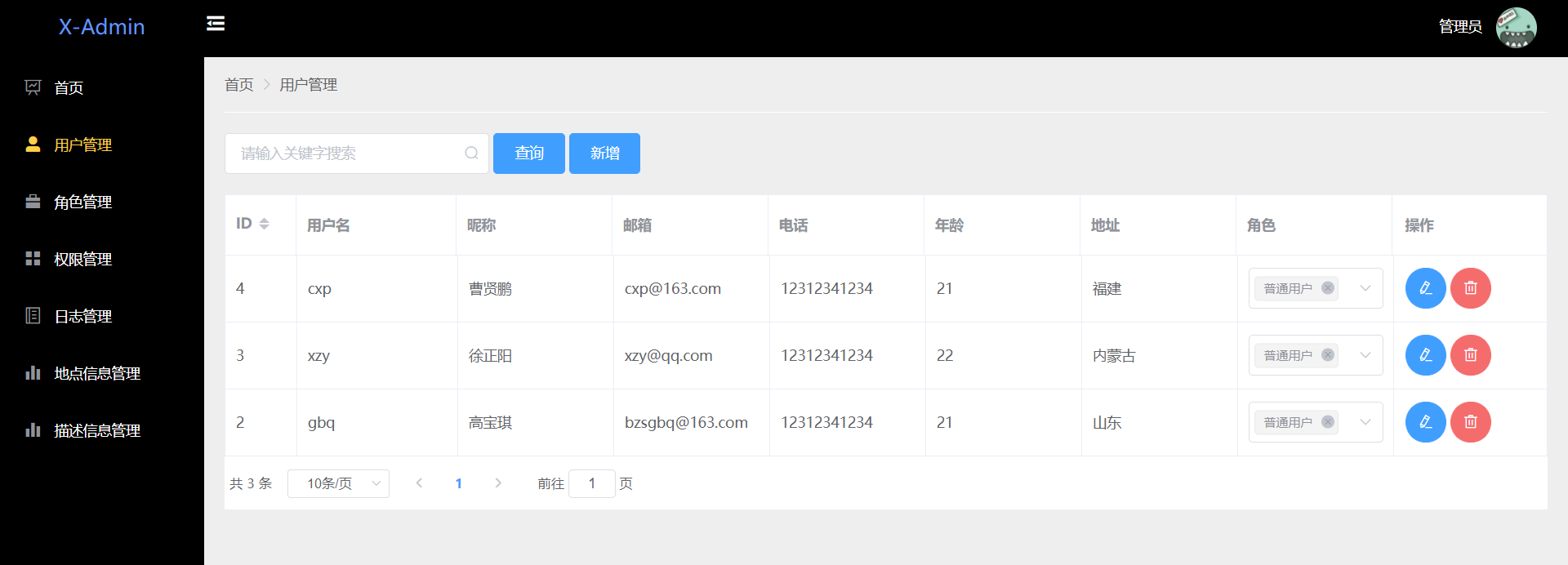
#### 获取当前中心点:



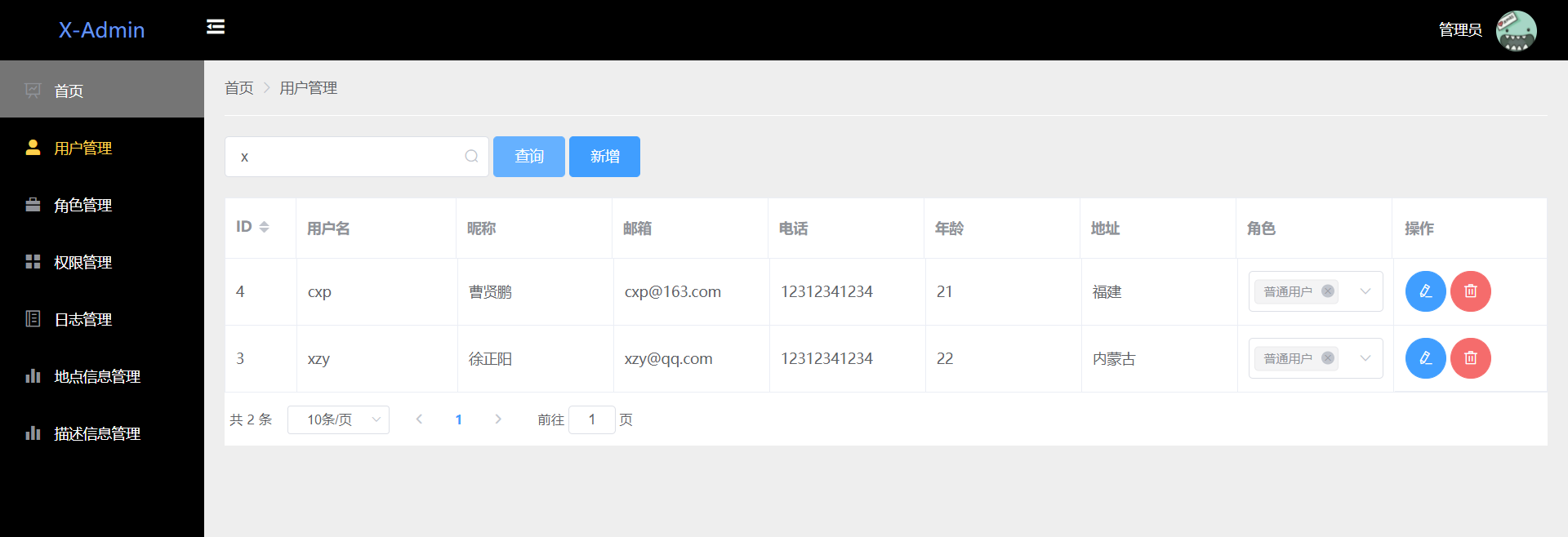
### 3.2.3 任务三(后台数据库管理(进阶))实现效果

(因为各个页面的基本操作(增删改查)过于相似, 所以只以"用户管理"页面为例进行详细展示)

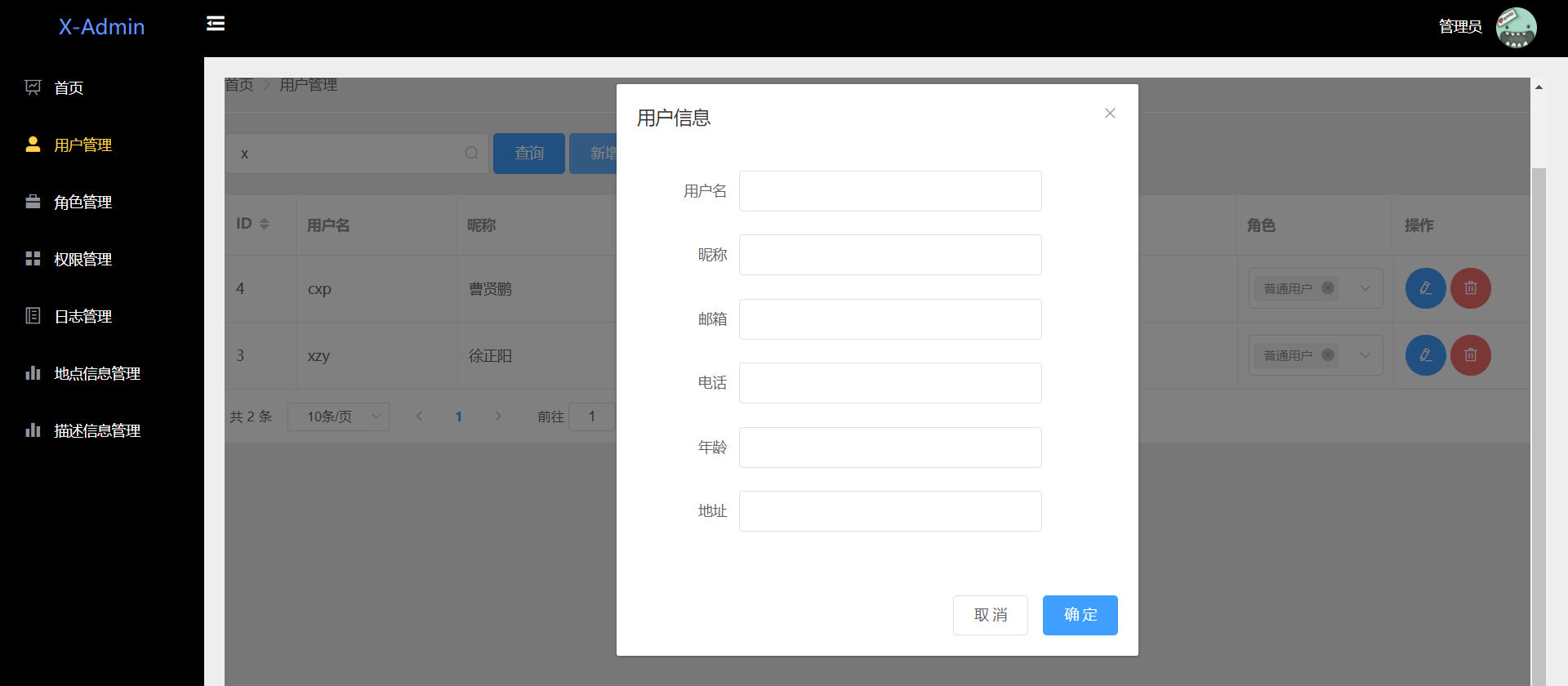
#### 1. 用户管理



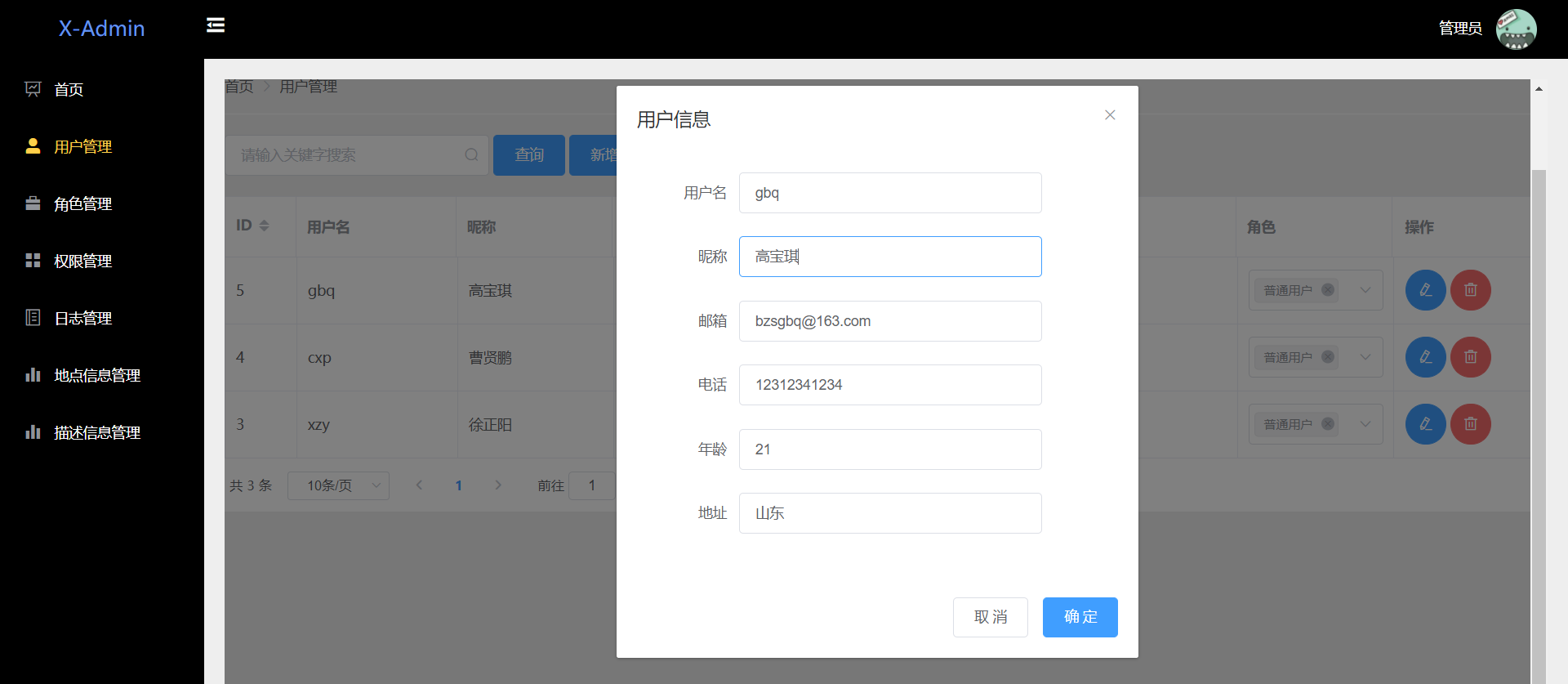
##### 查询:

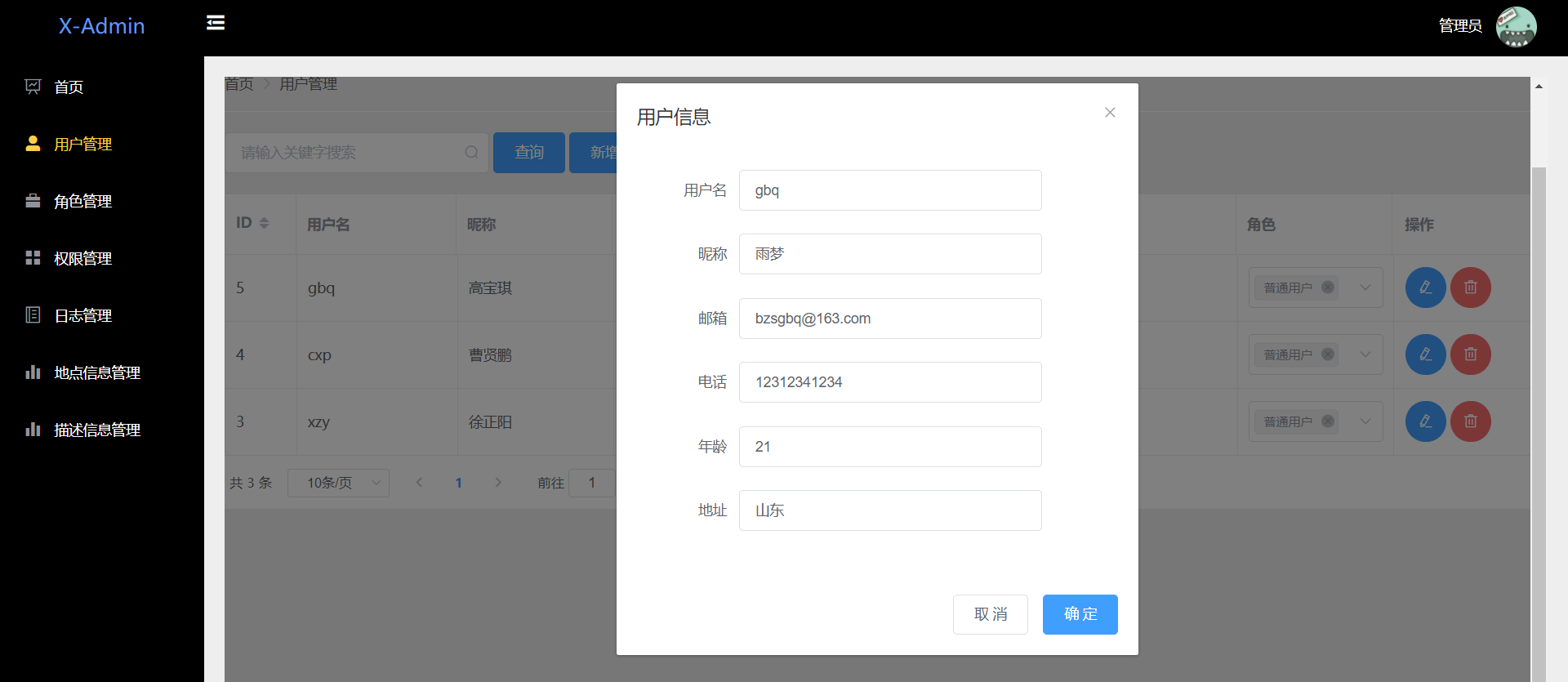


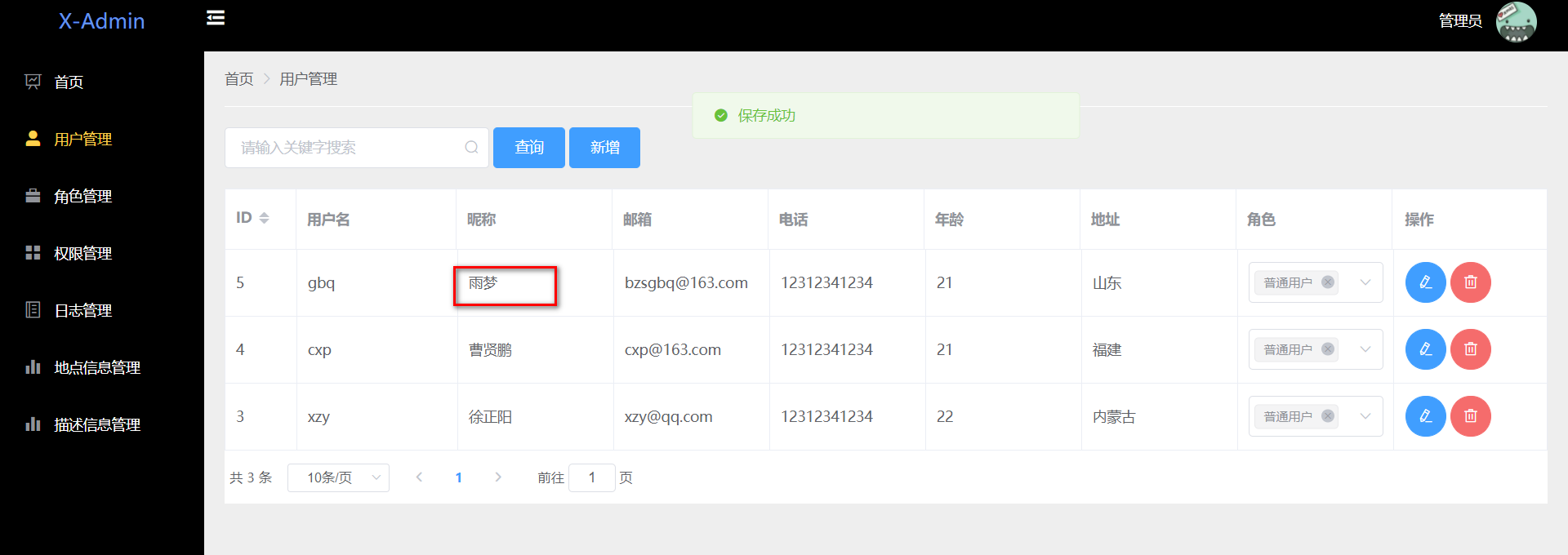
##### 新增:



##### 编辑:

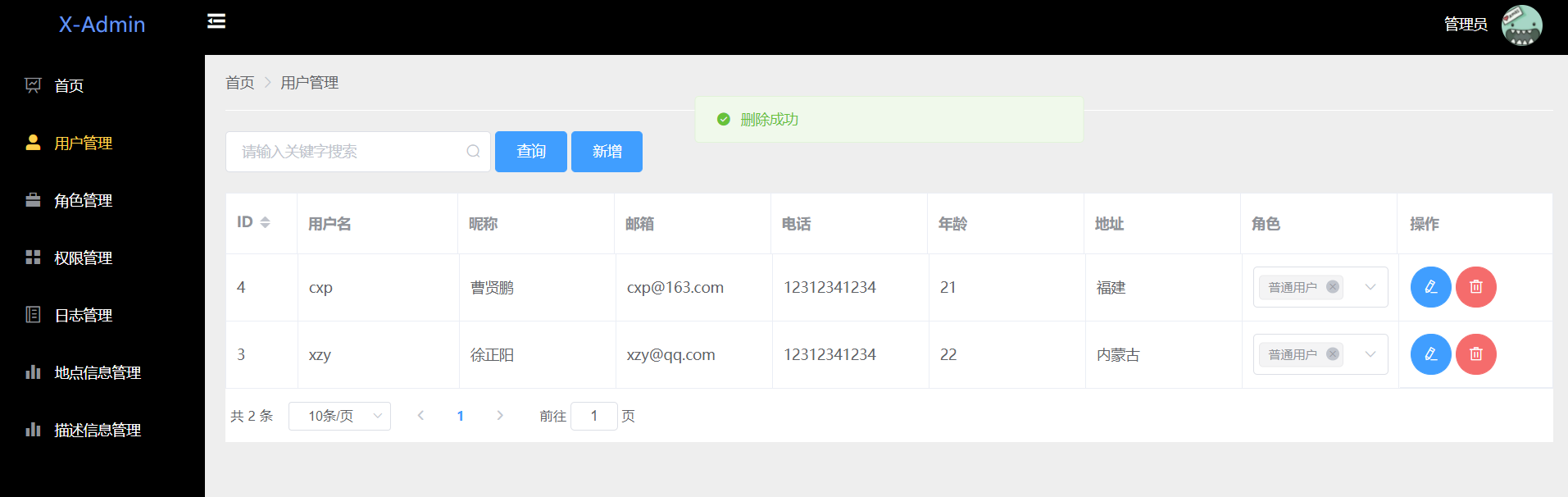






##### 删除:





#### 2. 角色管理



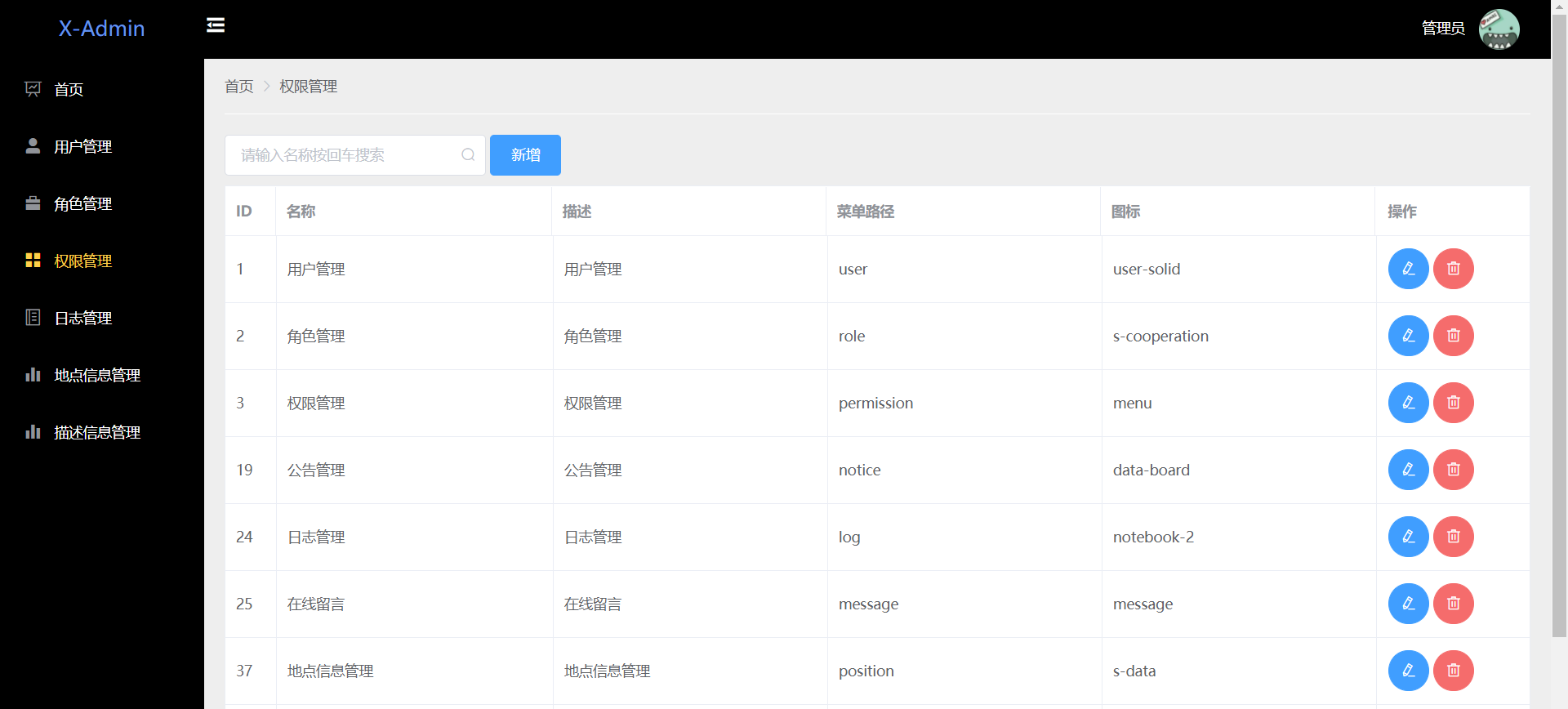
##### 超管权限设置:



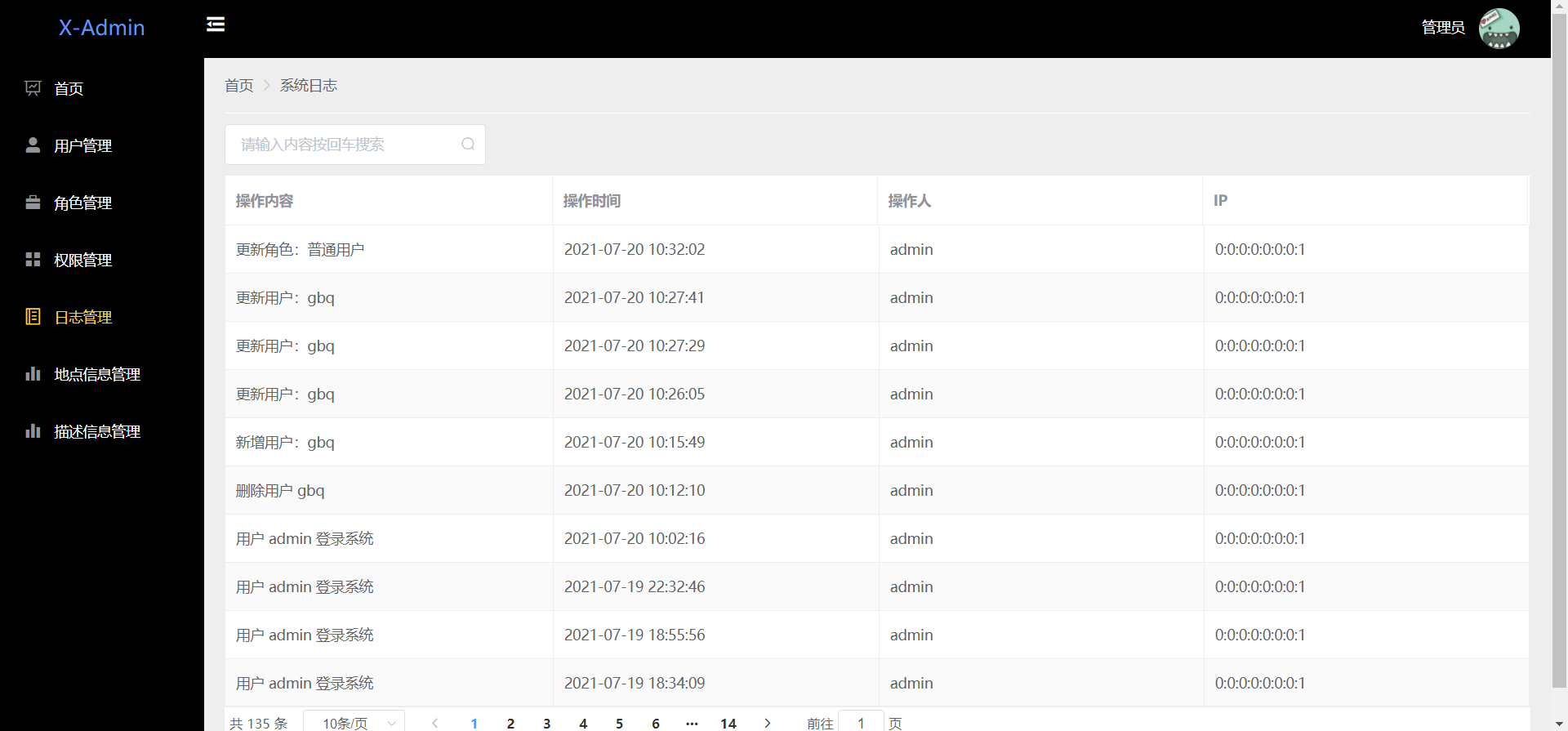
##### 普通用户权限设置:



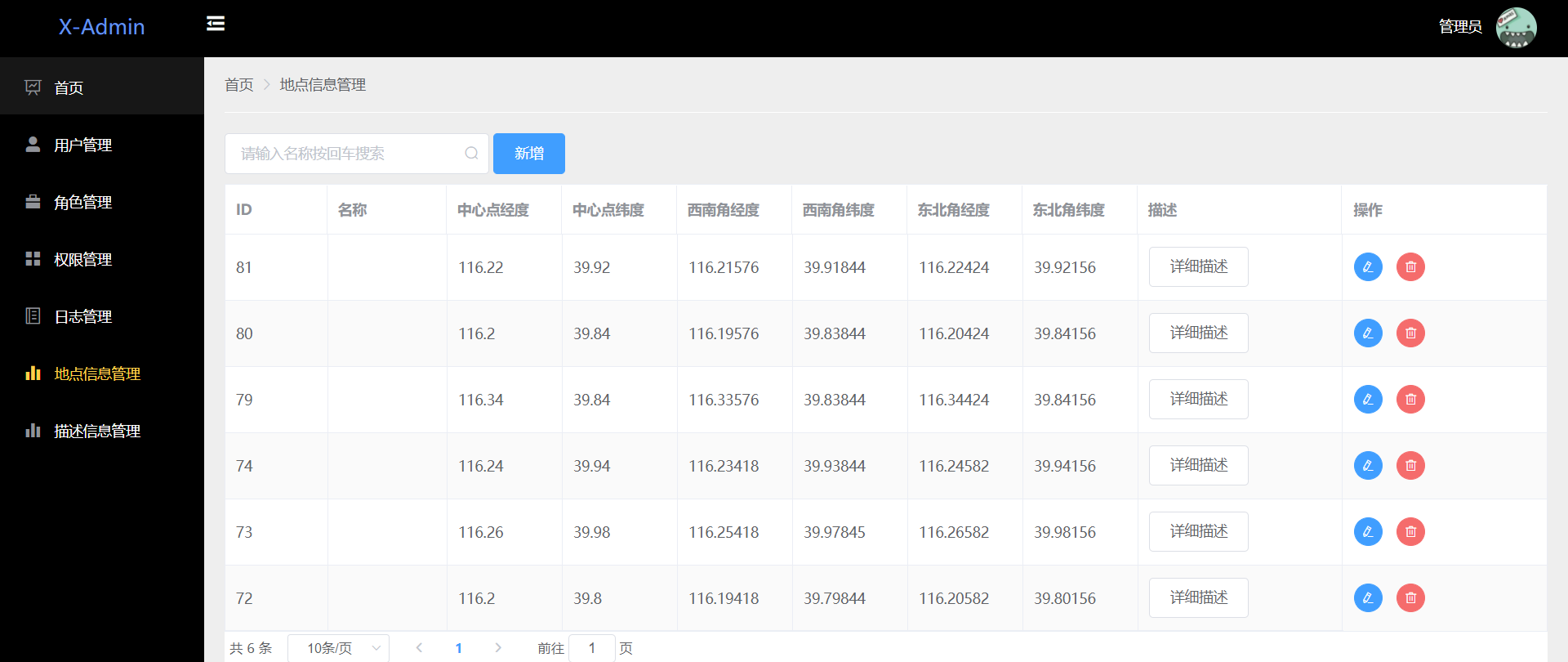
#### 3. 权限管理



#### 4. 日志管理

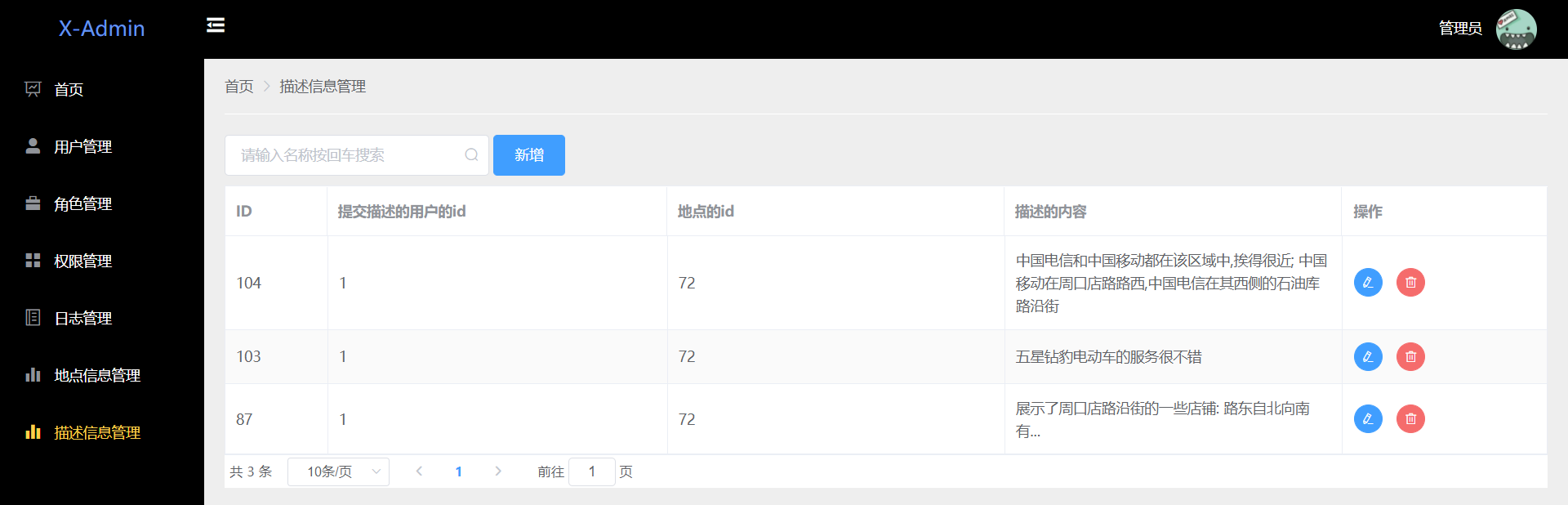


#### 5. 地点信息管理

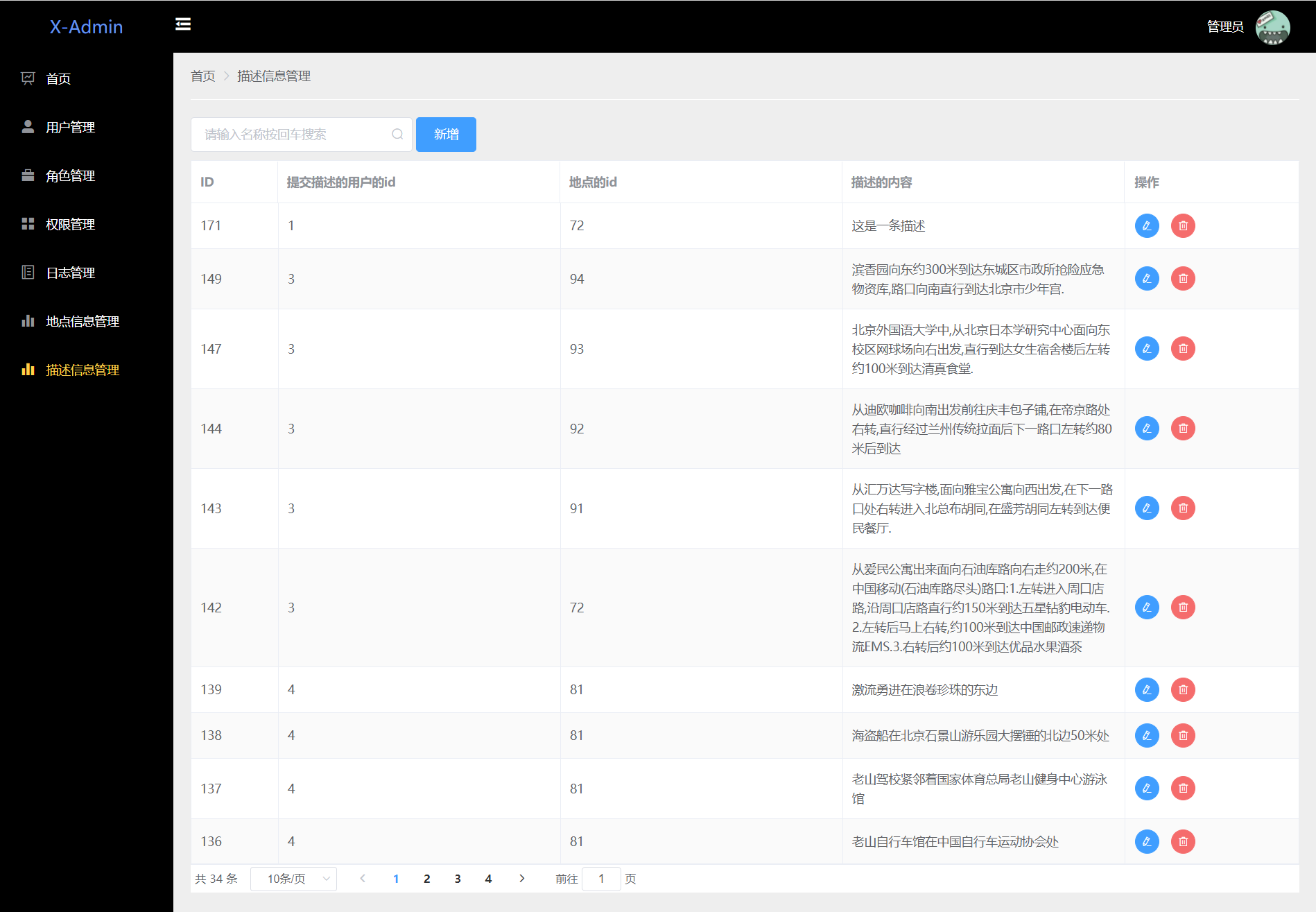


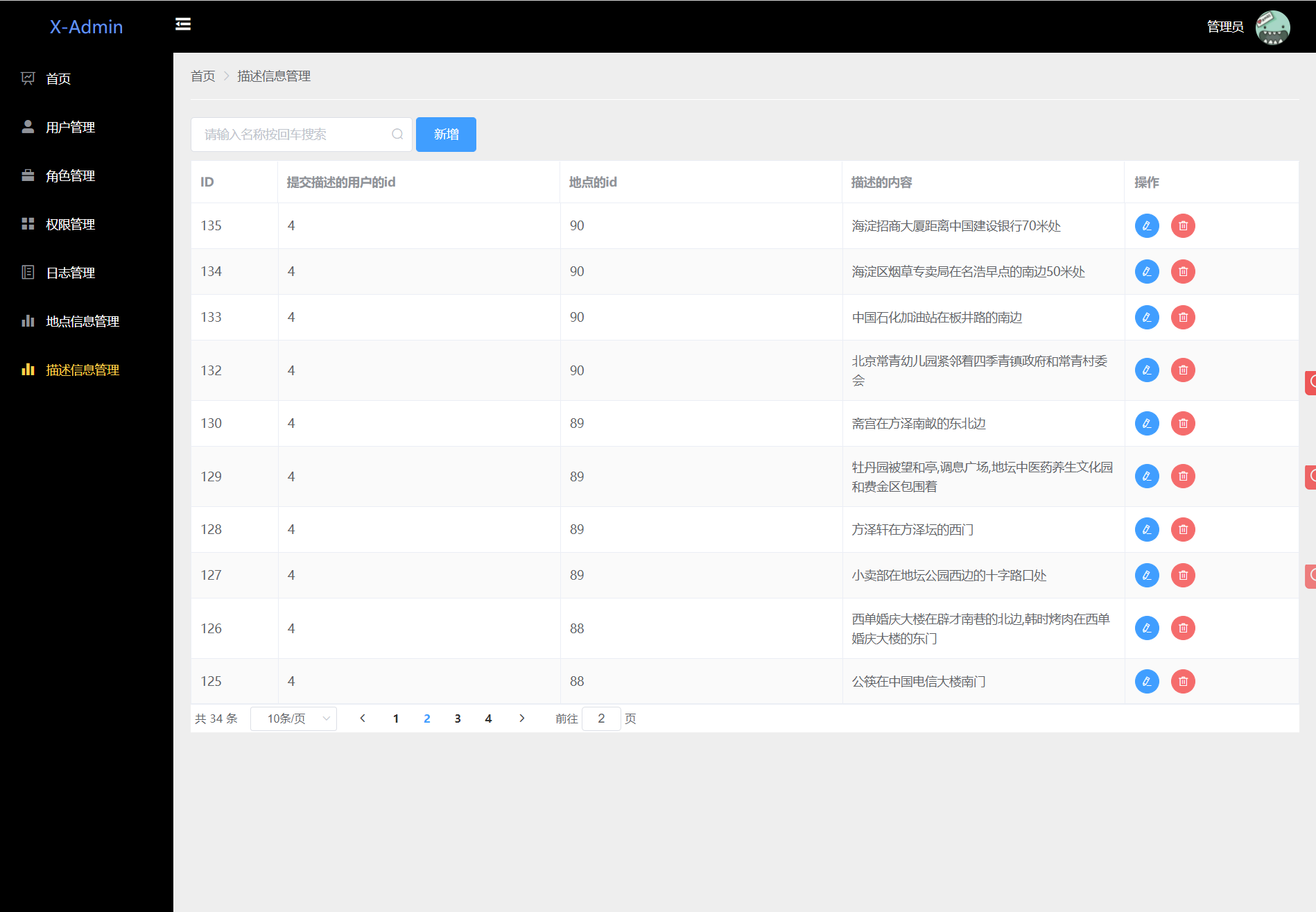


#### 6. 描述信息管理



### 3.2.4 任务四(收集更多的地图文本描述(进阶))实现效果





### 3.2.5 任务五(实现文本评价预筛选机制(进阶))实现效果

对于存在五个以上重复字符的评论，会禁止添加。

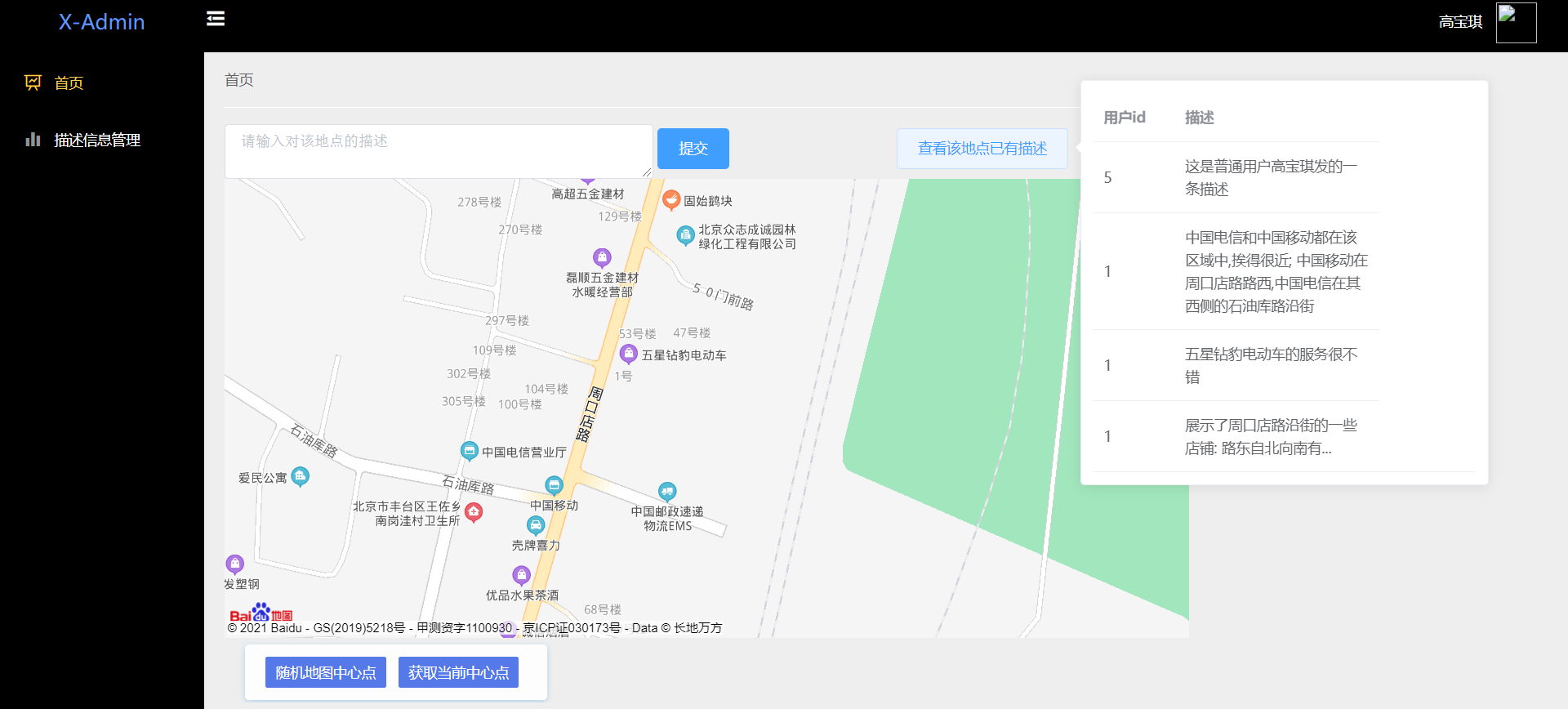




### 3.2.6 任务六(做出一个较完整的地图描述提交和管理系统(进阶))实现效果

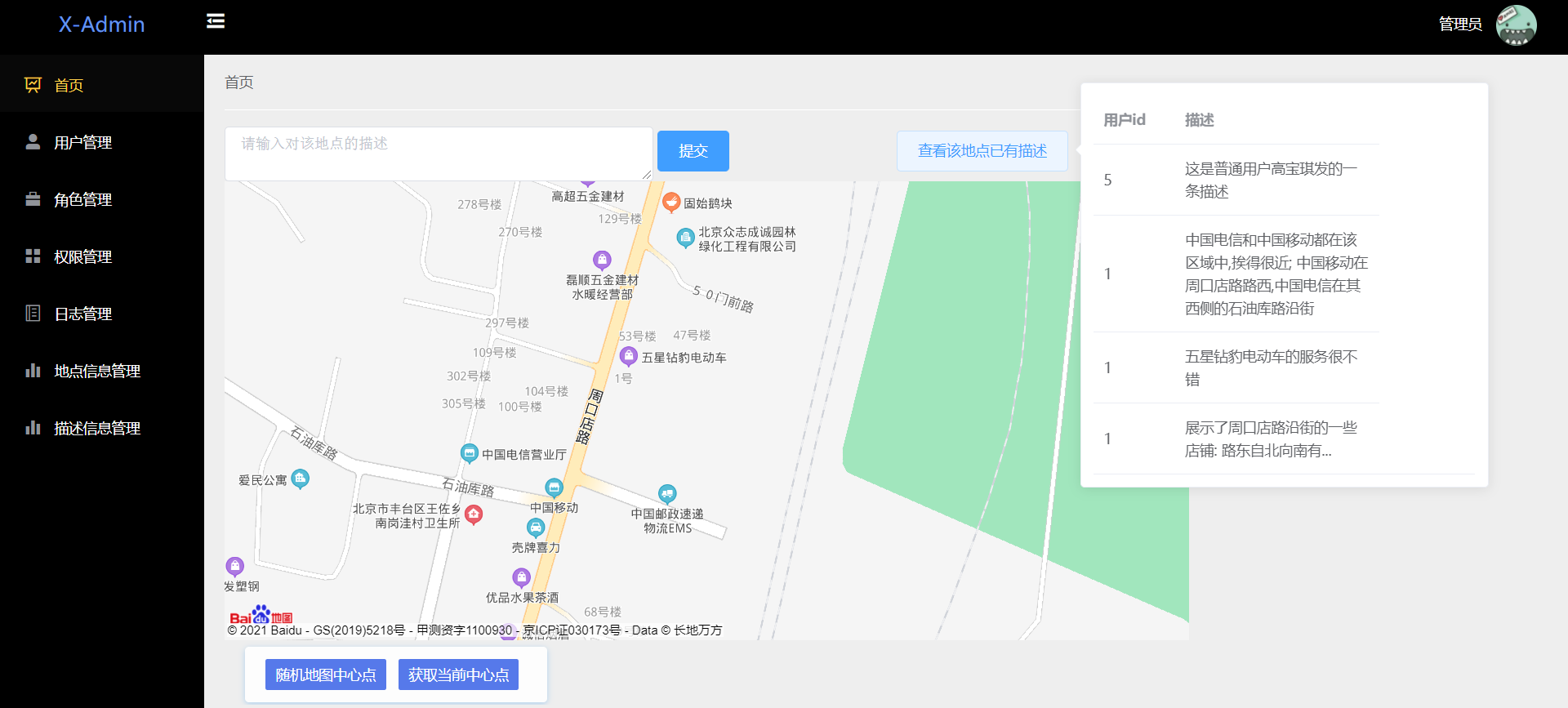
该系统不仅实现了地图描述提交功能,还实现了对各个地点、各个描述的管理。用户分为“超级管理员”和“普通用户”两种；他们都可以提交描述，但超管具有所有权限，对所有的地点和描述都可以进行操作；而普通用户只有对自己提交的描述进行操作的权限。

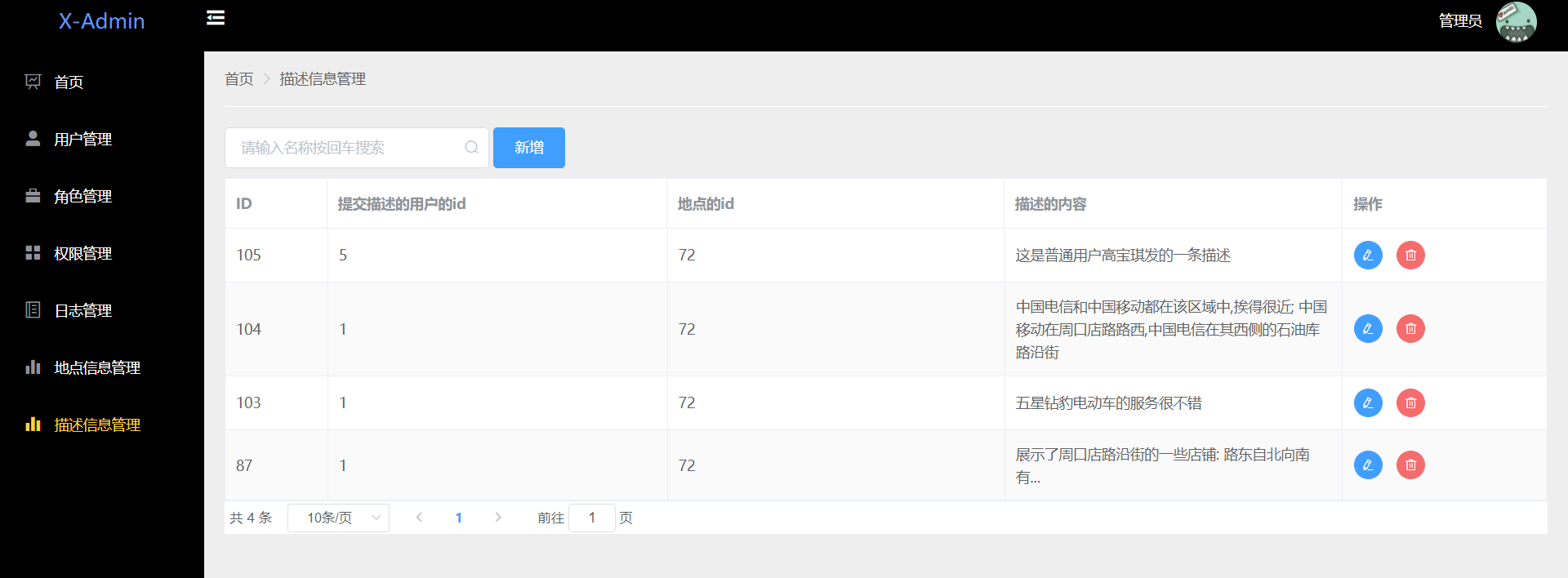
**普通用户操作界面：**





**超级管理员操作界面：**





## 3.3 项目难点分析

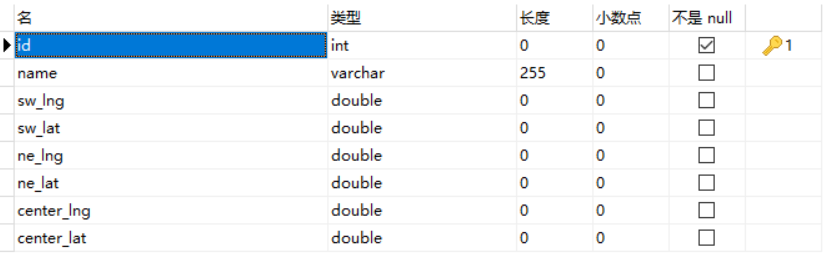
### 3.3.1 难点一 如何实现对一个地点可以添加多条描述，且能区分出每条描述是那个用户所添加。

为解决这个问题，我们在数据库中建立了：

**t\_user表：**



**position表：**



**description表：**



由此一来，就可以在尽可能节省空间的前提下，实现描述的多条存储；而且可以清楚每一条描述是那个用户对于那个地点提交的。

### 3.3.2 难点二 学会Vue特殊的函数调用方法。

如:

$.get("/api/position/" + "/center?pageNum=" + this.pageNum + "&pageSize=" + this.pageSize

+ "&lng=" + cen.lng.toFixed(***precision***) + "&lat=" + cen.lat.toFixed(***precision***)).then(res => {...}

是通过像网址一样的东西去调用Controller中的函数。

# 四、结论与展望

通过这段时间的设计与开发，总体完成了以下几项工作：

1. 对地图数据进行处理，并在前端页面上图形化操作，完成了基于Vue2.0 + ElementUI的页面设计。

2. 完成了基于SpringBoot+Mybatis-plus的Java后端。

3. 用Mysql数据库进行地图描述信息管理，实现在管理系统中对描述信息的新增，编辑，查询，删除功能。

4. 收集了一定数量的高质量地图描述。

5. 对系统进行测试，并解决了测试中发现的部分问题。

**项目的不足：**

用户点击提交按钮时小概率会出现无法成功提交信息到后台的情况，目前尚未完全解决，但通过优化提高了成功率。

**展望：**

随着近年来人工智能和大数据突飞猛进的发展，将地理信息技术和大数据结合成为不可阻挡的趋势。本系统再进一步完善后，可以为Map2Text任务的训练阶段收集数量多、质量高、形式多样的地图描述文本，为计算机自动完成地图到文本的转换收集训练数据，最终定义一种新概念的地图。

# 五、参考资源

https://lbsyun.baidu.com/jsdemo.htm#aCreateMap

https://www.w3school.com.cn/js/index.asp

https://www.w3school.com.cn/html/index.asp

https://element.eleme.cn/#/zh-CN

https://www.bilibili.com/video/BV1ey4y1h7ej?p=2

# 六、组员课题完成情况详细说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 组员一 | 班级 | 姓名 | 学号 | 贡献度 |
| 计科1802 | 高宝琪 | 2018040382 | 40% |
| 贡献描述 | 1. 实现了主页的通过pid定位和“查看该地点已有描述”按钮的功能 2. 实现了系统中“描述信息管理”和“地点信息管理页面” 3. 地图文本描述数据的收集 4. 实验报告摘要和第三章的编写 | | | |
| 组员二 | 班级 | 姓名 | 学号 | 贡献度 |
| 计科1802 | 曹贤鹏 | 2018040385 | 40% |
| 贡献描述 | 1. 地图文本描述页面主要功能的实现 2. 地图文本描述数据的收集 3. 文本评价预筛选机制的实现 4. 将项目上传GitHub，编写ReadMe文档 5. 实验报告第二章和第五章的编写 | | | |
| 组员三 | 班级 | 姓名 | 学号 | 贡献度 |
| 计科1802 | 徐正阳 | 2018040379 | 20% |
| 贡献描述 | 1. 地图文本描述数据的收集 2. 实现了系统中的用户管理、角色管理、权限管理和日志管理 3. 系统运行测试 4. 实验报告第一章和第四章的编写 | | | |