

地信 2021 级《地理信息系统开发实习》 课程设计报告

专 业： _____ 地理信息科学 _____
班 级： _____ 地信 21-1 班 _____
姓 名： _____ 马骁 _____
学 号： _____ 07212393 _____

中国矿业大学环境与测绘学院

2024.07

目录

1 实验过程.....	1
2 个人工作介绍.....	1
3 具体工作.....	2
3.1 总体介绍.....	2
3.2 系统功能界面开发.....	4
3.3 官方旅游路线推荐功能开发.....	5
3.3 AI 旅游助手开发.....	9
3.4 其它功能补充.....	14
4 总结.....	16

1 实验过程

疫情退去，社会逐渐回归正常状态。疫情期间的封闭生活让人们更加渴望出行，同时现在的科技发展也促使智慧旅游变得越来越重要。智慧旅游利用现代科技手段，如大数据、人工智能、物联网等，提供个性化、高效、安全的旅游服务。它不仅提升了游客的体验，还优化了旅游资源的配置和管理。

我们小组紧贴当下需求热点，开发出智行中国—融合 AI 和旅游规划的智慧旅游系统。前期工程我们手机各个大城市的旅游数据，如景区数据，酒店数据，用户评价数据，美食数据等基础数据，开发出系统原型，使用增量模式开发系统。逐步按模块添加功能。

首先是查询功能，是 GIS 系统的必备基础功能，提供了城市选择（以动画形式切换地图中心位置，并加载城市矢量边界），关键字搜索，标记中心点搜索周边 POI 数据（创新功能）利用了缓冲区分析，距离和面积查询。

第二是路线规划功能，可以选择出行方式如驾车、步行还可以选择路线规划模式，如最短时间，最低价格，躲避拥堵等。

第三是酒店价格查询功能，可以查询酒店即最近一个月的价格走势，为用户选择最经济的旅行方案。

第四是旅游路线推荐功能，可以通过在地图上标记点自定义旅游路线并发布，展示在地图上，使旅游方案更加灵活。

第五使官方旅游路线推荐功能，在系统中提供几条热度最高的旅行路线，供用户选择，并添加评论区功能，使用户能够互相交流。

最后是 AI 旅行助手功能，使用大模型接口提供了基础的日常对话，并提供地图文本生成，即输入文字在地图上生成相应的点标记，实现智能旅游地图。

小组成员共同开发系统，持续时间长达三个月，使用了各种 GIS 开发技术，并结合 AI 等，提升系统的可用性。

2 个人工作介绍

作为小组系统开发成员，我主要负责前端开发，完成了界面配色设计，动画设计，一部分功能开发，主要有：

- （1）系统主页面动画设计，功能页面配色设计与开发
- （2）官方旅游路线推荐模块开发，路线推荐+评论区
- （3）AI 旅游助手开发，日常对话+地图联动（文本—地图生成）
- （4）地铁图功能开发，系统定位功能开发

(5) 修改系统 bug，如功能界面布局，点标记 popup 样式等

3 具体工作

3.1 总体介绍

本系统名为“智行中国—融合 AI 和旅游规划的智慧旅游系统”系统功能结构图如下：

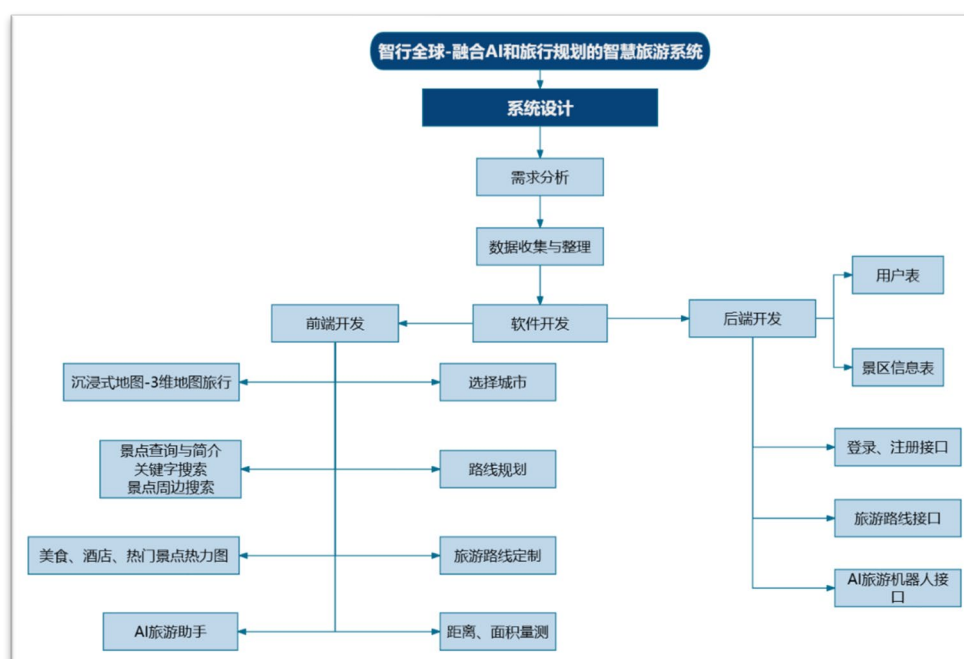


图 3-1-1 系统功能结构图

本系统是一个完整的 WebGIS 系统，包括前端开发和后端开发，并进行对接，后端服务主要有登录注册，景区信息，路线数据，AI 模型接口等服务，为系统开发提供了基础。前端包括上述六大模块，给用户一个良好的使用体验。

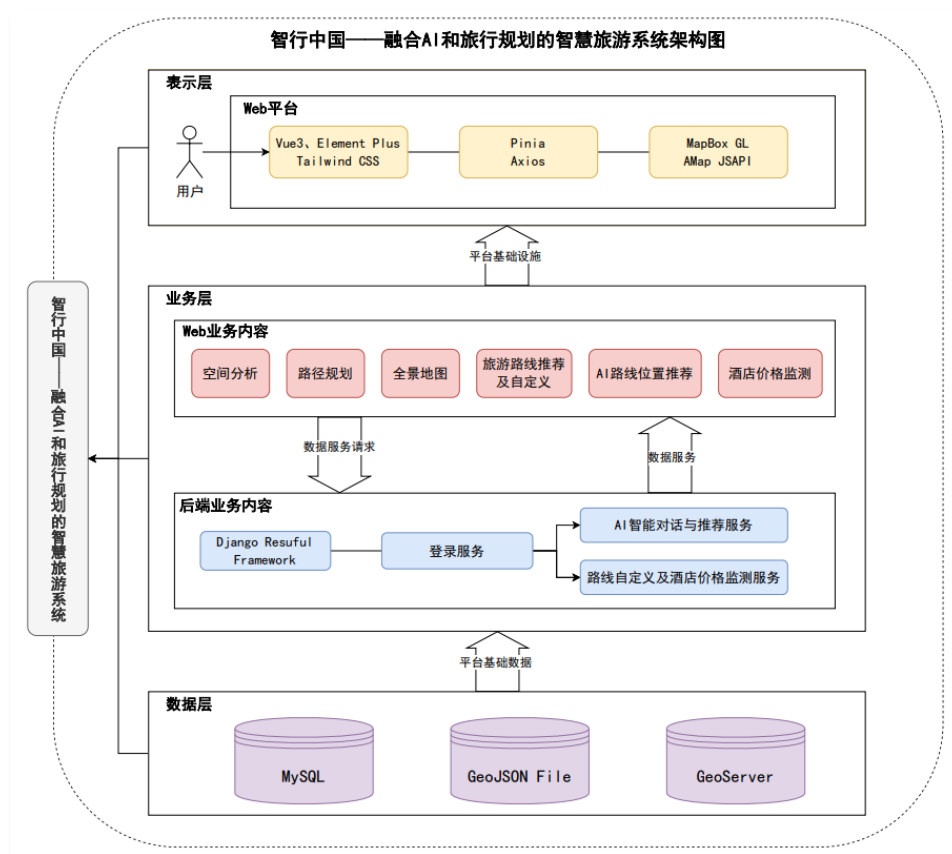


图 3-1-2 系统架构图

系统架构如上所示，前端使用技术栈为 Vue3+ElementPLUS（组件样式库），使用 Pinia 进行状态管理，实现单点登录，用户信息保存的功能。前后端通信使用 Axios 库，它基于 XHR 封装，是目前使用最广泛的异步请求库。地图组件库使用 MapBoxGL 它基于 WebGL 开发能够实现三维地图渲染，AMapJS API，它是高德地图的组件库，能够提供多种地图功能，查询，路径规划，定位等。

业务层包含空间分析、路径规划、全景地图、旅游路线推荐及自定义：根据用户偏好推荐旅游路线，并支持用户自定义路线、AI 路线位置推荐：利用 AI 技术为用户推荐旅游路线的最佳位置、酒店价格监测：监测酒店价格波动，为用户提供最优价格信息。

后端逻辑层使用 Python Django 框架，提供接口，数据层使用 MySQL 数据库，GeoJSON 文件存储地理数据，GeoServer。

3.2 系统功能界面开发

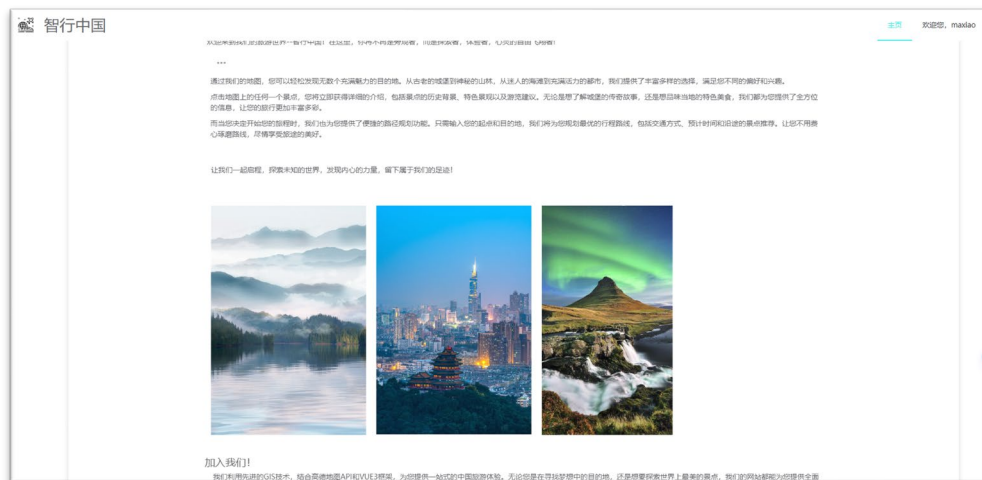


图 3-2-1 首页动画效果设计

系统首屏实现图片飞入效果，需要监听鼠标滚动事件，获取视口的高度，获取元素距离视口顶部的距离，并设置出发点，检测元素是否进入视口，添加 active 样式。实现函数如下：

```
1.  const reveal = () => {
2.      let reveals = document.querySelectorAll(".main-pic");
3.      let reveals1 = document.querySelectorAll(".main-pic span .el-image");
4.      for (let i = 0; i < reveals.length; i++) {
5.          let windowHeight = window.innerHeight;
6.          let revealTop = reveals[i].getBoundingClientRect().top;
7.          let revealPoint = -100;
8.          if (revealTop < windowHeight - revealPoint) {
9.              reveals[i].classList.add("active");
10.             reveals1[i].classList.add("active");
11.         } else {
12.             reveals[i].classList.remove("active");
13.             reveals1[i].classList.remove("active");
14.         }
15.     }
16. };
```

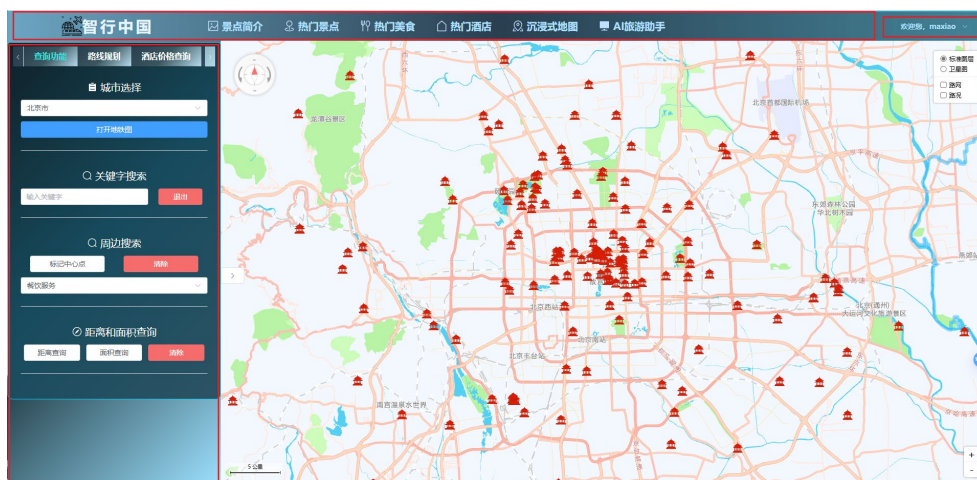


图 3-2-2 功能页布局、配色设计

使用 linear-gradient 进行渐变色的样式设计，提升界面美观性

3.3 官方旅游路线推荐功能开发



图 3-2-1 上海四日游路线推荐

为了实现推荐旅游路线加载，实现 `loadRoute` 函数，通过 `routeId` 和 `routes` 中的数据通过 `find` 函数进行 `id` 匹配实现数据的筛选，遍历路线中的每个点，添加标记和点击事件，并设置点的 `popup` 属性框。初始化驾车路线规划服务，设置起点、终点和途经点，进行路线规划，设置路线样式加载路线，并播放动画。

```
1.  const loadRoute = (routeId) => {
2.      clearMap();
3.      const route = routes.value.find(r => r.id === routeId);
4.      if (!route) return;
5.      const path = [];
6.      route.points.forEach(point => {
7.          const marker = new AMap.Marker({
```

```

8.         position: [point.lng, point.lat],
9.         title: point.name,
10.    });
11.    marker.on('click', () => {
12.        const infoWindow = new AMap.InfoWindow({
13.            content: `
14.                <div style='font-size: 14px;font-
15.                family:"Noto Serif SC", serif;max-width:400px;min-width:400px'>
16.                    <p>景点名称: ${point.name}</p>
17.                    <p>时间: ${point.time}</p>
18.                    <p>注意事项: ${point.attention}</p>
19.                </div>`,
20.            offset: new AMap.Pixel(0, -30),
21.        });
22.        infoWindow.open(refgmap.value, marker.getPosition());
23.    });
24.    markerList.value.push(marker);
25.    refgmap.value.add(markerList.value);
26.    driving = new AMap.Driving({
27.        map: refgmap.value,
28.        panel: 'panel',
29.        hideMarkers: true,
30.        showTraffic: false,
31.        isOutline: false
32.    });
33.
34.    const startLngLat = route.points[0];
35.    const endLngLat = route.points[route.points.length - 1];
36.    const waypoints = route.points.slice(1, route.points.length - 1).map(poi
nt => [point.lng, point.lat]);
37.
38.    driving.search(
39.        [startLngLat.lng, startLngLat.lat],
40.        [endLngLat.lng, endLngLat.lat],
41.        {
42.            waypoints: waypoints
43.        },
44.        function (status, result) {
45.            if (status === 'complete') {
46.                console.log('绘制驾车路线完成');
47.                const steps = result.routes[0].steps;
48.                const lineArr = [];
49.                for (let i = 0; i < steps.length; i++) {

```



```

50.         const path = steps[i].path;
51.         for (let j = 0; j < path.length; j++) {
52.             lineArr.push(path[j]);
53.         }
54.     }
55.
56.     animateRoute(lineArr);
57.     polylineCar = new AMap.Polyline({
58.         path: lineArr,
59.         strokeColor: "#AF5",
60.         strokeWeight: 1,
61.         strokeOpacity: 0.8,
62.         strokeStyle: "solid",
63.     });
64.     refgmap.value.add(polyline);
65.
66.     } else {
67.         console.log('获取驾车数据失败: ' + result);
68.     }
69. }
70. );
71.
72. refgmap.value.setFitView();
73. };

```

实现 animateRoute 函数，代码如下：

通过 AMap 插件 AMap.MoveAnimation 来实现地图上一辆车沿着指定路线移动，首先创建一个小车得到标记，创建一条线表示已经过的路线，当车移动时，同步更新已经过的路线，启动动画，让车沿着指定路线移动。

```

1.     const animateRoute = (lineArr) => {
2.         AMap.plugin(['AMap.MoveAnimation'], () => {
3.             carMarker = new AMap.Marker({
4.                 map: refgmap.value,
5.                 position: lineArr[0],
6.                 icon: "https://a.amap.com/jsapi_demos/static/demo-center-
v2/car.png",
7.                 offset: new AMap.Pixel(-13, -26),
8.             });
9.             passedPolyline = new AMap.Polyline({
10.                 map: refgmap.value,
11.                 strokeColor: "#AF5",
12.                 strokeWeight: 3,
13.             });
14.

```




图 3-2-3 评论区功能

实现评论区界面，使用户能够进行旅游相关的交流。

3.3 AI 旅游助手开发

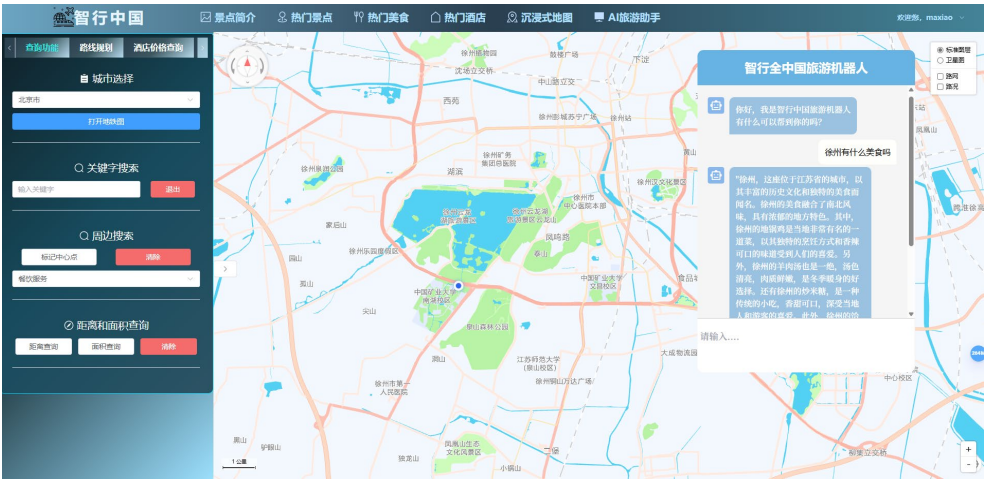


图 3-3-1 AI 旅游助手日常对话



图 3-3-2 AI 旅游助手—地图联动

构建 AI 对话机器人，代码如下：

首先导入使用的各个模块，定义 config 来配置 Axios，定义了 url 即请求地址。AibotClick 函数中对登录进行了校验。chatBot 函数初始化了聊天功能，获取发送按钮，输入框和聊天框，定义 createChatLi 函数创建聊天列表项，AIIocationFunc 函数用来进行文本-地图生成。generateResponse 函数生成 AI 的响应，异步请求 AI 接口，并处理响应数据，从本地获取景点数据并匹配 AI 返回的位置，调用 AIIocationFun 添加标记到地图。handleChat 函数处理用户发送的消息，创建用户消息列表，发送给 AI 生成响应。

```

1.   import { ref, inject } from 'vue';
2.   import axios from 'axios';
3.   import { refAMapInstance, refgmap } from '../views/备用/MapRef.js';
4.   import pinia from '@tools/globalStoreActive.js';
5.   import { useLoginStore } from "@tools/loginStore";
6.   import { ElMessage } from "element-plus";
7.
8.   export let AIIocation = [];
9.   export let AIIocationObj = [];
10.
11.   let AIMap = null;
12.   let markers = [];
13.
14.   let data = new FormData();
15.   let AiIsChecked = ref(false);
16.   const store = useLoginStore(pinia);
17.
18.   let config = {
19.     method: 'post',

```

```

20.     url: 'https://webgis-server.teduis.com/api/v1/ai/',
21.     headers: {
22.         'Authorization': 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2
tlb190eXB1IjoieWNjZXNzIiwiaXhwIjozIiwiaWF0IjE0IyLCJpYXQiOiJE3MTg1Mzc4MjIsImp0aSI6Ijk4M
2MSYzVhOTg3NDQyNzBiNDQzYmRjZmQ1ZDUxMjYxIiwidXNlcl9pZCI6MX0.D46B0ZBA05cy6vS1xyxm80ek
EdQGV7_Xw9vSa9FoGvQ',
23.         'Accept': '*/*',
24.         'content-type': 'application/x-www-form-urlencoded'
25.     },
26.     data: data
27. };
28.
29.     const AIBotClick = () => {
30.         if (!store.isLogin) {
31.             ElMessage({
32.                 message: '请先登录',
33.                 type: 'error'
34.             });
35.             return;
36.         }
37.         config.headers.Authorization = `Bearer ${store.token}`;
38.         AiIsChecked.value = !AiIsChecked.value;
39.     };
40.
41.     function chatBot() {
42.         const sendChatBtn = document.querySelector(".chat-input span");
43.         const chatInput = document.querySelector('.chat-input textarea');
44.         const chatbox = document.querySelector('.chatbox');
45.
46.         let userMessage;
47.         const inputInitHeight = chatInput.scrollHeight;
48.
49.         const createChatLi = (message, className) => {
50.             const chatLi = document.createElement("li");
51.             chatLi.classList.add("chat", className);
52.             let chatContent = className === "outgoing" ? `<p></p>` : `<span clas
s="material-symbols-outlined">smart_toy</span><p></p>`;
53.             chatLi.innerHTML = chatContent;
54.             chatLi.querySelector("p").textContent = message;
55.             return chatLi;
56.         };
57.
58.         const AILocationFunc = () => {
59.             const icon = {

```

```

60.         type: 'image',
61.         image: "https://mxorz12.cn/%E6%97%85%E8%A1%8C%E6%99%AF%E5%8C%BA.
png",
62.         size: [26, 26],
63.         anchor: 'bottom-center',
64.     });
65.     AIMap = new refAMapInstance.value.LabelsLayer({
66.         zooms: [0, 20],
67.         zIndex: 1010,
68.         collision: false
69.     });
70.     refgmap.value.add(AIMap);
71.
72.     AILocationObj.forEach((data) => {
73.         const { name, coordinates, description } = data;
74.         const marker = new refAMapInstance.value.LabelMarker({
75.             position: [coordinates.longitude, coordinates.latitude],
76.             icon,
77.             extData: {
78.                 name: name,
79.                 desc: description,
80.             },
81.         });
82.         markers.push(marker);
83.         marker.on('click', (e) => {
84.             const extData = e.target.getExtData();
85.             const infoWindow = new refAMapInstance.value.InfoWindow({
86.                 content: `
87.                 <div style='font-size: 14px;font-
family:"Noto Serif SC", serif;max-width:400px;min-width:400px'>
88.                     <p>景点名称: ${extData.name}</p>
89.                     <p>简介: ${extData.desc}</p>
90.                 </div>`,
91.                 offset: new refAMapInstance.value.Pixel(0, -30),
92.             });
93.             infoWindow.open(refgmap.value, e.target.getPosition());
94.         });
95.     });
96.     AIMap.add(markers);
97.     const xuzhouCoordinates = [117.284124, 34.205768];
98.     refgmap.value.setZoom(12);
99.     refgmap.value.panTo(xuzhouCoordinates);
100.
101.     });

```

```

102.     const generateResponse = (incomingChatLi) => {
103.         const messageElement = incomingChatLi.querySelector("p");
104.         async function fetchData() {
105.             try {
106.                 const response = await axios(config);
107.                 if (response.data.hasOwnProperty('location_name')) {
108.                     messageElement.textContent = JSON.stringify(response.data.reply);
109.                 } else {
110.                     messageElement.textContent = JSON.stringify(response.data);
111.                 }
112.                 const AILocation = response.data.location_name;
113.                 const res = await axios.get('../public/spot10.json');
114.
115.                 for (let location of res.data.locations) {
116.                     for (let aiLocation of AILocation) {
117.                         if (location.name === aiLocation) {
118.                             AILocationObj.push(location);
119.                         }
120.                     }
121.                 }
122.             } catch (error) {
123.                 console.log(error);
124.             }
125.         }
126.         fetchData().then(() => {
127.             AILocationFunc();
128.         });
129.     };
130.     const handleChat = () => {
131.         userMessage = chatInput.value.trim();
132.         data.append('message', userMessage);
133.         data.append('model', 'MoonshotAI');
134.         data.append('mode', 'json');
135.         if (!userMessage) return;
136.         chatInput.value = "";
137.         chatInput.style.height = `${inputInitHeight}px`;
138.
139.         chatbox.appendChild(createChatLi(userMessage, "outgoing"));
140.         setTimeout(() => {
141.             const incomingChatLi = createChatLi("思考中", "incoming");
142.             chatbox.appendChild(incomingChatLi);
143.             generateResponse(incomingChatLi);

```

```

144.         }, 500));
145.     };
146.     chatInput.addEventListener("input", () => {
147.         chatInput.style.height = `${inputInitHeight}px`;
148.         chatInput.style.height = `${chatInput.scrollHeight}px`;
149.     });
150.
151.     chatInput.addEventListener("keydown", (e) => {
152.         if (e.key === "Enter" && !e.shiftKey && window.innerWidth > 800) {
153.             e.preventDefault();
154.             handleChat();
155.         }
156.     });
157.     sendChatBtn.addEventListener("click", handleChat);
158. }
159. export { AIBotClick, AiIsChecked, chatBot };

```

3.4 其它功能补充

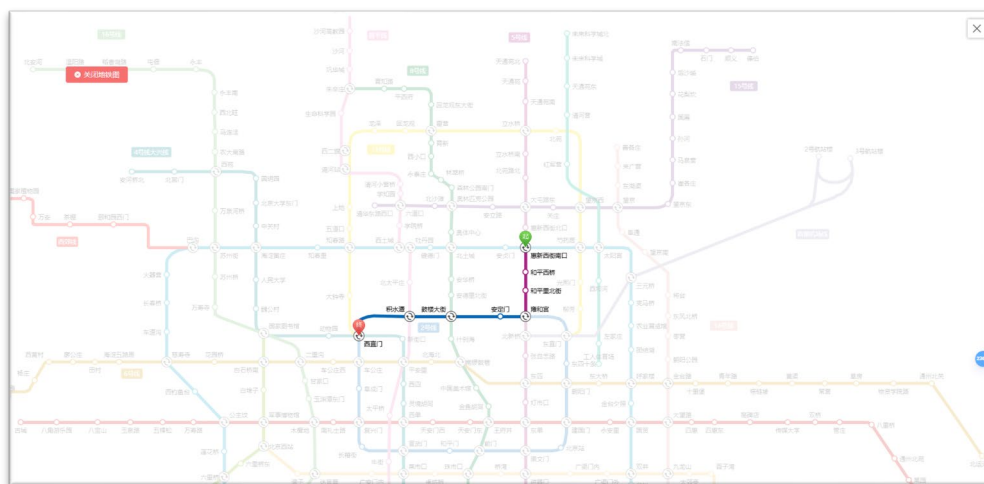


图 3-4-1 地铁图路径查询功能

使用高德地铁图 API 完成地铁路径查询功能开发，能够选择具体线路，起始点和结束点并搜寻最短路径。

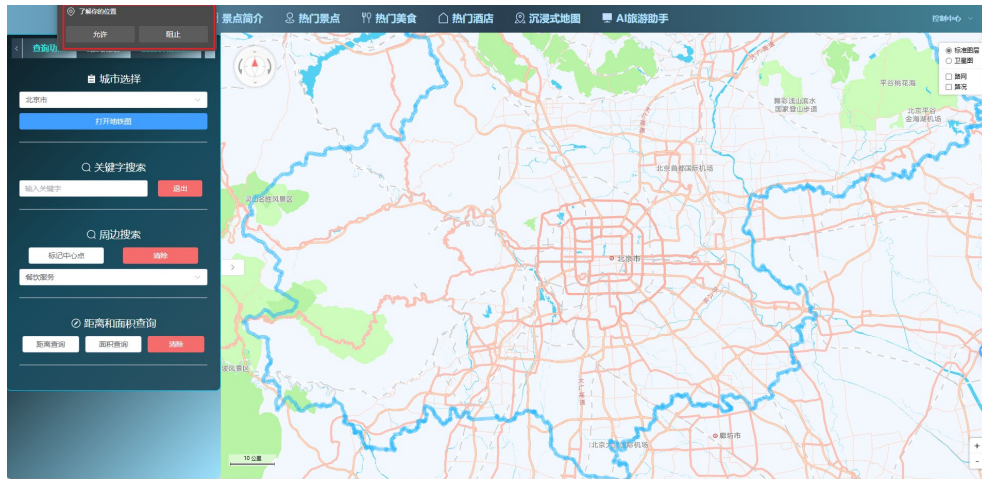


图 3-4-2 定位功能

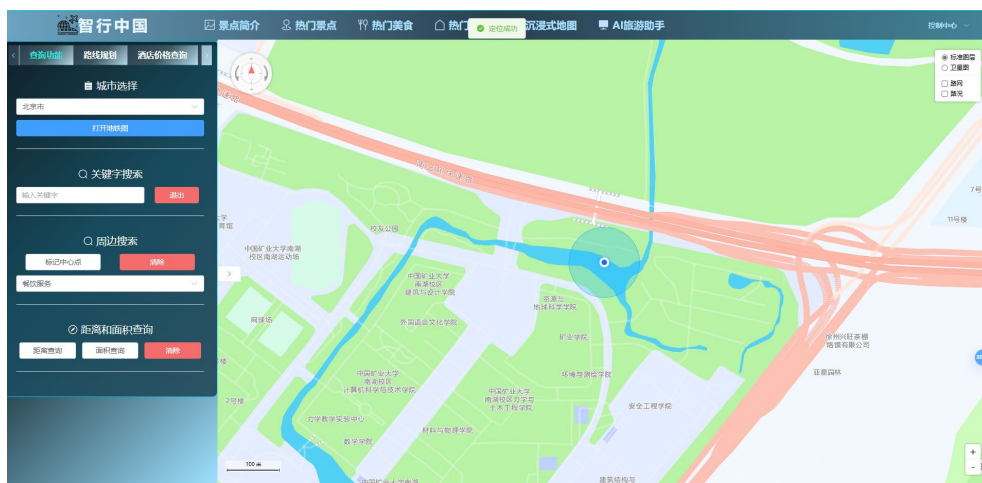


图 3-4-3 定位成功

定位功能使用 AMap.Geolocation 插件，通过 ElementPLUS 的 ElMessage 进行消息提醒，代码如下：

```
1. var geolocation = new AMap.Geolocation({
2.   enableHighAccuracy: true, //是否使用高精度定位，默认:true
3.   timeout: 10000,           //超过 10 秒后停止定位，默认: 5s
4.   position: 'RB',           //定位按钮的停靠位置
5.   offset: [10, 20],         //定位按钮与设置的停靠位置的偏移量，默认: [10, 20]
6.   zoomToAccuracy: true,     //定位成功后是否自动调整地图视野到定位点
7.
8. });
9. gmap.addControl(geolocation);
10. geolocation.getCurrentPosition(function (status, result) {
11.   if (status == 'complete') {
12.     console.log("定位结果");
13.     console.log(result);
14.     ElMessage({
15.       message: '定位成功',
16.       type: 'success',
```

```
17.         plain: true,
18.     });
19.     } else {
20.         ElMessage({
21.             message: '定位失败',
22.             type: 'warning',
23.             plain: true,
24.         });
25.     }
26. });
```

4 总结

通过这次系统开发，参加了全国大学生测绘学科创新开发比赛，我们在长达三个月的打磨中，终于使系统变得较为完整，功能全面。除了提供基础的 GIS 功能外，我们还优化了系统界面，精心设计了 UI，并加入了创新点——通过 AI 大模型实现文本-地图生成，充分体现了“智慧”一词的含义。

在这次系统开发中，我不仅锻炼了开发技术，能够熟练使用 Vue 框架开发 WebGIS 应用，还提升了解决 Bug 的能力。这次经历让我对整个项目的开发流程有了更深入的理解，也学会了如何在团队中高效协作。特别是面对技术难题时，通过不断学习和尝试，我逐渐找到了问题的解决方案，极大地提升了我的技术水平和自信心。

此外，这次比赛还让我深刻体会到创新的重要性。通过不断思考和尝试新的技术和方法，我们不仅提升了系统的功能性和用户体验，还让我们的作品在比赛中脱颖而出。这次经历无疑是我学习生涯中一次重要的历练，为我今后的学习和工作打下了坚实的基础。