基于WebGIS的

全民智慧化健身系统

详细设计说明书

（地信2019级）

组员： 谢逸云（学号**07192391**）

张文浩（学号**07192394**）

张思雨（学号**07192419**）

**中国矿业大学环境与测绘学院**

**2022年7月**

目录

[1 引言 3](#_Toc3868)

[1.1 编写目的 3](#_Toc3448)

[1.2 项目背景 3](#_Toc27499)

[1.3 定义 5](#_Toc23535)

[1.4 参考资料 5](#_Toc27390)

[2 系统架构 7](#_Toc23790)

[3 运行环境 8](#_Toc9973)

[3.1 硬件环境 8](#_Toc12745)

[3.2 软件环境 8](#_Toc15681)

[3.3 网络环境 9](#_Toc8947)

[3.4 软件使用人员 9](#_Toc30145)

[4 系统登录与注册功能模块 9](#_Toc23835)

[5 系统主界面 13](#_Toc1394)

[6 定位查询子系统 14](#_Toc14382)

[6.1 关键词查询 15](#_Toc2023)

[6.2 周边查询 16](#_Toc24421)

[6.3 热点查询 18](#_Toc18845)

[7 路径规划子系统 20](#_Toc10955)

[7.1 驾车路径规划 21](#_Toc4594)

[7.2 骑行路径规划 22](#_Toc15834)

[7.3 步行路径规划 24](#_Toc20928)

[8 运动方案子系统 25](#_Toc23585)

[8.1 骑行路线规划 25](#_Toc1008)

[8.2 散步路线规划 27](#_Toc11291)

[9 总结 28](#_Toc14208)

# 引言

## 编写目的

本文档对区域内健身场所空间位置数据与属性数据（包括健身器材信息、适宜人群、开放时间）进行数据挖掘及匹配并结合需求、区域分布特点、国家政策以及其他公共交通服务等，对运动场所定位查询、位置信息显示、路径规划、个性化运动方案设计等，总结出基于WebGIS全民智慧化健身系统需求分析报告，为系统的建设提供基础依据。

本文档主要是开发人员进行系统设计开发的主要依据，是数据分析、设计和软件系统开发的主要依据之一。

## 项目背景

2020年3月，中共中央办公厅、国务院办公厅印发《关于构建更高水平的全民健身公共服务体系的意见》，其中的一系列举措绘就了发展新蓝图，为未来我国全民健身发展指明了方向，明确了路径。

自中华人民共和国成立伊始，党和政府就十分重 视体育工作，毛泽东提出的“发展体育运动、增强人民体质”是党和政府对建设一个基于健康国民和健康社会的强大祖国的期盼和决心。改革开放 40 年，是我国经济快速发展的 40 年，也是我们重返国际舞台并逐步走向世界舞台中央的 40 年，在此过程中，中国需要让世界知道和了解，而奥运会和种种国际体育竞技场无疑是中国人民表现民族精神的重要舞台，因此竞技体育成为振奋民族精神的途径和让世界了解中国的窗口，因而受到国家的格外重视。像我国的经济发展速度一样，竞技体育水平也突飞猛进，2008 年北京奥运会，我国竞技运动水平站到了世界竞技体育的顶峰，同时，也宣告我国体育事业发展新时代的来临。竞技体育、健康体育、休闲娱乐体育全面协调发展的体育事业发展格局已经开启，在“人民对美好生活的向往就是我们的奋斗目标”宣言的感召下，健康体育尤其受到了关注。伴随着国家的快速发展，从“满足温饱”到“全面小康”，我国人民对美好生活的定义不断更新，不再局限于物质层面，而是从社会、文化、人民的精神与身心健康多方面出发。

相较于欧美，我国健身行业起步晚，主以80后、90后、00后为受众群体，年龄集中于41岁以下；地区健身资源分布不均衡、发展不协同，但产业未来远景可观，潜力巨大。由《2021年Q1中国年轻人运动健身白皮书》可知，智能可穿戴+运动健身逐渐成为中国年轻人重要的生活方式之一，但用户运动健身需求正在产生分化，不同类型的用户拥有不同的运动健身诉求，现阶段用户更期待智能可穿戴服务升级，尤其是运动数据的应用和反馈。在多种运动健身行为中，跑步为年轻人首选，其中室外跑步占比高达73.50%。此外，高消耗的球类运动、减肥塑形、骑行、登山、徒步等运动也占比较高。然而，在实际生活中却缺少提供有效的、准确的、专业的健身场所信息与交通路径规划；个性化设置运动路线以及提供运动指导信息；关于健身的分享、讨论、学习社区的功能集成软件产品。在Web GIS体系框架下，通过智慧化、数字化健身产业信息，搭建健身产业平台，利好明显。

本系统拟采用高德API，以重庆市为例，对重庆全市健身产所信息进行整合，可对健身场所信息添加、具体健身场所查询、定位缓冲区分析查询距离最近健身场所，同时，实现按出行方式实现最优路径规划、提供实时导航与路况可视化、选择运动方式设计个性化运动路径，并提供定位当前位置运动参考路径选择。

本系统分别设置管理员、客户的web端，设置不同使用权限，满足不同用户的使用需求。

## 定义

下面是在文档中的常用缩写语与术语的定义与解释。

本系统：基于WebGIS的全民智慧化健身系统

高德地图API：是为开发者免费提供的一套基于高德地图服务的应用接口。

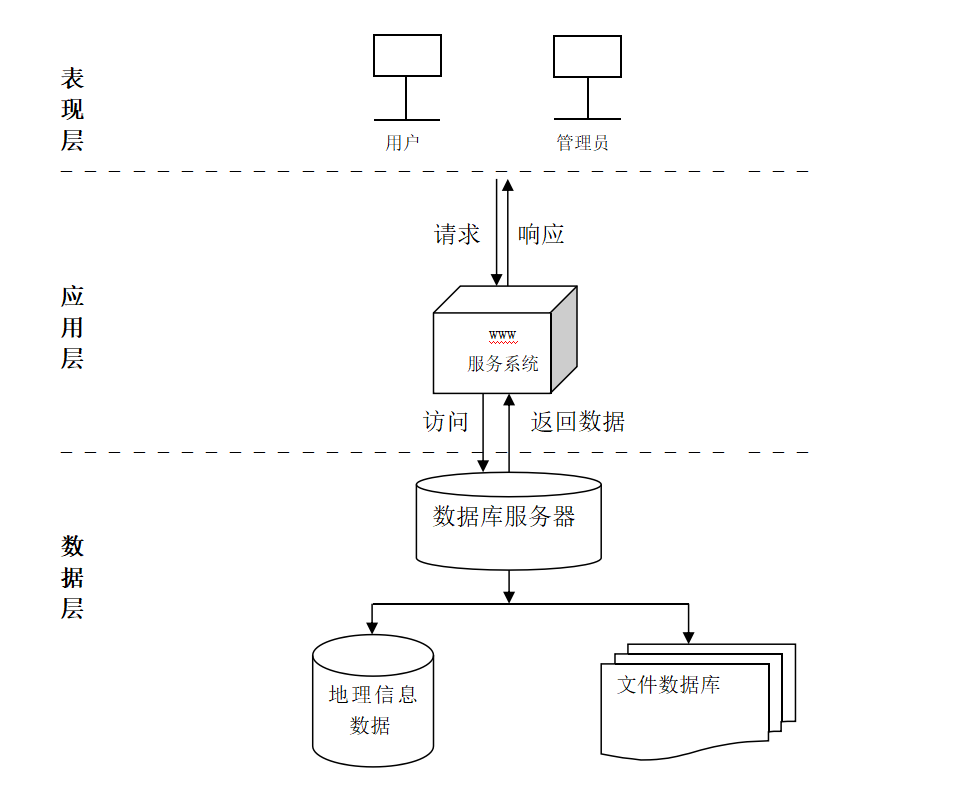
## 参考资料

本文档的主要参考资料包括：

* GB/T8566-1995《信息技术、软件生存期过程》
* GB/T16680-1996《软件文档管理指南》
* GB/T 7929—1995《1:500 1:1000 1:2000 1：10000地形图图式》(1995)
* 毕硕本，《地理信息系统软件工程的原理与方法》 科学出版社
* 戴健,史小强,程华.“十四五”时期我国全民健身发展的环境变化与战略转型[J/OL].体育学研究:1-12[2022-07-13].DOI:10.15877/j.cnki.nsic.2022052 6.001.
* 冯振伟,田丰,陈召,石立江.全民健身智慧化的条件诉求、阻滞因素与优化路径[J/OL].沈阳体育学院学报:1-7[2022-07-13].http://kns.cnki.net/kcms/ detail/21.1081.G8.20220705.1730.008.html
* 刘红建,高奎亭,徐百超.中国全民健身政策体系演进历程、优势特征及效能转化研究[J].体育学研究,2022,36(01):91-102.DOI:10.15877/j.cnki nsic. 20211222.002.
* 吴红波,郭敏,杨肖肖.基于地图API和GIS路径分析的城市公交车路网优化[J].北京交通大学学报,2022,46(01):69-78.
* 王辛岩,李庆华.基于GIS的最短路径分析[J].物流工程与管理,2012,34(06):42-43.
* 程思. 二维GIS中路径分析方法研究与实现[D].中南大学,2010.
* 赵春宇. 高性能并行GIS中矢量空间数据存取与处理关键技术研究[D].武汉大学,2006.
* Data Vision Group Takes Web GIS Transformation Award at Esri Partner Conference[J]. Manufacturing Close - Up,2022.
* [1]Oliazadeh S,Khoram M R,Abbasi M. Design and development of a web GIS for management and performance evaluation of agricultural farms of sahar food industries company (SFIC)[J]. Journal of Physics: Conference Series,2021,1918(4).

# 系统架构

本系统的运动爱好者和有健身需求者为用户主体，可进行数据上传管理、定位查询和位置显示等功能。用一览表及框图的形式说明本系统的系统元素（各层模块、子程序、公用程序等）的划分，扼要说明每个系统元素的标识符和功能，分层次地给出各元素之间的控制与被控制关系



# 运行环境

## 硬件环境

（1）服务器

采用两台服务器，一台用于数据库服务器，一台用做应用服务器，用来存放系统和组件。

机器基本配置（两台情况一样）：

内存：2GB以上；

CPU：双至强2.5GHZ以上；

硬盘：SCSI73G\*3块；

网卡：双千兆。

推荐采用READ5磁盘阵列，一实现系统和数据库的安全备份和灾难恢复。

（2）用户电脑

机器要求内存在512M以上，CPU要求在2.1GHz以上。

## 软件环境

（1）操作系统

从系统软件来看，在服务器端，选用Microsoft Window 2022 Server 。在客户端可以选用Windows 10 ，客户端 Microsoft Edge浏览器。

（2）数据库管理软件

本系统采用的是Microsoft 的SQL SERVER 2010 企业版软件。

（3）GIS平台

经过对软件的稳定性、与其它系统的融合、对数据库的支持、性价比等多个方面的考虑后，决定选用ESRI公司的ArcGIS Server 10版本。

（4）软件配置

系统软件选型表

|  |  |  |  |
| --- | --- | --- | --- |
| **软件分类** | | | **软件选型** |
| 系统软件 | 个人计算机  （服务器） | | Windows 10 |
| 数据库管理系统 | | | SQL Server 2010企业版/Access 2019 |
| GIS软件 | | GIS开发平台 | ArcGIS Server 10 |
| 其他GIS软件 | ArcGIS Desktop |
| 开发工具 | | | Visual Studio 2019 （asp.NET、C#）  Visual Studio Code （python） |

## 网络环境

把该网络连成局域网形式，数据存储在一台个人计算机（服务器）上，用户通过浏览器和服务器进行连接并访问该系统，进行相关的操作。

## 软件使用人员

本系统使用人员分为普通用户和系统管理员。

1. 普通用户：用户一般没有GIS系统的使用素养和充分的计算机知识。故本系统简单易操作，可使用定位查询、位置信息显示、交通路径规划、个性运动方案等功能。
2. 系统管理人员：熟悉GIS专业知识，对数据进行编辑更新，对系统进行维护，需要很高的计算机知识，对系统有比较深的了解，要求精通SQL SERVER数据库，能进行数据库的开发和备份等工作；熟悉网络维护和服务器操作系统维护具有一定的Web开发能力，同时需要进行仔细的系统使用和数据库管理培训。

# 系统登录与注册功能模块

模块的主要功能是：用户通过帐号与密码登录“全民智慧化系统”；同时用户还可以进行新用户注册功能。

实现思路如下：由于老师要求进行WEB端的实现，而连接数据库需要进行数据发布，小组成员无法进行服务器的购买因此小组成员选择通过将数据存入数据通过对象和数组的转换进行判断,进而完成注册和登录功能。

关键代码如下：

登录：

function enter() {

// 获取用户输入的账号

let tel = document.getElementById("tel").value;

console.log(tel);

// 获取用户输入的密码

let pas = document.getElementById("pas").value;

console.log(pas);

// 判断本地是否有数据 如果没有数据直接提示未注册

if (localStorage.length == 0) {

alert("您还未注册")

} else {

let teldata = [] // 创建一个数组 用于存储本地所有已存储的手机号

let pasdata = [] // 创建一个数组 用于存储本地所有已存储的密码

let iddata = [] // 创建一个数组 用于存储本地所有已存储的id

// 循环判断本地是否有次手机号

for (let i = 0; i < localStorage.length; i++) {

// 获取所有的key钥匙

let key = localStorage.key(i)

console.log(key);

// 通过key拿到对应的数据进行判断

let keydata = localStorage.getItem(key); // 拿到对应数据 只不过这时候是字符串

let keyinfo = JSON.parse(keydata) // 将字符串转化为对象的形式

console.log(keyinfo);

// 向数组中添加数据 我们通过下标i的方式添加 这样手机号我密码是对应的 不能通过push添加！！！ 不然手机号和密码是乱的

teldata[i] = keyinfo.tel

pasdata[i] = keyinfo.pas

iddata[i] = keyinfo.id

}

console.log(teldata);

console.log(pasdata);

console.log(iddata);

// 判断此手机号是否注册

if (teldata.indexOf(tel) < 0) { // indexof方法用户查看一个数组中是否有某个值，如果没有它会返回-1，有的话他会返回对应的下标

alert("此账号未注册")

} else {

let index = teldata.indexOf(tel) // 返回对应手机号的下标 我们通过下标去判断密码

if (pasdata[index] != pas) {

alert("密码错误")

} else {

console.log(iddata[index]);

// 定时器

setTimeout(function () {

// 跳转传参

window.location.href = "index.html"

}, 2000); }}}}

注册：

function affirm() {

// 获取账号

let tel = document.getElementById("tel").value;

console.log(tel);

// 获取密码

let pas = document.getElementById("pas").value;

console.log(pas);

if (localStorage.length == 0) { // 判断本地是否有数据 没有的话判断手机号和密码

// 动态向本地添加数据

let dataLength = localStorage.length // 获取现在已有数据的长度 这个长度用于拼接到本地存储的每个key值中 达到一个动态存储的效果 每条本地存储数据需要一个钥匙 也就是获取这条数据的名称 就是key

console.log(dataLength);

// 创建一个对象用于存储用户输入的数据

let data = {}

data.name = name; // 向对象添加昵称

data.tel = tel // 向对象添加手机号

data.pas = pas // 添加密码

data.id = dataLength // 添加用户唯一凭证ID

let info = JSON.stringify(data) // 将对象转化为字符串 因为本地存储只能存储字符串

console.log(info);

// 向本地存储数据 第一个参数就是key钥匙 第二个是我们要存储的数据

localStorage.setItem("key" + dataLength, info);

// 获取本地存储所有数据 查看是否存到本地

console.log(localStorage.valueOf());

// 当存储成功时 启动定时器 两秒钟后跳转到登录页面

}

else {

for (let i = 0; i < localStorage.length; i++) {

// 获取所有的key钥匙

let key = localStorage.key(i)

console.log(key);

// 通过key拿到对应的数据进行判断

let keydata = localStorage.getItem(key); // 拿到对应数据 只不过这时候是字符串

let keyinfo = JSON.parse(keydata) // 将字符串转化为对象的形式

console.log(keyinfo);

// 判断用户输入的信息是否存在

if (keyinfo.tel == tel) { // 判断本地存储的数据中是否有相同的手机号

alert("账号已注册")

break;

} else {

// 动态向本地添加数据

let dataLength = localStorage.length // 获取现在已有数据的长度 这个长度用于拼接到本地存储的每个key值中 达到一个动态存储的效果 每条本地存储数据需要一个钥匙 也就是获取这条数据的名称 就是key

console.log(dataLength);

// 创建一个对象用于存储用户输入的数据

let data = {}

data.tel = tel // 向对象添加手机号

data.pas = pas // 添加密码

data.id = dataLength // 添加用户唯一凭证ID

let info = JSON.stringify(data) // 将对象转化为字符串 因为本地存储只能存储字符串

console.log(info);

// 向本地存储数据 第一个参数就是key钥匙 第二个是我们要存储的数据

localStorage.setItem("key" + dataLength, info);

// 获取本地存储所有数据 查看是否存到本地

console.log(localStorage.valueOf());

// 当存储成功时 启动定时器 两秒钟后跳转到登录页面

setTimeout(function () {

window.location.href = "index.html"

}, 2000)

alert("存储成功,点击后跳转到登录页面")

break;}}}};

登录界面效果如图4-1、4-2所示。

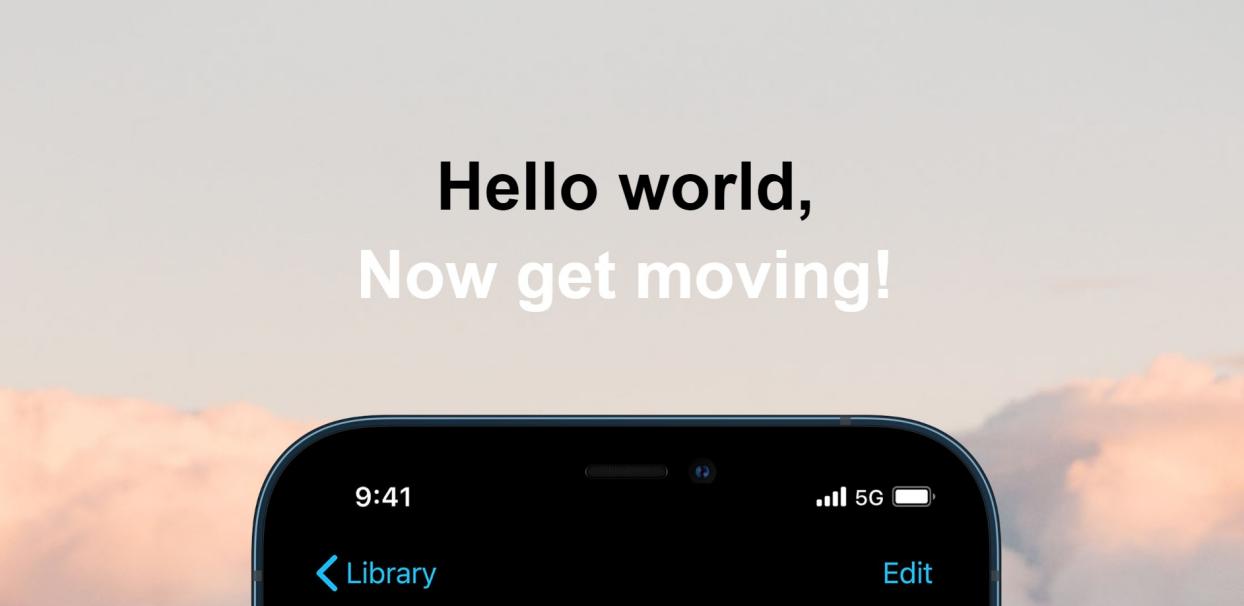


图4-1 登录界面效果

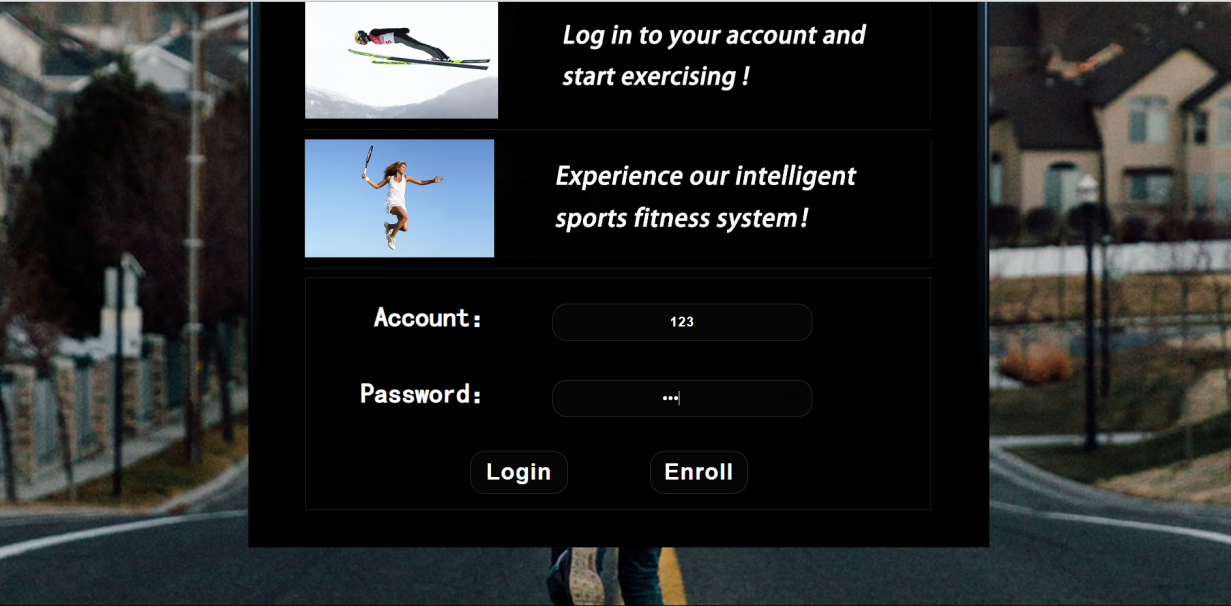


图4-2 登录界面效果

# 系统主界面

本系统主要功能分为定位查询子系统、路径规划子系统以及运动方案子系统三个主要部分，其中主页面展示了各子系统的功能概述，页面主要显示情况及各页面集成。主页面效果如图5-1、5-2、5-3所示。

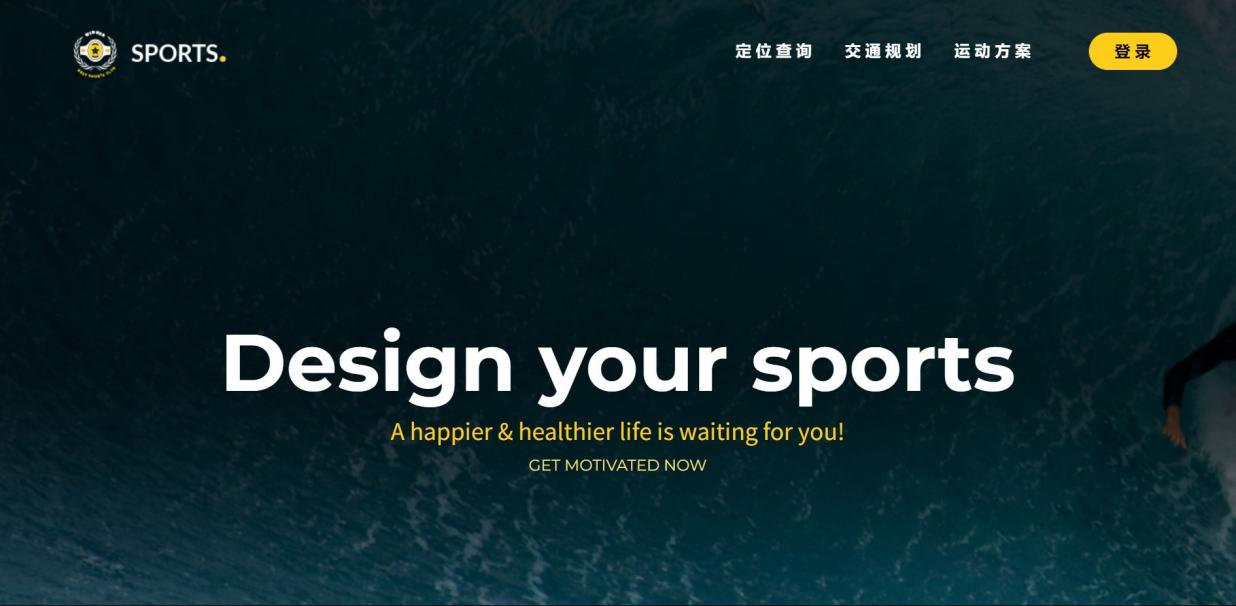


图5-1 主页面效果

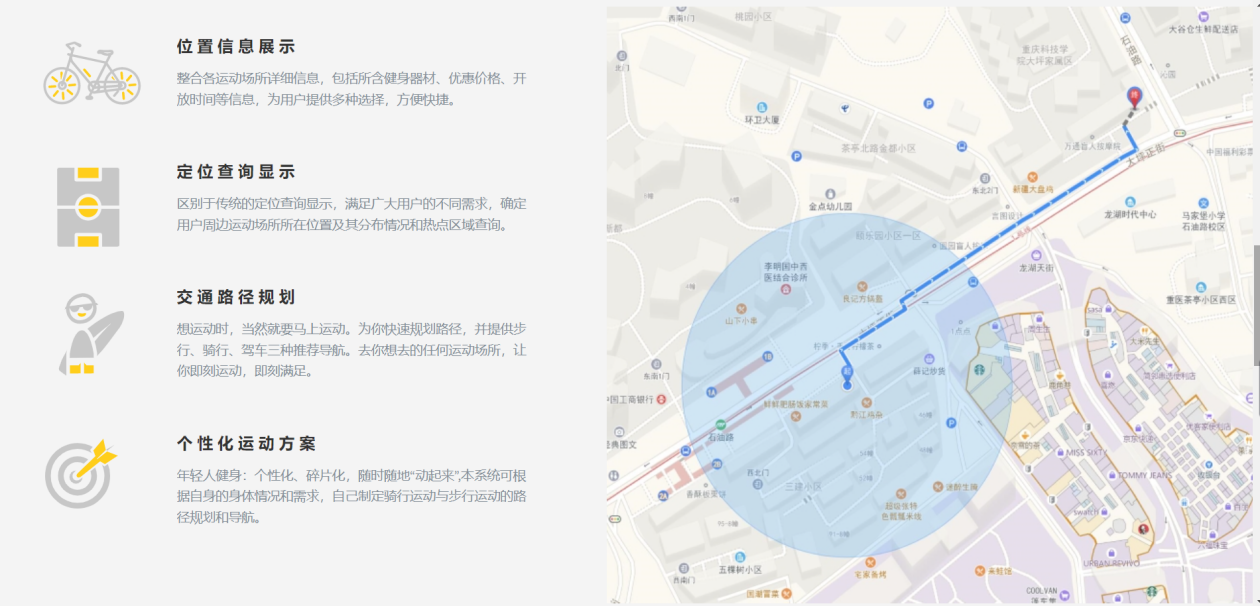


图5-2 主页面效果

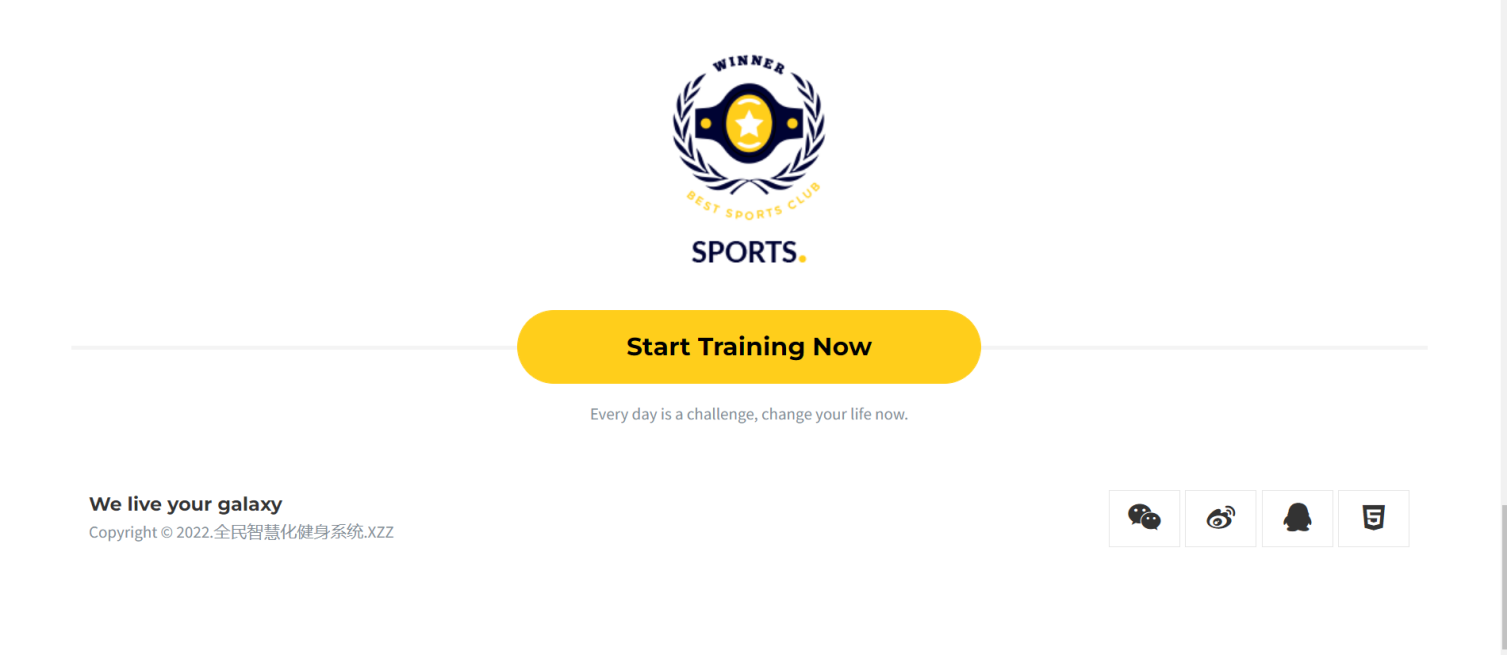


图5-2 主页面效果

# 定位查询子系统

定位查询子系统主要分为关键字查询、周边查询及热点查询三大功能。通过用户输入关键字、周边距离等进行查询显示。查询主页面如图6-1所示。



图6-1 查询主页面显示

## 关键词查询

1. 功能：

通过用户输入查询信息进行关键字检索，在页面左侧展示出查询结果，包括其地址、电话及店铺照片等信息，并在地图上添加mark进行定位，通过点击左侧的查询结果可以进行精准定位，方便用户筛选合适的健身场所。

1. 输入：

关键字

1. 输出：

检索结果及其精准定位

1. 关键代码：

//输入提示

var autoOptions = new AMap.Autocomplete({

city: "铁岭",

input: "tipinput"

});

AMap.plugin(['AMap.PlaceSearch', 'AMap.AutoComplete'], function () {

var auto = new AMap.AutoComplete(autoOptions);

var placeSearch = new AMap.PlaceSearch({

map: map

}); //构造地点查询类

auto.on("select", select);//注册监听，当选中某条记录时会触发

function select(e) {

placeSearch.setCity(e.poi.adcode);

placeSearch.search(e.poi.name); //关键字查询查询

}

});

function Search() {

AMap.service(["AMap.PlaceSearch"], function() {

//构造地点查询类

var placeSearch = new AMap.PlaceSearch({

pageSize: 5, // 单页显示结果条数

pageIndex: 1, // 页码

city: "铁岭", // 兴趣点城市

citylimit: true, //是否强制限制在设置的城市内搜索

map: map, // 展现结果的地图实例

panel: "panel", // 结果列表将在此容器中进行展示。

autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围

});

//关键字查询

placeSearch.search(tipinput.value);

});

}

1. 执行显示：



图6-2 执行结果显示

## 周边查询

1. 功能：

通过用户所在位置进行定位，通过用户输入查询信息及缓冲区半径进行检索，在页面左侧展示出查询结果，包括其地址、电话及店铺照片等信息，并在地图上添加mark进行定位，通过点击左侧的查询结果可以进行精准定位，方便用户其周边一定范围内的健身场所。

1. 输入：

关键字、缓冲区半径、授权定位信息

1. 输出：

检索结果及其精准定位

1. 关键代码：

function Scope()

{

AMap.plugin(["AMap.PlaceSearch"], function() {

//构造地点查询类

var placeSearch = new AMap.PlaceSearch({

type: tipinput.value, // 兴趣点类别

pageSize: 5, // 单页显示结果条数

pageIndex: 1, // 页码

//city: "吉林", // 兴趣点城市

//citylimit: true, //是否强制限制在设置的城市内搜索

map: map, // 展现结果的地图实例

panel: "panel", // 结果列表将在此容器中进行展示。

autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围

});

AMap.plugin('AMap.Geolocation', function() {

var geolocation = new AMap.Geolocation({

enableHighAccuracy: true,//是否使用高精度定位，默认:true

timeout: 10000, //超过10秒后停止定位，默认：5s

position:'RB', //定位按钮的停靠位置

offset: [10, 20], //定位按钮与设置的停靠位置的偏移量，默认：[10, 20]

zoomToAccuracy: true, //定位成功后是否自动调整地图视野到定位点

});

map.addControl(geolocation);

geolocation.getCurrentPosition(function(status,result){

if(status=='complete'){

onComplete(result)

}else{

onError(result)

}

});

});

function onComplete(data) {

var cpoint = data.position; //中心点坐标

placeSearch.searchNearBy('', cpoint, scope.value, function(status, result) {

});

}

});

}

1. 执行显示：



图6-3 执行结果显示

## 热点查询

1. 功能：

通过点击热点查询按钮，将地图上所标注的热点进行展示，展示信息包括位置、电话及类型，还包括基础的GIS功能，例如缓冲区分析及基本的路径查询功能，方便用户进行基础的查看分析，为用户对健身场所的选择起基础的指导作用。

1. 输入：

点击按钮，输入位置信息、缓冲区半径等

1. 输出：

检索结果及其精准定位及基本的路径规划

1. 关键代码：

function point()

{

var placeSearch = new AMap.PlaceSearch(); //构造地点查询类

var infoWindow=new AMap.AdvancedInfoWindow({});

map.on('hotspotover', function(result) {

placeSearch.getDetails(result.id, function(status, result) {

if (status === 'complete' && result.info === 'OK') {

placeSearch\_CallBack(result);

}

});

});

//回调函数

function placeSearch\_CallBack(data) { //infoWindow.open(map, result.lnglat);

var poiArr = data.poiList.pois;

var location = poiArr[0].location;

infoWindow.setContent(createContent(poiArr[0]));

infoWindow.open(map,location);

}

function createContent(poi) { //信息窗体内容

var s = [];

s.push('<div class="info-title">'+poi.name+'</div><div class="info-content">'+"地址：" + poi.address);

s.push("电话：" + poi.tel);

s.push("类型：" + poi.type);

s.push('<div>');

return s.join("<br>");

}

}

1. 执行显示：



图6-4 执行显示结果



图6-4 执行显示结果

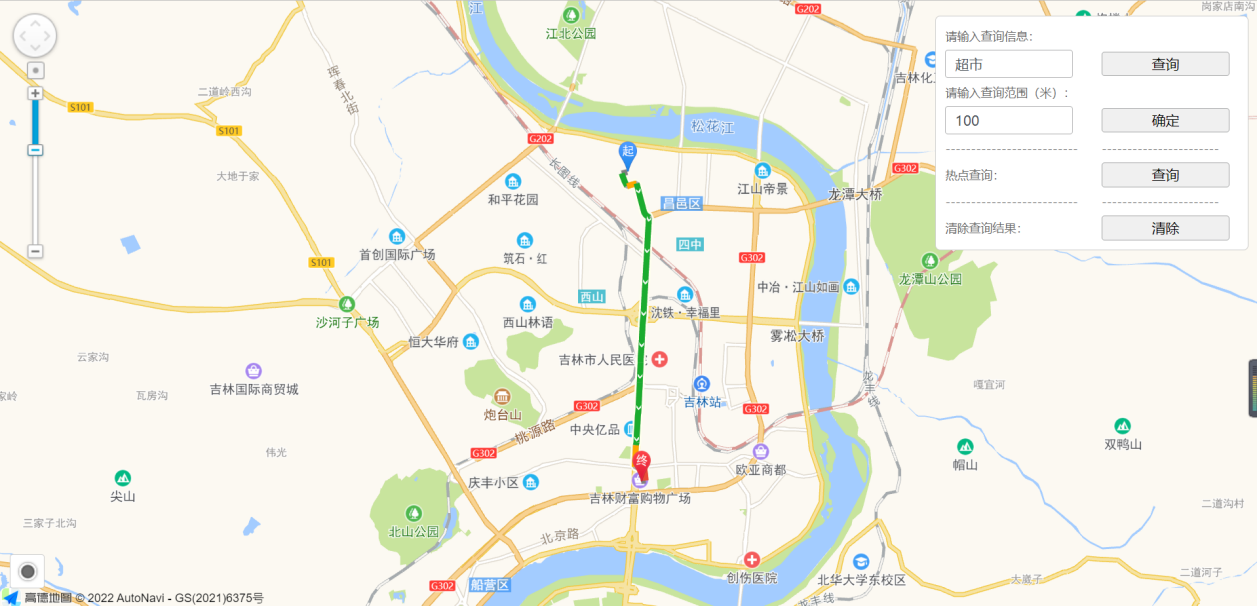


图6-4 执行显示结果

# 路径规划子系统

路径规划子系统，可根据用户的不同需求选择其对应交通出行方式，为用户提供合理的优化路径规划，并附有实时导航和预计花费时间等。规划主页面如图7-1、7-2所示。



图7-1 规划主页面

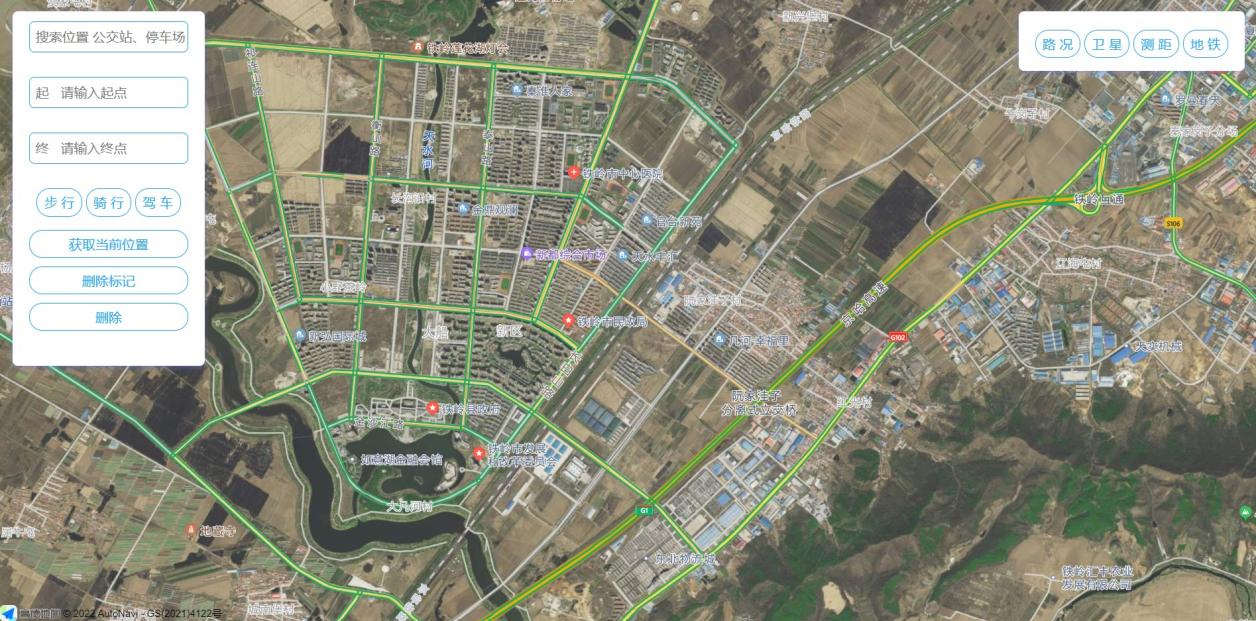


图7-2 规划主页面

## 驾车路径规划

1. 功能：

根据用户的驾车需求为用户提供合理的优化路径规划，并附有实时导航和预计花费时间等。通过关键字搜索确定起点与终点坐标，规划最优交通路径。并且提供实时路况展示、卫星图显示、定位用户当前位置等功能。

1. 输入：

起点、终点

1. 输出：

路径规划、时事路径情况及最优交通路径

1. 关键代码：

var edrive

function drive() {

AMap.plugin('AMap.Driving', function () {

var driving = new AMap.Driving({

map: map,

panel: "panel"

});

edrive = driving;

driving.search([

{ keyword: document.getElementById('start').value, city: '铁岭' },

{ keyword: document.getElementById('end').value, city: '铁岭' }

]);

});

}

function Dclear(edrive) {

if (edrive) {

edrive.clear();

}

}

1. 执行显示：

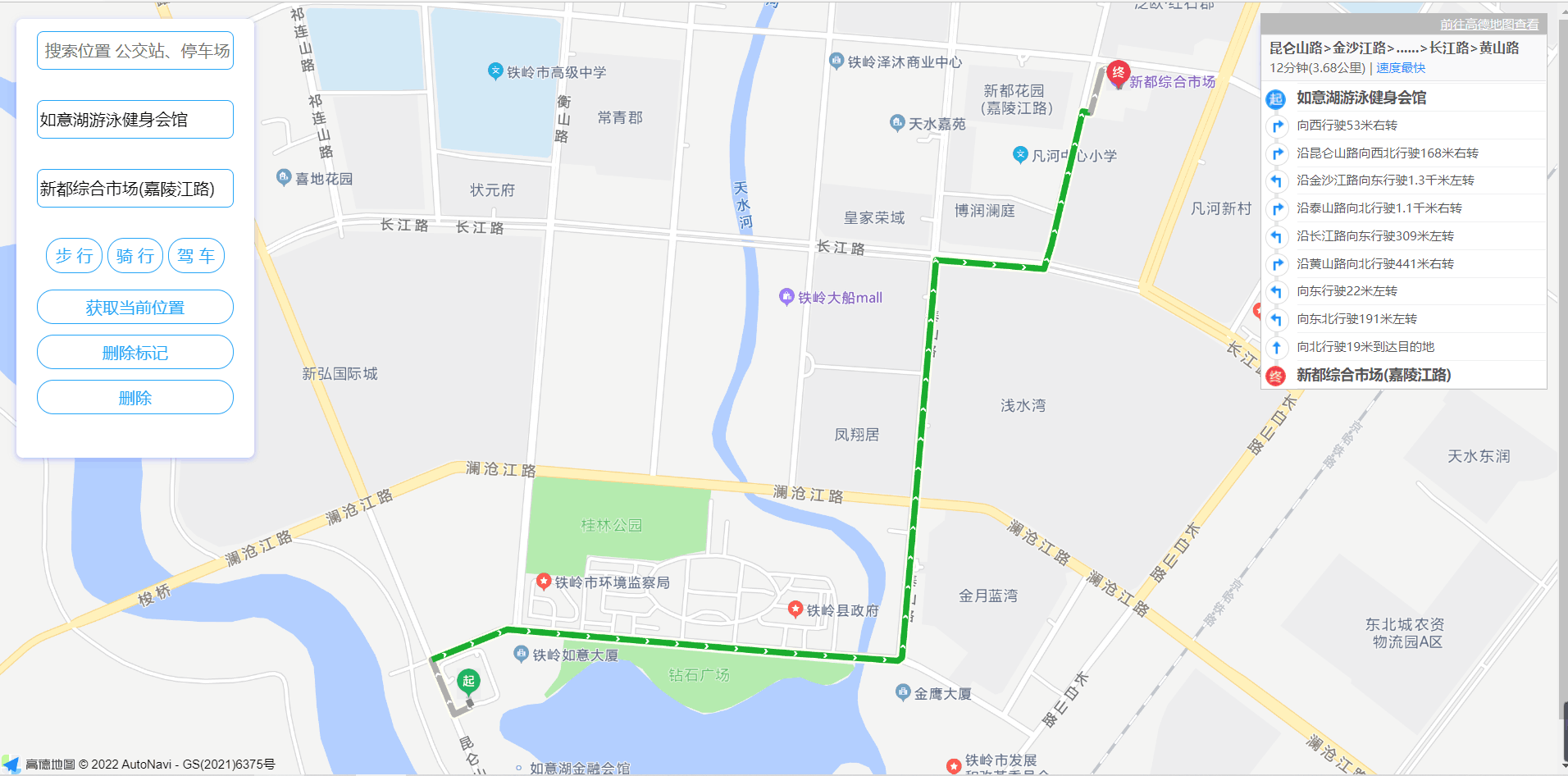


图7-3 执行显示结果

## 骑行路径规划

1. 功能：

根据用户的骑行需求为用户提供合理的优化路径规划，并附有实时导航和预计花费时间等。通过关键字搜索确定起点与终点坐标，规划最优交通路径。并且提供实时路况展示、卫星图显示、定位用户当前位置等功能。

1. 输入：

起点、终点

1. 输出：

路径规划、时事路径情况及最优交通路径

1. 关键代码：

var eride

function ride() {

AMap.plugin('AMap.Riding', function () {

var riding = new AMap.Riding({

map: map,

panel: "panel"

});

eride = riding;

riding.search([

{ keyword: document.getElementById('start').value, city: '铁岭' },

{ keyword: document.getElementById('end').value, city: '铁岭' }

]);

});

}

function Rclear(eride) {

if (eride) {

eride.clear();}}

1. 执行显示：

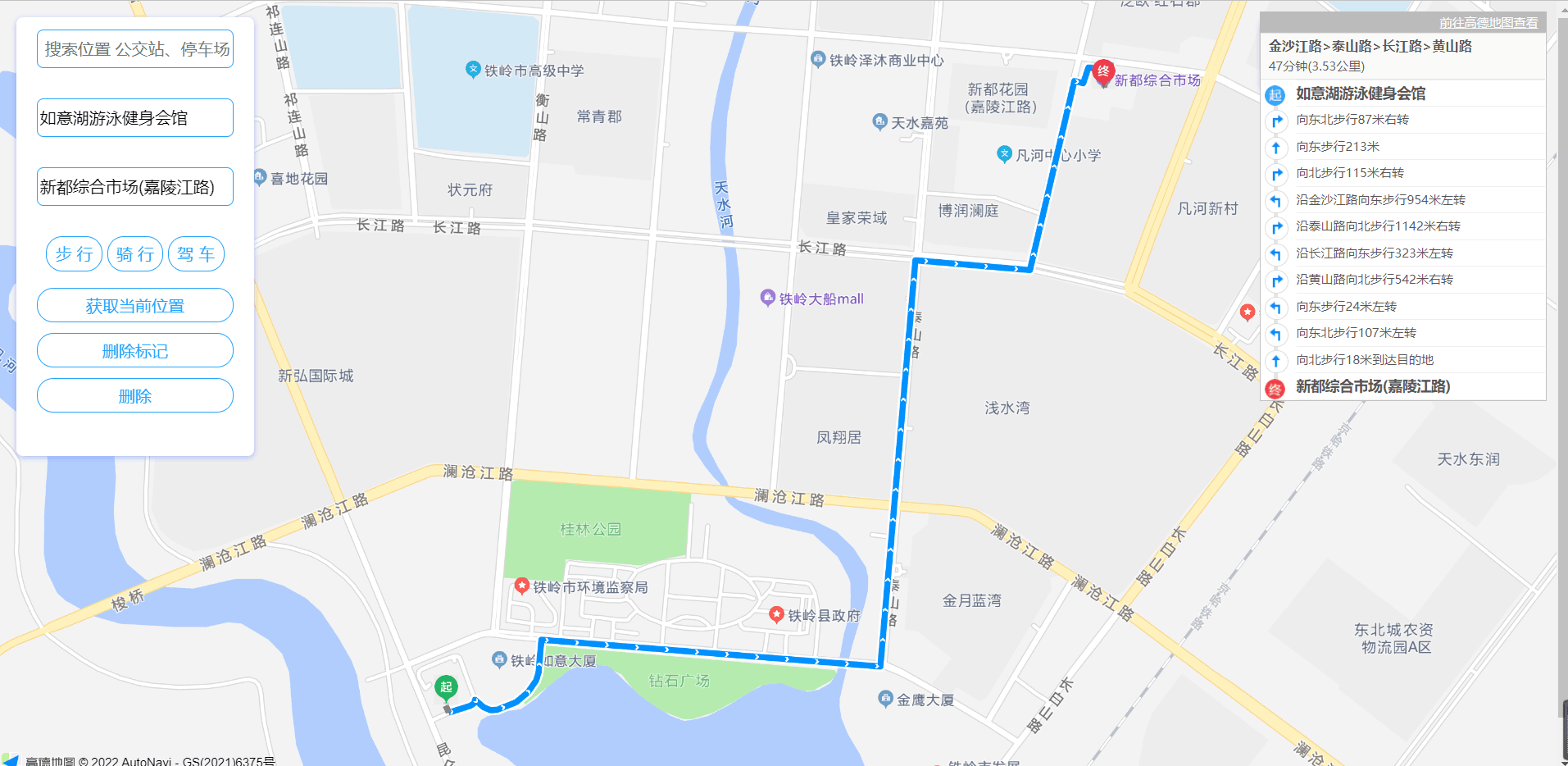


图7-4 执行显示结果

## 步行路径规划

1. 功能：

根据用户的步行需求为用户提供合理的优化路径规划，并附有实时导航和预计花费时间等。通过关键字搜索确定起点与终点坐标，规划最优交通路径。并且提供实时路况展示、卫星图显示、定位用户当前位置等功能。

1. 输入：

起点、终点

1. 输出：

路径规划、时事路径情况及最优交通路径

1. 关键代码：

var ewalk

function walk() {

AMap.plugin('AMap.Walking', function () {

var walking = new AMap.Walking({

map: map,

panel: "panel"

});

ewalk = walking;

walking.search([

{ keyword: document.getElementById('start').value, city: '铁岭' },

{ keyword: document.getElementById('end').value, city: '铁岭' }

]);

});

}

function Wclear(ewalk) {

if (ewalk) {

ewalk.clear();

}

}

1. 执行显示：

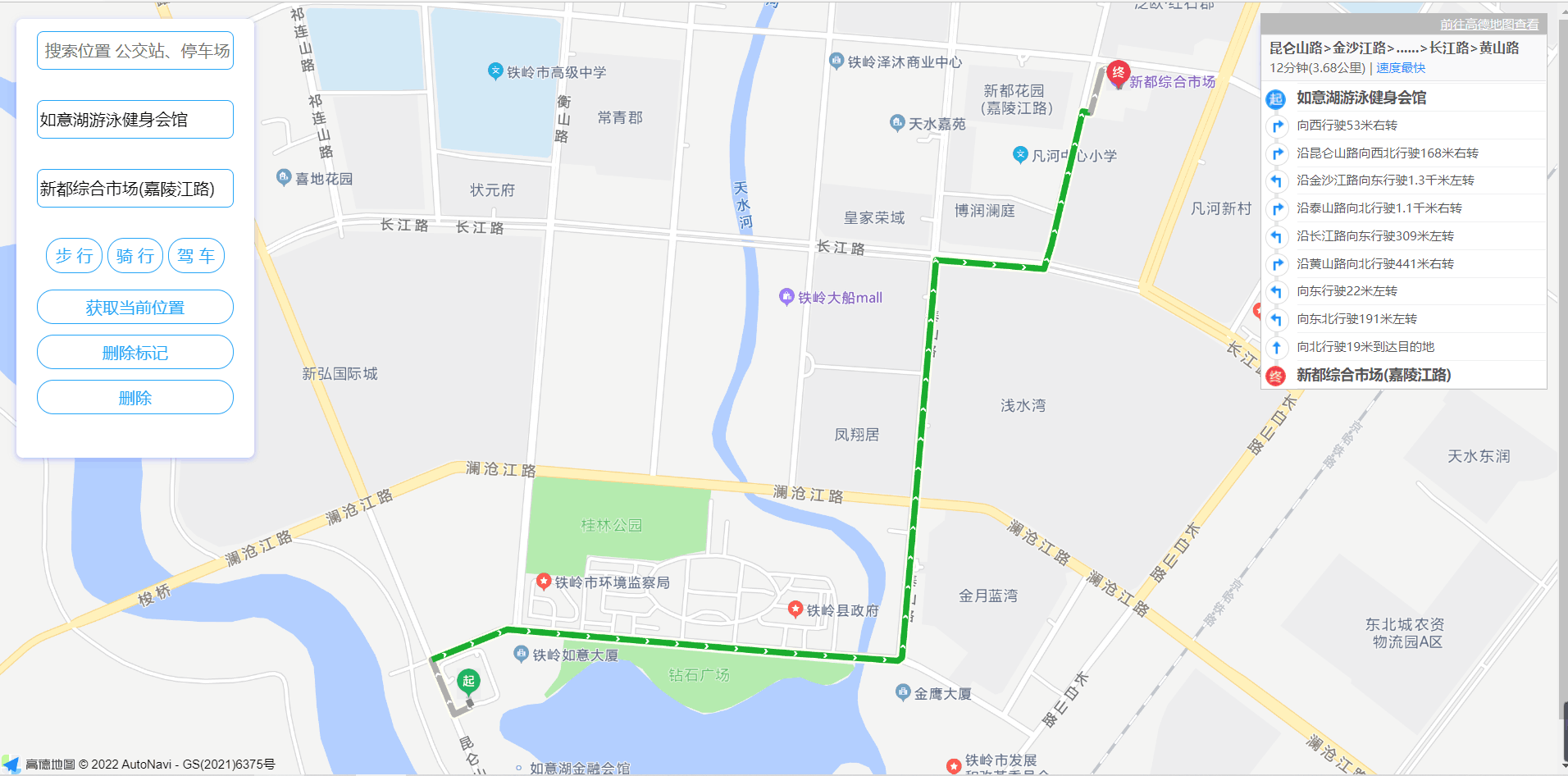


图7-5 执行结果显示

# 运动方案子系统

用户根据自己的需要选择运动方式，比如骑行或者散步来提供个性化的骑行或散步路径，自动定位当前位置为起点，用户自己选择终点，在图上标记出来。或者用户，没有从当前位置开始的意愿，可以自行输入起始点的位置获得路径。规划主页面如图8-1所示。



图8-1 规划主页面

## 骑行路线规划

1. 功能：

骑行路径规划提供个性化散步路径，自动定位当前位置为起点，用户自己选择终点，在图上标记出来。或者用户，没有从当前位置开始的意愿，可以自行输入起始点的位置获得散步路径。

1. 输入：

点击屏幕选择终点或输入起点、终点

1. 输出：

路径规划及最优交通路径

1. 关键代码：

var ride

function SearchR(){

map.plugin('AMap.Riding', function () {

var riding = new AMap.Riding({

map: map,

panel: "panel"

});

ride = riding

//根据起终点坐标规划骑行路线

riding.search([mya,myb],[wanta,wantb], function (status, result) {

// result即是对应的骑行路线数据信息

if (status === 'complete') {

log.success('绘制骑行路线完成')

} else {

log.error('骑行路线数据查询失败' + result)

}});})}

1. 执行显示：



图8-2 执行显示结果

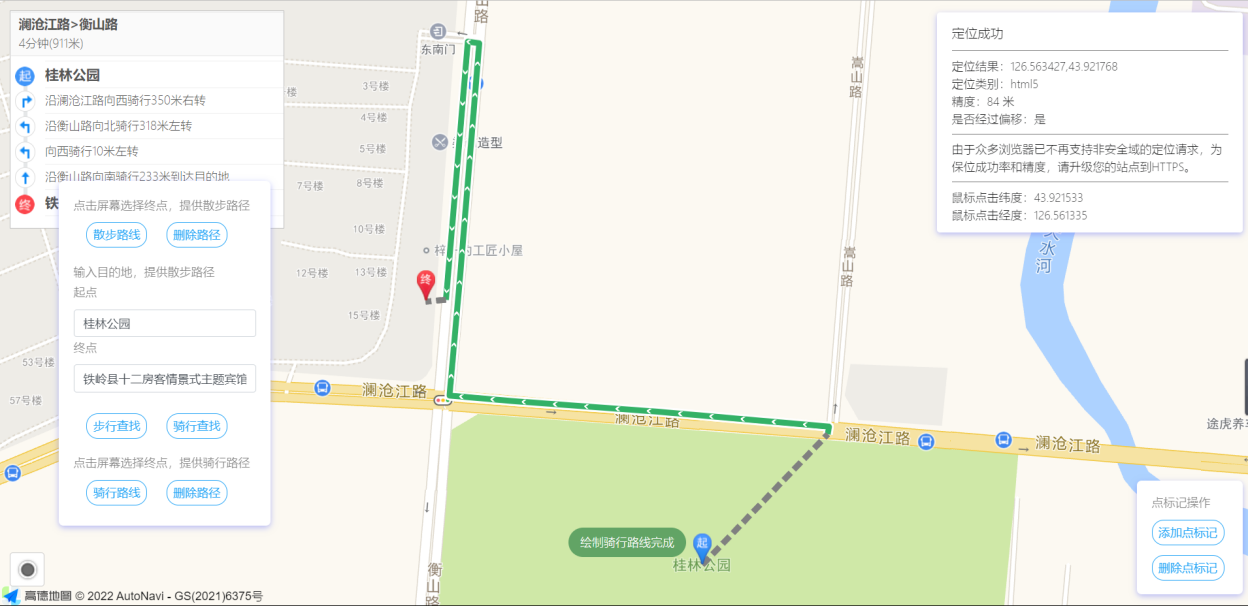


图8-3 执行显示结果

## 散步路线规划

1. 功能：

本子系统提供个性化散步路径，自动定位当前位置为起点，用户自己选择终点，在图上标记出来。或者用户，没有从当前位置开始的意愿，可以自行输入起始点的位置获得骑行路径。

1. 输入：

点击屏幕选择终点或输入起点、终点

1. 输出：

路径规划及最优交通路径

1. 关键代码：

var walk;

function SearchW(){

map.plugin('AMap.Walking', function () {

//步行导航

var walking = new AMap.Walking(walkOption)

console.log(wanta+wantb+""+mya+myb)

walk = walking

walking.search([mya,myb],[wanta,wantb], function(status, result) {

// result即是对应的不行路线数据信息，相关数据结构文档请参考 https://lbs.amap.com/api/javascript-api/reference/route-search#m\_RidingResult

if (status === 'complete') {

log.success('步行路线数据查询成功')

} else {

log.error('步行路线数据查询失败' + result)

}

});

});

// 步行导航

}

1. 执行显示：

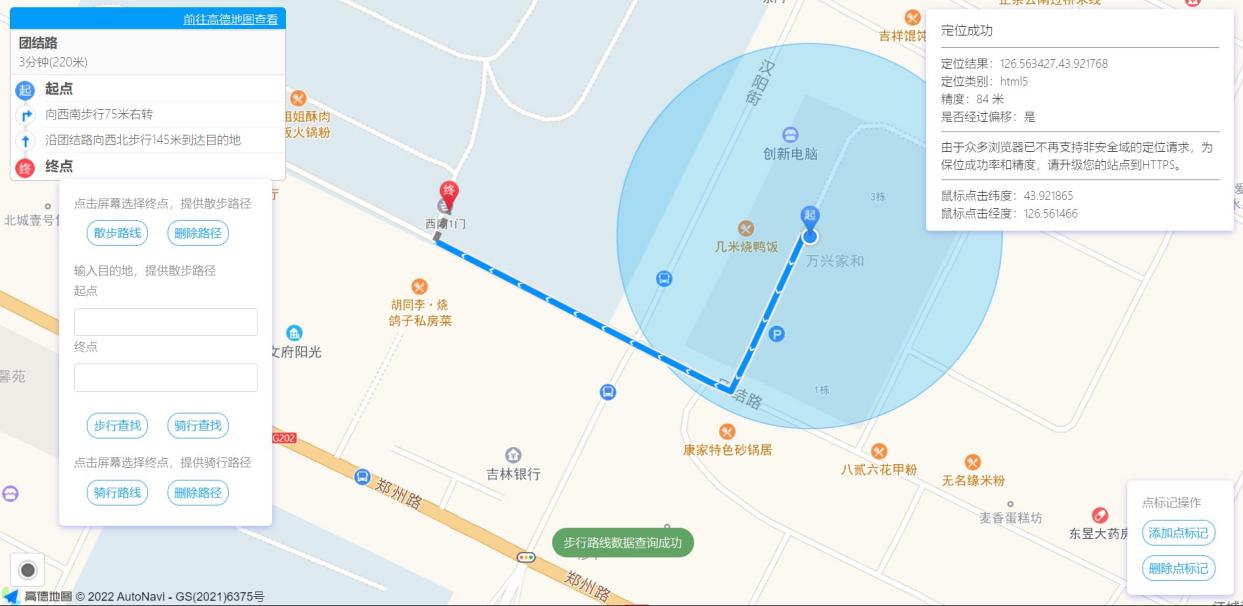


图8-4 执行显示结果

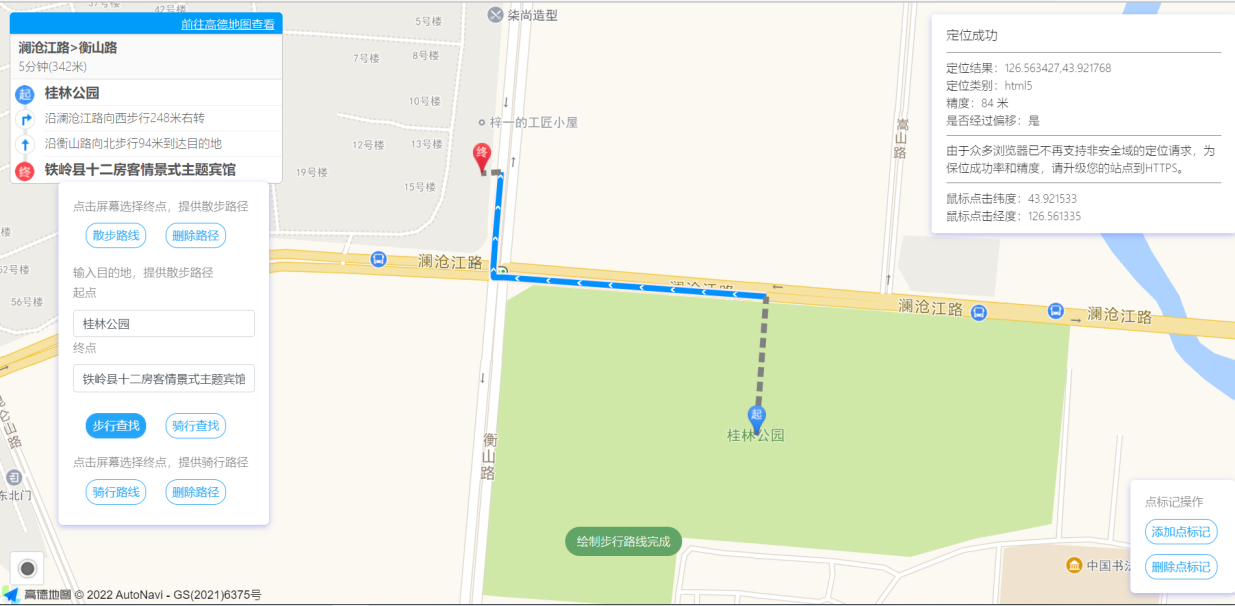


图8-5 执行显示结果

# 总结

本系统为基于WebGIS的全民智慧化健身系统，实现“全民智慧化健身”将公共服务信息、数据资源的一体化设计，结合WebGIS思想将健身场所位置信息可视化并实现相应功能，便利人们日常生活需要，迎合市场需求发展，为实现“健康中国”积极探索体育健康事业的创新发展之路，做出贡献。

本系统用户面向公共大众，系统中提供健身场所信息、信息查询检索、交通路径规划、个性化运动路线规划等功能，将具体专业功能以简单凝练高度概括的方式表达。