

****

**项目： 基于语音识别的植物分布查询系统**

**课程： 地理信息系统与网络技术**

**组员：邱洪发、王嘉奇、杨安邦、周青鑫**

**学院： 地球科学学院**

**目 录**

[一、项目概述 1](#_Toc29998916)

[二、需求分析 1](#_Toc29998917)

[1. 功能需求 1](#_Toc29998918)

[2. 系统界面需求 1](#_Toc29998919)

[3. 实用性需求 2](#_Toc29998920)

[三、总体设计 2](#_Toc29998921)

[1. 总体框架设计 2](#_Toc29998922)

[2. 界面设计 3](#_Toc29998923)

[3. 数据库设计 4](#_Toc29998924)

[四、关键技术 4](#_Toc29998925)

[1. 基于Python进行网络爬虫 4](#_Toc29998926)

[2. 对植物点坐标进行修正并与紫金港地块关联 5](#_Toc29998927)

[3. 基于高德API绘制地图相关功能 5](#_Toc29998928)

[4. 使用讯飞API进行语音语义识别 5](#_Toc29998929)

[5. 使用PhpStudy服务器联系客户端与MySQL数据库 5](#_Toc29998930)

[五、系统实现 5](#_Toc29998931)

[1. 主要代码 5](#_Toc29998932)

[2. 实现界面 17](#_Toc29998933)

[六、开发体会与分工 17](#_Toc29998934)

[1.开发体会 17](#_Toc29998935)

[2.人员分工： 20](#_Toc29998936)

## 

## 一、项目概述

生活中的每个细微之处都可能分布着各种植物。对于植物爱好者来说，若有一款应用可以将附近的植物根据其在现实世界中的位置对应地显示在地图上，那么用户只需根据地图上的描述就可以查看某一区域内的植物分布情况，极大地提升用户寻找植物的效率。

随着GIS技术的发展，许多LBS应用都嵌入了语音识别功能。语音识别包括读取用户语音、识别语音中蕴含的语义、根据语义进行查询几部分功能。语音识别可以大大简化用户的操作，提升用户使用体验。

本项目以浙江大学紫金港校区东区为例，为植物爱好者建立一个植物观赏点查询系统。同时加入智能语音识别系统，实现通过语音进行查询需求。此外，在实现查询的基础上，还加入了植物的科普功能，并实现时间尺度上的植物统计功能。

## 二、需求分析

### 1. 功能需求

（1）基本的地图显示与编辑：包括显示地图，更改地图底图，显示鹰眼图，显示比例尺等地图要素，显示实时路况，地图的缩放与平移等功能。

（2）植物分布点的可视化：根据现实世界的植物分布状况，在地图上对应的地方显示植物的分布情况。

（3）植物的科普：对于地图上每种植物，点击图标后在弹出的视图中显示植物的名称等主要信息，并提供跳转到对应百科的超链接，方便用户查看各种植物的信息。

（4）语音识别：用户可以通过语音告诉系统自己想查询的信息，例如“图书馆附近有什么植物？”，系统会将语音转换为文字，并识别文字信息进行后续的查询。

（5）其他辅助功能：通过刷新按钮改变每次显示的植物，在时间尺度上对植物数量进行统计等。

### 2. 系统界面需求

界面应足够简洁、美观。本系统主要面向园艺工作者、植物爱好者、大中小学生及学生家长，需要确保界面对用户具有一定的吸引力，保证用户一经上手就可以熟练操作。系统应提供一定的地图编辑工具，包括缩放条，底图转换工具等，方便用户根据自己的偏好调整地图的显示风格。

### 3. 实用性需求

目前市面上暂无针对植物的查询系统，本系统的查询功能可以帮助用户快速找到想要观赏的植物，对于植物爱好者来说十分实用。此外，语音识别功能方便了用户与系统的交互，简化了操作流程。系统集成的植物科普等相关功能让没有植物方面专业知识的用户在观赏中学习，极大地方便用户增进对身边各种植物的了解与认识。

## 三、总体设计

### 1. 总体框架设计

整个系统采用多层架构模式，主要由表现层、业务逻辑层、数据层三部分组成。

表现层使用PhpStorm作为开发环境，使用CSS、HTML、JavaScript、高德API、讯飞API进行开发。表现层组成客户端。

业务逻辑层使用WampServer充当服务器，表现层通过网络访问服务器。

数据层使用MySQL存储数据，服务器使用PHP访问数据库。业务逻辑层和数据层组成服务器端。

系统通信流程如下：当客户端进行操作时，客户端将用户的操作请求通过网络发送到服务器端，服务器检测到客户端的请求后，使用PHP向数据端发送查询请求。数据库将查询结果生成.json文件返回客户端，客户端进行解译后将结果进行显示。

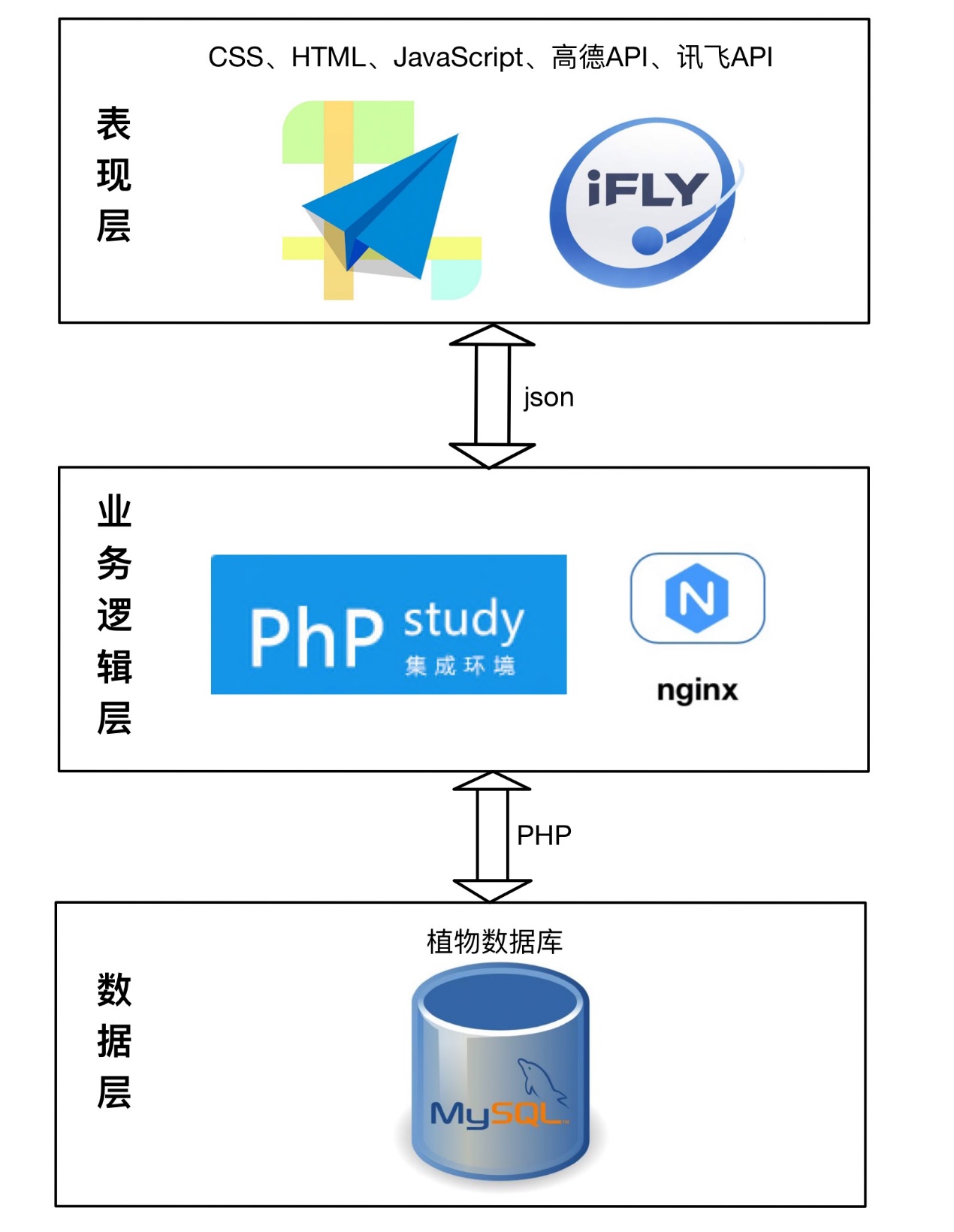


图 1 系统架构图

### 2. 界面设计

界面与一般的导航软件类似，但地图上多了植物点的位置。地图中上方为“开始识别”按钮，点击可输入语音，系统会将语音转换为文字。左上角由上自下分别为地图平移工具、定位按钮和地图缩放工具条，实习地图显示区域的变化。右上角为地图底图变换工具与路况显示工具，实习地图风格的变换。左下角为刷新按钮和地图比例尺，点击刷新按钮可改变地图上显示的数据点。右下角为鹰眼图，显示地图框中所显示的地图在整个区域中的位置。



图 2 用户界面

### 3. 数据库设计

本系统最主要的数据库为植物点数据表，植物点数据来源于花伴侣应用并经过投影修正，该表通过MySQL进行组织存储。植物数据表结构如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fields | Type | Primary Key | Foreign Key | Remark |
| id | int(5) | Yes | / | 植物点编号 |
| picid | Int(8) | / | / | 图片编号 |
| name | varchar(10) | / | / | 植物名 |
| creationTi | varchar(16) | / | / | 创建时间 |
| lat | decimal(10,8) | / | / | 纬度 |
| lng | decimal(10,7) | / | / | 经度 |
| pictureURL | Varchar(82) | / | / | 植物图片URL |
| zone | Varchar(7) | / | / | 所属地块名 |

表格 1 植物数据表

## 四、关键技术

### 1. 基于Python进行网络爬虫

网络爬虫技术可以自动地抓取万维网上的信息。花伴侣是一款自动识别并分享植物的移动应用，拥有用户上传的海量植物分布数据。项目使用Python编写爬虫程序，从花伴侣平台爬取紫金港东区范围内最近的两年记录的植物作为原始数据。

### 2. 对植物点坐标进行修正并与紫金港地块关联

系统的地图基础显示与编辑部分基于高德API实现，高德地图使用WGS84坐标系加密后的火星坐标系，因此需要使用高德坐标转换功能对非高德坐标系的植物点数据进行转换。此外，由于在客户端查询时会用到分区搜索查询，所以需要对紫金港东区范围进行划区，并通过ArcMap的空间连接等工具对植物点数据和对应的分区进行关联。

### 3. 基于高德API绘制地图相关功能

植物点数据需要展示在地图上，基于高德API实现地图的基本功能，包括地图与植物点的显示，更改地图底图，显示鹰眼图，显示比例尺等地图要素，显示实时路况，地图的缩放与平移，刷新显示的植物点等功能。

### 4. 使用讯飞API进行语音语义识别

科大讯飞API可以为开发者免费提供语音识别、语音合成、语音评测、声纹识别、人脸识别等服务。项目使用讯飞API对用户说的话进行语音识别并将语音信息转换为文字信息，再根据获得的文字信息提取关键词，将关键词转换为对应的SQL语句，从数据库获取查询结果。

### 5. 使用PhpStudy服务器联系客户端与MySQL数据库

phpStudy是一个PHP调试环境的程序集成包，该程序包集成最新的Apache+PHP+MySQL+phpMyAdmin+ZendOptimizer。项目使用PHP和数据库进行交互，实现对数据库的访问查询，将查询结果生成的json文件返回前端网页即可显示用户的查询结果。

## 五、系统实现

### 1. 主要代码

①Python爬虫代码pc.py：

**import** requests *#导入requests包***import** json  
**import** csv  
**import** datetime  
  
**def** getdata(n,jsonpath):  
 url = **'https://lv.nongbangzhu.cn/rest/classifylogs?lat=0.0&lng=0.0'** + \  
 **'&maxLat=30.311393798490386'** + \  
 **'&minLat=30.29586804332317'** + \  
 **'&maxLng=120.09256526827811'** + \  
 **'&minLng=120.08141532540324'** + \  
 **'&currentPage=0'** + \  
 **'&pageSize={}'**.format(n) + \  
 **'&maxId=0'** *# 请求表单数据* res = requests.get(url) *# Get方式获取网页数据  
 # 将Json格式字符串转字典* dict = json.loads(res.text)  
 str\_json = json.dumps(dict[**'payload'**][**'list'**])  
  
 **with** open(jsonpath, **'w'**, encoding=**'utf-8'**) **as** file:  
 file.write(str\_json)  
  
 file.close()  
  
**def** trans(jsonpath, csvpath):  
 json\_file = open(jsonpath, **'r'**, encoding=**'utf-8'**)  
 csv\_file = open(csvpath, **'w'**, newline=**''**, encoding=**'utf-8'**)  
 keys = []  
 writer = csv.writer(csv\_file)  
  
 json\_data = json\_file.read()  
 dic\_data = json.loads(json\_data)  
  
 keys = [**'id'**, **'picId'**, **'plant'**, **'creationTime'**, **'lat'**, **'lng'**, **'score'**, **'pictureURL'**]  
 *# 写入列名* writer.writerow(keys)  
  
 i=1  
 **for** dic **in** dic\_data:  
 **for** key **in** keys:  
 **if** key **not in** dic:  
 dic[key] = **''  
 if** key==**'creationTime'**:  
 *# 使用datetime* timeStamp = dic[**'creationTime'**]/1000  
 dateArray = datetime.datetime.fromtimestamp(timeStamp)  
 dic[**'creationTime'**] = dateArray.strftime(**"%Y-%m-%d %H:%M:%S"**)  
 writer.writerow([str(i),dic[**'id'**],dic[**'plant'**],dic[**'creationTime'**],dic[**'lat'**],dic[**'lng'**],dic[**'score'**],**'https://lv.nongbangzhu.cn/resources/image/'**+dic[**'picture'**]])  
 i+=1  
 **print**(**'一共爬取了%d个植物数据'** %(i-1))  
 json\_file.close()  
 csv\_file.close()  
  
**if** \_\_name\_\_==**'\_\_main\_\_'**:  
 getdata(10000,**'json.txt'**)  
 trans(**'json.txt'**, **'plant.csv'**)

②项目主界面代码index.html：

<!DOCTYPE **html**>  
<**html lang="en"**>  
  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**meta name="viewport" content="width=device-width, initial-scale=1.0"**>  
 <**meta http-equiv="X-UA-Compatible" content="ie=edge"**>  
 <**title**>植物语音导航</**title**>  
 <**link href="index.css" type="text/css" rel="stylesheet"**/>  
</**head**>  
  
<**body**>  
<**div id="panel"**></**div**>  
<**div id="container" tabindex="0" style="position**: **relative**;**"**>  
 <**div class="service-item service-item-taste"**>  
 <**div class="service-item-content service-item-taste-content"**>  
 <**div class="taste-content"**>  
 <**button class="taste-button ready-button" id="taste\_button"**>开始识别</**button**>  
 </**div**>  
 <**div class="start-taste"**>  
 <**div class="start-taste-left"**>  
 <**div class="time-box"**>  
 <**span class="start-taste-line"**>  
 <**hr class="hr hr1"**>  
 <**hr class="hr hr2"**>  
 <**hr class="hr hr3"**>  
 <**hr class="hr hr4"**>  
 <**hr class="hr hr5"**>  
 <**hr class="hr hr6"**>  
 <**hr class="hr hr7"**>  
 <**hr class="hr hr8"**>  
 <**hr class="hr hr9"**>  
 <**hr class="hr hr10"**>  
 </**span**>  
 <**span class="total-time"**><**span class="used-time"**>00: 00</**span**> / 01: 00</**span**></**span**>  
 </**div**>  
 <**div class="start-taste-button"**>  
 <**button class="taste-button start-button" onclick="***UpdateData*(*wordExtraction*());**"**>结束识别</**button**>  
 </**div**>  
 </**div**>  
 <**div class="end\_taste\_right"**>  
 <**div class="output-box" id="result\_output"**></**div**>  
 <**div class="end-button"**>  
 <**button class="query-button" id="confirm\_button" onclick="***UpdateData*(*wordExtraction*());**"**>确认</**button**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 </**div**>  
 <**div class="info2"** >点击地图上的点标记，查看植物信息</**div**>  
 <**button class="button1" id="refresh\_button" onclick="***refresh*()**"** ></**button**>  
</**div**>  
  
<**script src="hmac-sha256.js"**></**script**>  
<**script src="enc-base64-min.js"**></**script**>  
<**script src="jquery.js"**></**script**>  
<**script src="iat.js"**></**script**>  
<**script src="wordExtraction.js"**></**script**>  
<**script src="UpdateData.js"**></**script**>  
  
<**link rel="stylesheet" href="https://a.amap.com/jsapi\_demos/static/demo-center/css/demo-center.css"** />  
<**script src="https://webapi.amap.com/maps?v=1.4.2&key=608d75903d29ad471362f8c58c550daf&plugin=AMap.Walking"**></**script**>  
<**script src="https://a.amap.com/jsapi\_demos/static/demo-center/js/demoutils.js"**></**script**>  
<**script type="text/javascript" src="https://webapi.amap.com/demos/js/liteToolbar.js"**></**script**>  
<**script type="text/javascript" src="https://cache.amap.com/lbs/static/addToolbar.js"**></**script**>  
<**script src="https://webapi.amap.com/ui/1.0/main.js?v=1.0.11"**></**script**>  
<**script src="map.js"**></**script**>  
<**script src="navigation.js"**></**script**>  
  
</**body**>  
</**html**>

③地图绘制相关功能代码map.js：

*//创建地图***var *map*** = **new** AMap.Map(**'container'**, {  
 **resizeEnable**: **true**,  
 **center**:[120.087,30.3],  
 **zoom**:16,  
 */\*pitch:75, // 地图俯仰角度，有效范围 0 度- 83 度  
 viewMode:'3D' // 地图模式\*/*});  
  
  
  
**var *markerContent*** = **''** +  
 **'<div class="custom-content-marker">'** +  
 **' <img src="https://a.amap.com/jsapi\_demos/static/images/mass2.png">'** +  
 **'</div>'**;  
  
**var *infoWindow*** = **new** AMap.InfoWindow({**offset**: **new** AMap.Pixel(0, -30)});  
  
**var *position***;  
  
$(***document***).**ready**(**function**(){  
 *//默认获取当前月份的植物* **var** t=**new *Date***();  
 **var** month=t.getMonth()+1;  
 **fliter** = **"select \* from zjgplant where creationTi LIKE ('%/"** + month + **"/%')"**;  
 *UpdateData*(**fliter**);  
});  
  
**function** *markerClick*(e) {  
 ***infoWindow***.setContent(e.**target**.**content**);  
 ***infoWindow***.open(***map***, e.**target**.getPosition());  
 ***map***.setCenter(e.**target**.getPosition());  
}  
  
AMapUI.loadUI([**'control/BasicControl'**], **function**(BasicControl) {  
  
 *//图层切换控件* ***map***.addControl(**new** BasicControl.LayerSwitcher({  
 **position**: **'rt'** }));  
});  
  
AMap.plugin([  
 **'AMap.ToolBar'**,  
 **'AMap.Scale'**,  
 **'AMap.OverView'**,  
 **'AMap.MapType'**,  
], **function**(){  
 *// 在图面添加工具条控件，工具条控件集成了缩放、平移、定位等功能按钮在内的组合控件* ***map***.addControl(**new** AMap.ToolBar());  
  
 *// 在图面添加比例尺控件，展示地图在当前层级和纬度下的比例尺* ***map***.addControl(**new** AMap.Scale());  
  
 *// 在图面添加鹰眼控件，在地图右下角显示地图的缩略图* ***map***.addControl(**new** AMap.OverView({**isOpen**:**true**}));  
  
 *// 在图面添加类别切换控件，实现默认图层与卫星图、实施交通图层之间切换的控制* ***map***.addControl(**new** AMap.MapType());  
  
});

④语义识别代码wordExtraction.js：

**function** *wordExtraction*(){  
 *// var word = $('#taste-button start-button');  
 // alert(word);  
 //var word = document.getElementById("result\_output").innerText;* **var** word = **"白沙有什么植物"**;  
  
 **if**(word===**''**)  
 **return**;  
  
 **var** times = [**'1月'**, **'2月'**, **'3月'**, **'4月'**, **'5月'**, **'6月'**, **'7月'**, **'8月'**, **'9月'**, **'10月'**, **'11月'**, **'12月'**, **'春'**, **'夏'**, **'秋'**, **'冬'**];  
 **var** areas = [**'云峰'**,**'丹青'**,**'大食堂'**,**'翠柏'**,**'留学生公寓'**,**'白沙'**,**'校医院'**,**'游泳馆'**,**'体育馆'**,**'风雨操场'**,**'运动场'**,**'校友林'**,**'月牙楼'**,**'小剧场'**,**'西区理科实验楼'**,**'图书馆'**,**'蒙民伟楼'**,**'东区教学楼'**,**'西区教学楼'**,**'农生环团组'**,**'生科院'**,**'医药学院'**,**'启真湖'**];  
 **var** Out\_times = **new *Array***();  
 **var** Out\_areas = **new *Array***();  
  
 **var** OutSelect\_time=**''**;  
 **var** OutSelect\_area=**''**;  
  
 **var** i;  
 **for**(i = 0;i < times.**length**;i++)  
 {  
 **if**(word.indexOf(times[i])!=-1)  
 {  
 **if**(times[i] == **'春'**)  
 {  
 Out\_times[Out\_times.**length**] = **'3'**;  
 Out\_times[Out\_times.**length**] = **'4'**;  
 Out\_times[Out\_times.**length**] = **'5'**;  
 }  
 **else if**(times[i] == **'夏'**)  
 {  
 Out\_times[Out\_times.**length**] = **'6'**;  
 Out\_times[Out\_times.**length**] = **'7'**;  
 Out\_times[Out\_times.**length**] = **'8'**;  
 }  
 **else if**(times[i] == **'秋'**)  
 {  
 Out\_times[Out\_times.**length**] = **'9'**;  
 Out\_times[Out\_times.**length**] = **'10'**;  
 Out\_times[Out\_times.**length**] = **'11'**;  
 }  
 **else if**(times[i] == **'冬'**)  
 {  
 Out\_times[Out\_times.**length**] = **'12'**;  
 Out\_times[Out\_times.**length**] = **'1'**;  
 Out\_times[Out\_times.**length**] = **'2'**;  
 }  
 **else if**(times[i] == **'1月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'1'**;  
 }  
 **else if**(times[i] == **'2月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'2'**;  
 }  
 **else if**(times[i] == **'3月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'3'**;  
 }  
 **else if**(times[i] == **'4月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'4'**;  
 }  
 **else if**(times[i] == **'5月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'5'**;  
 }  
 **else if**(times[i] == **'6月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'6'**;  
 }  
 **else if**(times[i] == **'7月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'7'**;  
 }  
 **else if**(times[i] == **'8月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'8'**;  
 }  
 **else if**(times[i] == **'9月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'9'**;  
 }  
 **else if**(times[i] == **'10月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'10'**;  
 }  
 **else if**(times[i] == **'11月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'11'**;  
 }  
 **else if**(times[i] == **'12月'**)  
 {  
 Out\_times[Out\_times.**length**] = **'12'**;  
 }  
 }  
 }  
 **for**(i = 0; i < areas.**length**; i++)  
 {  
 **if**(word.indexOf(areas[i])!=-1)  
 {  
 Out\_areas[Out\_areas.**length**] = areas[i];  
 }  
 }  
  
 **for**(i = 0;i < Out\_areas.**length** ;i++)  
 {  
 OutSelect\_area += **"zone = '"** + Out\_areas[i] +**"' "**;  
 **if**(i < Out\_areas.**length** -1)  
 OutSelect\_area += **' OR'**;  
 }  
  
 **for**(i = 0;i < Out\_times.**length** ;i++)  
 {  
 OutSelect\_time += **" creationTi LIKE (\'%/"** + Out\_times[i] + **"/%')"** ;  
 **if**(i < Out\_times.**length** -1)  
 OutSelect\_time += **' OR'**;  
 }  
  
 **if**(OutSelect\_area===**''**&&OutSelect\_time===**''**){  
 *alert*(**"我不懂你在说啥。。。"**);  
 **return**;  
 }  
  
 **if**(OutSelect\_area===**''**||OutSelect\_time===**''**){  
 **fliter** =**"select \* from zjgplant where ("**+OutSelect\_area+OutSelect\_time+**")"**;  
 }  
 **else**{  
 **fliter** =**"select \* from zjgplant where ("**+OutSelect\_area+**") and ("**+OutSelect\_time+**")"**;  
 }  
  
 **return fliter**;  
  
}

⑤刷新植物点代码UpdateData.js：

**var *lat***,***lng***,***result***,***name***,***url***,***marker2***=[];  
**var *i1***,***i2***,***maxi***=20;  
  
**function** *UpdateData*(fliter){  
 ***i1***=1;  
 ***i2***=20;  
 $.**ajax**({  
 **url**:**'/plant/dbtojson.php'**,  
 **data**:{**q**:fliter},  
 success:**function**(data) {  
 ***console***.log(data);  
 }  
 });  
  
 $.getJSON(**"plant.json"**,**function**(data){  
 **var** jsonLength = 0;  
 **for**(**var** item **in** data){  
 jsonLength++;  
 }  
 ***maxi***=jsonLength;  
 });  
 *refresh*();  
}  
  
**function** *refresh*() {  
 ***map***.remove(***marker2***);  
  
 $.getJSON(**"plant.json"**,**function**(data){  
 $.each(data, **function**(i,row){  
 **if**(i<=***i2*** && i>=***i1*** ) {  
 $.each(row, **function** (j, field) {  
 **if** (j === **'lat'**) {  
 ***result*** = ***JSON***.stringify(field);  
 ***result***=***result***.replace(/"/,**""**);  
 ***result***=***result***.replace(/"/,**""**);  
 ***lat*** = ***result***;  
 }  
 **if** (j === **'lng'**) {  
 ***result*** = ***JSON***.stringify(field);  
 ***result***=***result***.replace(/"/,**""**);  
 ***result***=***result***.replace(/"/,**""**);  
 ***lng*** = ***result***;  
 }  
 **if** (j === **'name'**) {  
 ***result*** = ***JSON***.stringify(field);  
 ***result***=***result***.replace(/"/,**""**);  
 ***result***=***result***.replace(/"/,**""**);  
 ***name*** = ***result***;  
 }  
 **if** (j === **'pictureURL'**) {  
 ***result*** = ***JSON***.stringify(field);  
 ***result***=***result***.replace(/"/,**""**);  
 ***result***=***result***.replace(/"/,**""**);  
 ***url***=***result***;  
 }  
 });  
 *//坐标系转换* **gps**=[*parseFloat*(***lng***), *parseFloat*(***lat***)];  
 AMap.convertFrom(**gps**,**'gps'**, **function** (status, result) {  
 **if** (result.**info** === **'ok'**) {  
 **gps** = result.locations[0]; *// Array.<LngLat>* }  
 });  
 **var** marker = **new** AMap.Marker({  
  
 **position**: **gps**,  
 *// 将 html 传给 content* **content**: ***markerContent***,  
 *// 以 icon 的 [center bottom] 为原点* **offset**: **new** AMap.Pixel(-13, -30)  
 });  
 *//鼠标点击marker弹出自定义的信息窗体* **var** url1=**"https://baike.baidu.com/item/"**+***name***;  
 *encodeURIComponent*(url1);  
 marker.**content**=***name***+  
 **"<div ><img width='200px' src='"**+***url***+**"' ></div>"**+  
 **"<a href='https://baike.baidu.com/item/"**+***name***+**"'>详细信息</a>"**;  
 *//鼠标点击marker弹出自定义的信息窗体* marker.on(**'click'**, *markerClick*);  
 marker.emit(**'click'**, {**target**: marker});  
 *// 将 markers 添加到地图* ***map***.add(marker);  
 ***marker2***.push(marker);  
 }  
 });  
 });  
 ***console***.log(**"共查询到"**+***maxi***+**"条记录"**);  
 ***console***.log(**"现在是第"**+***i1***+**"到第"**+***i2***+**"条记录"**)  
  
 ***i1*** += 20;  
 ***i2***=(***i1***+19);  
 **if**(***i1***>***maxi***){  
 ***i1***=1;  
 ***i2***=***i1***+19;  
 }  
}

⑥数据库查询结果转json文件代码dbtojson.php：

**<?*php***$q **=** $\_GET[**'q'**];  
$mysql\_server\_name **= "localhost"**;  
$mysql\_username**="root"**;  
$mysql\_password**="123456"**;  
$mysql\_database**="plant"**;  
$json **= ''**;  
$data **=** array();  
  
*//判断是否为空。\_POST是post请求方式，\_GET是get请求方式。客户端请求***if**(isset($\_POST[**'name'**])){  
 $name **=** $\_POST[**'name'**];  
}  
*//连接数据库*$conn **=** mysqli\_connect($mysql\_server\_name,$mysql\_username,$mysql\_password,$mysql\_database);  
*// 检测连接***if** ($conn**->**connect\_error) {  
 **die**(**"连接失败: " .** $conn**->**connect\_error);  
}  
echo **"连接成功<br />"**;  
*//查询表*echo $q;  
$sql **=** $q;  
*//遍历表*$result **=** $conn**->**query($sql);  
  
**if**($result){  
 echo **"查询成功<br />"**;  
 **while** ($row **=** mysqli\_fetch\_array($result,MYSQLI\_ASSOC))  
 {  
 array\_push($data,$row);  
 }  
 $json **=** json\_encode($data,***JSON\_PRETTY\_PRINT***);*//把数据转换为JSON数据.* $file **=** fopen(**"../plant/plant.json"**,**"w"**);  
 echo fwrite($file,$json);  
 fclose($file);  
 echo **"done!"**;  
}**else**{  
 echo **"查询失败"**;  
}  
mysqli\_close($conn);  
**?>**

### 2. 实现界面

PhpStorm开发环境界面：

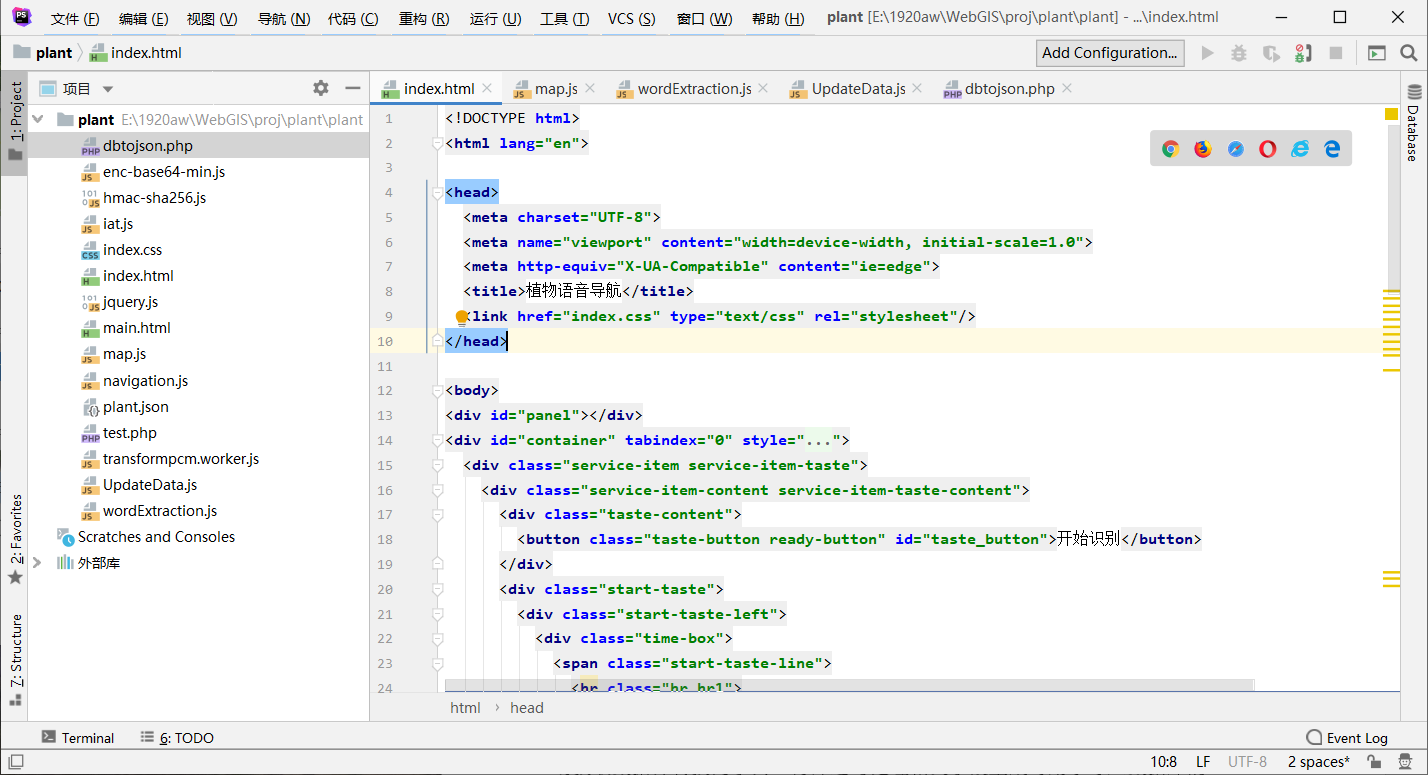


图 3 PhpStorm开发界面

## 六、开发体会与分工

### 1.开发体会

邱洪发：这次的开发算是GIS方面的一次综合大练兵，植物是我们组员感兴趣的内容，受制于数据来源，我打算学习爬虫技术爬取自己感兴趣的数据。形色APP的改版删除了遇见功能，于是我又重新找到了一款“花伴侣”软件，它带有地图以及数据的展示，正好可以爬取相关地理数据。于是使用fidder抓包软件抓取APP通信数据，分析数据包发现软件通过传递JSON数据显示植物，分析URL发现JSON数据的访问与获取很简单，于是我写了一个python脚本自动获取数据并转换为字典，再筛选与处理相关字段生成可用于导入数据库的CSV文件。  
 利用phpstudy集成软件部署Nginx1.15本地服务器与MySQL5.7.26数据库，将组员鑫鑫处理完毕的数据导入数据库。接下来如何进行php与数据库之间的通信令我困惑很久，PHP不同版本之间的语法差异十分巨大，通过反复的调试，最终成功实现了在HTML中使用ajax异步请求发送查询语句，并调用php5.6.9与服务器通信服务，更新本地的.json文件的功能。之所以没有采用传回json数据的方式是为了配合组员邦邦已经写好的JSON数据展示功能。  
此外，讯飞API是我的想法，简单学习web-api介绍后，参考他给的demo示例，为网页添加了语音流式识别的功能，在讯飞控制台添加热词以取得更好的识别效果。  
 为了项目的条理明确，我们将不同的功能封装为JS文件并在html中引用，同时我对整个网页的CSS样式进行了大量修改，使其尽量美观合理。但由于种种原因，界面还是非常简陋，原本的设想是参考花伴侣，每个数据点显示一张小图，点击获取更多信息。  
 因为开发时间极短，需要学习的新知识又多，安装调试软件还要占用不少时间，我们的应用不得不做了大量的简化。初始的想法包括加入用户注册登录、数据增删、图表展示、景观大道规划、3D导航视角、本地TTS语音导航、圈画感兴趣区域并实时调用爬虫接口等功能，但最终只能实现一个很粗糙的应用，令我有些遗憾。

周青鑫：这次的植物查询项目是本学期所有课程项目中难度最大的一个。除了以前接触过的html、JavaScript以及ArcMap相关操作之外，本次项目使用的数据库、服务器、相关接口以及部分其他语言都是以前没有接触过的，为开发带来不少难度。例如WampServer及PhpStorm开发环境的搭建与使用，各种数据坐标与高德在线地图坐标不一致带来的错位问题，数据入库及客户端与数据库的交互问题等都是不小的挑战。学习能力在人生的各个阶段都扮演着极其重要的角色，平时在课程中学习到的知识固然重要，但学习过程中养成的思维方式和学习能力对我们会有更大的帮助。虽然开发过程中困难不少，但在组长邱洪发的带领下，小组成员通力合作，最终顺利地完成了项目。项目的开发过程也是对课程的一次回顾，此外，我在解决问题的过程中也学到了不少新的东西。由于此次开发的大部分过程都是在考试周中进行的，时间紧迫，任务艰巨，有很多不够完善的地方，也有一些原先的设想未能实现，例如植物数据的统计分析，景观大道的规划，使用K-Means算法对植物分布情况进行聚类分析等。

杨安邦：本次开发对于我来说是将这学期所学的WebGIS知识综合运用的一次挑战，显而易见，将各部分内容组合到一起做出一款应用的难度是较高的，不光有对旧知识的回顾，也有新知识的学习使用。我们在实践开发过程中，遇到了许许多多的困难，比如网页前端展示和后端数据库的联系，这一问题迫使我们彻底更改开发环境，学习新的语言，用以实现动态网页。而每位成员在自己的分工内困难也是层出不穷，对不曾知晓的知识进行查阅学习，而往往细小的问题又难以获得帮助，需要自己反复尝试解决。但在这番学习与试验中，我的收获也颇为丰富，进一步熟悉了HTML的结构，掌握了JS这门语言，也了解了各种数据的格式和使用。回顾整个开发过程，虽然困难重重，但我们也都一一解决，在巩固课堂知识的基础上，拓展实践学习，最终做出了一个令自己较为满意的应用，对此我感到受益匪浅。

王嘉奇：本次开发我负责的主要是语义识别部分，先是通过讯飞API的语音识别功能将语音转换为文字，对转换出来的文字进行提取，提取出其中的关键字，将关键字插入到SQL查询语句的查询条件中。由于需要随时提取语音翻译过来的文本之中的关键文字，还需要对讯飞API的语音转换部分的结构有一定了解。而且本次项目使用了JavaScript、PHP、AJXA等开发语言，用到了新的数据库MySQL以及集成开发环境wampserver，拓宽了我的知识面，但是同时也发现不足之处，自己对于数据库部分的熟练度有待提高，对于JavaScript的理解也有待提升。

### 2.人员分工：

邱洪发：组织小组成员分工；使用Python爬取植物数据；使用讯飞API对语音进行识别并转换为文字；利用phpstudy集成软件部署Nginx1.15本地服务器与MySQL5.7.26数据库，实现前端与数据库的交互；参与展示

周青鑫：对紫金港东区进行区域划分；使用高德坐标转换Web API修正植物点数据坐标；撰写项目报告文档；制作PPT；参与展示

杨安邦：使用高德API显示地图及相关基本要素；设计地图相关UI对JSON数据进行展示

王嘉奇：使用讯飞API对转化的文本进行语义识别并转化为对应的SQL语句；参与展示