武汉纺织大学

Web应用开发课程设计

**基于eCharts的百度地图学生籍贯统计**

**学 院： 数学与计算机学院**

**班 级： 物联网11803**

**姓 名： 吴承霖**

**学 号： 1804280305**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月15日**

目 录

[1. 需求分析 1](#_Toc59298194)

[1.1. 显示百度地图 1](#_Toc59298195)

[1.2. 显示学生 1](#_Toc59298196)

[2. 系统设计 2](#_Toc59298197)

[2.1. 用例图 2](#_Toc59298198)

[2.2. ER图 2](#_Toc59298199)

[2.3. UML类图（Class Diagram） 3](#_Toc59298200)

[2.4. UML活动图（Activity Diagram） 3](#_Toc59298201)

[3. 系统实现 4](#_Toc59298202)

[3.1. 项目结构 4](#_Toc59298203)

[3.2. 配置文件 5](#_Toc59298204)

[3.2.1. application.properties文件 5](#_Toc59298205)

[3.2.2. SydController.java文件 6](#_Toc59298206)

[4.1. common的R.java 7](#_Toc59298207)

[4.2. Mapper层Mapper 8](#_Toc59298208)

[4.3. 接口映射文件SydMapper.xml 9](#_Toc59298209)

[4.4. Service层接口SydService.java 9](#_Toc59298210)

[4.5. entity层Syd.java 10](#_Toc59298211)

[5. 系统测试 10](#_Toc59298212)

[6. 系统总结 12](#_Toc59298213)

# 需求分析

设计一个在百度地图上统计学生籍贯所在地。实现在百度地图上的显示，学生的jiguan 显示在地图等功能，具体要求如下：

## 显示百度地图

打开网页显示百度

## 显示学生

每个来自不同地区的学生会显示在百度地图上面如图1-1，每一个在百度地图上的点都代表一个学生，每个点会显示学生的信息如图1-2。

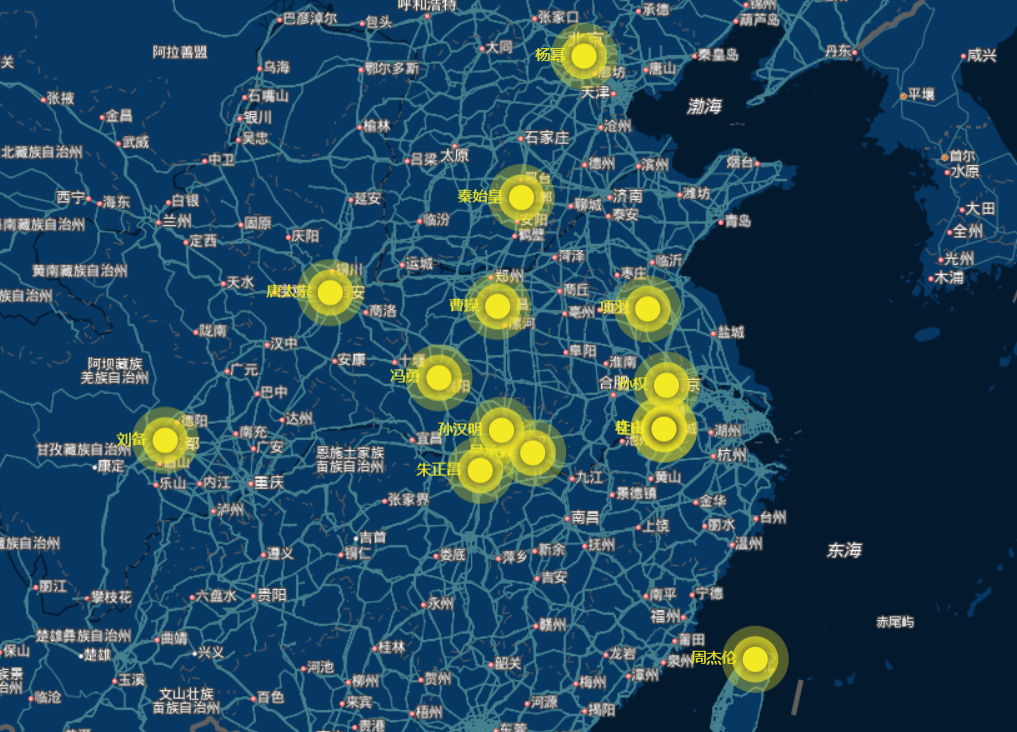


图1-1：需求界面



图1-2：学生信息

# 系统设计

## 用例图

例如下图2-1所示：

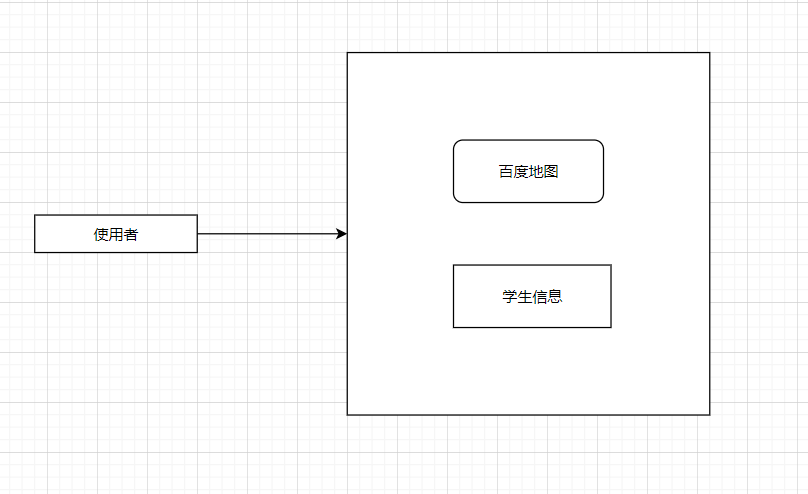
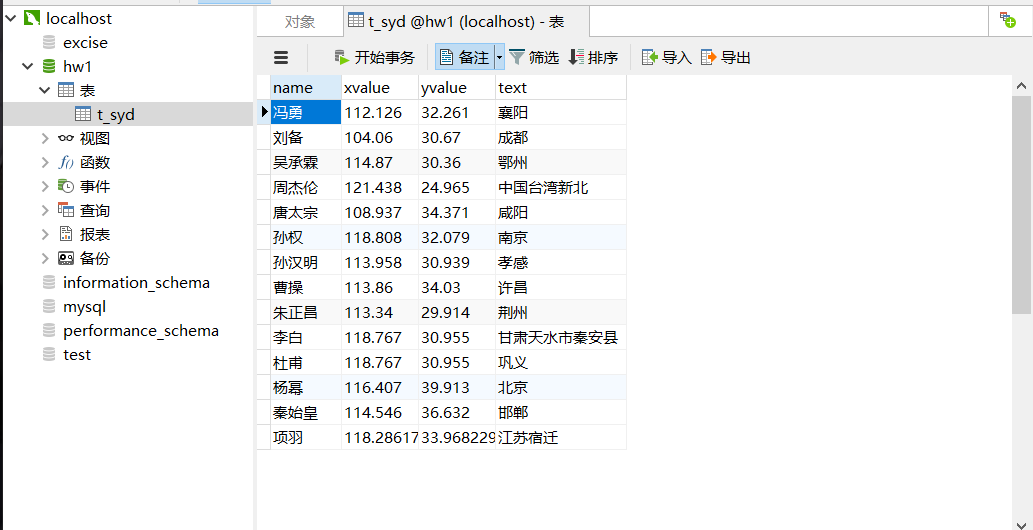


图2-1 用例图

## ER图

数据库设计1张表与ER图如图2-2所示：



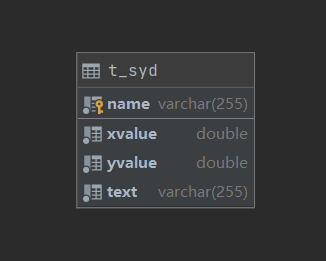


图2-2 数据库设计表和数据库ER图

## UML类图（Class Diagram）

对于用户功能模块，共设计如下6个类。

* 控制层类SydController类：前端控制器。
* Commonceng的R类：返回的结果
* mapper层接口SydMapper：主要定义数据操作的接口
* config层的MybatisPulsConfig类：配置Mybatis
* service层SydService接口： 业务逻辑层的接口
* 应用开关DituApplication类：启动spring boot开发的启动类。
* 各类的结构及类之间的关系如图2-3所示：

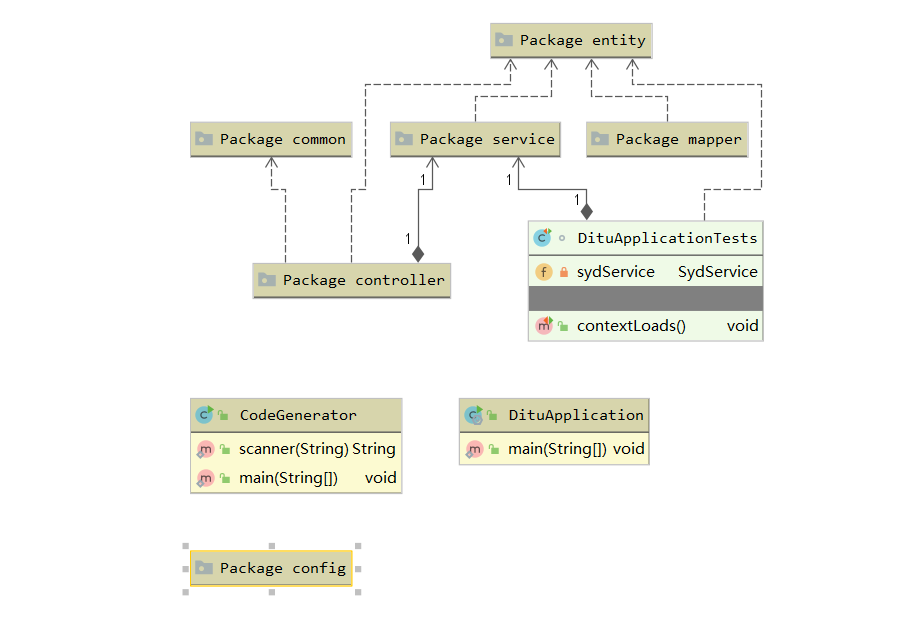
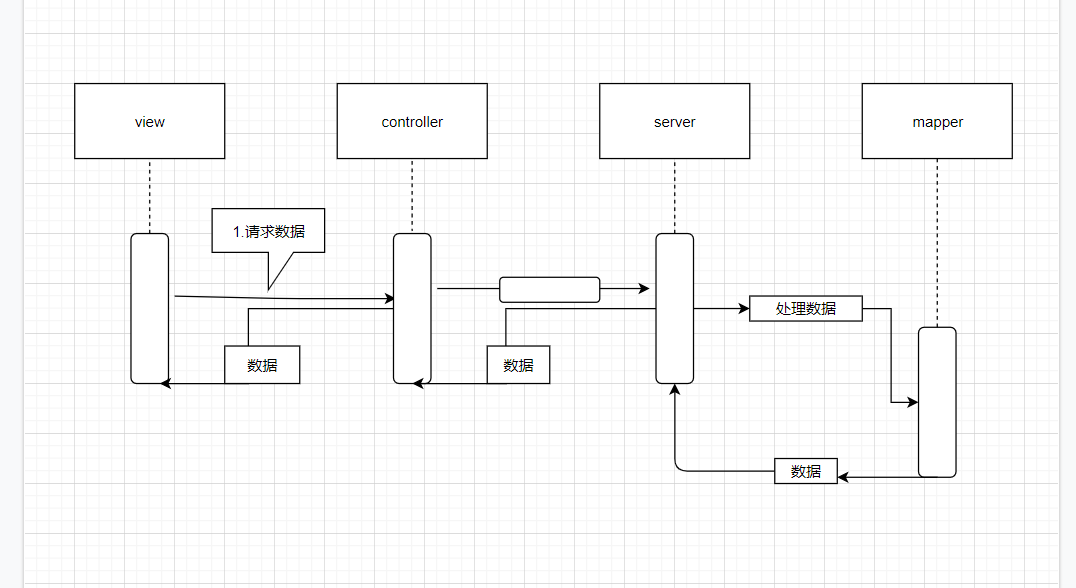


图2-3：UML类图

## UML时序图（Sequence Diagram）



## UML活动图（Activity Diagram）

用户会打开网页会看到百度地图，学生数据会在百度地图上形成一个散点图，具体活动图2-4所示。

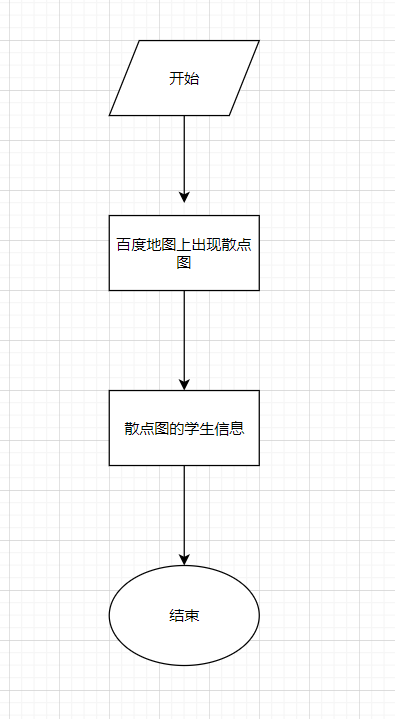


图2-4 显示流程图

# 系统实现

## 项目结构

项目结构如下图3-1所示：

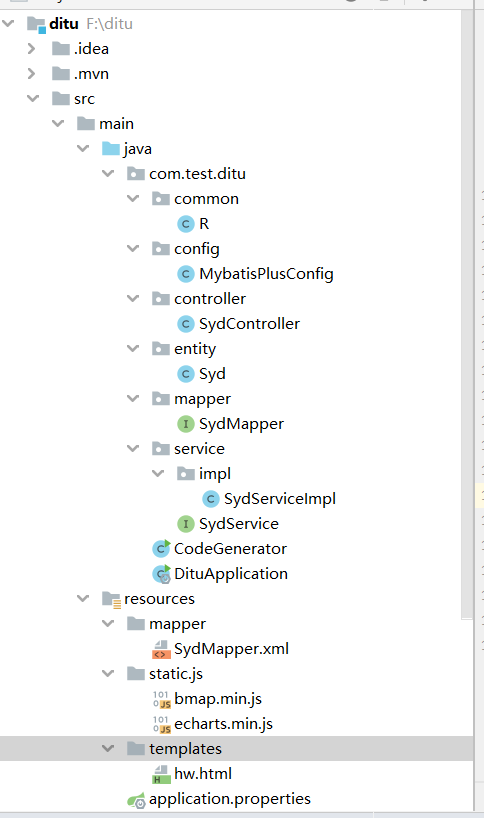


图3-1：系统结构

## 配置文件

### application.properties文件

该配置文件主要是配置数据库连接用户名和密码、mybatis配置检索路径和文件

*# 数据源*spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.url= jdbc:mysql://localhost:3306/hw1?useUnicode=true&useSSL=false&characterEncoding=utf8&serverTimezone=Asia/Shanghai  
spring.datasource.username=root  
spring.datasource.password=  
  
mybatis-plus.configuration.log-impl=org.apache.ibatis.logging.stdout.StdOutImpl  
mybatis-plus.configuration.map-underscore-to-camel-case=true  
mybatis-plus.type-aliases-package=com.test.ditu.entity  
mybatis-plus.mapper-locations=classpath:mapper/\*.xml

### SydController.java文件

前端控制器

1. **package** com.test.ditu.controller;  
     
     
   **import** com.test.ditu.common.R;  
   **import** com.test.ditu.entity.Syd;  
   **import** com.test.ditu.service.SydService;  
   **import** org.springframework.beans.factory.annotation.Autowired;  
   **import** org.springframework.stereotype.Controller;  
   **import** org.springframework.web.bind.annotation.GetMapping;  
   **import** org.springframework.web.bind.annotation.RequestMapping;  
     
   **import** org.springframework.web.bind.annotation.ResponseBody;  
   **import** org.springframework.web.bind.annotation.RestController;  
     
   **import** java.util.List;  
     
   */\*\*  
    \* <p>  
    \* 前端控制器  
    \* </p>  
    \*  
    \** ***@author*** *关注公众号：MarkerHub  
    \** ***@since*** *2020-12-17  
    \*/*@Controller  
   @RequestMapping(**"/syd"**)  
   **public class** SydController {  
     
    @Autowired  
    **private** SydService **sydService**;  
     
     
    @GetMapping(**"/all"**)  
    @ResponseBody  
    **public** R all(){  
    List<Syd> syds = **sydService**.getBaseMapper().selectList(**null**);  
    **return** R.*fail*(200,**"操作成功"**,syds);  
    }  
     
    @GetMapping(**"/"**)  
    **public** String to(){  
    **return "hw"**;  
    }  
   }

## common层的R.java

**package** com.test.ditu.common;  
  
**import** lombok.AllArgsConstructor;  
**import** lombok.Data;  
**import** lombok.NoArgsConstructor;  
**import** lombok.experimental.Accessors;  
  
**import** java.io.Serializable;  
  
*/\*\*  
 \* 这里我们用到了一个Result的类，这个用于我们的异步统一返回的结果封装。  
 \* 一般来说，结果里面有几个要素必要的  
 \* <p>  
 \* 是否成功，可用code表示（如200表示成功，400表示异常）  
 \* 结果消息  
 \* 结果数据  
 \*/*@Data  
@Accessors(chain = **true**)  
@AllArgsConstructor  
@NoArgsConstructor  
**public class** R **implements** Serializable {  
 **private int code**; *//200是正常 非200表示异常* **private** String **msg**;  
 **private** Object **data**;  
  
 **public static** R succ(**int** code,String msg, Object data){  
 R r = **new** R()  
 .setCode(code)  
 .setData(data)  
 .setMsg(msg);  
 **return** r;  
 }  
  
 **public static** R fail(**int** i, String message, Object data) {  
 R r = **new** R()  
 .setCode(i)  
 .setData(data)  
 .setMsg(message);  
 **return** r;  
 }  
}

## Mapper层Mapper 接口

**package** com.test.ditu.mapper;  
  
**import** com.test.ditu.entity.Syd;  
**import** com.baomidou.mybatisplus.core.mapper.BaseMapper;  
  
*/\*\*  
 \* <p>  
 \* Mapper 接口  
 \* </p>  
 \*  
 \** ***@author*** *关注公众号：MarkerHub  
 \** ***@since*** *2020-12-17  
 \*/***public interface** SydMapper **extends** BaseMapper<Syd> {  
  
}

## 接口映射文件SydMapper.xml

*<?***xml version="1.0" encoding="UTF-8"***?>***<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd"*>***<**mapper namespace="com.test.ditu.mapper.SydMapper"**>  
  
</**mapper**>

## Service层接口SydService

1. **package** com.test.ditu.service;  
     
   **import** com.test.ditu.entity.Syd;  
   **import** com.baomidou.mybatisplus.extension.service.IService;  
     
   */\*\*  
    \* <p>  
    \* 服务类  
    \* </p>  
    \*  
    \** ***@author*** *关注公众号：MarkerHub  
    \** ***@since*** *2020-12-17  
    \*/***public interface** SydService **extends** IService<Syd> {  
     
   }

## 4.5.entity层Syd.java

**package** com.test.ditu.entity;  
  
**import** com.baomidou.mybatisplus.annotation.TableName;  
**import** java.io.Serializable;  
**import** lombok.Data;  
**import** lombok.EqualsAndHashCode;  
**import** lombok.experimental.Accessors;  
  
*/\*\*  
 \* <p>  
 \*   
 \* </p>  
 \*  
 \** ***@author*** *关注公众号：MarkerHub  
 \** ***@since*** *2020-12-17  
 \*/*@Data  
@EqualsAndHashCode(callSuper = **false**)  
@Accessors(chain = **true**)  
@TableName(**"t\_syd"**)  
**public class** Syd **implements** Serializable {  
  
 **private static final long *serialVersionUID*** = 1L;  
  
 **private** String **name**;  
  
 **private** Double **xvalue**;  
  
 **private** Double **yvalue**;  
  
 **private** String **text**;  
  
  
}

# 5.前端配置

hw.Html的百度API调用和echarts的组件里的两个脚本文件

<**script type="text/javascript" src="http://api.map.baidu.com/api?v=2.0&ak=78hQX88IBEYa78XqOHE52w9ouaYFobrM"**></**script**>  
<**script src="/js/echarts.min.js"**></**script**>  
  
<**script src="/js/bmap.min.js"**></**script**>

百度自定义

**mapStyle**: { *// 百度地图自定义样式* **styleJson**: [{  
 **"featureType"**: **"land"**,*// 陆地* **"elementType"**: **"all"**,  
 **"stylers"**: {  
 **"color"**: **"#073763"** }  
 },{  
 **"featureType"**: **"water"**,*// 水系* **"elementType"**: **"all"**,  
 **"stylers"**: {  
 **"color"**: **"#073763"**,  
 **"lightness"**: -54  
 }  
 },{  
 **"featureType"**: **"highway"**,*// 国道与高速* **"elementType"**: **"all"**,  
 **"stylers"**: {  
 **"color"**: **"#45818e"** }  
 },{  
 **"featureType"**: **"boundary"**,*// 边界线* **"elementType"**: **"all"**,  
 **"stylers"**: {  
 **"color"**: **"#ffffff"**,  
 **"lightness"**: -62,  
 **"visibility"**: **"on"** }  
 },{  
 **"featureType"**: **"label"**,*// 行政标注* **"elementType"**: **"labels.text.fill"**,  
 **"stylers"**: {  
 **"color"**: **"#ffffff"**,  
 **"visibility"**: **"on"** }  
 },{  
 **"featureType"**: **"label"**,  
 **"elementType"**: **"labels.text.stroke"**,  
 **"stylers"**: {  
 **"color"**: **"#444444"**,  
 **"visibility"**: **"on"** }  
 }  
 ]  
}

# 系统测试

设计测试用例，给出程序每个功能模块的运行结果截图。

运行hw.work文件，会使服务运行，在浏览输入<http://localhost:8080/syd/打开界面如下图4-1>：

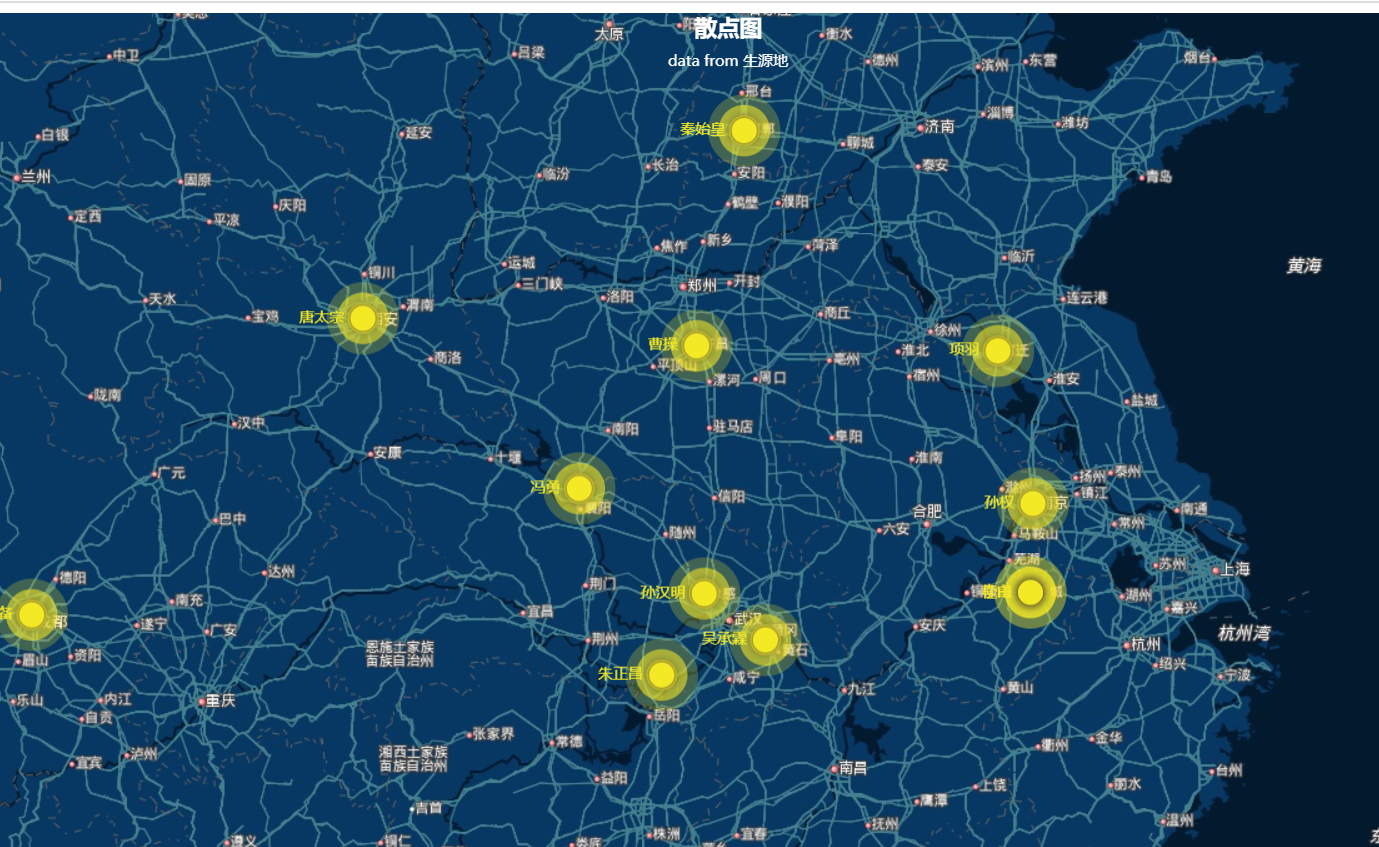


图4-1：主界面

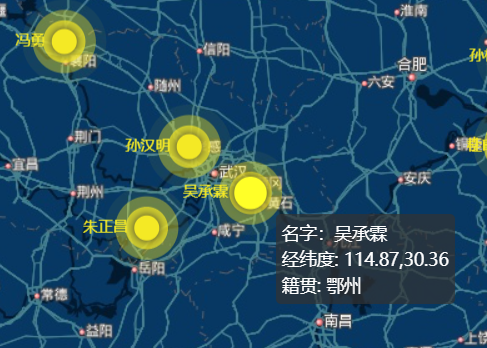


图4-2：显示学生学籍信息

# 系统总结

**7.1 echarts 图表的初始化和显示：**

echarts 对象：在标签式引入的方式中，有 echarts 对象，所有图表的初始化、事件的绑定...等等都通过 echarts 这个对象来处理。

一个简单的 echarts 图表显示总结起来有四步（可参考官网中 demo 的例子：http://echarts.baidu.com/doc/start.html）：

* 准备一个 DOM：
* <**div id="map-wrap"**>  
   *<!-- 这里以后是地图 -->*</**div**>

这是存放 echarts 图表的 DOM，**切记**在此需要指定 DOM 的 width 和 height 属性（即官网所说的”具备大小的 DOM“），否则图表不会显示的。

初始化图表： **var** bmapChart = **echarts**.init(***document***.getElementById(**'map-wrap'**));

组织 option（图表的核心对象：数据、样式...）：

这里就是定义所需要的图表，echarts 图表编程其实就是写 option ，组织数据、定义样式、各种控件...的显示，看 API 。echarts 就是基于 JSON 对象和数组的，option 对象的各个属性很规范。

加载数据： bmapChart.setOption(option);