武汉纺织大学

Web应用开发课程设计

**级联显示**

**学 院： 数学与计算机学院**

**班 级： 软件11801**

**姓 名： 陶佳乐**

**学 号： 1804240716**

**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](#_Toc22456)

[1.2主页面级联显示](#_Toc15319)

[2 系统设计](#_Toc21868)

[2.1用户用例图](#_Toc16712)

[2.2 ER图](#_Toc13261)

[2.3 UML类图（Class Diagram）](#_Toc12060)

[2.4 UML时序图（Sequence Diagram）](#_Toc9481)

[2.4.1 读取省份信息](#_Toc10858)

[2.4.2 读取大学信息](#_Toc18620)

[2.5 UML活动图（Activity Diagram）](#_Toc22054)

[2.5.1 读取省份信息](#_Toc12767)

[2.5.2 读取大学信息](#_Toc20020)

[3 系统实现](#_Toc20261)

[3.1 项目结构](#_Toc17930)

[3.2 配置文件](#_Toc14003)

[3.2.1 jdbc](#_Toc32289)

[3.2.2 web.xml文件](#_Toc20299)

[3.3 VO类scholol.java province.java](#_Toc7838)

[3.4 DAO类provinceDao.java schoolDao.java](#_Toc24038)

[3.5 tools类 jdbcutil.java](#_Toc1705)

[3.6 controller queryProvinceSchool.java](#_Toc18558)

[3.7前端页面 test.html js.js css.css](#_Toc1416)

[4 系统测试](#_Toc7918)

[5 系统总结](#_Toc18618)

# 1 需求分析

设计一个级联选择的功能，里面省份和省份大学的级联选择，数据都是来自于数据库。

当程序运行时，显示“欢迎使用\*\*\*\*超市收银系统，请登陆”，提示用户依次输入用户名及其密码，当用户名及密码不正确时，显示提示信息“用户名或密码不正确，请重新输入”，

## 1.1主菜单

主页面里面显示所有的数据

# 2 系统设计

## 2.1用户用例图

系统分为省份，省份大学两个角色，各角色的用户用例如下图2-1所示：

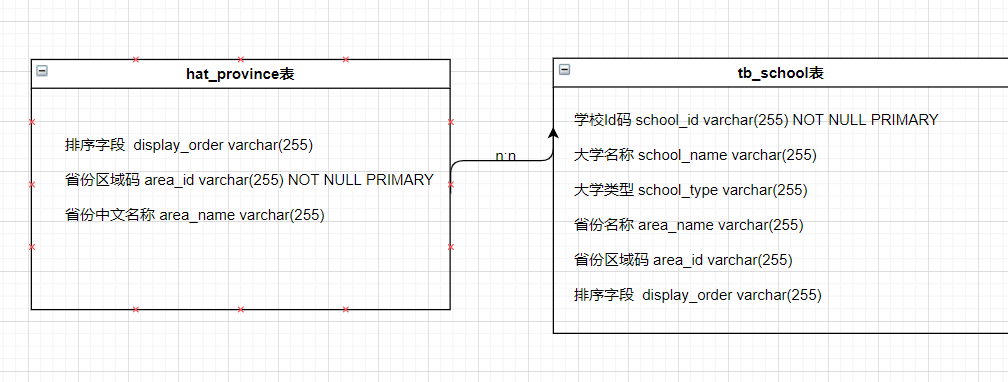
图2-1 用户用例图

## 2.2 ER图

数据库设计2张表，省份表，学校表，对应的ER图如图3-1所示。

%3C

图3-1 数据库ER图

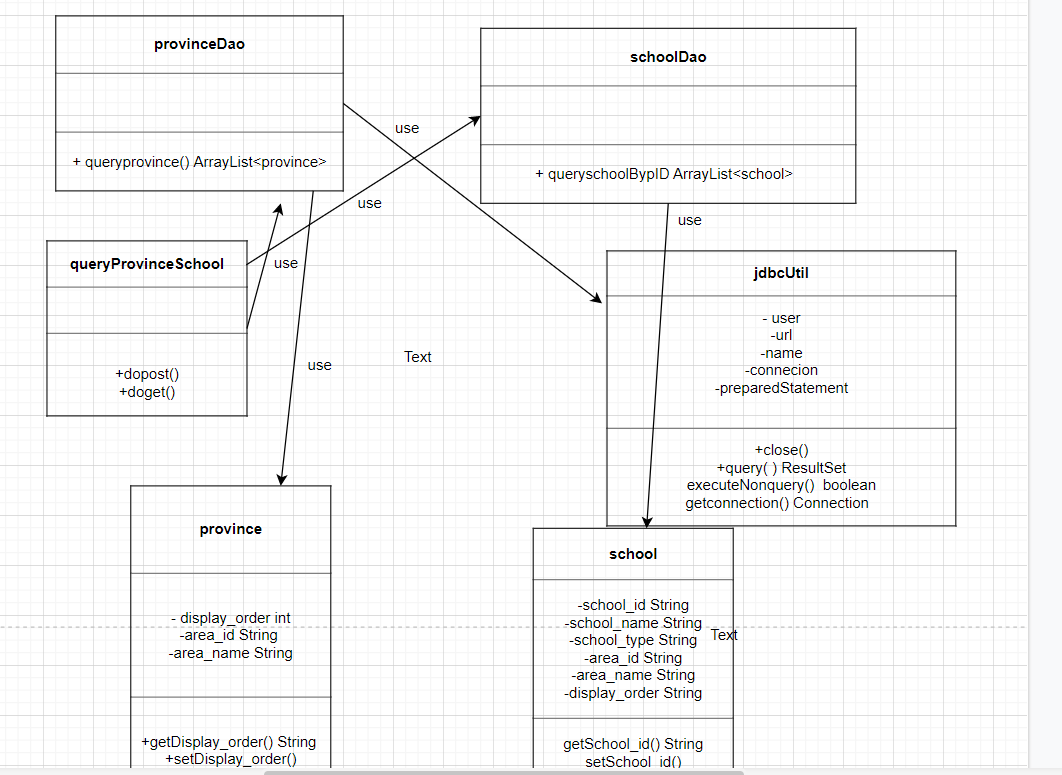


## 2.3 UML类图（Class Diagram）

对于用户功能模块，共设计如下6个类。

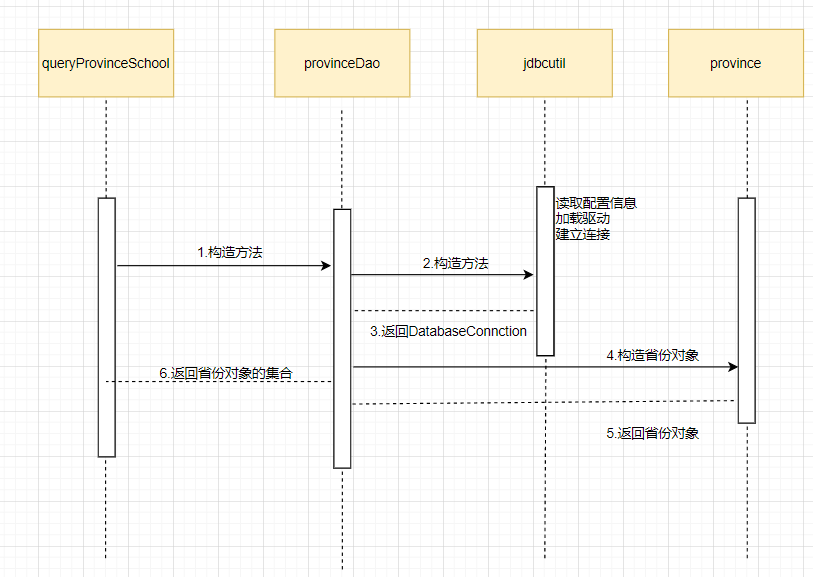
* 数据库连接类jdbcutil：连接数据库并获取连接对象，关闭连接对象。
* VO实体类school,province：与数据库结构进行映射的类。主要由属性，setter, getter方法组成，VO类中的属性与表中的字段相对应，每一个VO类的对象都表示表中的每一条记录
* DAO provinceDAO：主要定义操作的接口，定义一系列数据库的原子性操作，例如增删改查（通常称为CRUD）等。
* Controller类 queryProvinceSchool：主要接收前端的请求然后根据请求返回相应的数据各类的结构及类之间的关系如图2-5所示：

图2-5 用户功能类图

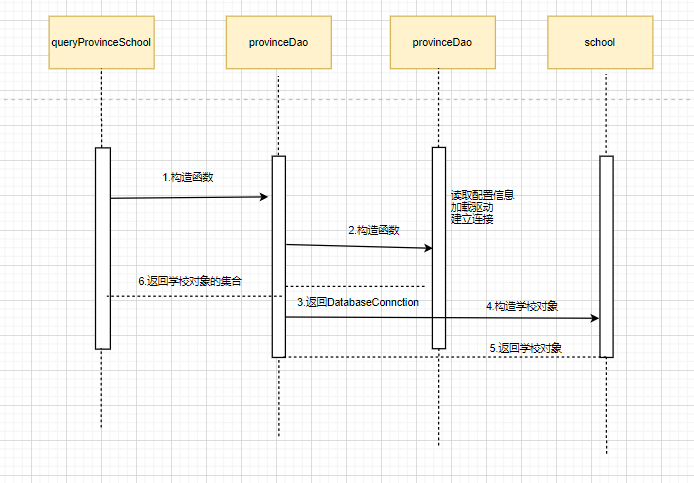


## 2.4 UML时序图（Sequence Diagram）

### 2.4.1 读取省份信息



### 2.4.2 读取学校信息



## 2.5 UML活动图（Activity Diagram）

### 2.5.1 读取省份和学校信息

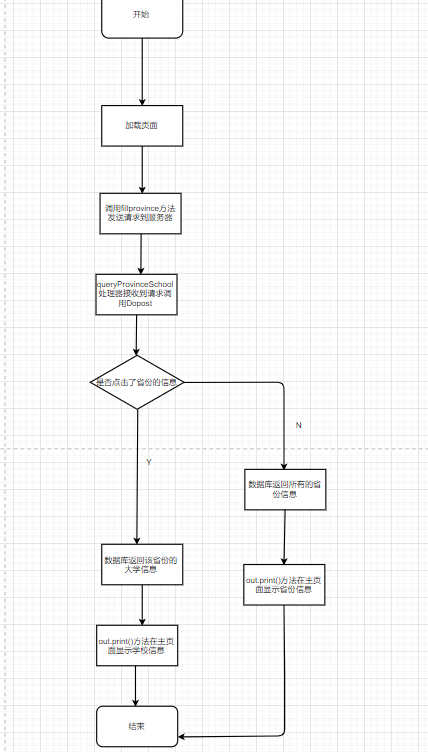
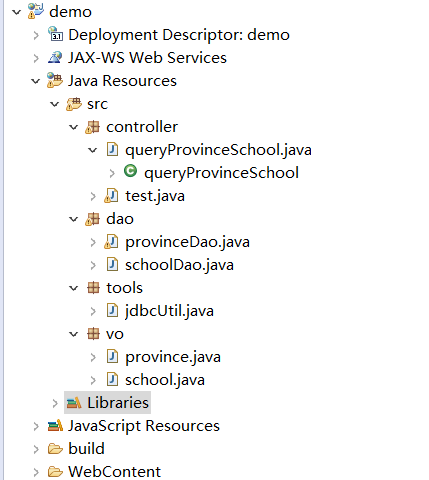


图2-2 读取省份和学校信息流程图

# 3 系统实现

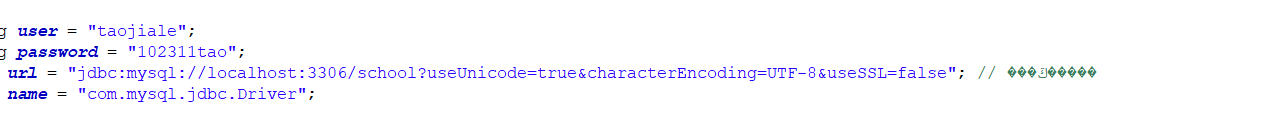
## 3.1 项目结构



## 3.2 配置文件

在项目下新建Source Folder，名为resources

### 3.2.1 jdbc



### 3.2.2 web.xml文件

描述该文件作用，并对里面的配置进行解释

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* version=*"3.1"*>

<display-name>demo</display-name>

<welcome-file-list>

<welcome-file>test.html</welcome-file>

</welcome-file-list>

### </web-app>

## 3.3 VO类school.java

学校类

**package** vo;

**public** **class** school {

**private** String school\_id;

**private** String school\_name;

**private** String school\_type;

**private** String area\_id;

**private** String area\_name;

**private** String display\_order;

**public** String getSchool\_id() {

**return** school\_id;

}

**public** **void** setSchool\_id(String school\_id) {

**this**.school\_id = school\_id;

}

**public** String getSchool\_name() {

**return** school\_name;

}

**public** **void** setSchool\_name(String school\_name) {

**this**.school\_name = school\_name;

}

**public** String getSchool\_type() {

**return** school\_type;

}

**public** **void** setSchool\_type(String school\_type) {

**this**.school\_type = school\_type;

}

**public** String getArea\_id() {

**return** area\_id;

}

**public** **void** setArea\_id(String area\_id) {

**this**.area\_id = area\_id;

}

**public** String getArea\_name() {

**return** area\_name;

}

**public** **void** setArea\_name(String area\_name) {

**this**.area\_name = area\_name;

}

**public** String getDisplay\_order() {

**return** display\_order;

}

**public** **void** setDisplay\_order(String display\_order) {

**this**.display\_order = display\_order;

}

**public** school() {

**super**();

}

**public** school(String school\_id, String school\_name, String school\_type, String area\_id, String area\_name,

String display\_order) {

**super**();

**this**.school\_id = school\_id;

**this**.school\_name = school\_name;

**this**.school\_type = school\_type;

**this**.area\_id = area\_id;

**this**.area\_name = area\_name;

**this**.display\_order = display\_order;

}

@Override

**public** String toString() {

**return** "school [school\_id=" + school\_id + ", school\_name=" + school\_name + ", school\_type=" + school\_type

+ ", area\_id=" + area\_id + ", area\_name=" + area\_name + ", display\_order=" + display\_order + "]";

}

}

## 3.4 VO类province.java

省份的类

**package** vo;

**public** **class** province {

**private** **int** display\_order;

**private** String area\_id;

**private** String area\_name;

**public** province(**int** display\_order, String area\_id, String area\_name) {

**super**();

**this**.display\_order = display\_order;

**this**.area\_id = area\_id;

**this**.area\_name = area\_name;

}

**public** **int** getDisplay\_order() {

**return** display\_order;

}

**public** **void** setDisplay\_order(**int** display\_order) {

**this**.display\_order = display\_order;

}

**public** String getArea\_id() {

**return** area\_id;

}

**public** **void** setArea\_id(String area\_id) {

**this**.area\_id = area\_id;

}

**public** String getArea\_name() {

**return** area\_name;

}

**public** **void** setArea\_name(String area\_name) {

**this**.area\_name = area\_name;

}

@Override

**public** String toString() {

**return** "province [display\_order=" + display\_order + ", area\_id=" + area\_id + ", area\_name=" + area\_name + "]";

}

**public** province() {

// **TODO** Auto-generated constructor stub

}

}

## 3.5 tools类 jdbcUtil.java

这个类的作用是连接数据库

## package tools;

## import java.sql.DriverManager;

## import java.sql.ResultSet;

## import java.sql.SQLException;

## import com.mysql.jdbc.Connection;

## import com.mysql.jdbc.PreparedStatement;

## public class jdbcUtil {

## private static final String user = "taojiale";

## private static final String password = "102311tao";

## public static final String url = "jdbc:mysql://localhost:3306/school?useUnicode=true&characterEncoding=UTF-8&useSSL=false"; // ���ݿ�����

## public static final String name = "com.mysql.jdbc.Driver";

## public Connection connection = null;

## public PreparedStatement preparedStatement = null;

## public jdbcUtil() {

## try {

## Class.forName(name);

## connection = (Connection) DriverManager.getConnection(url, user, password);// ��ȡ����

## } catch (Exception e) {

## e.printStackTrace();

## }

## }

## public void close() {

## try {

## this.connection.close();

## this.preparedStatement.close();

## } catch (SQLException e) {

## System.out.println("连接数据库失败");

## e.printStackTrace();

## }

## }

## public ResultSet query(String sql) {

## ResultSet resultSet = null;

## try {

## preparedStatement = (PreparedStatement) connection.prepareStatement(sql); // ׼��ִ�����

## resultSet = preparedStatement.executeQuery();

## } catch (Exception e) {

## System.out.println("执行语句失败");

## e.printStackTrace();

## }

## return resultSet;

## }

## public boolean executeNonquery(String sql) {

## boolean flag = false;

## try {

## preparedStatement = (PreparedStatement) connection.prepareStatement(sql);

## preparedStatement.executeUpdate();

## flag = true;

## } catch (Exception e) {

## System.out.println("�������ݿ�ʱ���ִ��󣡣�");

## e.printStackTrace();

## }

## return flag;

## }

## public Connection getconnection() throws SQLException {

## return connection;

## }

## 

## 

## }

## 3.6 dao类 provinceDao.java

可以通过这类拿到数据库里面省份的信息

package dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import tools.jdbcUtil;

import vo.province;

import vo.school;

public class provinceDao {

public ArrayList<province> queryprovince() throws SQLException{

ArrayList<province> SH=new ArrayList<province>();

jdbcUtil jdbc=new jdbcUtil();

String sql="select \* from hat\_province";

ResultSet rs=jdbc.query(sql);

while(rs.next()){

//首先新建一个school

province p=new province(rs.getInt("display\_order"),rs.getString("area\_id"),rs.getString("area\_name"));

SH.add(p);

}

return SH;

}

public provinceDao() {

// TODO Auto-generated constructor stub

}

}

## 3.7 dao类 schoolDao.java

可以通过这个类拿到数据里面学校的信息

package dao;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import tools.jdbcUtil;

import vo.school;

public class schoolDao {

public schoolDao() {

// TODO Auto-generated constructor stub

}

public ArrayList<school> queryschool() throws SQLException{

ArrayList<school> SH=new ArrayList<school>();

jdbcUtil jdbc=new jdbcUtil();

String sql="select \* from tb\_school";

ResultSet rs=jdbc.query(sql);

while(rs.next()){

//首先新建一个school

school s=new school(rs.getString("school\_id"),rs.getString("school\_name"),rs.getString("school\_type"),rs.getString("area\_id"),rs.getString("area\_name"),rs.getString("display\_order"));

SH.add(s);

}

return SH;

}

//根据省份id找到

public ArrayList<school> queryschoolBypID(String area\_id) throws SQLException{

ArrayList<school> SH=new ArrayList<school>();

jdbcUtil jdbc=new jdbcUtil();

String sql="select \* from tb\_school where area\_id="+area\_id;

ResultSet rs=jdbc.query(sql);

while(rs.next()){

//首先新建一个school

school s=new school(rs.getString("school\_id"),rs.getString("school\_name"),rs.getString("school\_type"),rs.getString("area\_id"),rs.getString("area\_name"),rs.getString("display\_order"));

SH.add(s);

}

return SH;

}

}

### 3.8.controller类 queryProvinceschool

这个类的作用是接收前端发送过来的请求 queryProvinceschool可以得到省份和学校的

信息

## package controller;

## import java.io.IOException;

## import java.io.PrintWriter;

## import java.sql.SQLException;

## import java.util.ArrayList;

## import javax.servlet.ServletException;

## import javax.servlet.annotation.WebServlet;

## import javax.servlet.http.HttpServlet;

## import javax.servlet.http.HttpServletRequest;

## import javax.servlet.http.HttpServletResponse;

## import com.google.gson.Gson;

## import dao.provinceDao;

## import dao.schoolDao;

## import vo.province;

## import vo.school;

## /\*\*

## \* Servlet implementation class queryProvinceSchool

## \*/

## @WebServlet("/queryProvinceSchool")

## public class queryProvinceSchool extends HttpServlet {

## private static final long serialVersionUID = 1L;

## 

## /\*\*

## \* @see HttpServlet#HttpServlet()

## \*/

## public queryProvinceSchool() {

## super();

## // TODO Auto-generated constructor stub

## }

## /\*\*

## \* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

## \*/

## protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

## // TODO Auto-generated method stub

## //response.getWriter().append("Served at: ").append(request.getContextPath());

## }

## /\*\*

## \* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

## \*/

## protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

## // TODO Auto-generated method stub

## //doGet(request, response);

## String area\_id= request.getParameter("area\_id");

## System.out.println("显示数据:");

## System.out.println(area\_id);

## String jsonStr ="";

## //分别得到城市和省份的数据

## schoolDao citydao=new schoolDao();

## provinceDao provincedao=new provinceDao();

## if(area\_id==null)//省份页面还没有选择

## {

## //将省份的信息用json表示

## 

## ArrayList<province> prolist ;

## try {

## prolist=provincedao.queryprovince() ;

## //将数组里面的数据放入map中

## // for(int i=0;i<prolist.size();i++)

## // {

## // map.put(String.valueOf(prolist.get(i).getPid()), prolist.get(i).getProvincial());

## // }

## jsonStr=new Gson().toJson(prolist);

## } catch (SQLException e) {

## // TODO Auto-generated catch block

## e.printStackTrace();

## }

## }

## else//选择了省份

## {

## try {

## ArrayList<school> slist=citydao.queryschoolBypID(area\_id);

## 

## slist.toString();

## //for(int i=0;i<clist.size();i++)

## //{

## //map.put(String.valueOf(clist.get(i).getPid()), clist.get(i).getProvincial());

## //}

## jsonStr=new Gson().toJson(slist);

## } catch (SQLException e) {

## // TODO Auto-generated catch block

## e.printStackTrace();

## }

## }

## 

## response.setCharacterEncoding("utf-8");

## response.setContentType("text/xml;charset=utf-8");

## response.setHeader("Cache-Control", "no-cache");

## //确保在取得PrintWriter对象之前先设置response的编码；顺序搞错将无法解决中文乱码问题！

## PrintWriter out = response.getWriter();

## 

## //PrintWriter out =response.getWriter();

## //System.out.println(map);

## System.out.println(jsonStr);//输入json字符串

## //String json = new Gson().toJson(map);

## out.print(jsonStr);

## out.flush();

## out.close();

## }

## 

## }

## 3.9 前端页面代码

## 3.9.1 test.html

<!DOCTYPE html>

<html lang=*"en"*>

<head>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<title>Document</title>

<link href=*"css/css.css"* rel=*"stylesheet"*>

<script src=*"js/jquery.min.js"*></script>

<script src=*"js/js.js"*></script>

</head>

<body>

<h1>选择学校</h1>

<div class=*"w"*>

<ul id=*"areaid"* class=*"province\_Style"*>

</ul>

</div>

<div class=*"w2"*>

<ul id=*"schoolid"* class=*"school\_Style"*></ul>

</div>

</body>

## </html>

**3.9.2 js代码**

**function** fillProvince() {

$.ajax({

type: "POST",

url: "queryProvinceSchool",

contentType: "application/x-www-form-urlencoded;charset=utf-8",

data: "",

dataType: "json",

success: **function**(response) {

//增加一个option

// provinceElement = document.getElementById("areaid");

//清空

//provinceElement.options.length = 0;

//provinceElement.add(new Option("请选择省份", ""));

//console.log(info[2].pid);

**for** (**var** index = 0; index < response.length; index++) {

//一个li的数值

//var str = "<li id='" + response[index].area\_id + "'>" + response[index].area\_name + "</li>";

str = "<li id='" + response[index].area\_id + "'>" + response[index].area\_name + "</li>";

$("#areaid").append("<li onclick='getarea\_id(this)' id='" + response[index].area\_id + "'>" + response[index].area\_name + "</li>");

}

}

});

}

**function** fillschool() {

$.ajax({

type: "POST",

url: "queryProvinceSchool",

contentType: "application/x-www-form-urlencoded;charset=utf-8",

data: { "area\_id": area\_id },

dataType: "json",

success: **function**(response) {

//增加一个option

//provinceElement = document.getElementById("school");

//清空

// provinceElement.options.length = 0;

// provinceElement.add(new Option("请选择学校", ""));

//console.log(info[2].pid);

**var** count = 0;

**for** (**var** index = 0; index < response.length; index++) {

count++;

$("#schoolid").append("<li id='" + response[index].school\_id + "'>" +"<span>"+ count +"</span>"+ "." + response[index].school\_name + "</li>");

}

}

});

}

window.onload = **function**() {

//首先调用函数,填充下拉框

//fillProvince();

fillProvince();

}

// $(function() {

// $(document).click(function(e) { // 在页面任意位置点击而触发此事件

// var v\_id = $(e.target).attr('id');

// })

// });

//拿到id的数值

**var** area\_id;

**function** getarea\_id(e) {

//清空所有的li

$("#schoolid").find("li").remove();

area\_id = e.id;

fillschool();

}

**3.9.3 css代码**

*.province\_Style* **li** {

list-style: *none*;

float: *left*;

margin-right: *10px*;

font-size: *13px*;

color: *black*;

}

*.province\_Style* **li***:hover* {

color: *aliceblue*;

background-color: *aqua*;

}

*.w* {

width: *1200px*;

height: *100px*;

margin: *0* *auto*;

border-bottom: *1px* *solid* *gray*;

border-top: *1px* *solid* *gray*;

}

*.w2* {

width: *1200px*;

height: *1000px*;

margin: *0* *auto*;

border-bottom: *1px* *solid* *gray*;

border-top: *1px* *solid* *gray*;

}

*.school\_Style* **li** {

width: *250px*;

list-style: *none*;

float: *left*;

font-size: *13px*;

color: *#4787be*;

}

*.school\_Style* **li** **span** {

color: *black*;

}

4 系统测试

点击山西省





点击湖北省



# 5 系统总结

在使用级联选择的时候，我用原生js取元素,然后添加数据没有效果，但是使用jq的时候可以将数据添加进去.

总体来说我认为我此次的作业不足之处就是没有使用到框架，我认为我对于框架还不是特别熟练，因此我就没使用框架来做这个题目。

因此我在今后的学习中应该要更加深层次的理解框架的意思。