武汉纺织大学

Web应用开发课程设计

**收银系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11804**

**姓 名： 黄溶**

**学 号： 1804280316**

**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](#_Toc59201084)

[1.1. 显示省份 1](#_Toc59201085)

[1.2. 显示大学 1](#_Toc59201086)

[2. 系统设计 1](#_Toc59201087)

[2.1. 用例图 1](#_Toc59201088)

[2.2. ER图 2](#_Toc59201089)

[2.3. UML类图（Class Diagram） 2](#_Toc59201090)

[2.4. UML时序图（Sequence Diagram） 4](#_Toc59201091)

[2.5. UML活动图（Activity Diagram） 4](#_Toc59201092)

[3. 系统实现 5](#_Toc59201093)

[3.1. 项目结构 5](#_Toc59201094)

[3.2. 配置文件 6](#_Toc59201095)

[3.2.1. application.properties文件 6](#_Toc59201096)

[3.2.2. application.yml文件 6](#_Toc59201097)

[3.2.3. UniversityController.java文件 7](#_Toc59201098)

[3.3. VO类University.java 8](#_Toc59201099)

[3.4. Mapper层UniversityMapper.java 9](#_Toc59201100)

[3.5. 接口映射文件UniversityMapper.xml 10](#_Toc59201101)

[3.6. Service层接口IUniversityService.java 11](#_Toc59201102)

[3.7. Service层实现声明的接口UniversityServiceImpl.java 11](#_Toc59201103)

[4. 系统测试 12](#_Toc59201104)

[5. 系统总结 13](#_Toc59201105)

# 需求分析

设计一个各省份大学级联查询系统，实现选择省份显示该省份包含的大学，具体要求如下：

## 显示省份

打开界面显示各个省份

## 显示大学

当点击某个省份时，显示该省包含的大学，如图所示：点湖北省时显示如下：每个学校前含有编号。



图1-1：需求界面

# 系统设计

## ER图

数据库设计1张表如图2-2所示：

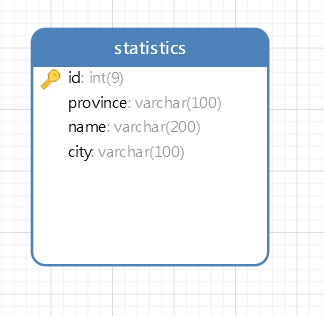


图2-2 数据库ER图

## 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UniversityController类：获取异步请求并发送数据。
* VO实体类University类：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。
* mapper层接口UniversityMapper类：主要定义数据操作的接口，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。
* service层 IUniversityService类:定义service层接口，声明方法。
* service层IUniversityServiceImpl类：调用mapper层，实现接口声明的方法。
* 应用开关TongjiApplication类：启动spring boot开发的应用。各类的结构及类之间的关系如图2-3所示：

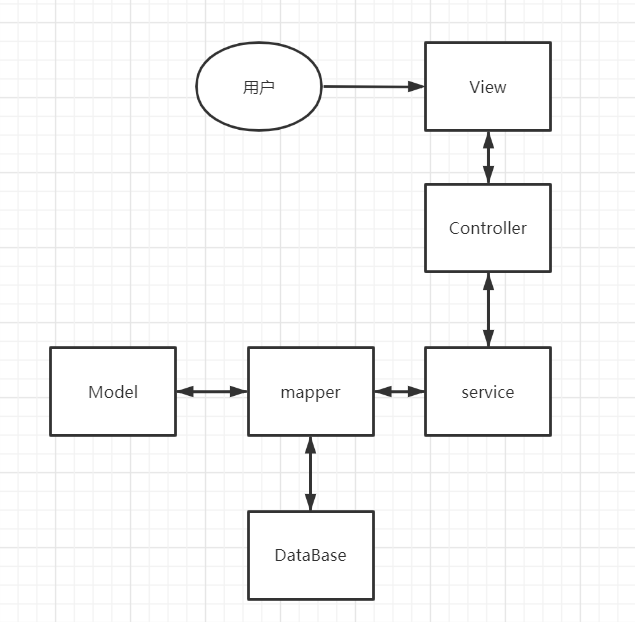


图2-3：UML类图

## UML时序图（Sequence Diagram）

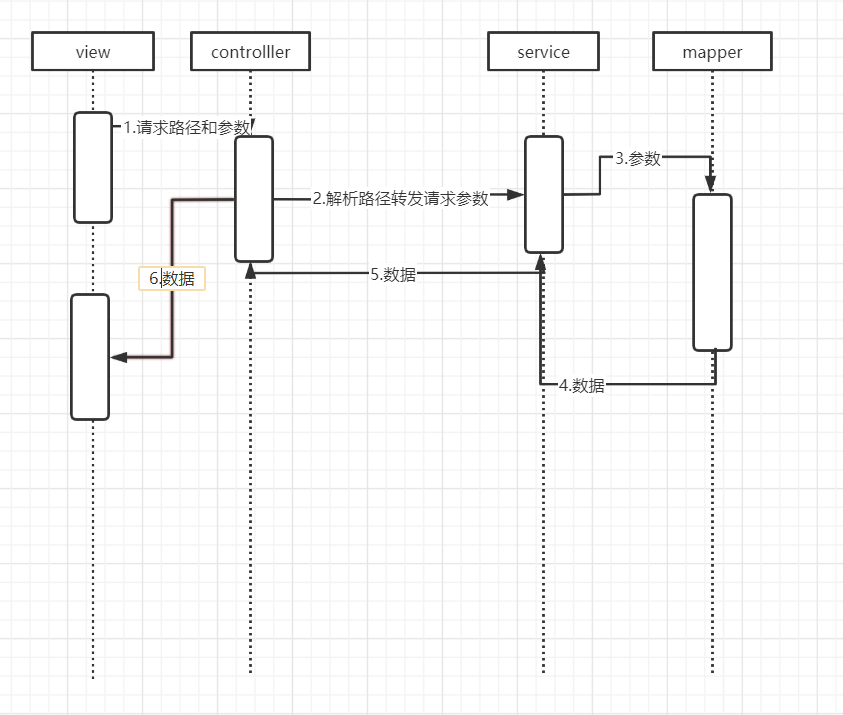


图2-4：时序图

## UML活动图（Activity Diagram）

用户会打开网页会看到各个省份名称，点击省份展示该省包含的大学，具体活动图2-5所示。

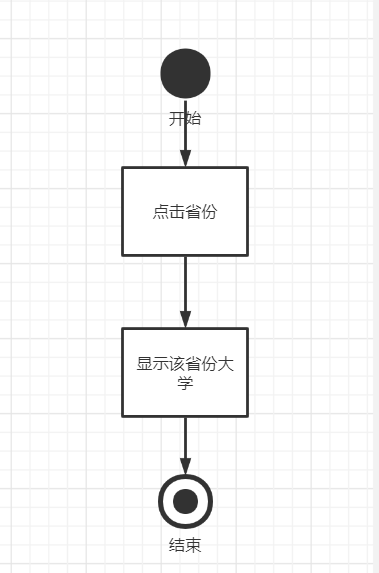


图2-5 登录流程图

# 系统实现

## 项目结构

项目结构如下图3-1所示：

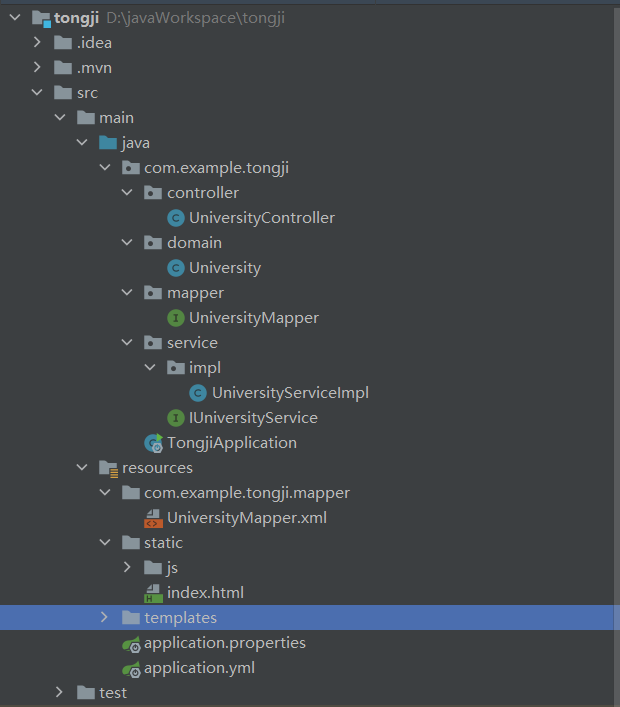


图3-1：系统结构

## 配置文件

在项目下新建Source Folder，名为resources

### application.properties文件

该配置文件主要是以文件形式保存view层的访问路径。

spring.thymeleaf.prefix=classpath:/templates/

spring.mvc.view.prefix=classpath:/templates/

spring.mvc.view.suffix=.html

### application.yml文件

配置应用运行的端口、数据库连接用户名和密码、mybatis配置检索路径和文件

# 数据源配置

spring:

datasource:

url: jdbc:mysql://134.175.39.56:3306/simple?useUnicode=true&characterEncoding=UTF-8&useSSL=false

username: root

password: qq123456@liangzai

driver-class-name: com.mysql.jdbc.Driver

# MyBatis 此处在使用了maperscan后 可以不需要了/

mybatis:

# 搜索指定包别名

typeAliasesPackage: com.example.tongji.domain

mapper-locations: classpath:com.example.tongji.mapper/\*\*/\*.xml

# 开发环境配置

server:

# 服务器的HTTP端口，默认为80

port: 80

servlet:

# 应用的访问路径

context-path: /

### UniversityController.java文件

控制器，捕获浏览器发来的请求

package com.example.tongji.controller;

import com.example.tongji.domain.University;

import com.example.tongji.service.IUniversityService;

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.PathVariable;

import org.springframework.web.bind.annotation.ResponseBody;

import java.util.List;

@Controller

public class UniversityController {

@Autowired

private IUniversityService universityService;

@GetMapping("/search/{prvince}") //请求路径 如：http://localhost/serar/湖北

@ResponseBody

public List<University> searchList(@PathVariable("prvince")String province){

return universityService.listU(province);

}

}

## VO类University.java

映射数据库的表

package com.example.tongji.domain;

/\*\*

\* 参数配置表 University

\*

\*/

public class University {

private Long id;

private String Province;

private String Name;

private String City;

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = id;

}

public String getProvince() {

return Province;

}

public void setProvince(String province) {

this.Province = province;

}

public String getName() {

return Name;

}

public void setName(String name) {

this.Name = name;

}

public String getCity() {

return City;

}

public void setCity(String city) {

this.City = city;

}

public University(Long id, String province, String name, String city) {

this.id = id;

this.Province = province;

this.Name = name;

this.City = city;

}

@Override

public String toString() {

return "University{" +

"id=" + id +

", Province='" + Province + '\'' +

", Name='" + Name + '\'' +

", City='" + City + '\'' +

'}';

}

}

## Mapper层UniversityMapper.java

声明使用数据库的方法

**package com.example.tongji.mapper;**

**import com.example.tongji.domain.University;**

**import java.util.List;**

**/\*\***

**\* 参数配置 数据层**

**\***

**\*/**

**public interface UniversityMapper {**

**//根据省名词查询大学**

**List<University> listU(String province);**

**}**

## 接口映射文件UniversityMapper.xml

描述该文件作用，并对里面的关键信息进行解释

<?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.example.tongji.mapper.UniversityMapper">

<resultMap type="com.example.tongji.domain.University" id="UniversityResult">

<id property="id" column="id" />

<result property="Province" column="province" />

<result property="Name" column="name" />

<result property="City" column="city" />

</resultMap>

<!--vo配置-->

<sql id="selectConfigVo">

SELECT id, province, name, city

FROM statistics

</sql>

<!--数据库查询-->

<select id="listU" parameterType="java.lang.String" resultMap="UniversityResult">

<include refid="selectConfigVo"/>

where province = #{province}

</select>

</mapper>

## Service层接口IUniversityService.java

声明方法，调用mapper层

/ package com.example.tongji.service;

import com.example.tongji.domain.University;

import java.util.List;

/\*\*

\* 参数配置 服务层

\*

\*/

public interface IUniversityService {

List<University> listU(String province);

}

## Service层实现声明的接口UniversityServiceImpl.java

package com.example.tongji.service.impl;

import com.example.tongji.domain.University;

import com.example.tongji.mapper.UniversityMapper;

import com.example.tongji.service.IUniversityService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

/\*\*

\* 参数配置 服务层实现

\*

\*/

@Service

public class UniversityServiceImpl implements IUniversityService {

@Autowired

private UniversityMapper universityMapper;

@Override

public List<University> listU(String province) {

return universityMapper.listU(province);

}

}

# 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TongjiApplication文件，会使服务运行，在浏览输入localhost打开界面如下图4-1：

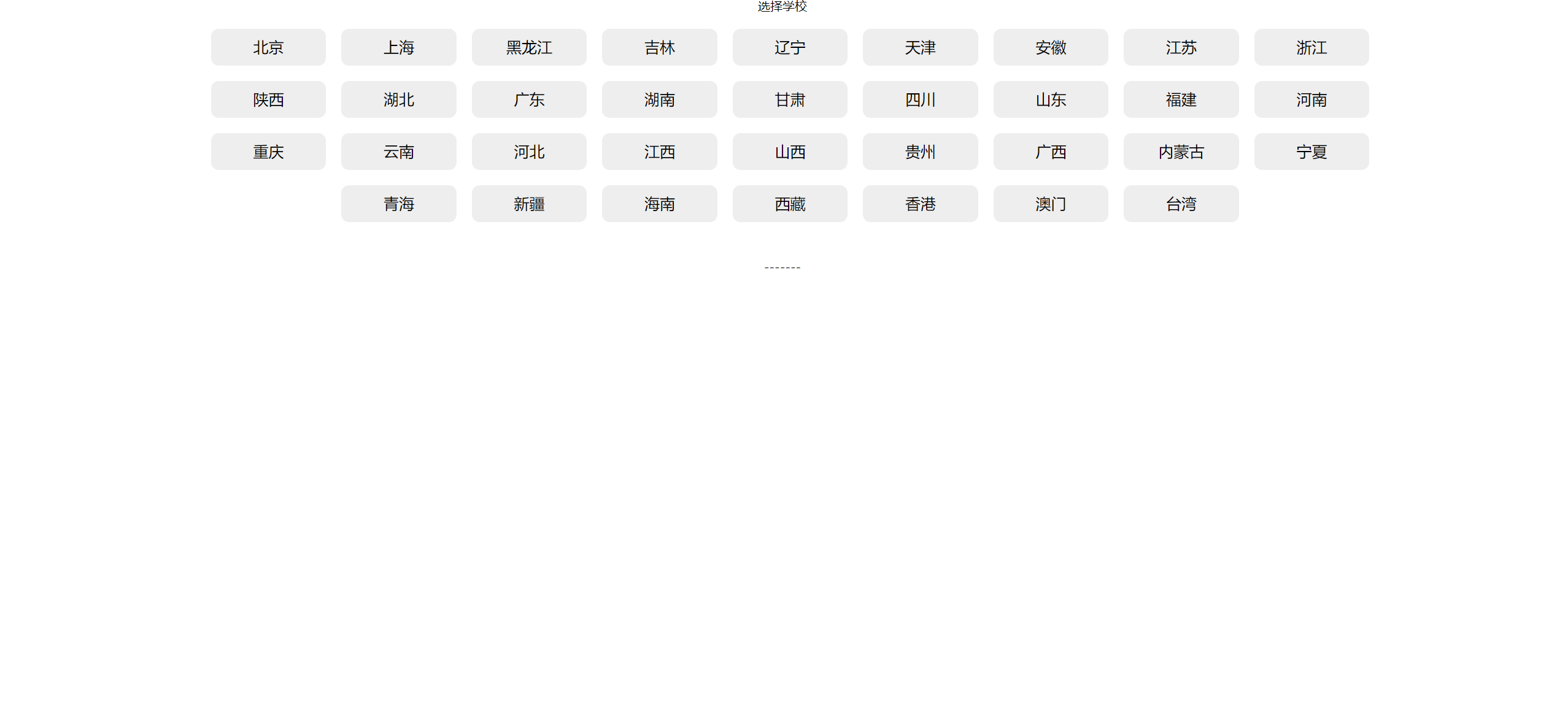


图4-1：主界面

选择北京，界面如下图4-2：



图4-2：显示北京高校

选择湖北，界面如下图4-3：



图4-3：湖北高校

# 系统总结

本系统使用了Springboot框架生成，SpringBoot是由Pivotal团队在2013年开始研发、2014年4月发布第一个版本的全新开源的轻量级框架。它基于Spring4.0设计，不仅继承了Spring框架原有的优秀特性，而且还通过简化配置来进一步简化了Spring应用的整个搭建和开发过程。另外SpringBoot通过集成大量的框架使得依赖包的版本冲突，以及引用的不稳定性等问题得到了很好的解决。

由于第一次尝试该框架，在环境配置时花费了很多时间，最终搞定。在编写java代码时，由于对springboot各层不够熟悉，导致程序无法执行，xml文件无法访问。在编写前端HTML代码时，由于对css的使用不太熟练，花费了大量的时间。但是经过不屈不挠的探索发现，终于大功告成，完成了实验任务。

本程序存在的问题还有：命名不规范、注释不合理、前端界面不友好等问题。在后续学习中我会加强对这几个方面的学习，写出更加规范的代码。