武汉纺织大学

Web应用开发课程设计

**商品信息增加系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11802**

**姓 名： 叶成**

**学 号： 1802220122**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年12月17日**

目 录

[1 需求分析 1](#_Toc4582)

[1.1分选项卡录入 1](#_Toc6313)

[1.2产地，计量单位，一级目录（ajax读取） 1](#_Toc32295)

[1.3选择一级目录后ajax读取二级目录 1](#_Toc14791)

[1.4点击保存后给出校验 1](#_Toc14791)

[1.5输入条形码后ajax唯一性校验 1](#_Toc14791)

[1.6校验通过后存入数据库 1](#_Toc14791)

[2 系统设计 2](#_Toc30285)

[2.1设计基本思想 2](#_Toc6011)

[2.2 uml类图 2](#_Toc218)

[3 系统实现 3](#_Toc13971)

[3.1 项目结构 3](#_Toc30810)

[3.2 配置文件 4](#_Toc24669)

[3.3 VO类 8](#_Toc31178)

[3.3.1 Goods.java文件 7](#_Toc4093)

[3.3.2 GoodsLocation.java文件 7](#_Toc15126)

[3.3.3 Payunit.java文件 7](#_Toc4093)

[3.3.4 Firstdirectory.java文件 7](#_Toc4093)

[3.3.5 Seconddirectory.java文件 7](#_Toc4093)

[3.4 DAO类 9](#_Toc7765)

[3.4.1 GoodsDao.java文件 7](#_Toc4093)

[3.4.2 GoodsLocationDao.java文件 7](#_Toc15126)

[3.4.3 PayunitDao.java文件 7](#_Toc4093)

[3.4.4 FirstdirectoryDao.java文件 7](#_Toc4093)

[3.4.5 SeconddirectoryDao.java文件 7](#_Toc4093)

[3.5 工具包 9](#_Toc30826)

[3.5.1 Jdbcutil.java文件 7](#_Toc4093)

[3.5.2 log4j2.xml文件 7](#_Toc4093)

[3.5.3 Spring-mvc.xml文件 7](#_Toc4093)

[3.6 服务层 9](#_Toc7997)

[3.6.1 GoodsController 9](#_Toc9988)

[3.6.2 GoodslocationController 10](#_Toc2207)

[3.6.3 PayunitController 9](#_Toc9988)

[3.6.4 DirectoryController 10](#_Toc2207)

[3.7 用户界面 11](#_Toc28650)

[3.6.1 index.html 9](#_Toc9988)

[3.6.2 index.css 10](#_Toc2207)

[4 系统测试 12](#_Toc31210)

[5 系统总结 12](#_Toc20663)

# 1 需求分析

设计一个商品信息增加的系统，该系统实现分选项卡录入、产地，计量单位，一级目录（ajax读取）、选择一级目录后ajax读取二级目录、点击保存后给出校验、输入条形码后ajax唯一性校验、校验通过后存入数据库这些功能，具体要求如下：

## 1.1分选项卡录入

当程序运行时，在网页上出现商品的基本信息的各个选项卡，数据来自本地数据库；

## 1.2产地，计量单位，一级目录（ajax读取）

当用户界面出现后，我们点击产地、计量单位、一级目录这些的时候会出现对应的选项，这些选项均来自于数据库，由ajax读取；

## 1.3选择一级目录后ajax读取二级目录

选完一级目录后，二级目录会根据一级目录的不同而发生不同的改变；

## 1.4点击保存后给出校验

点击保存后会给出校验信息，如果校验没通过会显示出错误信息，比如：商品码不能为空、条形码不能为空、名称不能为空、产地必须选择、计量单位不能为空、二级分类不能为空等等提示信息；

## 1.5输入条形码后ajax唯一性校验

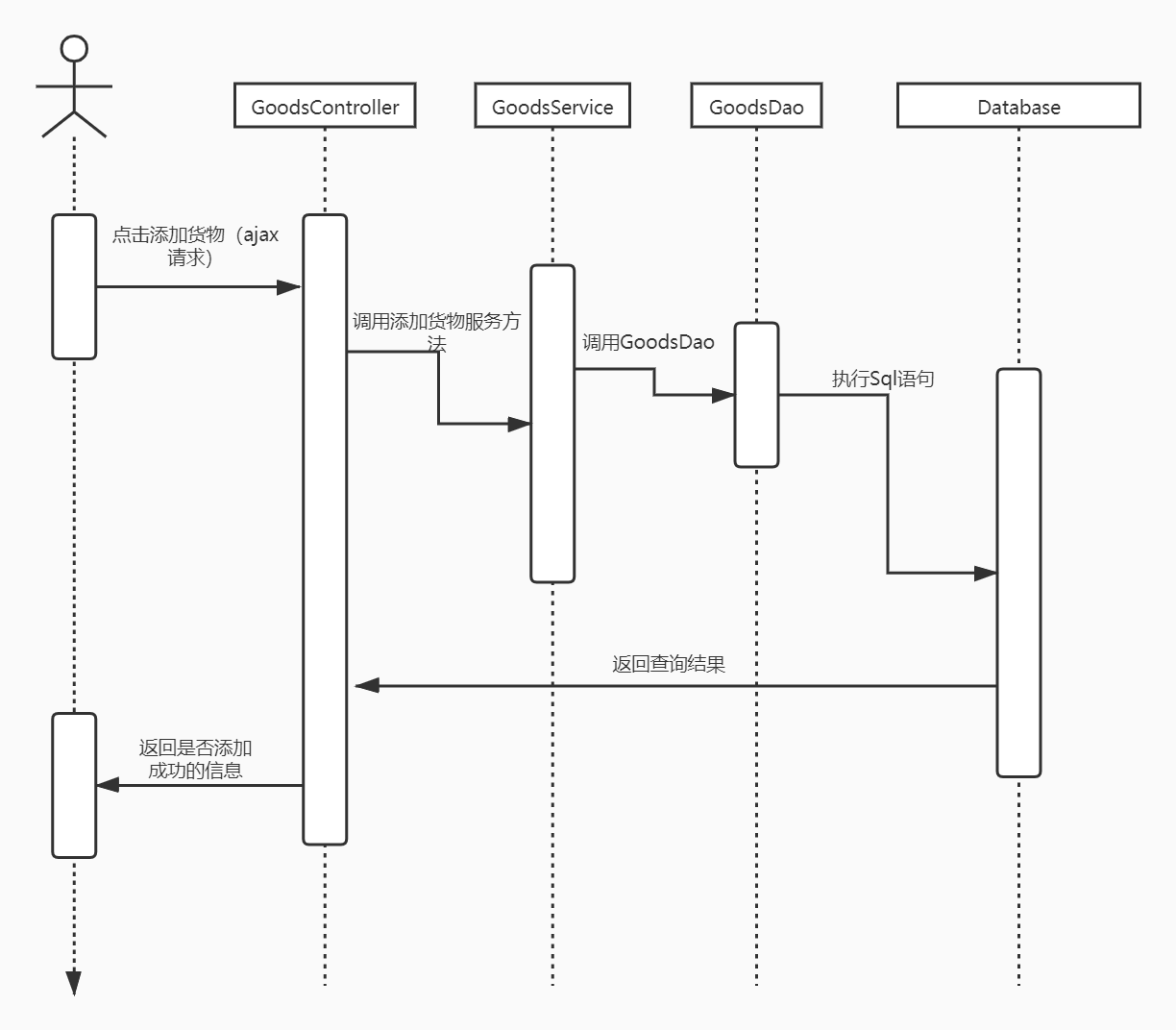
如果是唯一的那该增加的商品就保存成功，如果不是唯一的话那就会提示条形码已存在的提示信息；

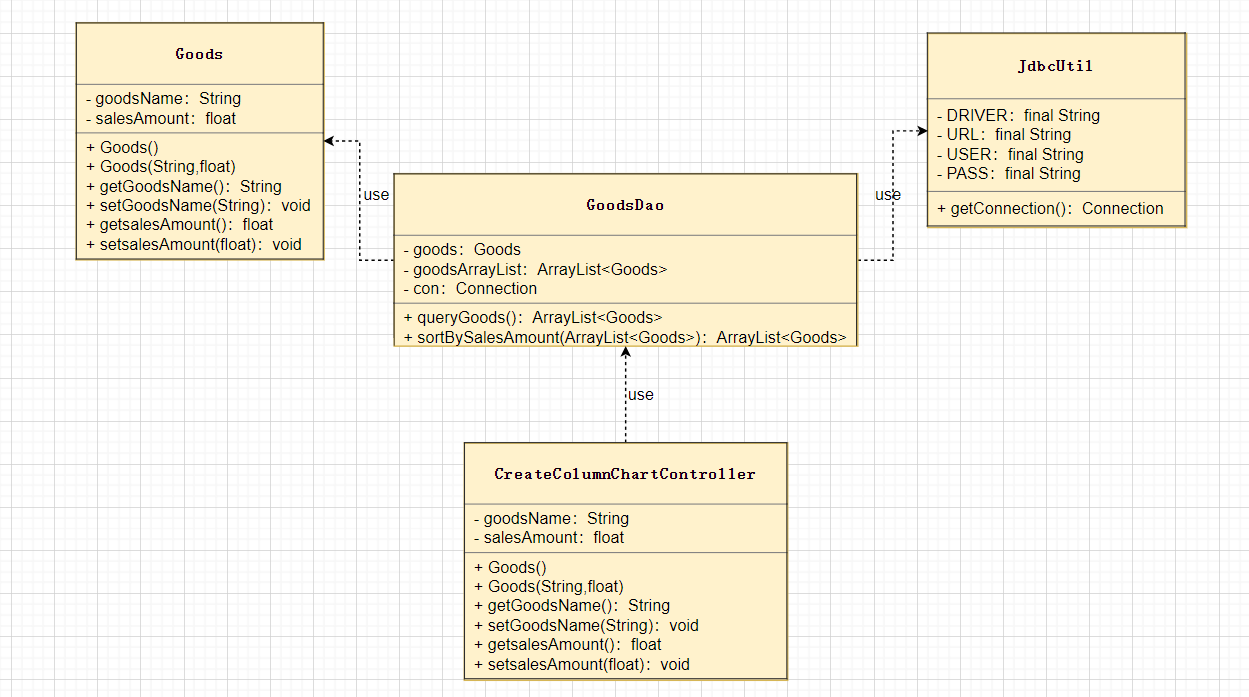
## 1.6校验通过后存入数据库

这里是所有信息都无误后将本产品的数据添加到数据库当中，这样，本系统就算完成了商品的信息增加功能了。

# **2 系统设计**

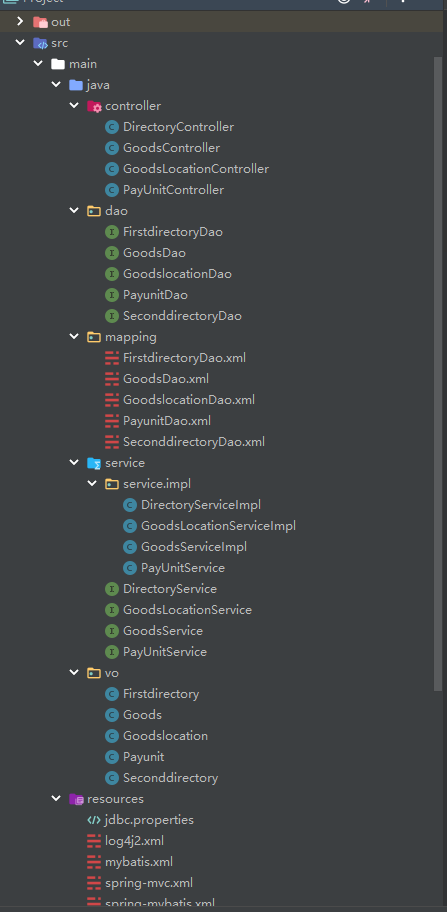
## 2.1设计基本思想

2.2 UML类图（Class Diagram）



# **3 系统实现**

## 3.1 项目结构



## 3.2 配置文件

web.xml文件：

<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xmlns="http://java.sun.com/xml/ns/javaee"  
 xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  
 id="WebApp\_ID" version="3.0">  
 <display-name>SSMDemo</display-name>  
 <welcome-file-list>  
 <welcome-file>index.html</welcome-file>  
 <welcome-file>index.jsp</welcome-file>  
 </welcome-file-list>  
 <!-- Spring和mybatis的配置文件 -->  
 <context-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath:spring-mybatis.xml</param-value>  
 </context-param>  
 <!-- log4j2的配置文件 -->  
 <context-param>  
 <param-name>log4jConfigLocation</param-name>  
 <param-value>classpath:log4j2.xml</param-value>  
 </context-param>  
  
 <!-- 编码过滤器 -->  
 <filter>  
 <filter-name>encodingFilter</filter-name>  
 <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>  
 <async-supported>true</async-supported>  
 <init-param>  
 <param-name>encoding</param-name>  
 <param-value>UTF-8</param-value>  
 </init-param>  
 </filter>  
 <filter-mapping>  
 <filter-name>encodingFilter</filter-name>  
 <url-pattern>/\*</url-pattern>  
 </filter-mapping>  
  
  
 <!-- Spring监听器 -->  
 <listener>  
 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>  
 </listener>  
 <!-- 防止Spring内存溢出监听器 -->  
 <listener>  
 <listener-class>org.springframework.web.util.IntrospectorCleanupListener</listener-class>  
 </listener>  
  
 <servlet>  
 <servlet-name>SpringMVC</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
 <init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>classpath:spring-mvc.xml</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 <async-supported>true</async-supported>  
 </servlet>  
 <servlet-mapping>  
 <servlet-name>SpringMVC</servlet-name>  
 <url-pattern>\*.do</url-pattern>  
 </servlet-mapping>  
  
  
</web-app>

## 3.3 VO类

**3.3.1 Goods.java**

商品信息类（商品货号、条形码、中文名、货品产地、计量单位、一级目录、二级目录）

public class Goods implements Serializable {  
 private String goodsid;  
 private String goodscode;  
 private String goodschrname;  
 private int goodslocation;  
 private int payunit;  
 private int firstdirectory;  
 private int seconddirectory;  
 private static final long serialVersionUID = 1L;  
  
 public Goods(String goodsid, String goodscode, String goodschrname, int goodslocation, int payunit, int firstdirectory, int seconddirectory) {  
 this.goodsid = goodsid;  
 this.goodscode = goodscode;  
 this.goodschrname = goodschrname;  
 this.goodslocation = goodslocation;  
 this.payunit = payunit;  
 this.firstdirectory = firstdirectory;  
 this.seconddirectory = seconddirectory;  
 }  
 public String getGoodsid() {  
 return goodsid;  
 }  
 public void setGoodsid(String goodsid) {  
 this.goodsid = goodsid;  
 }  
 public String getGoodscode() {  
 return goodscode;  
 }  
 public void setGoodscode(String goodscode) {  
 this.goodscode = goodscode;  
 }  
 public String getGoodschrname() {  
 return goodschrname;  
 }  
 public void setGoodschrname(String goodschrname) {  
 this.goodschrname = goodschrname;  
 }  
 public int getGoodslocation() {  
 return goodslocation;  
 }  
 public void setGoodslocation(int goodslocation) {  
 this.goodslocation = goodslocation;  
 }  
 public int getPayunit() {  
 return payunit;  
 }  
 public void setPayunit(int payunit) {  
 this.payunit = payunit;  
 }  
 public int getFirstdirectory() {  
 return firstdirectory;  
 }  
 public void setFirstdirectory(int firstdirectory) {  
 this.firstdirectory = firstdirectory;  
 }  
 public int getSeconddirectory() {  
 return seconddirectory;  
 }  
 public void setSeconddirectory(int seconddirectory) {  
 this.seconddirectory = seconddirectory;  
 }  
 @Override  
 public String toString() {  
 return "Goods{" +  
 "goodsid='" + goodsid + '\'' +  
 ", goodscode='" + goodscode + '\'' +  
 ", goodschrname='" + goodschrname + '\'' +  
 ", goodslocation=" + goodslocation +  
 ", payunit=" + payunit +  
 ", firstdirectory=" + firstdirectory +  
 ", seconddirectory=" + seconddirectory +  
 '}';  
 }  
}

**3.3.2 Goodslocation.java**

货品产地类

public class Goodslocation implements Serializable {  
 private Integer id;  
 private String goodslocation;  
 private static final long serialVersionUID = 1L;  
  
 public Goodslocation(Integer id, String goodslocation) {  
 this.id = id;  
 this.goodslocation = goodslocation;  
 }  
 public Integer getId() {  
 return id;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public String getGoodslocation() {  
 return goodslocation;  
 }  
 public void setGoodslocation(String goodslocation) {  
 this.goodslocation = goodslocation;  
 }  
 public static long getSerialVersionUID() {  
 return serialVersionUID;  
 }  
 @Override  
 public String toString() {  
 return "Goodslocation{" +  
 "id=" + id +  
 ", goodslocation='" + goodslocation + '\'' +  
 '}';  
 }  
}

**3.3.3 Payunit.java**

计量单位类

public class Payunit implements Serializable {  
 private Integer id;  
 private String payunit;  
 private static final long serialVersionUID = 1L;  
  
 public Payunit(Integer id, String payunit) {  
 this.id = id;  
 this.payunit = payunit;  
 }  
 public Integer getId() {  
 return id;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public String getPayunit() {  
 return payunit;  
 }  
 public void setPayunit(String payunit) {  
 this.payunit = payunit;  
 }  
 public static long getSerialVersionUID() {  
 return serialVersionUID;  
 }  
 @Override  
 public String toString() {  
 return "Payunit{" +  
 "id=" + id +  
 ", payunit='" + payunit + '\'' +  
 '}';  
 }  
}

**3.3.4Firstdirectory.java**

一级目录类

public class Firstdirectory implements Serializable {  
 private Integer id;  
 private String name;  
 private static final long serialVersionUID = 1L;  
  
 public Firstdirectory(Integer id, String name) {  
 this.id = id;  
 this.name = name;  
 }  
 public Integer getId() {  
 return id;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 public static long getSerialVersionUID() {  
 return serialVersionUID;  
 }  
  
 @Override  
 public String toString() {  
 return "Firstdirectory{" +  
 "id=" + id +  
 ", name='" + name + '\'' +  
 '}';  
 }  
}

**3.3.5Seconddirectory.java**

二级目录类

public class Seconddirectory implements Serializable {  
 private Integer id;  
 private Integer fid;  
 private String name;  
 private static final long serialVersionUID = 1L;  
  
 public Seconddirectory(Integer id, Integer fid, String name) {  
 this.id = id;  
 this.fid = fid;  
 this.name = name;  
 }  
 public Integer getId() {  
 return id;  
 }  
 public void setId(Integer id) {  
 this.id = id;  
 }  
 public Integer getFid() {  
 return fid;  
 }  
 public void setFid(Integer fid) {  
 this.fid = fid;  
 }  
 public String getName() {  
 return name;  
 }  
 public void setName(String name) {  
 this.name = name;  
 }  
 @Override  
 public String toString() {  
 return "Seconddirectory{" +  
 "id=" + id +  
 ", fid=" + fid +  
 ", name='" + name + '\'' +  
 '}';  
 }  
}

## 3.4 DAO类

### **3.4.1 GoodsDao.java**

package dao;  
  
import vo.Goods;  
public interface GoodsDao {  
 int deleteByPrimaryKey(String goodsid);  
 int insert(Goods record);  
 int insertSelective(Goods record);  
 Goods selectByPrimaryKey(String goodsid);  
 int updateByPrimaryKeySelective(Goods record);  
 int updateByPrimaryKey(Goods record);  
}

**3.4.2 GoodslocationDao.java**

从数据库获取货品产地信息

package dao;  
  
import vo.Goodslocation;  
import java.util.ArrayList;  
public interface GoodslocationDao {  
 ArrayList<Goodslocation> selectAllLocation();  
}

**3.4.3 PayunitDao.java**

从数据库获取计量单位信息

package dao;  
  
import vo.Payunit;  
import java.util.ArrayList;  
public interface PayunitDao {  
 ArrayList<Payunit> getAllUnit();  
 int deleteByPrimaryKey(Integer id);  
 int insert(Payunit record);  
 int insertSelective(Payunit record);  
 Payunit selectByPrimaryKey(Integer id);  
 int updateByPrimaryKeySelective(Payunit record);  
 int updateByPrimaryKey(Payunit record);

}

**3.4.4 FirstdirectoryDao.java**

从数据库获取一级目录信息

package dao;  
  
import vo.Firstdirectory;  
import java.util.ArrayList;  
public interface FirstdirectoryDao {  
 ArrayList<Firstdirectory> getAllFirstDirectory();  
 int deleteByPrimaryKey(Integer id);  
 int insert(Firstdirectory record);  
 int insertSelective(Firstdirectory record);  
 Firstdirectory selectByPrimaryKey(Integer id);  
 int updateByPrimaryKeySelective(Firstdirectory record);  
 int updateByPrimaryKey(Firstdirectory record);  
}

**3.4.5 SeconddirectoryDao.java**

从数据库获取二级目录信息

package dao;  
  
import org.apache.ibatis.annotations.Param;  
import vo.Seconddirectory;  
import java.lang.reflect.Array;  
import java.util.ArrayList;  
public interface SeconddirectoryDao {  
 ArrayList<Seconddirectory> getAllSecondDirectory(@Param("f\_id") int f\_id);  
 int deleteByPrimaryKey(Integer id);  
 int insert(Seconddirectory record);  
 int insertSelective(Seconddirectory record);  
 Seconddirectory selectByPrimaryKey(Integer id);  
 int updateByPrimaryKeySelective(Seconddirectory record);  
 int updateByPrimaryKey(Seconddirectory record);  
}

## 3.5 工具包

## **3.5.1 JdbcUtil.java**

连接数据库

driver=com.mysql.jdbc.Driver  
url=jdbc:mysql://101.132.227.58:3306/goodsManager?useUnicode=true&characterEncoding=utf-8&useSSL=false&serverTimezone=UTC  
username=itoffice  
password=1234  
#定义初始连接数?  
initialSize=0   
#定义最大连接数  
maxActive=20   
#定义最小空闲  
minIdle=1   
#定义最长等待时间?  
maxWait=60000   
#验证连接是否可用，使用的SQL语句  
validationQuery =SELECT 1

## **3.5.2 log4j2.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<!--日志级别以及优先级排序: OFF > FATAL > ERROR > WARN > INFO > DEBUG > TRACE > ALL -->  
<!-- status配置Log4j2启动和加载配置文件时的日志输出登记 -->  
<Configuration status="info">  
 <!--appenders:定义输出内容,输出格式,输出方式,日志保存策略等,常用其下三种标签[console,File,RollingFile] -->  
 <!--Appender可以理解为日志的输出目的地 -->  
 <Appenders>  
 <!--console :控制台输出的配置 -->  
 <Console name="Console" target="SYSTEM\_OUT">  
 <!--  
 %d 输出日志时间点的日期或时间  
 %t 输出产生该日志事件的线程名  
 %p 输出优先级，即DEBUG,INFO,WARN,ERROR,FATAL  
 %F 输出日志消息产生时所在的文件名称  
 %c 输出所属的类目,通常就是所在类的全名  
 %L 输出代码中的行号  
 -->  
 <PatternLayout pattern="%d{YYYY-MM-dd HH:mm:ss} %-5p [%c,%L]- %msg%n" />  
 </Console>  
  
 <RollingFile name="RollingFile" filename="log/ZycTest.log"  
 filepattern="${logPath}/%d{YYYYMMddHHmmss}-fargo.log">  
 <PatternLayout pattern="%d{YYYY-MM-dd HH:mm:ss} %-5p [%c,%L] - %msg%n" />  
 <Policies>  
 <SizeBasedTriggeringPolicy size="100 MB" />  
 </Policies>  
 <DefaultRolloverStrategy max="20" />  
 </RollingFile>  
 </Appenders>  
 <!--定义logger，只有定义了logger并引入的appender，appender才会生效 -->  
 <Loggers>  
 <!-- Logger节点用来单独指定日志的形式，可以为通过name属性设置指定包下的class指定不同的日志级别等 ,  
 可以设置Logger的additivity="false"只在自定义的Appender中进行输出 -->  
 <logger name="dao" level="debug" additivity="false">  
 <appender-ref ref="Console" />  
 </logger>  
 <!-- Root节点用来指定项目的根日志，如果没有单独指定Logger，那么就会默认使用该Root日志输出 -->  
 <Root level="info">  
 <AppenderRef ref="Console" />  
 <AppenderRef ref="RollingFile" />  
 </Root>  
 </Loggers>  
</Configuration>

## **3.5.3 spring-mvc.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:aop="http://www.springframework.org/schema/aop"  
 xmlns:tx="http://www.springframework.org/schema/tx"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans  
 http://www.springframework.org/schema/beans/spring-beans.xsd  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/mvc  
 http://www.springframework.org/schema/mvc/spring-mvc.xsd  
 http://www.springframework.org/schema/tx  
 http://www.springframework.org/schema/tx/spring-tx.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
  
 <mvc:annotation-driven />  
 <!-- 扫描类包，将标注Spring注解的类自动转化Bean，同时完成Bean的注入 -->  
 <context:component-scan base-package="controller" />  
 <context:component-scan base-package="service" />  
  
 <!--配置视图解析器 -->  
 <!-- ViewResolver 视图解析器 用于将返回的ModelAndView对象进行分离  
 InternalResourceViewResolver：用于支持Servlet、JSP视图解析；  
 viewClass：JstlView表示JSP模板页面需要使用JSTL标签库，classpath中必须包含jstl的相关jar包； prefix和suffix：查找视图页面的前缀和后缀（前缀[逻辑视图名]后缀），  
 比如传进来的逻辑视图名为hello，则该该jsp视图页面应该存放在“/hello.jsp”； -->  
 <bean  
 class="org.springframework.web.servlet.view.InternalResourceViewResolver">  
 <property name="prefix" value="/"></property> <!-- 视图放在webroot/下 -->  
 <property name="suffix" value=".jsp"></property>  
 </bean>  
</beans>

## 3.6 服务层

**3.6.1 GoodsController.java**

package controller;

import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import service.GoodsService;  
import vo.Goods;  
import java.util.HashMap;  
import java.util.Map;  
  
@Controller  
public class GoodsController {  
  
 @Autowired  
 private GoodsService goodsService;  
 @RequestMapping(value = "/insertGoods.do",method = RequestMethod.POST)  
 @ResponseBody  
 public Map<String,Object> insertGoods(String goodsid,  
 String goodscode,  
 String goodschrname,  
 int goodslocation,  
 int payunit,  
 int firstdirectory,  
 int seconddirectory){  
 Map<String,Object> map = new HashMap<String, Object>();  
 int flag = goodsService.insertGoods(  
 new Goods(goodsid,  
 goodscode,  
 goodschrname,  
 goodslocation,  
 payunit,  
 firstdirectory,  
 seconddirectory));  
 if(flag == 1){  
 map.put("code",0);  
 map.put("msg","添加成功");  
 }else{  
 map.put("code",-1);  
 map.put("msg","添加失败");  
 }  
 return map;  
 }  
}

**3.6.2 GoodsLocationController.java**

package controller;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import service.GoodsLocationService;  
import vo.Goodslocation;

import java.util.ArrayList;  
  
@Controller  
public class GoodsLocationController {  
  
 @Autowired  
 private GoodsLocationService goodsLocationService;  
  
 @RequestMapping(value = "/getLocation.do",method = RequestMethod.POST)  
 @ResponseBody  
 public ArrayList<Goodslocation> getAllLocation(){  
  
 System.out.println("Locations");  
  
 return goodsLocationService.getLocationList();  
 }  
}

**3.6.3PayUnitController.java**

package controller;

import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import service.PayUnitService;  
import vo.Payunit;

import java.util.ArrayList;  
  
@Controller  
public class PayUnitController {  
 @Autowired  
 private PayUnitService payUnitService;  
  
 @RequestMapping(value = "/getAllPayUnit.do",method = RequestMethod.POST)  
 @ResponseBody  
 public ArrayList<Payunit> getAllPayUnit(){  
 System.out.println("PayUnit");  
 return payUnitService.getAllUnit();  
 }  
}

**3.6.4 DirectoryController.java**

package controller;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.ResponseBody;  
import service.DirectoryService;  
import vo.Firstdirectory;  
import vo.Seconddirectory;  
  
import java.util.ArrayList;  
  
@Controller  
public class DirectoryController {  
  
 @Autowired  
 private DirectoryService directoryService;  
  
  
 @RequestMapping(value = "/getFirstDirectory.do",method = RequestMethod.POST)  
 @ResponseBody  
 public ArrayList<Firstdirectory> getFirst(){  
 return directoryService.getAllFirstDirectory();  
 }  
  
 @RequestMapping(value = "/getSecondDirectory.do",method = RequestMethod.POST)  
 @ResponseBody  
 public ArrayList<Seconddirectory> getSecond(int f\_id){  
 return directoryService.getAllSecondDirectory(f\_id);  
 }  
}

## 3.7 用户界面

### **3.7.1 index.html**

网页文件

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <link rel="stylesheet" href="css/index.css">  
 <script src="js/vue.js"></script>  
 <script src="js/jquery.min.js"></script>  
 <script src="js/index.js"></script>  
 <title>货物添加</title>  
</head>  
  
<body>  
 <div>  
 <div id="win-title">  
 <span>商品信息</span>  
 </div>  
 <div id="win-content">  
 <div class="win-content-1">  
 <div class="type1">  
 <strong>商品货号</strong>  
 <input placeholder="请输入商品货号" id="goodsId">  
 </div>  
 <div class="type1">  
 <strong>条形码</strong>  
 <input style="width: 310px;" placeholder="请输入条形码" id="goodsCode">  
 </div>  
 <div class="type2" style="margin-top: 10px;">  
 <strong>商品中文名称</strong>  
 <input placeholder="请输入商品中文名称" id="goodsChrName">  
 </div>  
 <div class="type1" style="margin-top: 10px;" id="goodsLocation">  
 <strong>货品产地</strong>  
 <select id="location">  
 <option value="0">--请选择--</option>  
 <option v-for="location in locations" :value="location.id">{{location.goodslocation}}</option>  
 </select>  
 </div>  
 <div class="type1" style="margin-top: 10px;" id="payUnit">  
 <strong>计量单位</strong>  
 <select id="pay\_unit">  
 <option value="0">--请选择--</option>  
 <option v-for="unit in units" :value="unit.id">{{unit.payunit}}</option>  
 </select>  
 </div>  
 <div class="type1" style="margin-top: 10px;" id="firstDirectory">  
 <strong>一级目录</strong>  
 <select id="first">  
 <option value="0">--请选择--</option>  
 <option v-for="sb in sbs" :value="sb.id">{{sb.name}}</option>  
 </select>  
 </div>  
 <div class="type1" style="margin-top: 10px;" id="secondDirectory">  
 <strong>二级目录</strong>  
 <select id="second">  
 <option value="0">--请选择--</option>  
 <option v-for="name in names" :value="name.id">{{name.name}}</option>  
 </select>  
 </div>  
 <div class="btns">  
 <button class="btn" id="goodsSave">商品保存</button>  
 <button class="btn">取消</button>  
 <button class="btn">退出</button>  
 </div>  
 </div>  
 </div>  
 </div>  
</body>  
  
</html>

**3.7.2 index.js**

$(document).ready(**function**() {  
 **var** app1 = **new** Vue({  
 el: "#goodsLocation",  
 data: {  
 locations: []  
 }  
 });  
  
 **var** app2 = **new** Vue({  
 el: "#payUnit",  
 data: {  
 units: []  
 }  
 });  
 **var** app3 = **new** Vue({  
 el: "#firstDirectory",  
 data: {  
 sbs: []  
 }  
 });  
 **var** app4 = **new** Vue({  
 el: "#secondDirectory",  
 data: {  
 names: []  
 }  
 });  
  
 $.ajax({  
 type: "post",  
 url: "getLocation.do",  
 data: "",  
 dataType: "json",  
 success: **function**(response) {  
 app1.locations = response;  
 }  
 });  
 $.ajax({  
 type: "post",  
 url: "getAllPayUnit.do",  
 data: "",  
 dataType: "json",  
 success: **function**(response) {  
 app2.units = response;  
 }  
 });  
  
 $.ajax({  
 type: "post",  
 url: "getFirstDirectory.do",  
 data: "",  
 dataType: "json",  
 success: **function**(response) {  
 app3.sbs = response;  
 }  
 });  
  
 $("#first").change(**function**(e) {  
 **if** ($("#first").val() == "0") {  
 alert("一级目录不可为空");  
 } **else** {  
 **var** f\_id = $("#first").val();  
 $.ajax({  
 type: "post",  
 url: "getSecondDirectory.do",  
 data: "f\_id=" + f\_id,  
 dataType: "json",  
 success: **function**(response) {  
 app4.names = response;  
 }  
 });  
 }  
 });  
  
 $("#goodsSave").click(**function**(e) {  
 **var** goodsId = $("#goodsId").val();  
 **var** goodsCode = $("#goodsCode").val();  
 **var** goodsChrName = $("#goodsChrName").val();  
 **var** payUnit = $("#pay\_unit").val();  
 **var** location = $("#location").val();  
 **var** first = $("#first").val();  
 **var** second = $("#second").val();  
 **if** (  
 $("#goodsId").val() != "" &&  
 $("#goodsCode").val() != "" &&  
 $("#goodsChrName").val() != "" &&  
 $("#location").val() != "0" &&  
 $("#pay\_unit").val() != "0" &&  
 $("#first").val() != "0" &&  
 $("#second").val() != "0") {  
 alert("数据正确");  
 $.ajax({  
 type: "post",  
 url: "insertGoods.do",  
 data: "goodsid=" + goodsId +  
 "&goodscode=" + goodsCode +  
 "&goodschrname=" + goodsChrName +  
 "&goodslocation=" + location +  
 "&payunit=" + payUnit +  
 "&firstdirectory=" + first +  
 "&seconddirectory=" + second,  
 dataType: "json",  
 success: **function**(response) {  
 alert(response.msg);  
 }  
 });  
 } **else** {  
 alert("数据存在为空情况");  
 }  
 });  
});

**3.7.3 index.css**

网页的样式文件

body {  
 text-align: center;  
}  
  
#win-title {  
 position: absolute;  
 top: 0;  
 left: 0;  
 width: 100%;  
 height: 40px;  
 text-align: center;  
 line-height: 40px;  
 background: cornflowerblue;  
 color: white;  
}  
  
#win-content {  
 width: 1000px;  
 height: 500px;  
 position: absolute;  
 top: 50px;  
 left: 50%;  
 margin-left: -500px;  
 margin-top: -;  
}  
  
.win-content-1 {  
 width: 900px;  
 height: 400px;  
 position: absolute;  
 top: 50%;  
 left: 50%;  
 margin-top: -200px;  
 margin-left: -450px;  
 text-align: center;  
}  
  
.type1 {  
 width: 400px;  
 height: 40px;  
 display: inline-block;  
}  
  
.type1 strong {  
 width: 100px;  
}  
  
.type1 input {  
 display: inline;  
 width: 295px;  
 height: 25px;  
}  
  
.type1 input:focus {  
 outline: 0;  
}  
  
.type1 select {  
 display: inline;  
 width: 305px;  
 height: 32px;  
}  
  
.type2 {  
 width: 900px;  
 height: 40px;  
}  
  
.type2 input {  
 display: inline;  
 width: 700px;  
 height: 25px;  
 margin-right: 30px;  
}  
  
.type2 input:focus {  
 outline: 0;  
}  
  
.btns {  
 float: right;  
 margin-right: 90px;  
}  
  
.btn {  
 color: white;  
 background: rgb(81, 124, 204);  
 border-radius: 10px;  
 width: max-content;  
 height: 30px;  
 border: 0;  
 padding-left: 10px;  
 padding-right: 10px;  
 padding-top: 5px;  
 padding-bottom: 5px;  
 margin-top: 10px;  
}  
  
.btn:focus {  
 outline: 0;  
}  
  
.btn:active {  
 transform: scale(0.98);  
}

}

**3.7.4 jquery-3.5.1.min.js**

**3.7.5 vue.js**

# **4 系统测试**

**4.1 输入商品货号，条形码，中文名，货品产地，计量单位，**

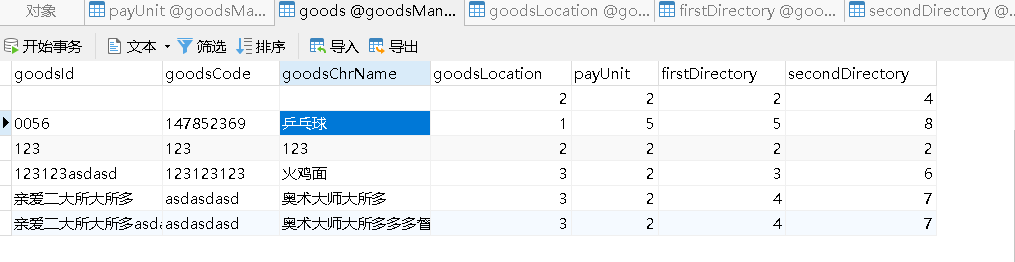
**一级目录和二级目录（这里我们先输入正确的信息）**







我们可以看到，乒乓球这个商品已经添加成功，下面我们来看数据库是否添加成功了，我们刷新一下数据库，结果如下：（可以看到乒乓球的信息已经添加成功）



**4.2 当填写的信息不全时，会弹出“数据存在为空情况”的字样，如下图所示：**

只要数据信息有一项为空就会出现下面的情况：



# **5 系统总结**

主要问题为下拉框的设计和上两行的设计不同且二级目录是根据一级目录的不同而发生改变的，当一级目录访问完数据库且选择了之后，二级目录才会去访问数据库，这个先后顺序不能搞错，否则显示出来的效果就不会如题目的要求一般出现一级目录不同二级目录也会随之发生改变的情况，系统存在的不足：当输入的信息不完全时，没有像想象的一样显示出没填写的地方能够显示出在哪里出现的问题。

课程收获：跟着聂老师又学习了一学期，掌握了不少的框架结构，学习到了许多关于 web应用开发的知识，在学习的同时也认识到了自己还有很多不足和需要继续学习的地方，这学期的自己相对以前的自己又有了不小的进步。

课程建议：老师做的已经非常非常的好了，认真负责，经常抽空给我们撰写教案且每周都有跟新，我们在学习的过程中又进一步学习到了老师的编程思想对我们的启发很大，目前还没有什么独特的建议，老师做的已经没什么能够挑剔的地方了。

发展方向：软件工程师

自评成绩：*90*