武汉纺织大学

Web应用开发课程设计

**基于bootstrap的查询显示页面**

**学 院： 数学与计算机学院**

**班 级： 软件11802班**

**姓 名： 刘颖**

**学 号： 1801230203**

**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](#_Toc22456)

[1.1用户登陆 1](#_Toc6247)

[1.2主菜单 1](#_Toc15319)

[1.3各项功能 1](#_Toc14458)

[1.3.1 收银 1](#_Toc1897)

[1.3.2 查询统计 1](#_Toc7940)

[1.3.3 商品维护 2](#_Toc4651)

[1.3.4 修改密码 2](#_Toc30015)

[1.3.5 数据导出 3](#_Toc5914)

[1.3.6 退出 3](#_Toc3637)

[1.3.7 \*\*\*\* 3](#_Toc6415)

[2 系统设计 3](#_Toc21868)

[2.1用户用例图 3](#_Toc16712)

[2.2 ER图 4](#_Toc13261)

[2.3 UML类图（Class Diagram） 4](#_Toc12060)

[2.4 UML时序图（Sequence Diagram） 6](#_Toc9481)

[2.4.1 系统登录 6](#_Toc10858)

[2.4.2 \*\*\*\*\*\*模块 6](#_Toc18620)

[2.4.3 \*\*\*\*\*\*模块 6](#_Toc30352)

[2.5 UML活动图（Activity Diagram） 6](#_Toc22054)

[2.5.1 登陆 6](#_Toc12767)

[2.5.2 收银 7](#_Toc20020)

[2.5.3 \*\*\*模块 8](#_Toc19945)

[2.5.4 \*\*\*\*模块 8](#_Toc26400)

[3 系统实现 9](#_Toc20261)

[3.1 项目结构 9](#_Toc17930)

[3.2 配置文件 9](#_Toc14003)

[3.2.1 jdbc.properties文件 9](#_Toc32289)

[3.2.2 log4j.properties文件 9](#_Toc20299)

[3.2.3 mybatis-config.xml文件 10](#_Toc19038)

[3.3 VO类User.java 10](#_Toc7838)

[3.4 DAO接口类IUserDAO.java 11](#_Toc24038)

[3.5 接口映射文件UserMapper.xml 11](#_Toc1705)

[3.6 工具包Util 11](#_Toc18558)

[3.6.1 MybatisUtils.java 11](#_Toc16083)

[3.6.2 MD5Util.java 12](#_Toc6085)

[3.7 服务层UserService.java 13](#_Toc19150)

[3.8 用户界面Driver.java 14](#_Toc25352)

[3.9 \*\*\*\*\*\*.java 14](#_Toc1416)

[4 系统测试 14](#_Toc7918)

[5 系统总结 14](#_Toc18618)

# 1 需求分析

设计一个查询显示页面，使用bootstrap table组件，实现网页的数据来自于数据库、可进行组合查询、某些字段支持模糊查询、分页显示功能，具体要求如下：

## 1.1主菜单

打开项目进入主页面，如下：



## 1.3各项功能

### 1.3.1 数据查询

将数据库中的数据导入bootstrap table中显示

### 1.3.2 查询统计

可根据商品编号、条形码、所有公司三个条件任意组合对数据库中的数据进行查询。

### 1.3.3 模糊查询

当用户通过input框进行查询时，可显示所有含有输入信息的记录，即可以只输入部分信息。

### 1.3.4 分页显示

当数据库中的数据记录很多时，将记录分成很多页，用户可通过分页导航读取显示需要的某页数据。也可选择每页显示多少条数据。

# 2 系统设计

## 2.1 ER图

数据库设计1张表，商品表，表对应的ER图如图3-1所示。

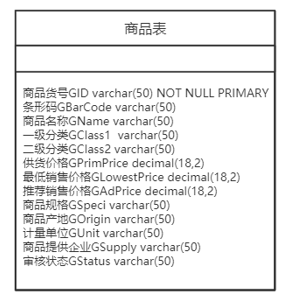


图3-1 数据库ER图

## 2.3 UML类图（Class Diagram）

对于功能模块，共设计如下6个类。

* 数据库连接类DatabaseConnection：连接数据库并获取连接对象，关闭连接对象。
* VO实体类Goods、Pages：与数据库结构进行映射的类。主要由属性，构造函数，setter, getter,Tostring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
* Controller类SearchInn：收集来自网页的用户输入，呈现数据库的记录，动态创建网页。
* DAO实现类SearchInDaoo：DAO接口的真实实现类，主要完成具体数据库操作，但不负责数据库的打开和关闭。
* 业务逻辑实现类databaseConnection：对于数据层的原子操作进行整合。还要负责数据库的打开与关闭

各类的结构及类之间的关系如图2-5所示：

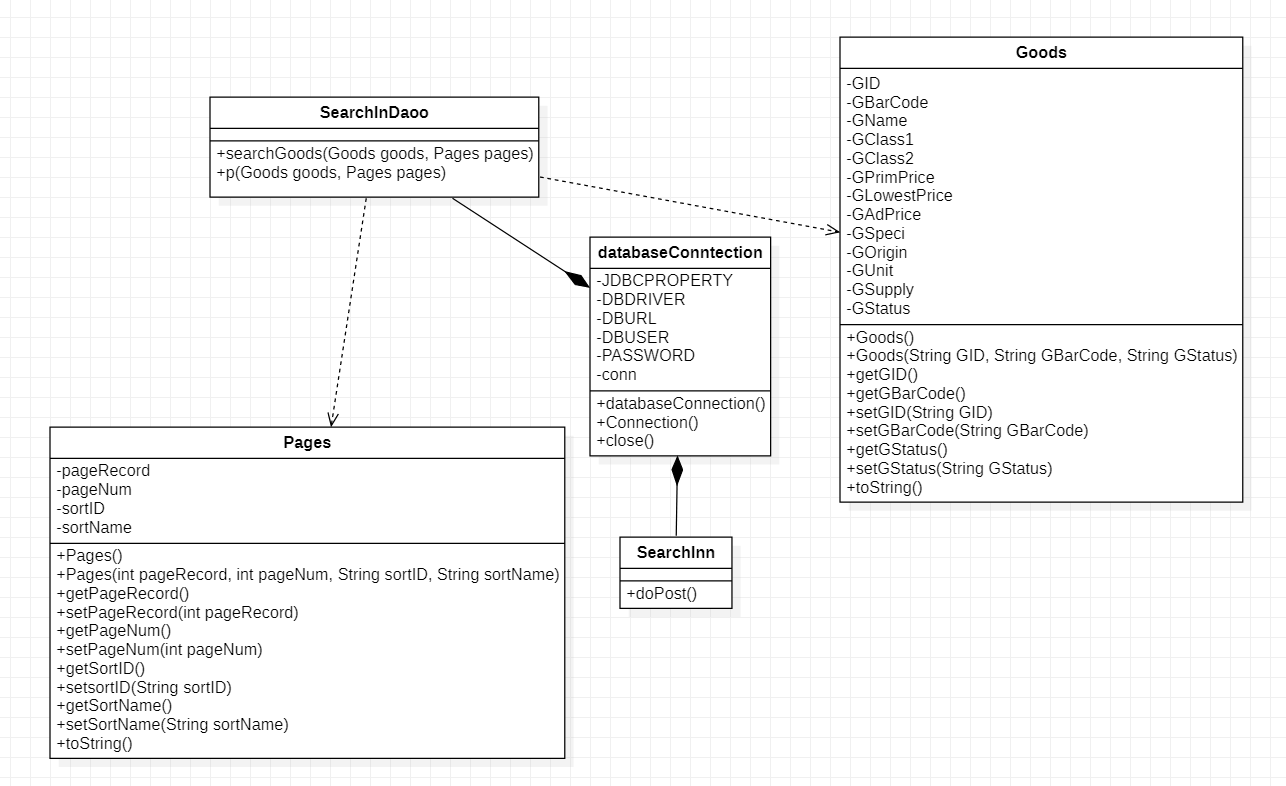
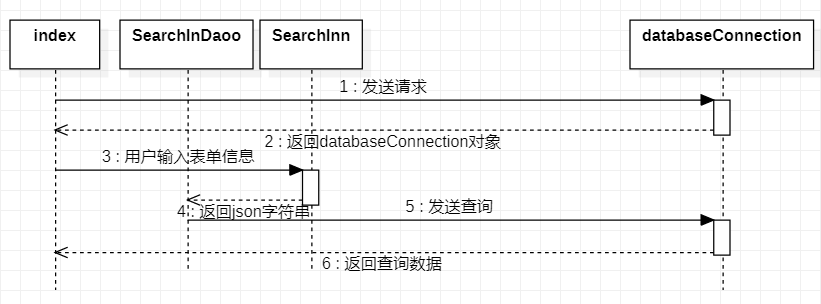


图2-3 类图

## 2.4 UML时序图（Sequence Diagram）



## 2.5 UML活动图（Activity Diagram）

### 2.5.1 用户查询

用户查询可任意输入商品货号、条形码、选择公司或不输入信息，网页将信息表单提交后数据库返回所要查询的数据，具体流程如图2-5-1所示。

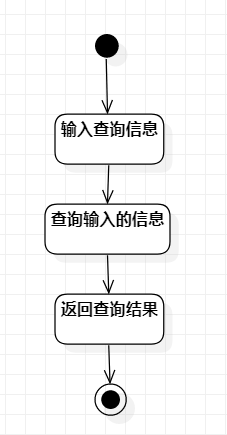


图2-5-1 数据查询流程图

### 2.5.2分页导航

分页导航允许用户选择每页显示的记录条数和查看那一页的数据，网页将信息表单提交后数据库返回页数以及修改分页导航按钮的状态，具体流程如图2-5-2所示。

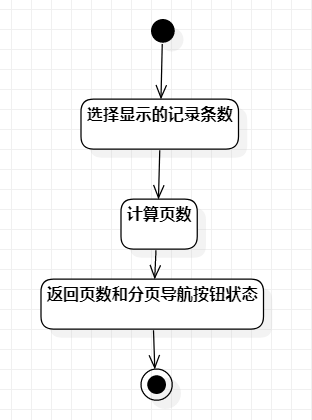
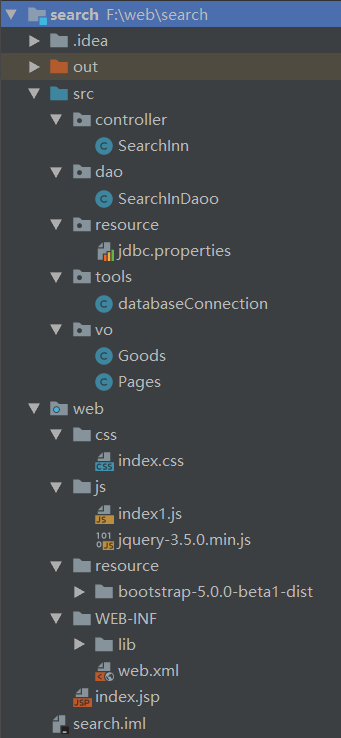


图2-5-1 分页显示流程图

# 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

jdbc.properties文件

该配置文件主要是以文件形式保存数据库的驱动类名称，连接数据库的URL地址，访问数据库的用户名及对应的密码，程序运行时会读取该文件相关信息，避免硬编码，当相关信息发生变化时，只需修改配置文件而不用修改源代码，增加程序的可扩展型。

DBDRIVER=com.mysql.jdbc.Driver

DBURL=jdbc:mysql://localhost:3306/search?userUnicode=true&characterEncoding=UTF-8

DBUSER=root

PASSWORD=LiuY

## 3.3 VO类Goods.java

实体类，数据库结构映射

package vo;

public class Goods {

private String GID;

private String GBarCode ;

private String GName ;

private String GClass1 ;

private String GClass2 ;

private double GPrimPrice ;

private double GLowestPrice ;

private double GAdPrice ;

private String GSpeci ;

private String GOrigin ;

private String GUnit ;

private String GSupply ;

private String GStatus ;

private String company;

public Goods() {

}

public Goods(String string, String rsString, String s, String string1, String rsString1, double aDouble, double rsDouble, double v, String s1, String string2, String rsString2, String s2, String string3) {

super();

}

public Goods(String GID, String GBarCode, String GName, String GClass1, String GClass2, double GPrimPrice, double GLowestPrice, double GAdPrice, String GSpeci, String GOrigin, String GUnit, String GSupply, String GStatus, String company) {

this.GID = GID;

this.GBarCode = GBarCode;

this.GName = GName;

this.GClass1 = GClass1;

this.GClass2 = GClass2;

this.GPrimPrice = GPrimPrice;

this.GLowestPrice = GLowestPrice;

this.GAdPrice = GAdPrice;

this.GSpeci = GSpeci;

this.GOrigin = GOrigin;

this.GUnit = GUnit;

this.GSupply = GSupply;

this.GStatus = GStatus;

this.company = company;

}

public String getGID() {

return GID;

}

public String getGBarCode() {

return GBarCode;

}

public String getGName() {

return GName;

}

public String getGClass1() {

return GClass1;

}

public String getGClass2() {

return GClass2;

}

public double getGPrimPrice() {

return GPrimPrice;

}

public double getGLowestPrice() {

return GLowestPrice;

}

public double getGAdPrice() {

return GAdPrice;

}

public String getGSpeci() {

return GSpeci;

}

public String getGOrigin() {

return GOrigin;

}

public String getGUnit() {

return GUnit;

}

public String getGSupply() {

return GSupply;

}

public String getGStatus() {

return GStatus;

}

public String getCompany() {

return company;

}

public void setGID(String GID) {

this.GID = GID;

}

public void setGBarCode(String GBarCode) {

this.GBarCode = GBarCode;

}

public void setGName(String GName) {

this.GName = GName;

}

public void setGClass1(String GClass1) {

this.GClass1 = GClass1;

}

public void setGClass2(String GClass2) {

this.GClass2 = GClass2;

}

public void setGPrimPrice(double GPrimPrice) {

this.GPrimPrice = GPrimPrice;

}

public void setGLowestPrice(double GLowestPrice) {

this.GLowestPrice = GLowestPrice;

}

public void setGAdPrice(double GAdPrice) {

this.GAdPrice = GAdPrice;

}

public void setGSpeci(String GSpeci) {

this.GSpeci = GSpeci;

}

public void setGOrigin(String GOrigin) {

this.GOrigin = GOrigin;

}

public void setGUnit(String GUnit) {

this.GUnit = GUnit;

}

public void setGSupply(String GSupply) {

this.GSupply = GSupply;

}

public void setGStatus(String GStatus) {

this.GStatus = GStatus;

}

public void setCompany(String company) {

this.company = company;

}

@Override

public String toString() {

return "Goods{" +

"GID='" + GID + '\'' +

", GBarCode='" + GBarCode + '\'' +

", GName='" + GName + '\'' +

", GClass1='" + GClass1 + '\'' +

", GClass2='" + GClass2 + '\'' +

", GPrimPrice=" + GPrimPrice +

", GLowestPrice=" + GLowestPrice +

", GAdPrice=" + GAdPrice +

", GSpeci='" + GSpeci + '\'' +

", GOrigin='" + GOrigin + '\'' +

", GUnit='" + GUnit + '\'' +

", GSupply='" + GSupply + '\'' +

", GStatus='" + GStatus + '\'' +

", company='" + company + '\'' +

'}';

}

}

## 3.3 VO类Pages.java

实体类，数据库结构映射

package vo;

import java.io.Serializable;

import java.util.HashMap;

public class Pages implements Serializable {

private int pageRecord;//每页的记录数

private int pageNum;//页数

private String sortID;

private String sortName;

public Pages() {

}

public Pages(int pageRecord, int pageNum, String sortID, String sortName) {

this.pageRecord = pageRecord;

this.pageNum = pageNum;

this.sortID = sortID;

this.sortName = sortName;

}

public static Pages getPageParams(HashMap<String, Object> mapPage) {

int pageRecord = Integer.parseInt((String) mapPage.get("pageRecord"));

int pageNum = Integer.parseInt((String) mapPage.get("pageNum"));

String sortID = (String) mapPage.get("GID");

String sortName = (String) mapPage.get("desc");

return new Pages(pageRecord,pageNum,sortID,sortName);

}

public int getPageRecord() {

return pageRecord;

}

public void setPageRecord(int pageRecord) {

this.pageRecord = pageRecord;

}

public int getPageNum() {

return pageNum;

}

public void setPageNum(int pageNum) {

this.pageNum = pageNum;

}

public String getSortID() {

return sortID;

}

public void setsortID(String sortID) {

this.sortID = sortID;

}

public String getSortName() {

return sortName;

}

public void setSortName(String sortName) {

this.sortName = sortName;

}

@Override

public String toString() {

return "Pages{" +

"pageRecord=" + pageRecord +

", pageNum=" + pageNum +

", sortID='" + sortID + '\'' +

", sortName='" + sortName + '\'' +

'}';

}

}

## 3.4 DAO类SearchInDaoo.java

package dao;

import tools.databaseConnection;

import vo.Goods;

import vo.Pages;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

//

public class SearchInDaoo {

//数据导出到table+模糊查询

public ArrayList<Goods> searchGoods(Goods goods, Pages pages){

ArrayList<Goods> list=new ArrayList<Goods>();//结果存放数组

StringBuffer condition=new StringBuffer();//查询条件

databaseConnection dbc=new databaseConnection();

Connection con =dbc.getConnection();

if(goods.getGID()!=null&&!"".equals(goods.getGID())){

condition.append(" and GID like '% ").append(goods.getGID()).append(" %' ");

}

if (goods.getGBarCode()+""!=null&&!"".equals(goods.getGBarCode())){

condition.append(" and GBarCode like '% ").append(goods.getGBarCode()).append(" %' ");

}

if (goods.getGSupply()+""!=null&&!"".equals(goods.getGSupply())){

condition.append(" and GSupply like '% ").append(goods.getGSupply()).append(" %' ");

}

int begin = pages.getPageRecord()\*(pages.getPageNum()-1);

String sql=" select \* from search ";

sql=sql+" where 1=1 ";

sql=sql+condition+ " order by " + pages.getSortID()+"limit" + " , "+pages.getPageRecord();

System.out.println(sql);

try {

PreparedStatement pst = con.prepareStatement(sql);

ResultSet rs=pst.executeQuery();

while (rs.next()){

goods.setGID(rs.getString("GID"));

goods.setGBarCode(rs.getString("GBarCode"));

goods.setGName(rs.getString("GName"));

goods.setGClass1(rs.getString("GClass1"));

goods.setGClass2(rs.getString("GClass2"));

goods.setGPrimPrice(rs.getDouble("PrimPrice"));

goods.setGLowestPrice(rs.getDouble("GLowestPrice"));

goods.setGAdPrice(rs.getDouble("GAdPrice"));

goods.setGSpeci(rs.getString("GSpeci"));

goods.setGOrigin(rs.getString("GOrigin"));

goods.setGUnit(rs.getString("GUnit"));

goods.setGSupply(rs.getString("GSupply"));

goods.setGStatus(rs.getString("GStatus"));

list.add(goods);

}

}catch (Exception e){

e.printStackTrace();

}

return list;

}

//页数输出

public int p(Goods goods,Pages pages){

int num=0;

ArrayList<Goods> list=new ArrayList<Goods>();//结果存放数组

databaseConnection dbc=new databaseConnection();

Connection con =dbc.getConnection();

String sql=" select count(GID) from search ";

System.out.println(sql);

try {

PreparedStatement pst = con.prepareStatement(sql);

ResultSet rs=pst.executeQuery();

if (rs.next()){

num = rs.getInt("count(GID)");

}

}catch (Exception e){

e.printStackTrace();

return 0;

}

return num;

}

}

## 3.5 Servlet SearchInn.java

package controller;

import com.google.gson.Gson;

import dao.SearchInDaoo;

import vo.Goods;

import vo.Pages;

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 java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

@WebServlet(name = "SearchInn" ,urlPatterns="/SearchInn.do")

public class SearchInn extends HttpServlet {

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

String GID = request.getParameter("GID");

String jsonStr = "";

SearchInDaoo dao=new SearchInDaoo();

if (GID==null){

ArrayList<Goods> list = dao.searchGoods();

jsonStr=new Gson().toJson(list);

}

response.setContentType("text/html;charset=uft-8");

PrintWriter out = response.getWriter();

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

out.println(jsonStr);

out.flush();

out.close();

}

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

}

}

## 3.6 接口映射文件web.xml

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

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_4\_0.xsd"

version="4.0">

<servlet>

<servlet-name>SearchInn</servlet-name>

<servlet-class>controller.SearchInn</servlet-class>

</servlet>

</web-app>

## 3.6 工具包tools

### 3.6.1 MybatisUtils.java

描述该文件作用，并对里面的关键代码进行解释

package tools;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.sql.Connection;

import java.sql.DriverManager;

import java.util.Properties;

public class databaseConnection {

private static final String JDBCPROPERTY="jdbc.properties";

private static String DBDRIVER = "";

private static String DBURL = "";

private static String DBUSER = "";

private static String PASSWORD = "";

private Connection conn;//数据库连接对象

static {

try{

Properties property = new Properties();

InputStream is=databaseConnection.class.getResourceAsStream("/"+JDBCPROPERTY);

property.load(new InputStreamReader(is,"UTF-8"));

is.close();

DBDRIVER=property.getProperty("DBDRIVER");

DBURL = property.getProperty("DBURL");

DBUSER = property.getProperty("DBUSER");

PASSWORD=property.getProperty("PASSWORD");

Class.forName(DBDRIVER);

}catch (Exception e){

e.printStackTrace();

}

}

public databaseConnection(){

try {

this.conn= DriverManager.getConnection(DBURL,DBUSER,PASSWORD);

}catch (Exception e){

e.printStackTrace();

}

}

public Connection getConnection(){

return this.conn;

}

public void close(){

if (this.conn!=null){

try {

this.conn.close();

}catch (Exception e){

e.printStackTrace();

}

}

}

}

## 3.8 查询界面index.jsp

网页主页面

}

}

<html>

<head>

<!-- Required meta tags -->

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<title>查询显示</title>

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

<link rel="stylesheet" href="resource/bootstrap-5.0.0-beta1-dist/css/bootstrap.css">

<script src="${pageContext.request.contextPath}js/jquery-3.5.0.min.js"></script>

<script src="${pageContext.request.contextPath}js/index.js"></script>

</head>

<body>

<div class="header">

<!--导航栏-->

<div class="nav">

<ul>

<li><a href="#">首页</a></li>

<li><a href="#">商品管理</a></li>

<li><a href="#">订单管理</a></li>

<li><a href="#">财务管理</a></li>

<li><a href="#">报关管理</a></li>

<li><a href="#">数据维护</a></li>

<li><a href="#">系统管理</a></li>

</ul>

</div>

<div class="user">

当前用户：刘颖

</div>

<div class="role">

武汉市海欢商贸有限公司:系统管理员

</div>

</div>

<div class="panel panel-default pre-scrollable">

<div class="panel-heading">

<h4>商品总览</h4>

<div class="form-group">

<span>

<input id="GID" name="GID" class="in" type="text" class="form-control " placeholder="请输入商品货号">

<input id="GBarCode" name="GBarCode" class="in" type="text" class="form-control " placeholder="请输入条形码">

</span>

<button id="" type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

所有公司

</button>

<button id="btnsearch" class="btn btn-success">查询</button>

<button id="btnclean" class="btn btn-danger">清空数据</button>

<div id="GCompany" name="GCompany" class="dropdown-menu">

<a class="dropdown-item" href="#">Link 1</a>

<a class="dropdown-item" href="#">Link 2</a>

<a class="dropdown-item" href="#">Link 3</a>

</div>

</div>

</div>

<div class="panel-body" style="overflow-y: scroll;">

<table id="sTable" name="sTable" class="table table-light" data-toggle="table" data-height="530" data-striped="true"

data-classes="table table-hover" data-toolbar="#toolbar" data-side-pagination="server" data-pagination="true"

style="height: 100px; ">

<!--caption>商品总览</caption-->

<thead style="background-color: greenyellow;">

<tr style="background-color: greenyellow;">

<th></th>

<th>商品货号</th>

<th>条形码</th>

<th>商品名称</th>

<th>一级分类</th>

<th>二级分类</th>

<th>供货价格</th>

<th>最低销售价格</th>

<th>推荐销售价格</th>

<th>商品规格</th>

<th>商品产地</th>

<th>计量单位</th>

<th>商品提供企业</th>

<th>审核状态</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

</div>

</div>

<div class="paging bottom ">

<div class="pageRecord page "> 每页显示

<select id="pageRecord">

<option selected>5</option>

<option>10</option>

<option>20</option>

<option>30</option>

</select> 条记录，共

<span id="total"></span>条数据，

<span id="pageNum">1</span>页/<span id="pageP"></span>页

</div>

<div class="navig">

<nav aria-label="Page navigation">

<ul class="pagination">

<li class="page-item disabled">

<a class="page-link" href="#" tabindex="-1" aria-disabled="true">

<< </a>

</li>

<li class="page-item disabled">

<a class="page-link" href="#" tabindex="-1" aria-disabled="true">

< </a>

</li>

<li class="page-item active" aria-current="page">

<a class="page-link" href="#"> 1 </a>

</li>

<li class="page-item">

<a class="page-link" href="#" tabindex="-1" aria-disabled="true">2</a>

</li>

<li class="page-item">

<a class="page-link" href="#">3</a>

</li>

<li class="page-item">

<a class="page-link" href="#">4</a>

</li>

<li class="page-item">

<a class="page-link" href="#">5</a>

</li>

<li class="page-item">

<a class="page-link" href="#">></a>

</li>

<li class="page-item">

<a class="page-link" href="#">>></a>

</li>

</ul>

</nav>

</div>

</div>

</body>

</html>

## 3.8 级联样式表index.css

div#navigation{

background:#0c63e4;

color: #f0f0f0;

}

.nav-link{

color: #f0f0f0;

}

li#role{

float: right;

}

h4{

background-color: gainsboro;

}

caption {

caption-side: top;

color: #000000;

background-color: gainsboro;

}

.form-control{

width: 20%;

}

.panel-body{

height: 40%;

overflow-y:scroll;

}

thead th {

background-color: aquamarine;

}

.nav {

float: left;

width: 700px;

text-align: center;

}

.user {

float: left;

padding-left: 10px;

width: 200px;

line-height: 30px;

}

.role {

float: right;

padding-right: 10px;

line-height: 30px;

}

.nav ul {

list-style: none;

}

.nav ul li {

float: left;

display: block;

height: 14px;

}

.nav ul li a {

display: block;

text-decoration: none;

color:#fff;

font-size: 13px;

text-align: center;

height: 30px;

padding: 0 18px;

line-height: 30px;

}

.header {

margin: center;

height: 30px;

width: 100%;

background-color: #418be8;

color:#fff;

font-size: 13px;

font-weight: 700;

line-height: 30px;

padding-left: 240px;

}

.in {

display: block;

float: left;

}

.bottom{

height: 100px;

margin: 10px;

float: left;

width: 100%;

}

.page {

float: left;

height: 100px;

line-height: 100px;

}

.navig {

padding: 30px;

display: inline;

float: right;

text-align: right;

}

## 3.9 index.js

主页面响应脚本

var pageRecord = null;

var pageNum = null;

var pageP=null;

function ajaxrequest(){

pageRecord=$('#pageRecord').val();

pageNum=$('#pageNum').text();

var queryParams={

GID:$('#GID').val(),

GBarCode:$('#GBarCode'),

GSupply:$('#GSupply')

};

//

var pageParams={

pageRecord:pageRecord,

pageNum:pageNum,

GID:'GID'

}

var data={

queryParams:JSON.stringify(queryParams),

pageParams:JSON.stringify(pageParams)

};

$ajax({

type:"post",

url:"SearchInn.do",

data:data,

success:function(response){

var rows = response.rows;

var total = response.total;

pageP = (Math.ceil(total/pageRecord));

$('#total').text(total);

$('#pageP').text(pageP);

$('#tbody').empty();

$each(rows,function(index,row){

var s = JSON.stringify(row);

var str="<tr data='"+s+"'>";

str=str+'<td><input type="checkbox" value='+row.GID+'></td>';

str=str+'<td>'+row.GID+'</td>';

str=str+'<td>'+row.GBarCode+'</td>';

str=str+'<td>'+row.GName+'</td>';

str=str+'<td>'+row.GClass1+'</td>';

str=str+'<td>'+row.GClass2+'</td>';

str=str+'<td>'+row.GSupply+'</td>';

str=str+'<td>'+row.GPrimPrice+'</td>';

str=str+'<td>'+row.GLowestPrice+'</td>';

str=str+'<td>'+row.GAdPrice+'</td>';

str=str+'<td>'+row.GOrigin+'</td>';

str=str+'<td>'+row.GUnit+'</td>';

str=str+'<td>'+row.GStatus+'</td>';

str=str+'</tr>';

$("tbody").append(str);

});

},

error:function(){

alert("error");

}

});

}

function clean(){

$('#GID').val();

$('#GBarCode').val();

$('#GSupply').val();

}

function back(){

if(pageNum<=1){

alert("已经到第一页了！");

}else{

$('#pageNum').text(parseInt($('#pageNum').text())-1);

ajaxrequest();

}

}

function forward(){

if(pageNum>=pageP){

alert("已经到最后一页了！");

}else{

$('#pageNum').text(parseInt($('#pageNumber').text())+1);

ajaxrequest();

}

}

$(document).ready(function(){

ajaxrequest();

$('#checkbox').change(function(){

if($('#checkbox').prop('checked')){

$('tbody tr input:checkbox').attr("checked",true);

}else{

$('tbody tr input:checkbox').attr("checked",false);

}

});

$('.page-item').click(function(){

$('.page-item').removeClass('active');

$(this).addClass('active');

})

$('#btnsearch').click(function(){

ajaxrequest();

});

$('#btnclean').click(function(){

clean();

});

$('#pageRecord').change(function(){

ajaxrequest();

});

$('#back').click(function(){

back();

});

$('forward').click(function(){

forward();

});

$('#one').click(function(){

$('#pageNum').text($(this).text());

ajaxrequest();

});

$('#two').click(function(){

$('#pageNum').text($(this).text());

ajaxrequest();

});

$('#three').click(function(){

pageNum=3;

$('#pageNum').text($(this).text());

ajaxrequest();

});

$('#four').click(function(){

pageNum=4;

$('#pageNum').text($(this).text());

ajaxrequest();

});

$('#five').click(function(){

pageNum=5;

$('#pageNum').text($(this).text());

ajaxrequest();

});

})

# 4 系统测试



# 5 总结

本次作业是我第一次使用bootstrap框架完成网页，在写前端的过程中对bootstrap的提供的组件不是很熟悉，经常不知道该用bootstrap的哪个组件，和自己使用的组件有什么功能。这些问题我通过查看官方文档、学习网站，参考博客等方式解决。对css的基础知识不熟，在调整页面时经常要花费更多的时间，通过菜鸟、w3c网站以及博客、视频等途径学习css基础知识，解决了布局和排版上的问题。在使用ajax实现表单提交时，经常出现错误信息，经过查找相关博客的方式解决。由于知识和技术上的欠缺，本网页功能较为简单，在网页基础上还可以增加更多功能和响应事件以提高使用体验。此外网页设计上也有提高空间。由于对bootstrap应用不熟，在前端页面的标签和属性不够简洁，不利于后期维护。

通过本门课程我学习了网页的前端和后端开发技术，知道了一个网站是如何实现的，我们平时经常浏览的网页功能是如何实现的。在平时的学习和作业完成的过程中，我经常遇到很多问题，通过不断的发现问题和解决问题，我实践的能力有了进步。也通过课程的学习知道了一个专业发展方向，未来会考虑向前端发展。