

**Java课程设计报告**

**题目：**\*\*\*\*

\*\*\*\*

**专业班级：**软件技术19\*\*班

**学生学号：**居中

**学生姓名：**居中

**完 成 时间：2020-5-22**

**软件技术专业Java课程设计任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学生姓名** | **\*\*\*\*\*** | **专业班级** | | **\*\*\*\*\*** | **学号** | | **\*\*\*\*\*** |
| **题目** | **药店销售管理系统** | | | | | | |
| **课题性质** | **A** | | **课题来源** | | | **D** | |
| **指导教师** | **\*\*\*\*\*** | | **同组姓名** | | | **无** | |
| **主要内容** | **面向药店销售管理人员，利用文件、数据库存储或固定的用户名、密码，验证之后进入药店销售管理系统，实现药品输入，按照条件的查询，按照条件的修改和按照条件的删除。** | | | | | | |
| **任务要求** | **（1）实现一个登陆验证，验证用户输入的用户名、密码是否正确，正确进入药店销售管理系统，不正确给出提示，并要求重新输入；**  **（2）密码验证正确后进入药店销售系统，实现对药品信息录入，按照条件的查询，按照条件的修改和按照条件的删除**  **（3）系统能够使药品信息的永久保存，可以将信息存储在数据库中，也可以存储在文件中；**  **（4）主界面可以通过控制台实现，也可以通过图形界面实现（菜单或窗口按钮）；**  **（5）设置合理的数据类型，功能符合课题逻辑。** | | | | | | |
| **参考文献** | [1]刘继承等. Java8程序设计及实验[M]. 清华大学出版社, 2018.09.  [2] <https://www.w3cschool.cn/java/> | | | | | | |
| **审查意见** | **指导教师签字：**  **2020年05月11日** | | | | | | |

目录

[1 需求分析 1](#_Toc9344803)

[2 概要设计 2](#_Toc9344804)

[2.1功能模块结构图 2](#_Toc9344805)

[2.2业务流程图 2](#_Toc9344806)

[3 运行环境 3](#_Toc9344807)

[4 开发工具和编程语言 3](#_Toc9344808)

[5 详细设计 4](#_Toc9344809)

[5.1数据库设计 4](#_Toc9344810)

[5.2登录 4](#_Toc9344811)

[5.3主界面 4](#_Toc9344812)

[5.4添加学生成绩信息 4](#_Toc9344813)

[5.5查询学生成绩信息（或按照…条件查询） 4](#_Toc9344814)

[5.6修改学生成绩信息（或按照…条件修改学生信息） 5](#_Toc9344815)

[5.7删除学生成绩信息（或按照条件删除） 5](#_Toc9344816)

[6 系统测试 5](#_Toc9344817)

[7 心得体会 5](#_Toc9344818)

[8 参考文献 6](#_Toc9344819)

## 1 需求分析

药店销售管理系统是为了帮助药店管理员信息化的管理药品，可以对药品进行增加，修改，删除以及条件查询等功能。

## 2 概要设计

### 2.1功能模块结构图

根据需求分析设计功能模块，主要功能由下图展示。

图2-1学生成绩信息管理

### 2.2业务流程图

（1）登录系统模块

系统启动后展示的是登陆页面，在登陆页面需要输入正确的用户信息，包括用户名和密码，在输入错误的时候会提示响应的错误（用户名不存在或者密码错误）

不成功

输入用户名

输入密码

成功

登录

系统主界面

图2-2登录流程图

（2）添加药品信息流程图

在列表页面点击添加药品信息进入药品信息添加页面，输入相关信息后点击保存，保存前会有相应的提示，如果有信息没有输入的话将无法保存

输入相关信息

点击保存信息

验证是否填完

否

是

验证成功后跳转列表页面

（3）查询药品信息流程图

在列表页面上方，可以输入相应的药品名称，这里实现模糊查询，只输入其中一个字就可以实现查询，查询结果直接在页面展示

输入药品名称

点击搜索按钮

列表展示查询结果

（4）修改药品信息流程图

在列表页面的操作列种点击对应的修改后，将跳转到修改页面，展示选择的药品信息，

可以直接在表单中修改相应的信息后点击保存，这里保存会验证是否已经输入完成

点击修改按钮

跳转修改页面

修改相关信息

点击保存信息

验证是否填完

否

是

返回到列表页面

（5）删除药品信息流程图

直接点击列表操作列的删除按钮，页面提示是否删除该信息，确定后将删除该信息

点击删除按钮

提示是否删除

否

误操作

是

直接删除

## 3 运行环境

（1）硬件环境:PC机内存 2G

（2）软件环境：操作系统：windowsAll

## 4 开发工具和编程语言

（1）开发环境：Eclipse,mysql（列出你所用的开发环境和数据库）；

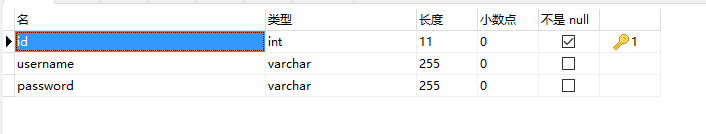
（2）编程语言：Java语言

## 5 详细设计

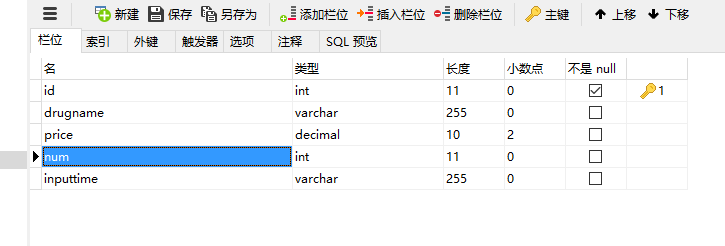
### 5.1数据库设计（存储文件设计）

（1）用户表

用户表保存登陆用户信息，用户名和密码

（2）药品信息表

保存录入的药品的信息，包括药品名称，药品的价格，药品的数量以及录入的时间

****

### 5.2类的设计

（1）用户类

三个属性id为表的主键，username是用户名，password是密码

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

（2）药品信息类

有五个字段，id为主键，drugname为药品名称，price药品的价格，num为药品的数量，inputtime为录入的时间

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

public String getDrugname() {

return drugname;

}

public void setDrugname(String drugname) {

this.drugname = drugname;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

public Integer getNum() {

return num;

}

public void setNum(Integer num) {

this.num = num;

}

public String getInputtime() {

return inputtime;

}

public void setInputtime(String inputtime) {

this.inputtime = inputtime;

}

（3）工具类（连接数据库）

package com.util;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

/\*\*

\* 数据库连接工具

\*/

public class DBUtil {

public static String db\_url = "jdbc:mysql://localhost:3306/drug";

public static String db\_user = "用户名";

public static String db\_pass = "密码";

public static Connection getConn () {

Connection conn = null;

try {

Class.forName("com.mysql.jdbc.Driver");//加载驱动

conn = DriverManager.getConnection(db\_url, db\_user, db\_pass);

} catch (Exception e) {

e.printStackTrace();

}

return conn;

}

/\*\*

\* 关闭连接

\* @param state

\* @param conn

\*/

public static void close (Statement state, Connection conn) {

if (state != null) {

try {

state.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void close (ResultSet rs, Statement state, Connection conn) {

if (rs != null) {

try {

rs.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if (state != null) {

try {

state.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if (conn != null) {

try {

conn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void main(String[] args) throws SQLException {

Connection conn = getConn();

PreparedStatement pstmt = null;

ResultSet rs = null;

String sql ="select \* from drug";

pstmt = conn.prepareStatement(sql);

rs = pstmt.executeQuery();

if(rs.next()){

System.out.println("空");

}else{

System.out.println("不空");

}

}

}

（4）工具类（列表分页）

package com.util;

/\*

@description

@anther Administrator

@creater 2020-04-28 9:34

\*/

public class PagingVO {

/\*\*

\* 分页信息 pojo类

\*/

//当前页码,默认第一页

private int curentPageNo = 1;

//总页数

private int totalCount;

//页面容量

private int pageSize=5;

//上一页

private int upPageNo;

//下一页

private int nextPageNo;

//要前往的页码,默认0

private int toPageNo = 0;

public void setToPageNo(Integer toPageNo) {

//新一页

this.toPageNo = (toPageNo-1) \* pageSize ;

//设置跳转后当前的页码

setCurentPageNo(toPageNo);

}

public Integer getTopageNo() {

return toPageNo;

}

public int getCurentPageNo() {

return curentPageNo;

}

//设置当前页码

public void setCurentPageNo(int curentPageNo) {

if (curentPageNo != 1) {

this.upPageNo = curentPageNo - 1;

}

this.nextPageNo = curentPageNo + 1;

this.curentPageNo = curentPageNo;

}

public int getTotalCount() {

return totalCount;

}

public void setTotalCount(int totalCount) {

if (totalCount%pageSize > 0) {

this.totalCount = (totalCount/pageSize)+1;

} else {

this.totalCount = totalCount/pageSize;

}

}

public int getPageSize() {

return pageSize;

}

public void setPageSize(int pageSize) {

this.pageSize = pageSize;

}

public int getUpPageNo() {

return upPageNo;

}

public void setUpPageNo(int upPageNo) {

this.upPageNo = upPageNo;

}

public int getNextPageNo() {

return nextPageNo;

}

public void setNextPageNo(int nextPageNo) {

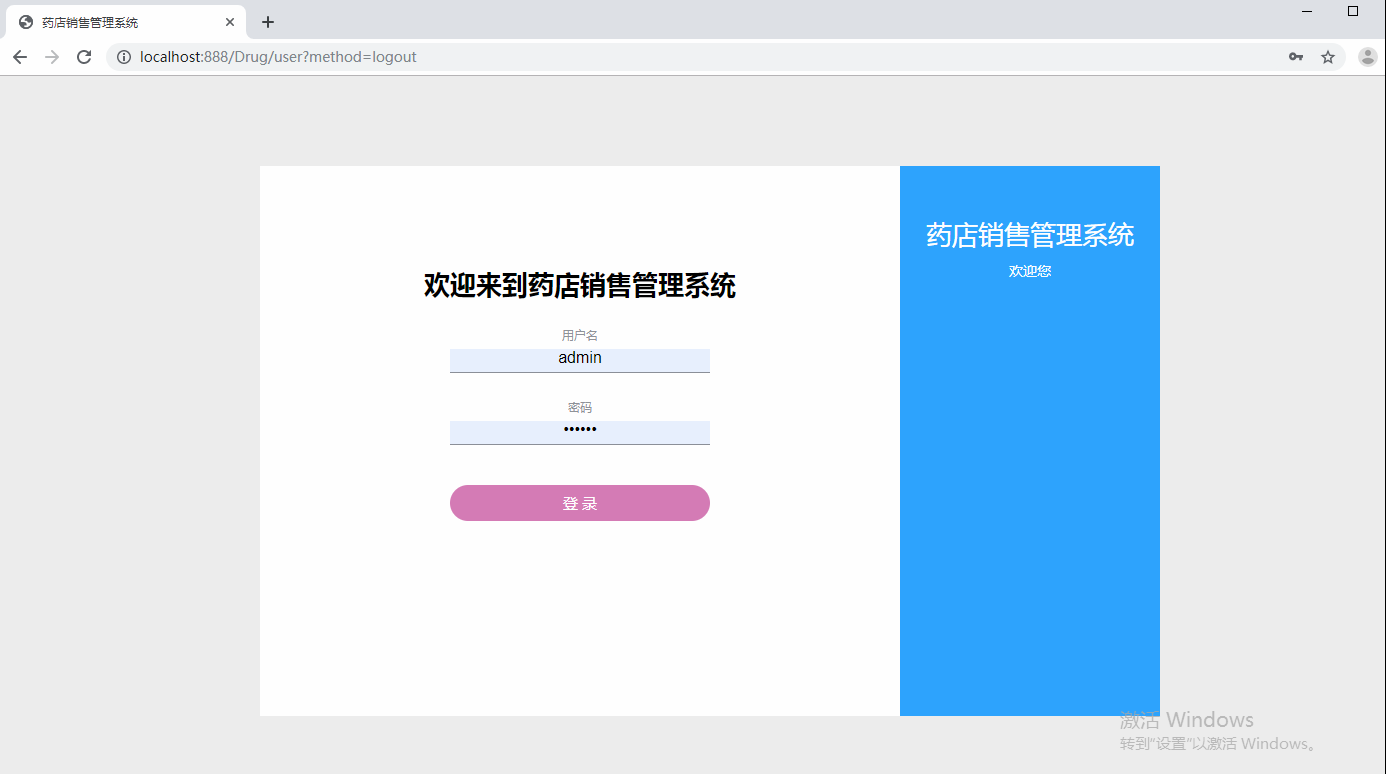
this.nextPageNo = nextPageNo;

}

}

### 5.3登录

系统启动后会加载登陆页面，输入用户名和密码在js中判断是否输入，在servlet中调用userDao中的登陆方法判断用户名是否存在以及密码是否正确



Js代码：

function login() {

if ($("#username").val() == '' || $("#password").val() == '') {

alert("请输入用户名和密码")

return false

}

$.ajax({

url:"user?method=login",

method:'post',

data:{'username':$("#username").val(),'password':$("#password").val()},

success:function(res){

if(res=='验证成功'){

//登陆成功跳转

alert(res)

window.location = "drug?method=getAll";

}else{

alert(res)

}

},

error:function (err) {

console.log(err)

}

})

return false

}

Servlet代码：

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

request.setCharacterEncoding("utf-8");// 放在第一行

response.setCharacterEncoding("utf-8");

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

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

UserDao ud = new UserDao();

switch (method) {

case "login":

//ajax请求的响应方式

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

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

response.getWriter().write(ud.login(username, password, request));

break;

case "logout":

//ajax退出

request.getSession().removeAttribute("userinfo");

request.getRequestDispatcher("index.jsp").forward(request, response);

break;

default:

break;

}

}

UserDao代码：

public String login(String username,String password, HttpServletRequest request){

String sql = "select \* from user where username='"+username+"'";

Connection conn = DBUtil.getConn();

Statement state = null;

ResultSet rs = null;

User user = new User();

Boolean flag = false;

try {

state = conn.createStatement();

rs = state.executeQuery(sql);

while(rs.next())

{

flag = true;

user.setId(rs.getInt("id"));

user.setUsername(rs.getString("username"));

user.setPassword(rs.getString("password"));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

if(!flag){

return "用户名不存在";

}else{

if(user.getPassword().equals(password)){

//存到session中

request.getSession().setAttribute("userinfo", user);

return "验证成功";

}else{

return "密码错误";

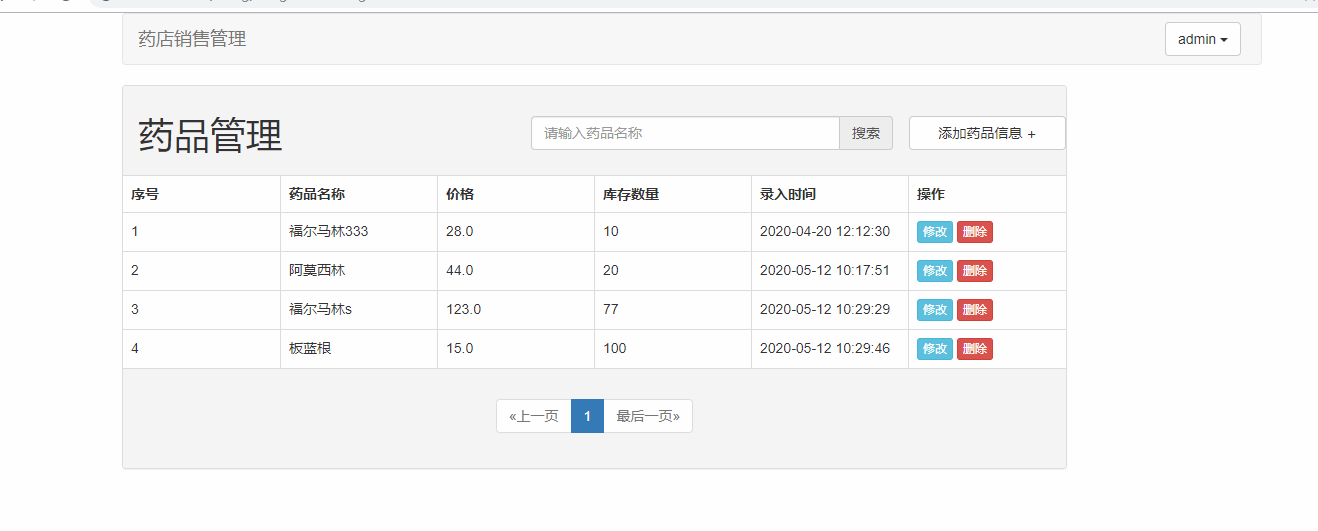
}

}

}

### 5.4主界面

登陆成功后在js中访问列表数据并在servlet中跳转到列表页面，在servlet中通过DrugDao中查询数据库所有的药品信息后保存在request作用域中，然后跳转页面，通过jsp的request作用域展示所有数据



Js代码：

window.location = "drug?method=getAll";

servlet代码：

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

request.setCharacterEncoding("utf-8");// 放在第一行

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

response.setCharacterEncoding("utf-8");

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

//根据不同的参数，调用不同的方法

DrugDao dd = new DrugDao();

switch (method) {

case "addPage":

request.getRequestDispatcher("/jsp/drug/add.jsp").forward(request, response);

break;

case "add":

add(request,response);

break;

case "editPage":

Integer id = Integer.parseInt(request.getParameter("id"));

Drug d = dd.getDrugById(id);

request.setAttribute("drug", d);

request.getRequestDispatcher("/jsp/drug/edit.jsp").forward(request, response);

break;

case "edit":

edit(request,response);

break;

case "delete":

Integer id2 = Integer.parseInt(request.getParameter("id"));

dd.deleteById(id2);

//重定向到列表

response.sendRedirect("drug?method=getAll");

break;

case "getAll":

getAll(request,response);

break;

case "findByName":

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

List<Drug> list = dd.findByField("drugname", "%"+findByName+"%");

request.setAttribute("nlist", list);

request.getRequestDispatcher("/jsp/drug/showDrug.jsp").forward(request, response);

break;

default:

break;

}

}

private void getAll(HttpServletRequest request, HttpServletResponse response) {

// TODO Auto-generated method stub

//分页

String strpage = request.getParameter("page");

Integer page = 0;

if(strpage==null){

page = 0;

}else{

page = Integer.parseInt(strpage);

}

//用户

List<Drug> vlist = null;

PagingVO pagingVO = new PagingVO();

DrugDao dd = new DrugDao();

//设置总页数

pagingVO.setTotalCount(dd.getCount());

if(page==null || page==0){

pagingVO.setToPageNo(1);

vlist = dd.getAllByPage(1);

}else{

pagingVO.setToPageNo(page);

vlist = dd.getAllByPage(page);

}

request.setAttribute("nlist",vlist);

request.setAttribute("pagingVO", pagingVO);

try {

request.getRequestDispatcher("/jsp/drug/showDrug.jsp").forward(request, response);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

DrugDao代码：

/\*\*

\* 分页查找所有

\*/

public List<Drug> getAllByPage(int toPageNo){

PagingVO pagingVO = new PagingVO();

pagingVO.setToPageNo(toPageNo);

List<Drug> list = new ArrayList<>();

String sql = "select \* from drug limit "+pagingVO.getTopageNo()+","+pagingVO.getPageSize();

Connection conn = DBUtil.getConn();

Statement state = null;

ResultSet rs = null;

try {

state = conn.createStatement();

rs = state.executeQuery(sql);

while(rs.next())

{

Drug d = new Drug();

d.setId(rs.getInt("id"));

d.setDrugname(rs.getString("drugname"));

d.setNum(rs.getInt("num"));

d.setPrice(rs.getDouble("price"));

d.setInputtime(rs.getString("inputtime"));

list.add(d);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

return list;

}

/\*\*

\* 查找个数

\*/

public int getCount(){

boolean f = false;

String sql = "select count(\*) from drug";

Connection conn = DBUtil.getConn();

Statement state = null;

ResultSet rs = null;

int a = 0;

try {

state = conn.createStatement();

rs = state.executeQuery(sql);

while(rs.next())

{

//打印的就是总记录数。把检索结果看成只有一跳记录一个字段的表

a = rs.getInt(1);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

if (a > 0) {

f = true;

}

return a;

}

### 5.5添加药品信息

在列表页面上方可以点击添加药品信息，点击后跳转到添加信息页面，在完全输入所有信息后点击保存，判断是否输入完成，成功后跳转列表页面

jsp代码：

<form class="form-horizontal" role="form" action="${pageContext.request.contextPath}/drug?method=edit" id="editfrom" method="post" onsubmit="return submit()">

<div class="form-group">

<label class="col-sm-2 control-label">药品id</label>

<div class="col-sm-10">

<input readonly="readonly" type="number" class="form-control" id="id" name="id" placeholder="" value="${drug.id}">

<input hidden name='inputtime' value='${drug.inputtime }'>

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">药品名称</label>

<div class="col-sm-10">

<input type="text" required="required" class="form-control" id="drugname" name="drugname" placeholder="请输入药品名称" value="${drug.drugname}">

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">价格</label>

<div class="col-sm-10">

<input type="number" required="required" class="form-control" id="price" name="price" placeholder="请输入价格" value="${drug.price}">

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">数量</label>

<div class="col-sm-10">

<input type="number" required="required" class="form-control" id="num" name="num" placeholder="请输入数量" value="${drug.num}">

</div>

</div>

<div class="form-group" style="text-align: center">

<button class="btn btn-default" type="submit">提交</button>

<button class="btn btn-default" type="reset">重置</button>

</div>

</form>Servlet代码：

private void add(HttpServletRequest request, HttpServletResponse response) {

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

Double price = Double.parseDouble(request.getParameter("price"));

Integer num = Integer.parseInt(request.getParameter("num"));

String inputtime = DateUtils.getCurrentTime();

Drug d = new Drug();

d.setDrugname(drugname);

d.setInputtime(inputtime);

d.setNum(num);

d.setPrice(price);

DrugDao dd = new DrugDao();

dd.insert(d);

//重定向到列表

try {

response.sendRedirect("drug?method=getAll");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

DrugDao代码：

/\*\*

\* 新增

\*/

public boolean insert(Drug drug){

boolean f = false;

String sql = "insert into drug(drugname,price,num,inputtime) values('"+drug.getDrugname()+"'"

+ ","+drug.getPrice()+","+drug.getNum()+",'"+drug.getInputtime()+"')";

Connection conn = DBUtil.getConn();

Statement state = null;

int a = 0;

try {

state = conn.createStatement();

a = state.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

if (a > 0) {

f = true;

}

return f;

}

### 5.6查询药品信息

在列表页面可以根据药品名称进行查询，查询结果会直接在列表中展示



Servlet代码：

case "findByName":

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

List<Drug> list = dd.findByField("drugname", "%"+findByName+"%");

request.setAttribute("nlist", list);

request.getRequestDispatcher("/jsp/drug/showDrug.jsp").forward(request, response);

break;

DrugDao代码：

/\*\*

\* 按照字段查询结果

\*/

public List<Drug> findByField(String field,String value){

List<Drug> list = new ArrayList<>();

String sql = "select \* from drug where "+field+" like '"+value+"'";

Connection conn = DBUtil.getConn();

Statement state = null;

ResultSet rs = null;

try {

state = conn.createStatement();

rs = state.executeQuery(sql);

while(rs.next())

{

Drug d = new Drug();

d.setId(rs.getInt("id"));

d.setDrugname(rs.getString("drugname"));

d.setNum(rs.getInt("num"));

d.setPrice(rs.getDouble("price"));

d.setInputtime(rs.getString("inputtime"));

list.add(d);

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

return list;

}

### 5.7修改药品信息

直接在列表的操作列中点击对应行的修改按钮，将通过id去请求servlet中查找到对应的药品信息后将药品信息存在jsp作用域的request中跳转到修改页面，修改药品信息保存会验证是否填写完整，验证通过后修改信息跳转到列表页面。

****

Jsp代码：

<form class="form-horizontal" role="form" action="${pageContext.request.contextPath}/drug?method=edit" id="editfrom" method="post" onsubmit="return submit()">

<div class="form-group">

<label class="col-sm-2 control-label">药品id</label>

<div class="col-sm-10">

<input readonly="readonly" type="number" class="form-control" id="id" name="id" placeholder="" value="${drug.id}">

<input hidden name='inputtime' value='${drug.inputtime }'>

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">药品名称</label>

<div class="col-sm-10">

<input type="text" required="required" class="form-control" id="drugname" name="drugname" placeholder="请输入药品名称" value="${drug.drugname}">

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">价格</label>

<div class="col-sm-10">

<input type="number" required="required" class="form-control" id="price" name="price" placeholder="请输入价格" value="${drug.price}">

</div>

</div>

<div class="form-group">

<label class="col-sm-2 control-label">数量</label>

<div class="col-sm-10">

<input type="number" required="required" class="form-control" id="num" name="num" placeholder="请输入数量" value="${drug.num}">

</div>

</div>

<div class="form-group" style="text-align: center">

<button class="btn btn-default" type="submit">提交</button>

<button class="btn btn-default" type="reset">重置</button>

</div>

</form>Servlet代码：

case "editPage":

Integer id = Integer.parseInt(request.getParameter("id"));

Drug d = dd.getDrugById(id);

request.setAttribute("drug", d);

request.getRequestDispatcher("/jsp/drug/edit.jsp").forward(request, response);

break;

private void edit(HttpServletRequest request, HttpServletResponse response) {

Integer id = Integer.parseInt(request.getParameter("id"));

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

Double price = Double.parseDouble(request.getParameter("price"));

Integer num = Integer.parseInt(request.getParameter("num"));

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

String[] field1 = {"drugname","price","num","inputtime"};

String[] field2 = {drugname,price.toString(),num.toString(),inputtime};

DrugDao dd = new DrugDao();

dd.update(field1, field2, id);

//重定向到列表

try {

response.sendRedirect("drug?method=getAll");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

DrugDao代码：

/\*\*

\* 单个通过id查询

\*/

public Drug getDrugById(Integer id){

Drug d = new Drug();

String sql = "select \* from drug where id="+id;

Connection conn = DBUtil.getConn();

Statement state = null;

ResultSet rs = null;

try {

state = conn.createStatement();

rs = state.executeQuery(sql);

while(rs.next())

{

d.setId(rs.getInt("id"));

d.setDrugname(rs.getString("drugname"));

d.setNum(rs.getInt("num"));

d.setPrice(rs.getDouble("price"));

d.setInputtime(rs.getString("inputtime"));

}

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

return d;

}

/\*\*

\* 修改\*/

public boolean update(String []strlist,String []strlist1,Integer id) {

String sql = "update drug set ";

for(int i=0;i<strlist.length;i++)

{

if(i!=strlist.length-1)

sql+=strlist[i]+"='" + strlist1[i] + "',";

else sql+=strlist[i]+"='" + strlist1[i] + "' where id=" + id + "";

}

Connection conn = DBUtil.getConn();

Statement state = null;

boolean f = false;

int a = 0;

try {

state = conn.createStatement();

a = state.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

if (a > 0) {

f = true;

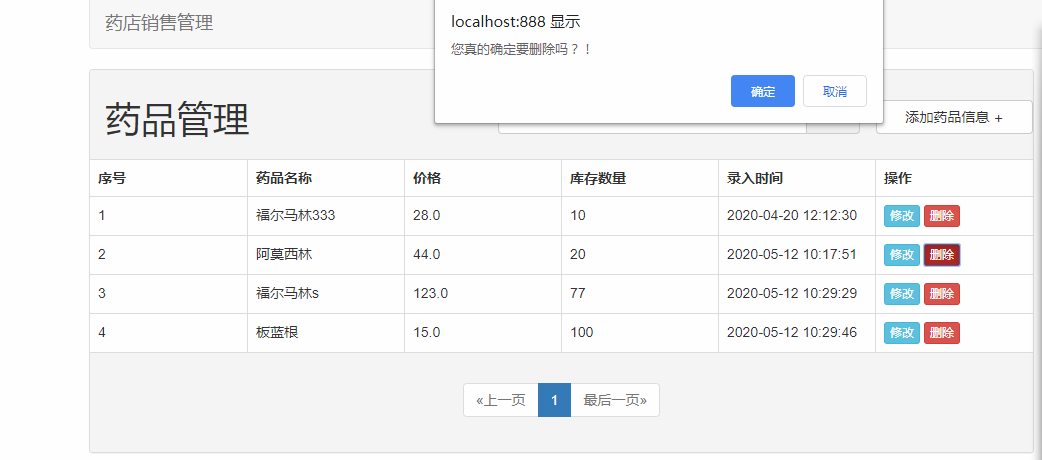
}

return f;

}

### 5.7删除药品信息（或按照条件删除）

直接在页面列表的操作列点击删除按钮，点击后在js中给出提示是否删除，防止无操作，点击确定后将删除对应id的数据后重新跳转列表页面。



Js代码：

function confirmd(id) {

var msg = "您真的确定要删除吗？！";

if (confirm(msg)==true){

location.href='drug?method=delete&id='+id;

}else{

return false;

}

}

Servlet代码:

case "delete":

Integer id2 = Integer.parseInt(request.getParameter("id"));

dd.deleteById(id2);

//重定向到列表

response.sendRedirect("drug?method=getAll");

break;

DrugDao代码：

/\*\*

\* 删除

\*

\* @return

\*/

public boolean deleteById (Integer id) {

boolean f = false;

String sql = "delete from drug where id =" + id + "";

Connection conn = DBUtil.getConn();

Statement state = null;

int a = 0;

try {

state = conn.createStatement();

a = state.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

} finally {

DBUtil.close(state, conn);

}

if (a > 0) {

f = true;

}

return f;

}

## 6系统测试

**6.1登录模块**

**（1）成功登录**

**用户名：admin**

**密码**： 111111

**结果**： 

（2）不能成功登录

**用户名：a111**

**密码**： 111111

**结果**：

(3)密码错误

用户名:admin

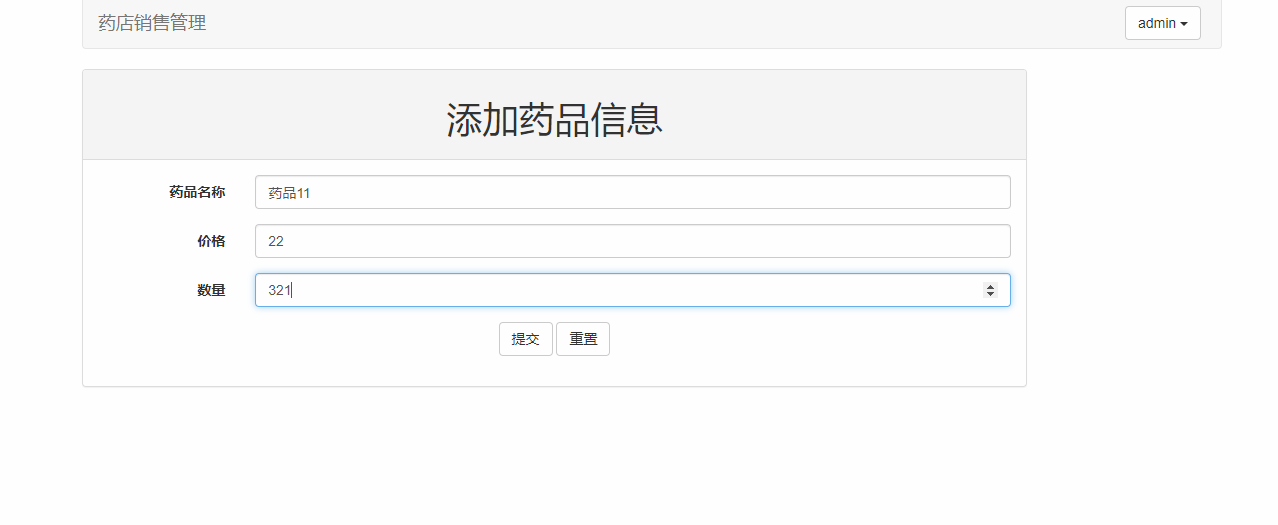
密码：123456

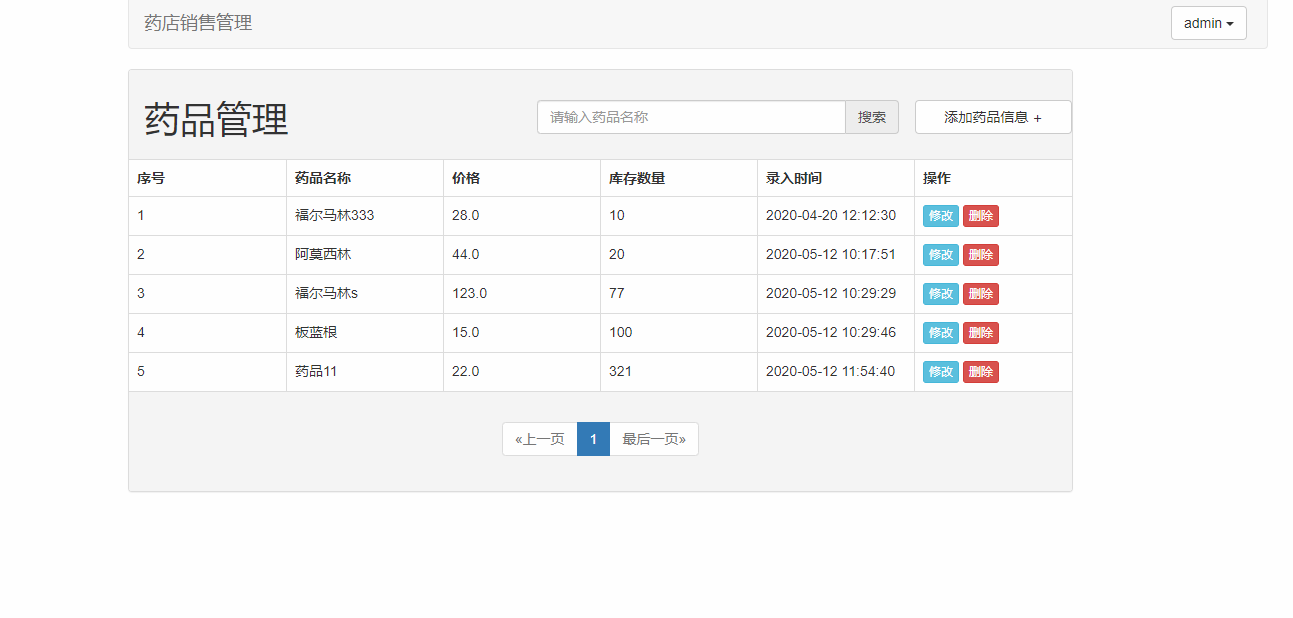
结果：

6.2 新增模块

（1）录入完全

结果：





（2）录入不完全

结果：





6.3 修改模块

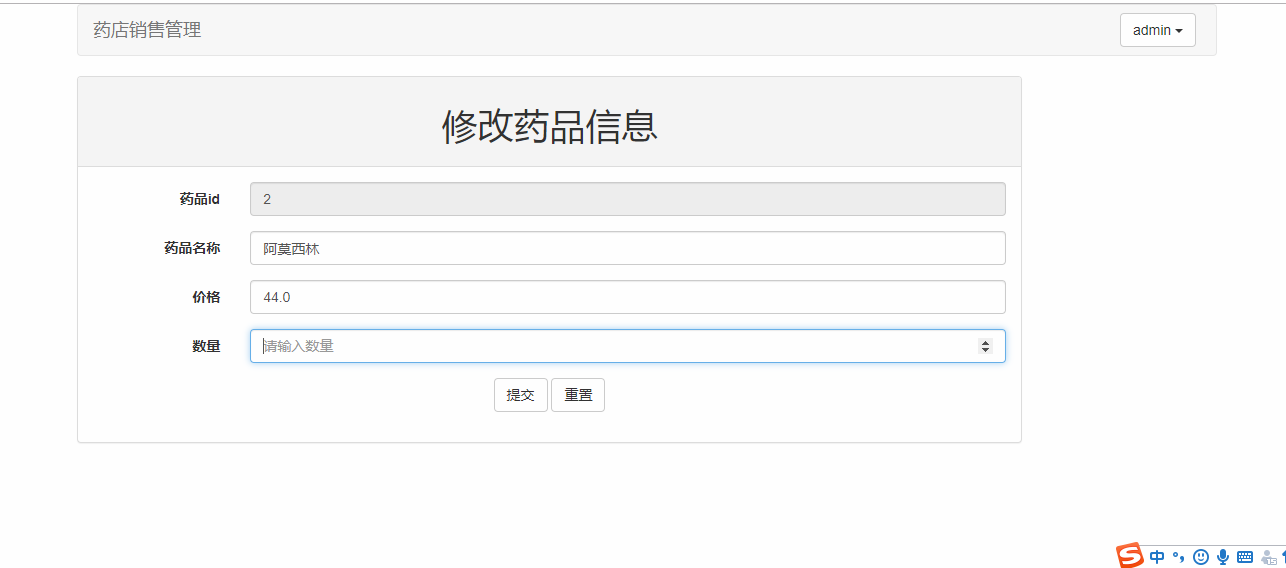
（1）录入完全

结果：



（2）录入不完全

结果：





## 7心得体会

给出本次课程设计在做系统和撰写设计报告中遇到的困难，解决方法，以及心得体会

## 8参考文献（自己修改）

[1]刘继承, 王社伟, 宋敏. Java8程序设计及实验[M]. 清华大学出版社, 2018.

[2]谷志峰, 琚伟伟. Java程序设计基础教程[M]. 电子工业出版社, 2016.

[3]王宗亮. Java程序设计任务驱动式实训教程.第2版[M]. 清华大学出版社, 2016.

[4] 王宗亮. Java程序设计任务驱动式实训教程[M]. 清华大学出版社, 2016.

**中英国际学院学院课程设计成绩评价表**

课程名称：Java课程设计

设计题目：学生成绩信息管理系统

专业：软件技术班级：1901姓名：\*\*\*学号：201\*\*\*\*\*

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 评审项目 | 分数 | 满分标准说明 |
| 1 | 内容 |  | 思路清晰；语言表达准确，概念清楚，论点正确；实验方法科学，分析归纳合理；结论严谨，设计有应用价值。任务饱满，做了大量的工作。满分20分。 |
| 2 | 创新 |  | 内容新颖，题目能反映新技术，对前人工作有改进或突破，或有独特见解。满分10分。 |
| 3 | 完整性、实用性 |  | 整体构思合理，理论依据充分，设计完整，实用性强。满分10分。 |
| 4 | 数据准确、可靠 |  | 数据准确，公式推导正确。满分10分。 |
| 5 | 规范性 |  | 设计格式、绘图、图纸、实验数据、标准的运用等符合有关标准和规定。满分10分。 |
| 6 | 纪律性 |  | 能很好的遵守各项纪律，设计过程认真。满分20分。 |
| 7 | 答辩 |  | 准备工作充分，回答问题有理论依据，基本概念清楚。主要问题回答简明准确。在规定的时间内作完报告。满分20分。 |
| 总分 |  | | |
| 综  合  意  见 | 指导教师年月日 | | |