# JAVA 课程设计任务书

## 一 题目

学生信息管理系统

## 二 目的要求

灵活应用所学Java基础知识，独立完成需求分析、系统设计和编码等系统设计全过程的综合实践能力。巩固、深化学生的理论知识，提高实践工程应用水平，并在此过程中培养了严谨的科学态度和良好的工作作风。

## 三 主要内容及技术要求

前端：

1. HTML,JS,CSS,Bootstrap

后台：

2. Java基础

3.Tomcat简单使用（Servlet）

4. Http协议的了解

5.Servlet知识

6. 数据存储：XML（Dom4j读取）

7.单元测试：Junit5

## 四 主要参考资料

## 《Java程序设计使用教程》

程 序 设 计 报 告 单

# 课程设计报告

# 功能描述

1. 登陆界面：验证账号密码是否正确，背景图片轮播
2. 操作总界面：实现增删改查
3. 增加学生信息界面：增加信息数据
4. 查找学生信息界面
   1. 根据学号查找：返回学生信息
   2. 查找全部： 返回学生信息
5. 修改学生信息界面：修改信息数据
6. 删除学生信息界面：删除信息数据

# 类的设计

前台网页：

1. AddOrUpdaTeStudent.html :添加页面
2. DeleteStudent.html：删除页面
3. Index.html：总页面
4. Left.html：左侧选择栏
5. Login.html：登陆界面
6. LookStudentOne.html：根据学号查找学生界面
7. Right.html：右侧存放操作界面
8. Top.html：头部标题界面

Css：

Style.css：实现背景图片轮播

JS:

Ready.js：获取url传递过来的数据，判断是添加操作还是更改操作，因为两个页面是一样的

后台根据传递过来的不同标识符来执行不同的操作

后台：

Servlet交互类：

AddServlet：增加和删除学生信息操作

LoginServlet：页面判断账号密码是否正确

DeleteServlet：删除学生信息操作

LookServlet：查找学生信息操作

信息存储对象类：

Student：一个对象实体的属性和方法

增删查改的实现类：

StudentOperator：操作接口

Studentsx：继承接口，实现方法

工具类：

XMLUtil读取xml和写入xml操作

单元检测类：

TestStudent：验证方法是否正确

# 代码实现

Index.html:（主界面）

<!DOCTYPE html>

<html>

<head>

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

<title>学生管理系统</title>

<!-- Bootstrap -->

<!--Bootstrap的CDN-->

<link rel=*"stylesheet"* href=*"https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css"*>

<script src=*"https://cdn.bootcss.com/jquery/2.1.1/jquery.min.js"*></script>

<script src=*"https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"*></script>

<title>教务系统</title>

</head>

<div class=*"container"*>

<frameset rows=*"15%, 85%"* >

<frame src=*"top.html"* >

<frameset cols=*"180,\*"*>

<frame src=*"left.html"*>

<frame src=*"right.html"* name=*"showframe"*>

</frameset>

<noframes>

<body>

<p>对不起，您的浏览器不支持框架！</p>

</body>

</noframes>

</frameset>

</div>

</html>

Left.html：（左侧选择栏）

<!DOCTYPE html>

<html>

<head>

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

<title>选择操作</title>

<link rel=*"stylesheet"* href=*"https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css"*>

<script src=*"https://cdn.bootcss.com/jquery/2.1.1/jquery.min.js"*></script>

<script src=*"https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"*></script>

<head>

<body>

<div class=*"container"*>

<div class=*"btn-group-vertical"*>

<button type=*"button"* class=*"btn btn-success btn-lg"* style="margin-top:*20px*;" onclick="top.showframe.location.href='AddOrUpdateStudent.html?name=Add'">增加学生信息<span class=*"badge"*></scan></button>

<div class=*"btn-group"*>

<button type=*"button"* class=*"btn btn-primary dropdown-toggle btn-lg"* data-toggle=*"dropdown"* style="margin-top:*20px*;">

查找学生信息<span class=*"caret"*></scan>

</button>

<ul class=*"dropdown-menu"* role=*"menu"*>

<li><a href=*"LookStudentOne.html"* target=*"showframe"*>根据学号查找</a></li>

<li><a href=*"/Student/LookServlet"* target=*"showframe"*>查找全部</a></li>

</ul>

</div>

<button type=*"button"* class=*"btn btn-warning btn-lg"* style="margin-top:*20px*;" onclick="top.showframe.location.href='AddOrUpdateStudent.html?name=Change'">修改学生信息</button>

<button type=*"button"* class=*"btn btn-danger btn-lg"* style="margin-top:*20px*;" onclick="top.showframe.location.href='DeleteStudent.html'">删除学生信息<span class=*"badge"*></scan></button>

</div>

</div>

</body>

</html>

AddOrUpdateStudent.html：（增加和更改的页面）

<!DOCTYPE html>

<html>

<head>

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

<link rel=*"stylesheet"* href=*"https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css"*>

<script src=*"https://cdn.bootcss.com/jquery/2.1.1/jquery.min.js"*></script>

<script src=*"https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"*></script>

<script type=*"text/javascript"* src=*"js/ready.js"*></script>

<script type=*"text/javascript"*>

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

**var** name=UrlParm.parm("name");

document.getElementById("field＿id").value=name;

};

</script>

</head>

<body>

<div class=*"container"*>

<h2>添加学生信息</h2>

<form role=*"form"* action=*"/Student/AddServlet"* method=*"Post"*>

<div class=*"form-group"*>

<label for=*"number"*>请填写学生学号:</label>

<input type=*"text"* class=*"form-control input-lg"* id=*"id"* name=*"id"* placeholder=*"学生 学号"*>

</div>

<div class=*"form-group"*>

<label for=*"name"*>请填写学生姓名:</label>

<input type=*"text"* class=*"form-control input-lg"* name=*"name"* id=*"name"* placeholder=*"学生 姓名"*>

</div>

<div class=*"form-group"*>

<label>学生性别:</label><br/>

<label class=*"radio-inline"*><input type=*"radio"* name=*"gender"* value=*"男"*/>男</label>

<label class=*"radio-inline"*><input type=*"radio"* name=*"gender"* value=*"女"*/>女</label>

</div>

<div class=*"form-group"*>

<label for=*"age"*>请填写学生年龄:</label>

<input type=*"text"* class=*"form-control input-lg"* name=*"age"* id=*"age"* placeholder=*"学生 年龄"*>

</div>

<div class=*"form-group"*>

<label for=*"phone"*>请填写联系方式:</label>

<input type=*"text"* class=*"form-control input-lg"* name=*"phone"* id=*"phone"* placeholder=*"联系 方式"*>

</div>

<div class=*"form-group"*>

<label for=*"QQ"*>请填写QQ:</label>

<input type=*"text"* class=*"form-control input-lg"* name=*"QQ"* id=*"QQ"* placeholder=*"学生 QQ"*>

</div>

<div class=*"form-group"*>

<label for=*"adress"*>请填写居住地址:</label>

<input type=*"text"* class=*"form-control input-lg"* name=*"adress"* id=*"adress"* placeholder=*"居住 地址"*>

</div>

<input type=*"hidden"* name=*"field＿name"* id=*"field＿id"* value=*"value"* />

<button type=*"submit"* class=*"btn btn-default input-lg"*>Submit</button>

</form>

</div>

</body>

</html>

DeleteStudent.html:(删除界面)

<!DOCTYPE html>

<html>

<head>

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

<link rel=*"stylesheet"* href=*"https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css"*>

<script src=*"https://cdn.bootcss.com/jquery/2.1.1/jquery.min.js"*></script>

<script src=*"https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"*></script>

<title>选择操作</title>

</head>

<body>

<div class=*"container"*>

<div class=*"panel panel-default "*>

<div class=*"panel-heading"*>

<div class=*"form-group"*>

<form role=*"form"* action=*"/Student/DeleteServlet"* method=*"Get"*>

<div class=*"form-group"*>

<label for=*"number"*>请填写学生学号:</label>

<input type=*"text"* class=*"form-control input-lg"* id=*"id"* name=*"id"* placeholder=*"学生 学号"*>

</div>

<div class=*"col-lg-5 pull-right"* align=*"right"*>

<button type=*"submit"* class=*"btn btn-default input-lg"*>Submit</button>

</div>

</form>

</div>

</div>

<!-- <div class="panel-body"><div id="demo" class="collapse in">Panel Content</div></div> -->

</div>

</body>

</body>

</html>

LookStudentOne.html:(根据学号查找学生信息界面)

<!DOCTYPE html>

<html>

<head>

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

<link rel=*"stylesheet"* href=*"https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css"*>

<script src=*"https://cdn.bootcss.com/jquery/2.1.1/jquery.min.js"*></script>

<script src=*"https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"*></script>

</head>

<body>

<div class=*"container"*>

<div class=*"panel panel-default "*>

<div class=*"panel-heading"*>

<form role=*"form"* action=*"/Student/LookServlet"* method=*"Get"*>

<div class=*"form-group"*>

<label for=*"number"*>请填写学生学号:</label>

<input type=*"text"* class=*"form-control input-lg"* id=*"id"* name=*"id"* placeholder=*"学生 学号"* />

</div>

<div class=*"col-lg-5 pull-right"* align=*"right"*>

<button type=*"submit"* class=*"btn btn-default input-lg"*>Submit</button>

</div>

</form>

</div>

<!-- <div class="panel-body"><div id="demo" class="collapse in">Panel Content</div></div> -->

</div>

</body>

</html>

AddServlet.java:(增加或者修改学生信息)

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

request.setCharacterEncoding("utf-8");

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

Student student=**new** Student();

student.setAdress(request.getParameter("adress"));

student.setAge(Integer.*parseInt*(request.getParameter("age")));

student.setGender(request.getParameter("gender"));

student.setId(request.getParameter("id"));

student.setName(request.getParameter("name"));

student.setPhone(request.getParameter("phone"));

student.setQQ(request.getParameter("QQ"));

String choise=request.getParameter("field＿name");

**if**("Add".equals(choise)) {

**new** studentsx().addStudent(student);

response.getWriter().write("<script language=javascript>alert('修改成功！');</script>");

}**else** **if**("Change".equals(choise)) {

**if**(**new** studentsx().updateStudent(student)) {

response.getWriter().write("<script language=javascript>alert('修改成功！');</script>");

}**else** {

response.getWriter().write("<script language=javascript>alert('修改失败，重新输入！');;</script>");

}

}

}

LoginServlet.html：（验证账号密码）

@Override

**protected** **void** doGet(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {

//获取服务器内置的账号密码信息

String name=**this**.getInitParameter("name");

String pwd=**this**.getInitParameter("pwd");

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

//response.setHeader("content-type", "text/html");

resp.setContentType("text/html;charset=utf-8");//和上面代码等价。推荐使用此方法

PrintWriter out = resp.getWriter();

**if**(req.getParameter("username").equals(name)&&req.getParameter("password").equals(pwd)) {

resp.getWriter().write("<script language=javascript>alert('登陆成功！');window.location = 'index.html';</script>");

}**else** {

//out.println("<script type='text/javascript'> alert('账号或者密码错误')</script>");

//resp.sendRedirect("/Student/Login.html");

//因为无法提示信息后再跳转 故不用页面跳转，sendRedirect是页面跳转

resp.getWriter().write("<script language=javascript>alert('账号或者密码错误，请重新输入');window.location = 'Login.html';</script>");

}

}

DeleteServlet.java:(删除的操作)

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

request.setCharacterEncoding("utf-8");

Student student=**new** Student();

student.setId(request.getParameter("id"));

**boolean** delete=**new** studentsx().deleteStudent(student);

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

**if**(delete) {

response.getWriter().write("<script language=javascript>alert('删除成功！');top.showframe.location.href='DeleteStudent.html';</script>");

}**else** {

response.getWriter().write("<script language=javascript>alert('删除失败！');top.showframe.location.href='DeleteStudent.html';</script>");

}

}

LookServlet.java:（查找操作）

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

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

String referer=request.getHeader("referer");

**if**(referer.contains("/Student/LookStudentOne.html")) {

Student student=**new** Student();

student.setId(request.getParameter("id"));

student=**new** studentsx().findOne(student);

response.getWriter().write(student.toString());

}**else** **if**(referer.contains("/Student/left.html")){

StringBuilder stringBuilder=**new** StringBuilder();

List<Student> list=**new** studentsx().findAll();

**for**(Student student:list) {

stringBuilder.append(student.toString()).append("<br />");

}

response.getWriter().write(stringBuilder.toString());

}

}

Student.java:(学生信息个体)

**public** **class** Student {

**private** String id;

**private** String name;

**private** String gender;

**private** **int** age;

**private** String phone;

**private** String QQ;

**private** String adress;

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** String getPhone() {

**return** phone;

}

**public** **void** setPhone(String phone) {

**this**.phone = phone;

}

**public** String getQQ() {

**return** QQ;

}

**public** **void** setQQ(String qQ) {

QQ = qQ;

}

**public** String getAdress() {

**return** adress;

}

**public** **void** setAdress(String adress) {

**this**.adress = adress;

}

@Override

**public** String toString() {

**return** "Student [id=" + id + ", name=" + name + ", gender="

+ gender + ", age=" + age + ", phone=" + phone + ", QQ=" + QQ

+ ", adress=" + adress + "]";

}

}

StudentOperator.java:(操作数据接口)

**public** **interface** StudentOperator {

**public** **void** addStudent(Student student); //增加联系人

**public** **boolean** updateStudent(Student student); //修改联系人

**public** **boolean** deleteStudent(Student student); //根据ID删除联系人

**public** List<Student> findAll(); //查询所有联系人

**public** Student findOne(Student student);

}

Studentsx.java：（数据操作的实现类）

package Student;

import static org.junit.jupiter.api.Assertions.assertFalse;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import org.dom4j.Document;

import org.dom4j.DocumentHelper;

import org.dom4j.Element;

import org.dom4j.Node;

public class studentsx implements StudentOperator {

@Override

public void addStudent(Student student) {

try

{

File file=new File("D:/Students.xml");

Document doc=null;

Element rootElem=null;

if(!file.exists()) {

doc=DocumentHelper.createDocument();

rootElem=doc.addElement("Students");

}else {

doc=XMLUtil.getDocument();

rootElem=doc.getRootElement();

}

Element studentElement=rootElem.addElement("Student");

studentElement.addAttribute("id", student.getId());

studentElement.addElement("name").setText(student.getName());

studentElement.addElement("gender").setText(student.getGender());

studentElement.addElement("age").setText(student.getId());

studentElement.addElement("phone").setText(student.getPhone());

studentElement.addElement("QQ").setText(student.getQQ());

studentElement.addElement("adress").setText(student.getAdress());

XMLUtil.writeXML(doc);

}catch (Exception e) {

e.printStackTrace();

throw new RuntimeException(e);

}

}

@Override

public boolean updateStudent(Student student) {

Document doc=XMLUtil.getDocument();

Element studentOne=(Element) doc.selectSingleNode("//Student[@id='"+student.getId()+"']");

if(studentOne==null) {

return false;

}else {

studentOne.element("name").setText(student.getName());

studentOne.element("gender").setText(student.getGender());

studentOne.element("age").setText(student.getAge()+"");

studentOne.element("phone").setText(student.getPhone());

studentOne.element("QQ").setText(student.getQQ());

studentOne.element("adress").setText(student.getAdress());

try {

XMLUtil.writeXML(doc);

} catch (Exception e) {

e.printStackTrace();

throw new RuntimeException(e);

}

}

return true;

}

@Override

public boolean deleteStudent(Student student) {

Document doc=XMLUtil.getDocument();

Element element=(Element) doc.selectSingleNode("//Student[@id='"+student.getId()+"']");

if(element==null) {

return false;

}else {

element.detach();

try {

XMLUtil.writeXML(doc);

return true;

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

throw new RuntimeException();

}

}

}

@Override

public List<Student> findAll() {

// TODO Auto-generated method stub

Document document=XMLUtil.getDocument();

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

List<Node> stuList = document.selectNodes("//Student");

for(Node node :stuList) {

Element element=(Element)node;

Student student=new Student();

setStudent(student, element);

list.add(student);

}

return list;

}

@Override

public Student findOne(Student student) {

Document document=XMLUtil.getDocument();

Element studentOne=(Element) document.selectSingleNode("//Student[@id='"+student.getId()+"']");

if(studentOne==null) {

return student;

}

setStudent(student, studentOne);

return student;

}

private void setStudent(Student student, Element studentOne) {

student.setAdress(studentOne.elementText("adress"));

student.setAge(Integer.parseInt(studentOne.elementText("age")));

student.setGender(studentOne.elementText("gender"));

student.setId(studentOne.attributeValue("id"));

student.setName(studentOne.elementText("name"));

student.setPhone(studentOne.elementText("phone"));

student.setQQ(studentOne.elementText("QQ"));

}

}

XMLUtil.java:(XML文件的读取和写入)

package Student;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import org.dom4j.Document;

import org.dom4j.DocumentException;

import org.dom4j.io.OutputFormat;

import org.dom4j.io.SAXReader;

import org.dom4j.io.XMLWriter;

public class XMLUtil {

//读取xml文件

public static Document getDocument(){

try {

Document document=new SAXReader().read(new File("D:/Students.xml"));

return document;

} catch (DocumentException e) {

// TODO Auto-generated catch block

e.printStackTrace();

throw new RuntimeException(e);

}

}

public static void writeXML(Document doc) throws IOException {

try {

FileOutputStream fileOutputStream=new FileOutputStream("D:/Students.xml");

OutputFormat format=OutputFormat.createPrettyPrint();

format.setEncoding("utf-8");

XMLWriter writer = new XMLWriter(fileOutputStream,format);

writer.write(doc);

writer.close();

} catch (UnsupportedEncodingException e) {

e.printStackTrace();

throw new RuntimeException(e);

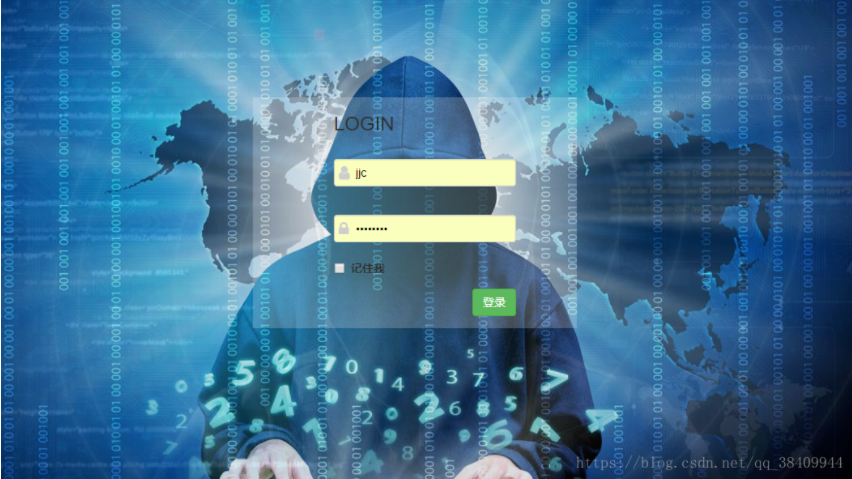
}

}

}

# 效果展示

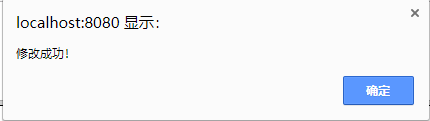
登陆界面：



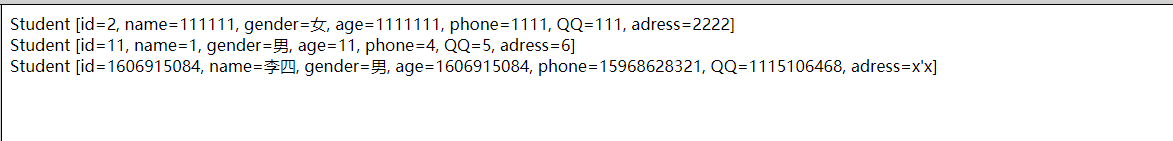
操作界面：



添加后显示：



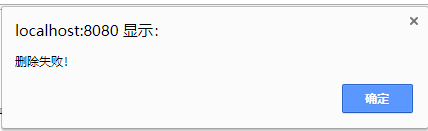
查找后的数据显示：



删除数据成功后：

# IJ{83%J}JXU19]P]J$DU@7W

删除失败后：



# 小结

最近期末的Java大作业大部分都是要Gui（Swing技术） 书本教的也是 Swing技术，但是那界面实在是不忍直视，如果真的要做桌面应用之类的 根本轮不到Java的Swing，所以我觉得还是不要让Java干他不擅长的事 比较好，刚好最近学了Tomcat就 决定做一个简单的学生管理系统，目的无非是为了完成任务还有巩固一下 Eclipse中配置Tomcat JavaWeb项目。

需要注意的是 单元检测是真的好用，可以局部测试，可以写一个类测试一下看是否正确而不用跑整个项目来检测这么一个类。

做完了之后感觉真的是简陋无疑缺少了很多功能，不过这几天碰到的坑坑洼洼还是让我 受益匪浅的。