摘 要

随着科技的进步，互联网在当今这个社会发展迅猛，每个行业之间都在运用网络技术，随着网络的发展，人们的生活状态相比较以往有着巨大的改变。在这个信息爆炸的时代，互联网的迅速发展，同学之间基本都在用即时聊天工具和电子邮件日常通讯。信息化的普及让同学之间联系更加紧密，建立同学信息库，实现同学之间的信息化。

同学录管理系统开发工具用的是MyEcalipse，它是一个能够在网上登录注册的一个网站，还可以使用他的一系列功能。目标是实现两个模块，一个是管理员模块，一个是学生模块。管理员模块有大致以下的几个功能：更换密码，学校、学生、班级、照片以及论坛等信息进行管理。学生模块主要功能有：在线注册登录、个人信息管理、创建班级信息、加入班级以及班级信息管理功能。其中学生部分的对所在班级对它的里面内容的管理，包括：班级论坛、照片这些内容的添加以及班级里同学的信息查看。

关键词：同学录管理系统；B/S；JSP；MySQL数据库

Abstract

With the progress of science and technology, the Internet is developing rapidly in this society, the use of network technology in every industry, with the development of the network, people's living conditions have changed greatly compared with the past.In this era of information explosion, the rapid development of the Internet, students are basically using instant messaging tools and E-mail daily communication.The popularity of information technology to make students more closely linked to the establishment of student information database, the realization of information between students.

Student records management system development tool is MyEcalipse, it is a sign on the Internet to register a website, you can also use a series of functions. the goal is to achieve two modules, one is the administrator module, one is the student module.The administrator module has the following functions: change passwords, schools, students, classes, photos and forums and other information management.The main functions of the student module are: online registration login, personal information management, the creation of class information, to join the class and class information management functions.Some of the students in the class on the contents of its internal management, including: class forums, photos of the contents of the class and the students to add information to view.

**Keywords:** Classmates; B/S; JSP;MySql database

目 录

1 引言 7

1.1 课题研究意义及背景 7

1.2 论文的目的及设计要求 7

1.3 研究现状及设计目标 8

1.3.1 基于Java的同学录现状 8

1.3.2 基于Java的同学录优缺点分析 8

1.4 现行研究存在的问题及解决方法 9

1.4.1 现行研究存在的问题 9

1.4.1 解决办法 9

1.5 开发环境 9

2 系统分析 10

2.1 需求分析 10

2.1.1 功能需求分析 10

2.1.2 技术可行性分析 11

2.1.3 性能需求分析 11

2.2 系统业务主要流程 11

3 系统设计 13

3.1 系统体系结构 13

3.2 系统功能结构 13

3.3 数据库表设计 14

4 系统实现 20

4.1 系统主要模块的实现 20

4.1.1 系统登录功能界面 20

4.1.2 修改登录密码功能界面 22

4.1.3 学生注册功能界面 23

4.1.4 学校信息管理功能界面 24

4.1.5 注册学生信息管理功能界面 25

4.1.6 班级信息管理功能界面 25

4.1.7 照片信息管理功能界面 26

4.1.8 论坛信息管理功能界面 26

4.1.9 学生管理个人信息功能界面 27

4.1.10 学生创建班级功能界面 27

4.1.11 学生管理创建班级信息功能界面 29

4.1.12 学生管理班级论坛信息功能界面 29

4.1.13 学生管理班级照片信息功能界面 31

4.1.14 学生查看班级通讯录功能界面 33

5 系统测试 20

5.1 实例测试的研究与选择 34

5.2 实例测试表格 34

6 结论与展望 36

参考文献 37

致 谢 38

1. 引言
   1. 课题研究意义及背景

当今世界，人们生活的方方面面都渗透着信息化产业 ，中国经济的迅速发展，信息产业也发展的很迅猛。科技的不断创新，使中国信息行业成为支柱产业。中国经济水平的提高，人们的文化水平也在提高。对待新事物的接受能力更快了，网络小到小孩，大到老人都在使用。由于这个行业发展的迅猛，竞争是不可避免的。人工管理模式基本被取代，大部分公司现今运用的都是信息管理模式，效率是人工模式不可比拟的。

在这个互联网时代，计算机能够辅助人们完成很多自身那已完成任务。计算机能够快速高效的处理数据，并且确保它的准确度，把人员的效率得到巨大提高，由于计算机的参与，错误率也降低了很多。学生对同学录的管理也应该是用相应的系统，所以很有必要开发个这样的系统，这篇论文就是介绍基于Java的同学录的设计开发过程。它能使学生同学之间的联系通过信息化的管理，更加快捷、高效。这个系统能够系统化、规范化学生信息，提高同学之间的沟通效率。

* 1. 论文的目的及设计要求

基于Java的同学录信息管理系统主要的目的就是把学生的同学信息规范化管理。设计思想为：管理员登录系统时与数据库的信息比对成功，登录管理员系统，进入的页面可以查看或删除学生信息、查看或删除班级信息、查看或删除学生信息、照片信息以及空间信息的查看或删除，他能够对以下信息进行合理的管控，学生加入或新建班级、通过班级的论坛进行交流等流程。

本设计将系统分为两个模块：管理员模块可以管理学校信息，对它增加或删除，删除图片信息，删除空间信息。学生模块则能够加入或新建班级、对班级的空间信息、照片信息和通讯信息进行查看。

* 1. 开发环境

操作系统：Windows 7

开发语言：JSP、Java语言

数据库：MySQL

WEB服务器：Tomcat 7

开发工具：MyEclipse

客户端浏览器：IE

1. 系统分析
   1. 需求分析
      1. 功能需求分析
2. 登录认证：为了能够保证登录同学录系统信息的安全和信息的一致，登录时用户信息与数据库中的信息比对，如果信息一致，就会进入成功，如果比对不正确，将会返回错误提示，重新输入用户信息。

2、学校管理功能：此功能是一个系统管理员模式下的功能，管理员可以添加学校信息并传入数据库，也可以删除数据库中的学校的信息。

3、学生管理功能：此功能是一个系统管理员模式下的功能，其功能类似于学校管理功能，即管理所有的学生信息，包括查看和删除学生信息。

4、班级管理功能：此功能是一个管理员功能，能够查看系统所有班级信息，还可以对当前系统所有的班级进行删除。

5、照片管理功能：此功能是一个系统管理员模式下功能，能够查看所有的班级照片信息，还可以删除照片信息。

6、论坛管理功能：此功能是一个系统管理员模式下功能，管理员可以查看或者清除选中论坛信息。

7、创建班级功能：此功能是一个学生功能，学生可以创建一个以前同学没创建过的班级。

8、班级管理功能：此功能是一个系统学生模式下功能，学生可以对自己的班级信息查看，包括管理帖子信息查看、照片信息的查看和班级通讯录的查看。

* + 1. 技术可行性分析

该同学录管理系统使用开发软件是MyEcalipse和MySql数据库，在开发是运用了Spring+Struts+Hibernate(SSH)框架和JSP等技术，进行的开发。还运用了MVC设计模式，提高了程序开发效率，在程序发现问题时容易维护，而且拓展其它功能时，更加方便，而且利用控制器将模型与视图分离减少了它们之间的耦合性。

* + 1. 性能需求分析

1. 系统处理能力：同学录管理系统涉及到诸多功能的操作。不同的用户可能会同时操作，很多人一起操作的时候，多人一起操作的时候，系统处理过程比较繁杂，所以并发能力要提上去，不然系统会出问题。
2. 稳定性：这个系统是一个小型系统，但是还是要保证整个程序运行时安全性和稳定性，而且在用户发出请求是，服务器快速反应，需要配置比较高的系统来运行。
3. 处理时间：理想状态应该是24h为学生提供服务。
   1. 系统主要业务流程

在明确了一个系统的业务流程，能够使一个系统有良性的开发流程。基于Java的同学录主要的业务就是实现管理员发布或维护学校信息、班级信息、学生信息、论坛信息以及照片信息，学生的在线注册登录、创建班级以及对自己班级的论坛、照片和通讯信息进行管理。本系统的业务流程如下：



基于Java的同学录流程图

1. 系统设计
   1. 系统体系结构

基于Java的同学录管理系统运用浏览器和服务器（B/S）的体系结构，在浏览器上，实现同学录的界面。B/S结构简单有效，不需要实现复杂的软件，就能够实现强大的功能，不仅节约了资源，还能够减少成本的投入。通过浏览器发出请求，请求传到前端接收了请求以后，然后将请求传给数据库，数据库中将数据进行比对，然后将结果返回，最后在浏览器上面把返回的信息显示给用户。

* 1. 系统功能结构

本系统将分为两种登录状态，一种是管理员登录，实现对程序内部的整体内容进行管控。另一种是学生登录，对系统内部功能的使用。登录时会在数据库对身份进行比对，比对成功让用户正确的登录，合理使用所拥有的功能，功能如下：

1．系统管理员：

a、学生管理功能，通过对数据库学生信息表数据的读取，显示学生信息，也能对数据库中的信息进行删除。

b、学校管理功能，通过对数据库里面的管理员添加的学校的信息表数据的读取，可以对学校的信息在数据库中进行查看和增删。

c、班级管理功能，通过对数据库所有学生创建班级数据的读取，查找班级，还可以进行增删操作。

d、照片管理功能，通过对数据库中所有学生发布照片数据的读取，能够查看照片信息，还能删除照片信息。

e、论坛管理功能，通过对数据库中所有学生发布的帖子数据进行读取，可以查看论坛信息，还可以进行删除。

2．学生

a、注册登录功能，学生必须先注册，把学生信息保存的数据库，才能实现登陆。

b、创建班级功能，学生进入系统后建立一个以前所在班，录入到数据库中。

c、班级管理功能，对本人创建的班级信息管理，包括其他人发表上去帖子，照片，还有班级通讯录。

如图3.1为系统功能结构图。



如图3.1系统功能结构图

* 1. 数据库表设计

根据对同学录系统的整体的分析，基于Java的同学录的表有系统管理员、学校、学生、班级、照片、加入班级、论坛帖子、回复以及留言的这些信息表，各个表结构描述如下：

1、系统管理员表t\_admin

如下表3.1：

* + - * 1. 管理员信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| userId | int | 30 | Y | Y | 管理员ID |
| userName | Varchar2 | 30 | N | Y | 用户名 |
| userPw | Varchar2 | 30 | N | Y | 密码 |

2、院校信息表t\_yuanxiao

如下表3.2：

* + - * 1. 院校信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 院校ID |
| chinaProvince | Varchar2 | 30 | N | Y | 省份 |
| chinaCity | Varchar2 | 30 | N | Y | 城市 |
| mingcheng | Varchar2 | 30 | N | Y | 名称 |

3、班级信息表t\_banji

如下表3.3：

* + - * 1. 班级信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 班级ID |
| yuanxiaoId | int | 30 | N | Y | 院校ID |
| chinaProvince | Varchar2 | 30 | N | Y | 省份 |
| chinaCity | Varchar2 | 30 | N | Y | 城市 |
| mingcheng | Varchar2 | 30 | N | Y | 名称 |
| chuangjianUserId | int | 30 | N | Y | 学生ID |
| chuangjianshi | Date | 30 | N | Y | 创建时间 |

4、学生信息表t\_xuesheng

如表3.4：

* + - * 1. 学生信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 学生ID |
| xuehao | Varchar2 | 30 | N | Y | 学号 |
| xingming | Varchar2 | 30 | N | Y | 姓名 |
| xingbie | Varchar2 | 30 | N | Y | 性别 |
| nianling | int | 30 | N | Y | 年龄 |
| loginname | Varchar2 | 30 | N | Y | 帐号 |
| loginpw | Varchar2 | 30 | N | Y | 密码 |
| xiandizhi | Varchar2 | 30 | N | Y | 地址 |
| xiandanwei | Varchar2 | 30 | N | Y | 单位 |
| xiandianhua | Varchar2 | 30 | N | Y | 电话 |

5、留言信息表t\_liuyan

如表3.5：

* + - * 1. 留言信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| liuyan\_id | int | 30 | Y | Y | 留言ID |
| liuyan\_content | Varchar2 | 30 | N | Y | 内容 |
| liuyan\_date | Varchar2 | 30 | N | Y | 留言时间 |
| banjiId | int | 30 | N | Y | 班级ID |
| liuyan\_user | int | 30 | N | Y | 留言用户 |

6、照片信息表t\_zhaopian

如表3.6：

* + - * 1. 照片信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 照片ID |
| title | Varchar2 | 50 | N | Y | 标题 |
| fujian | Varchar2 | 50 | N | Y | 附件 |
| shijian | Varchar2 | 50 | N | Y | 时间 |
| shangchuanzhe | Varchar2 | 50 | N | Y | 上传用户 |
| banjiId | int | 50 | N | Y | 班级ID |

7、论坛帖子信息表t\_zhuti

如表3.7：

* + - * 1. 论坛帖子信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 论坛帖子ID |
| title | Varchar2 | 30 | N | Y | 标题 |
| content | Varchar2 | 30 | N | Y | 内容 |
| fujian | Varchar2 | 50 | N | Y | 附件 |
| fujianYuanshiming | Varchar2 | 50 | N | Y | 附件原始名 |
| user\_id | int | 20 | N | Y | 学生ID |
| banji\_id | int | 20 | N | Y | 班级ID |
| shijian | Date | 20 | N | Y | 发布时间 |
| shifouding | Varchar2 | 20 | N | Y | 是否置顶 |
| shifoujing | Varchar2 | 20 | N | Y | 是否精华 |

8、回复信息表t\_huifu

如表3.8：

* + - * 1. 回复信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 回复ID |
| user\_id | int | 30 | N | Y | 学生ID |
| zhuti\_id | int | 30 | N | Y | 主题ID |
| title | Varchar2 | 30 | N | N | 标题 |
| content | Varchar2 | 30 | N | N | 内容 |
| shijian | Varchar2 | 30 | N | N | 时间 |

9、加入的班级信息表t\_jiaru

如表3.9：

* + - * 1. 加入班级信息表

| 列名 | 类型 | 长度 | 主键 | 非空 | 说明 |
| --- | --- | --- | --- | --- | --- |
| id | int | 30 | Y | Y | 加入班级ID |
| userId | int | 30 | N | Y | 学生ID |
| banjiId | int | 20 | N | Y | 班级ID |

1. 系统实现及测试
   1. 系统登录功能模块

本系统的首页是登陆界面，登录界面输入用户信息，将在数据库中进行对比，查看用户的信息是否正确，只有输入的正确，比对成功才可以进入，如果用户账号或密码错误，那就进不去还会有错误提醒返出来，登录会失败。

角色判断代码：

if(userType==0){

String sql="from TAdmin where userName=? and userPw=?";

Object[] con={userName,userPw};

List adminList=adminDAO.getHibernateTemplate().find(sql,con);

if(adminList.size()==0){

result="no";

}

else{

WebContext ctx = WebContextFactory.get();

HttpSession session=ctx.getSession();

TAdmin admin=(TAdmin)adminList.get(0);

session.setAttribute("userType", 0);

session.setAttribute("admin", admin);

result="yes";

}

}

if(userType==1){

String sql="from TUser where loginname=? and loginpw=? and del='no'";

Object[] con={userName,userPw};

List userList=userDAO.getHibernateTemplate().find(sql,con);

if(userList.size()==0){

result="no";

}

else{

WebContext ctx = WebContextFactory.get();

HttpSession session=ctx.getSession();

TUser user=(TUser)userList.get(0);

session.setAttribute("userType", 1);

session.setAttribute("user", user);

result="yes";

}

}

其实现页面如图4.1所示。



系统登录界面

* 1. 修改登录密码功能模块

管理员信息通过与数据库的验证进入系统之后，点击进入改密码的按键，系统跳转到改密码子的界面。在这里必须知道以前的密码，才能够添加全新的密码，然后才能把密码修改成功。

主要代码如下：

function check(){

var userPwReal="${sessionScope.admin.userPw}";

if(document.formPw.userPw.value !=userPwReal){

alert("原密码不正确");

return ;

}

if(document.formPw.userPw1.value ==""){

alert("新密码不能空");

return ;

}

document.getElementById("indicator").style.display="block";

loginService.adminPwEdit(document.formPw.userPw1.value,callback);

}

function callback(data){

document.getElementById("indicator").style.display="none";

alert("修改成功");

}

其实现页面如图6.2所示。



修改登录密码界面

* 1. 学生注册功能模块

在登录页面点击“注册”按钮，可以打开学生注册信息填写页面，在这里需要填写或选择学生学号、姓名、性别、年龄以及密码信息，如果这里的一定写的选项没有写如信息，会出现一个提示。

代码入下：

function check(){

if(document.formAdd.xuehao.value==""){

alert("请输入学号");

return false;

}

if(document.formAdd.xingming.value==""){

alert("请输入姓名");

return false;

}

if(document.formAdd.loginname.value==""){

alert("请输入账号");

return false;

}

document.formAdd.submit();

}

其实现页面如图6.3所示。



学生注册界面

* 1. 学校信息管理功能模块

管理员点击左侧菜单“查询学校信息”可以根据学校所在地和名称进行查询，在结果页面中，可以把所有学校的信息一列一列依次排下去，然后管理员就能够选择实现确定的的学校，把这个以前添加的学校删除了。如下图4.4以及4.5所示。

代码入下：

public String yuanxiaoRes(){

System.out.println(chinaProvince+"&&&"+chinaCity);

StringBuffer sql=new StringBuffer("from TYuanxiao where del='no'");

if(chinaProvince.equals("选择")==false){

sql.append(" and chinaProvince like '%"+chinaProvince.trim()+"%'");

}

if(chinaCity.equals("选择")==false){

sql.append(" and chinaCity like '%"+chinaCity.trim()+"%'");

}

sql.append(" and mingcheng like '%"+mingcheng.trim()+"%'");

sql.append(" order by chinaProvince,chinaCity");

List yuanxiaoList=yuanxiaoDAO.getHibernateTemplate().find(sql.toString());

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("yuanxiaoList", yuanxiaoList);

return ActionSupport.SUCCESS;

}



学校查询界面

代码入下：

public String yuanxiaoDel(){

TYuanxiao yuanxiao=yuanxiaoDAO.findById(id);

yuanxiao.setDel("yes");

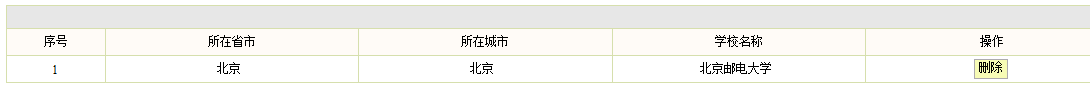
yuanxiaoDAO.attachDirty(yuanxiao);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg", "信息删除成功");

return "msg";

}



学校查询结果和学校管理界面

假如必要增加学校信息，可以点击学校添加按钮跳转学校添加页面，在这里需要输入学校信息的编号、姓名、选择性别、年龄以及密码等信息，如果这里的一定写的选项没有写如信息，会出现一个提示。如下图4.6以及4.7所示。

代码入下：

public String yuanxiaoAdd(){

TYuanxiao yuanxiao=new TYuanxiao();

yuanxiao.setChinaProvince(chinaProvince);

yuanxiao.setChinaCity(chinaCity);

yuanxiao.setMingcheng(mingcheng);

yuanxiao.setDel("no");

yuanxiaoDAO.save(yuanxiao);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg", "信息添加成功");

return "msg";

}



学校添加界面



学校非空提示界面

* 1. 注册学生信息管理功能模块

在管理员界面的学生管理页面之中，会把所有学生他们的信息一列一列的显示出来，清楚的呈现出所有的学生信息，其中学生信息包括学号、姓名、性别、年龄、帐号、密码、现住址、工作单位以及联系方式等信息，管理员如果有必要，可以选择不合理的学生信息，然后把他们清除掉。其实现页面如图4.8所示。

代码入下：

public String tongxunluAll(){

System.out.println(banjiId+"HHH");

String sql="from TJiaru where banjiId="+banjiId;

List jiaruList=jiaruDAO.getHibernateTemplate().find(sql);

for(int i=0;i<jiaruList.size();i++){

TJiaru jiaru=(TJiaru)jiaruList.get(i);

jiaru.setUser(userDAO.findById(jiaru.getUserId()));

}

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("jiaruList", jiaruList);

return ActionSupport.SUCCESS;

}



学生管理界面

* + 1. 班级信息管理功能界面

在管理员界面的班级管理页面之中，会把所有班级信息一列一列的显示出来，清楚的呈现出所有的班级信息，并可以选择指定的班级，然后把这个指定的进行清除。其实现页面如图4.9所示。

代码入下：

public String banjiDel(){

TBanji banji=banjiDAO.findById(id);

banjiDAO.delete(banji);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg", "信息删除成功");

return "msg";

}



班级信息管理界面

* + 1. 照片信息管理功能界面

管理员点击菜单栏里照片信息管理按钮，能够显示出所有的照片信息，在照片管理页面中，会把所有照片信息一列一列的显示出来，清楚的呈现出所有的照片信息，并可以把指定的照片清除掉。如图4.10所示。

代码入下：

public String zhaopianDel(){

TZhaopian zhaopian=zhaopianDAO.findById(id);

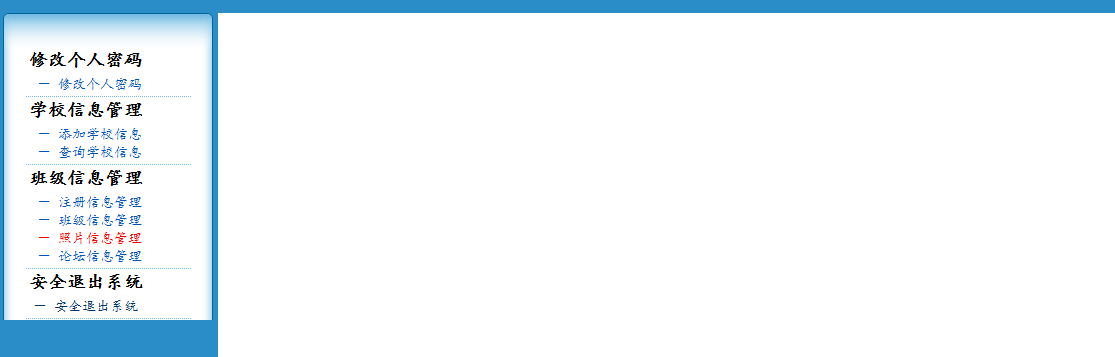
zhaopianDAO.delete(zhaopian);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg", "信息删除完毕");

return "msg";

}



照片信息管理界面

* + 1. 论坛信息管理功能界面

管理员点击菜单栏里的论坛信息管理按钮，能够显示出所有的帖子信息，在论坛管理页面中，会把所有的帖子信息一列一列的显示出来，清楚的呈现出所有的帖子，点击信息进入论坛主题详细页面可以进行附件下载或删除的操作。如下图4.11以及图4.12所示。

代码入下：

public String zhutiAll(){

String s2="from TZhuti where shifouding='是'";

List zhuti\_list\_all=zhutiDAO.getHibernateTemplate().find(s2);

String s3="from TZhuti where shifouding='否' order by shijian desc";

List zhuti\_list\_fouzhiding=zhutiDAO.getHibernateTemplate().find(s3);

for(int i=0;i<zhuti\_list\_fouzhiding.size();i++){

TZhuti zhuti=(TZhuti)zhuti\_list\_fouzhiding.get(i);

zhuti\_list\_all.add(zhuti);

}

for(int j=0;j<zhuti\_list\_all.size();j++){

TZhuti zhuti=(TZhuti)zhuti\_list\_all.get(j);

zhuti.setUser(userDAO.findById(zhuti.getUserId()));

zhuti.setHuifushu(huifuDAO.getHibernateTemplate().find("from THuifu where zhutiId="+zhuti.getId()).size());

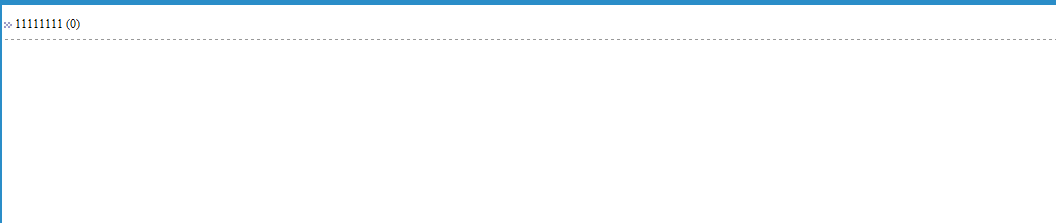
}

HttpServletRequest req=ServletActionContext.getRequest();

req.setAttribute("zhuti\_list\_all", zhuti\_list\_all);

return ActionSupport.SUCCESS;

}



论坛信息管理界面

* + 1. 学生管理个人信息功能界面

学生在通过验证后，登录到学生模式下，能够对自己的以前填写的个人信息完成改动，在这里系统将当前登录客户信息填充到修改页面。如下图4.13所示。

代码入下：

public String userEdit(){

TUser user=userDAO.findById(id);

user.setXuehao(xuehao);

user.setXingming(xingming);

user.setXingbie(xingbie);

user.setNianling(nianling);

user.setLoginname(loginname);

user.setLoginpw(loginpw);

user.setXiandizhi(xiandizhi);

user.setXiandanwei(xiandanwei);

user.setXiandianhua(xiandianhua);

user.setDel("no");

userDAO.attachDirty(user);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg1", "修改成功，重新登录后生效");

return "msg1";

}



学生个人信息管理界面

* + 1. 学生创建班级功能界面

学生登录到学生模式之后，导航栏里有一个“创建新的班级”按钮，可以进入创建班级页面，在该页面需要填写或选择班级所在省市、城市、院校以及班级名称。

代码入下：

public String banjiAdd(){

Map session=ActionContext.getContext().getSession();

TUser user=(TUser)session.get("user");

TBanji banji=new TBanji();

banji.setChinaProvince(chinaProvince);

banji.setChinaCity(chinaCity);

banji.setYuanxiaoId(yuanxiaoId);

banji.setMingcheng(mingcheng);

banji.setChuangjianUserId(user.getId());

banji.setChuangjianshi(new SimpleDateFormat("yyyy-MM-dd HH:mm").format(new Date()));

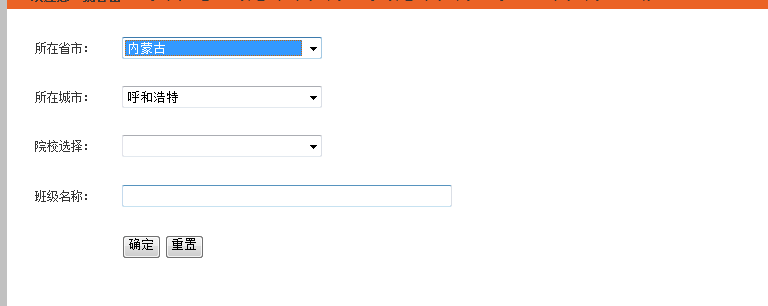
banjiDAO.save(banji);

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg1", "班级创建成功");

return "msg1";

}如下图4.14所示。



创建班级界面

* + 1. 学生管理创建班级信息功能界面

学生登录到学生模式之后，在首页点击导航栏“我创建的班级”菜单可以进入班级信息列表中，点击进入可以查看班级详情，如下图6.16所示。

代码入下：

public String banjiMine(){

Map session=ActionContext.getContext().getSession();

TUser user=(TUser)session.get("user");

StringBuffer sql=new StringBuffer("from TBanji where chuangjianUserId="+user.getId());

List banjiList=banjiDAO.getHibernateTemplate().find(sql.toString());

for(int i=0;i<banjiList.size();i++){

TBanji banji=(TBanji)banjiList.get(i);

banji.setYuanxiao(yuanxiaoDAO.findById(banji.getYuanxiaoId()));

banji.setChuangjianUser(userDAO.findById(banji.getChuangjianUserId()));

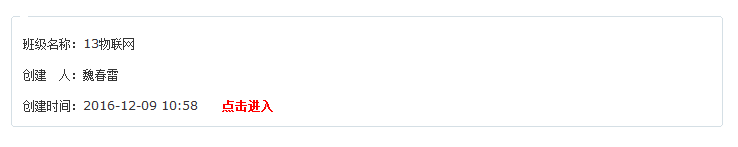
}

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("banjiList", banjiList);

return ActionSupport.SUCCESS;

}



创建的班级界面

* + 1. 学生管理班级论坛信息功能界面

学生登录学生模式之后，能够对自己创建的班级论坛信息点击查看，进入班级详情页面后点击班级论坛可以查看班级所有的论坛信息，点击可查看详细内容，点击发布帖子按钮可以发帖，在进入帖子详细内容后，还能够对班级帖子查看自己感兴趣的，并且还能够进行回复。

代码入下：

public String zhutiAdd(){

TZhuti zhuti=new TZhuti();

zhuti.setTitle(title);

zhuti.setContent(content);

zhuti.setFujian(fujian);

zhuti.setFujianYuanshiming(fujianYuanshiming);

zhuti.setShijian(new SimpleDateFormat("yyyy-MM-dd").format(new Date()) );

zhuti.setUserId(userId);

zhuti.setBanjiId(banjiId);

zhuti.setShifouding("否");

zhuti.setShifoujing("否");

zhutiDAO.save(zhuti);

HttpServletRequest req=ServletActionContext.getRequest();

req.setAttribute("msg", "成功发布新帖");

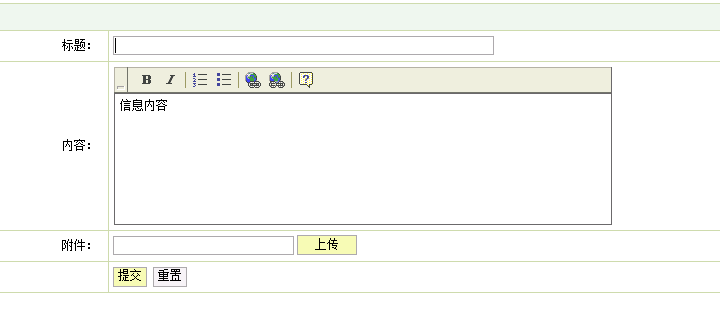
return "msg";

}

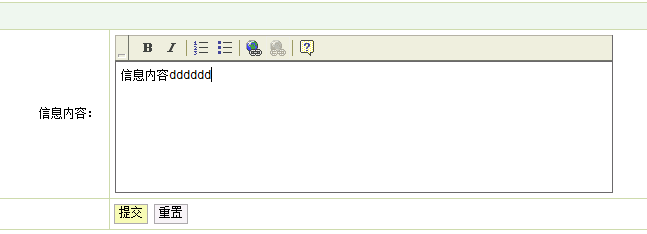
如下图4.15、4.16、4.17、所示。



班级论坛界面



发布帖子界面



回复帖子界面

* + 1. 学生管理班级照片信息功能界面

学生登录到学生模式之后，可以在自己创建的班级里面的照片信息进行查看，进入班级详情页面后点击班级照片可以查看班级所有的照片信息，点击上传照片，可以把自己的照片分享到班级相册。

代码入下：

public String zhaopianAdd(){

Map session=ActionContext.getContext().getSession();

TUser user=(TUser)session.get("user");

TZhaopian zhaopian=new TZhaopian();

zhaopian.setTitle(title);

zhaopian.setFujian(fujian);

zhaopian.setShijian(new SimpleDateFormat("yyyy-MM-dd HH:mm").format(new Date()));

zhaopian.setShangchuanzhe(user.getXingming());

zhaopian.setBanjiId(banjiId);

zhaopianDAO.save(zhaopian);

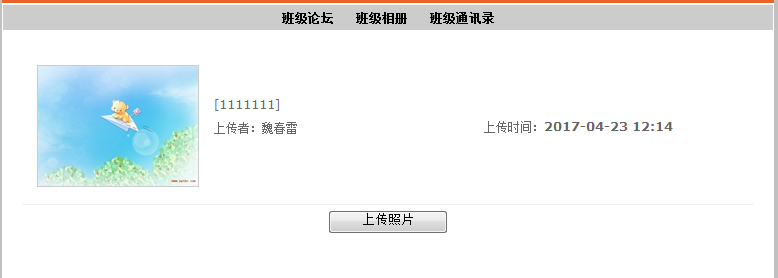
Map request=(Map)ServletActionContext.getContext().get("request");

request.put("msg", "照片上传成功");

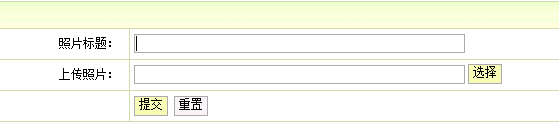
return "msg";

}

如下图4.18、4.19所示。



班级相册界面



上传照片界面

* + 1. 学生查看班级通讯录功能界面

学生登录学生模式之后，点击通讯录，可以班级中所有同学的个人信息，在这里将当前登录学生当前进入的班级通讯录信息显示在通讯录信息列表中。

代码入下：

public String tongxunluAll(){

System.out.println(banjiId+"HHH");

String sql="from TJiaru where banjiId="+banjiId;

List jiaruList=jiaruDAO.getHibernateTemplate().find(sql);

for(int i=0;i<jiaruList.size();i++){

TJiaru jiaru=(TJiaru)jiaruList.get(i);

jiaru.setUser(userDAO.findById(jiaru.getUserId()));

}

Map request=(Map)ServletActionContext.getContext().get("request");

request.put("jiaruList", jiaruList);

return ActionSupport.SUCCESS;

}

如下图4.20所示。



班级通讯录信息界面

1. 系统测试
   1. 测试的意义和目的

测试是对系统在真实环境下测试，与需求定义作比较，检验功能是否能够正常实现。系统的开发过程中，错误随时可能出现。为了保证系统的最终能够完整的实现，应该在设计编码等开发阶段完成之前，系统测试时应按照需求而设计实施，并在实际环境进行。

因为人们能力终究有限的，不可能做到100%完美，在测试的时候，很可能会有遗留的BUG没发现，尤其是在编写程序阶段还会出现许多许多的错误。这些错误与缺陷如果在系统运用时暴露出来，修改这些错误往往会耗费很多人力和金钱，对后面用户的使用造成影响，那样程序容易不被用户接纳。因此系统的BUG 应该尽量在前期发现，发现的越早，越是容易修改，而且前期的投入也小，代价就会在合理的接受范围之内。

系统在投入使用前，对系统的测试，是保证系统质量的关键。测试的目的就是提早发现，提早改正。

* 1. 实例测试表格

对管理员模块和学生模块进行测试，测试结果如下表4.22,4.23所示：

表4.22管理员模块测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 验证过程 | 预期结果 | 实际结果 | 结论 |
| 系统管理员  登录 | 系统管理员登录时输入  账号：a密码：a | 进入系统主界面 | 进入系统主界面 | 通过 |
| 新学校添加 | 进入学校添加页面输入  学校信息 | 添加成功，返回成功提醒 | 添加成功，返回成功提醒 | 通过 |
| 学校查看 | 进入学校管理页面 | 成功显示所有学校列表 | 成功显示所有学校列表 | 通过 |
| 学校删除 | 进入学校管理页面，点击删除按钮 | 删除成功，返回成功提示 | 删除成功，返回成功提示 | 通过 |
| 学生删除 | 进入学生管理页面，点击删除按钮 | 删除成功，返回成功提示 | 删除成功，返回成功提示 | 通过 |
| 照片查看 | 进入照片管理页面 | 成功显示所有照片列表 | 成功显示所有照片列表 | 通过 |
| 照片删除 | 进入照片管理页面，点击删除按钮 | 删除成功，返回成功提示 | 删除成功，返回成功提示 | 通过 |
| 班级查看 | 进入班级管理页面 | 成功显示所有班级列表 | 成功显示所有班级列表 | 通过 |
| 班级删除 | 进入班级管理页面，点击删除按钮 | 删除成功，返回成功提示 | 删除成功，返回成功提示 | 通过 |
| 帖子查看 | 进入帖子管理页面 | 成功显示所有帖子列表 | 成功显示所有帖子列表 | 通过 |
| 帖子删除 | 进入帖子管理页面，点击删除按钮 | 删除成功，返回成功提示 | 删除成功，返回成功提示 | 通过 |

表4.23学生模块测试表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项 | 验证过程 | 预期结果 | 实际结果 | 结论 |
| 学生登录 | 学生登录时输入  账号：\*\*\*密码：\*\*\* | 进入学生页面 | 进入学生页面 | 通过 |
| 学生查看 | 进入学生管理页面 | 成功显示所有学生列表 | 成功显示所有学生列表 | 通过 |
| 学生注册 | 进入学生注册页面输入  学生编号：\*\*\*  姓名：\*\*\*  选择性别：\*  年龄：\*\*  密码：\*\*\*\* | 注册成功，出现注册成功的提示 | 注册成功，出现注册成功的提示 | 通过 |
| 照片查看 | 进入照片管理页面 | 成功显示所有照片列表 | 成功显示所有照片列表 | 通过 |
| 新照片添加 | 进入照片添加页面输入  照片信息 | 添加成功，返回成功提示 | 添加成功，返回成功提示 | 通过 |
| 新班级添加 | 学生进入班级添加页面  输入班级信息 | 添加成功，返回成功提示 | 添加成功，返回成功提示 | 通过 |
| 帖子查看 | 进入帖子管理页面 | 成功显示所有帖子列表 | 成功显示所有帖子列表 | 通过 |
| 新帖子发布 | 进入帖子添加页面  输入帖子信息 | 发布成功，返回成功提示 | 发布成功，返回成功提示 | 通过 |
| 帖子回复 | 进入帖子回复页面  输入回复信息 | 回复成功，返回成功提示 | 回复成功返回成功提示 | 通过 |

1. 结论与展望

这篇论文主要写了基于Java的同学录管理系统的系统分析、系统设计、系和系统测试。系统模块分成两个模块，管理员模块的学生模块，区分开不用的模块对系统操作，每个模块都进行了详细的介绍。测试使用了黑盒测试检测个了每个模块的功能。本系统的实现能够使同学之间的联系更加紧密，不用再为找不到以前的同学而发愁，这个系统的完成不仅方便了其他人，而且也是对自己的提升，通过这是对项目的实践，对这方面的知识也有了更加深入的了解。希望这个系统能够很好的服务大家。

通过这次系统的设计，发现自己对JSP技术的了解不够深入，而且这个系统的开发，开始的时候很是慌乱。所以在动手做这个同学录之前，先在网上看看类似的系统的体系构造。

这次的实践我的感触很深，认识到自己有很多不足之处，在以后也要保持学习的劲头，不能懈怠。

# 参考文献

1. [唐友国](http://search.dangdang.com/book/search_pub.php?category=01&key2=%CC%C6%D3%D1%B9%FA" \t "_blank)，[湛洪波](http://search.dangdang.com/book/search_pub.php?category=01&key2=%D5%BF%BA%E9%B2%A8" \t "_blank).JSP网站开发详解，北京：电子工业出版社，2008
2. 冯艳奎，赵德奎.JSP实用案例教程，清华大学出版社，2004
3. 姜 仲，李绪成.Java语言程序设计，东北大学出版社，2007
4. 史嘉权.数据库系统概论，清华大学出版社，2006
5. 许令波． 深入分析Java Web技术内幕 [M]，北京：电子工业出版社，2012
6. 张峋，[杨三成](http://searchb.dangdang.com/?key=&key2=%D1%EE%C8%FD%B3%C9&medium=01&category_path=01.00.00.00.00.00)． 关键技术 JSP与JDBC应用详解 [M]，北京：中国铁道出版社，2010
7. 林信良． JSP & Servlet学习笔记 [M]，北京：清华大学出版社，2012