****

****

**软 件 学 院**

**三级课题报告**

**课题名称： 高校运动会管理系统研发-用户管理**

**专业： 软件工程**

**班级： 卓越161**

**学号： 201619140220**

**学生姓名： 崔烁豪**

**指导老师： 韩玉民**

2018年 3月25日

**中原工学院软件学院**

**三级实践课**题任务书

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 崔烁豪 | 班级 | 卓越161班 |
| 题目 | 高校运动会管理系统研发-用户管理 | | |
| **设**  **计**  **任**  **务** | 运动会是高校重要的大型集体活动，参加人员最多、涉及面最广、影响最大，是展示学校风采和提高凝聚力的有效平台，目前许多高校的二级学院也开展了二级学院运动会。但运动会的组织既烦琐又复杂，传统的人工管理效率低、易出错，高校运动会的管理也是教育信息化的组成部分，因此，开发基于Web的高校运动会管理系统，解决人工管理的效率低、准确性差、难于管理的问题，提高信息化管理水平。  系统主要功能包括：  **1.学习各项活动组织方式与竞赛规则。**  **2.构建系统数据库。**  **3.系统管理子系统，主要功能：**  （1）人员管理：包括运动员、裁判员、志愿者、管理人员等信息管理。  （2）新闻公告管理。  （3）比赛项目管理：包括项目信息及其竞赛规则管理。  （4）比赛记录管理：历届运动会竞赛记录管理。  **4.报名与分组子系统，主要功能：**  （1）个人报名管理。  （2）团体报名管理：按学院、班级参加的多人集体项目报名管理。  （3）分组管理：如按跑道对运动员进行自动智能分组。  （4）报名统计汇总。  **5.成绩管理子系统，主要功能：**  （1）成绩录入。  （2）成绩排名管理：包括排名、记分、下一阶段（复赛、决赛）选手自动选拔等。  （3）成绩查询。  （4）奖项管理：包括奖项统计、优秀组织奖、优秀运动员奖等奖项管理。  （5）最新记录管理：统计新的记录项目和成绩。  开发工具：MySQL和 MyEclipse | | |
| **时**  **间**  **进**  **度** | 第1周（2018-03-05~2018-03-11）：对需求进行深入分析，并设计出架构与数据库，仔细理解需求，规划功能。  第2周（2018-03-12~2018-03-18）：学习并完成每一步的开发，在过程中发现问题并改进问题，学习使用第三方的API来完成系统，并进行下一步的功能该进，并优化界面。  第3周（2018-03-19~2018-03-25）：调试测试，通过实际用户的操作，提出建议并修改，撰写报告。 | | |
| **主要**  **参考**  **文献**  **与原**  **始资**  **料** | [1]郭丽.Java Web程序设计[M].中原工学院软件学院出版,2018.  [2]欧二强.Java编程手记[M].清华大学出版社，2013.  [3]何欣.网页设计与制作技能基础教学[M].科学出版社，2017.  [4]罗杰S.普莱斯曼.软件工程实践者的研究方法[M]. 郑人杰译,机械工业出版社017.  [5]刘瑞新.SQL Server数据库技术及应用教程[M].电子工业出版社，2012. | | |

# 摘要

在计算机时代日益发展的今天,计算机网络正快速融入这个社会的每一个领域。在体育领域中，计算机在运动会管理方面的应用也越来越深入。现在各大高校已经很普及了，高校运动会管理系统便应运而生。高校运动会管理系统是这个时代的产物。高校运动会管理系统，便是为了提高运动会管理的效率，实现整个运动会的集中管理而设计的。

高校运动会管理系统，主要是用于提高运动会管理的效率，更方便、更系统的管理高校运动会。本系统集报名管理、比赛管理于一身，实现运动会管理自主化、智能化，达到提高管理效率和质量，满足不同层次学校的不同要求，节省人力资源的目的。而且，本系统的另一个重要特点是界面友好、易于操作，有很好的可视化效果。

本文共5章，第1章主要介绍了高校运动会管理系统开发的背景，研究意义及系统功能，第2章是对高校运动会管理系统的系统分析，第3章是分析本系统数据库的设计，第4章是对系统的几大功能的具体实现进行阐述，第5章介绍在软工实践课题三完成过程中的一些心得体会，以及对此次实践成果的评价，进行总结。

本系统采用Java、Servlet作为编程语言，MySQL为后台数据库，Jsp实现动态网页的制作，以确保系统安全保密，易于维护。

关键字：高校；运动会；管理系统；实践课题。

至少3个关键字，要能反映题目

# 目录

[摘要 I](#_Toc517462564)

[目录 1](#_Toc517462565)

[第1章 概述 1](#_Toc517462566)

[1.1 课题背景 1](#_Toc517462567)

[1.2 运动会管理系统的现状 1](#_Toc517462568)

[1.3 运动会管理系统的功能 1](#_Toc517462569)

[1.4 运动会管理系统的作用 2](#_Toc517462570)

[第2章 系统分析 3](#_Toc517462571)

[2.1问题描述 3](#_Toc517462572)

[2.2技术分析 3](#_Toc517462573)

[2.3需求分析 3](#_Toc517462574)

[第3章 数据库分析与设计 5](#_Toc517462575)

[3.1 实体分析 5](#_Toc517462576)

[3.2关系设计 6](#_Toc517462577)

[3.3数据表设计 7](#_Toc517462578)

[第4章 系统实现 9](#_Toc517462579)

[4.1用户登录流程与权限 9](#_Toc517462580)

[4.2功能实现 10](#_Toc517462581)

[4.3系统的发布 26](#_Toc517462582)

[第5章 结束语 27](#_Toc517462583)

[指导老师评语 28](#_Toc517462584)

# 第1章 概述

本章主要阐述高校运动会管理系统研发的开发背景、应用现状及开发意义。

## 1.1 课题背景

进入21世纪，随着计算机时代的飞速发展，计算机网络正快速融入这个社会的每一个领域。随着计算机的普及，各个学科都在探讨如何利用这个新科技资源更好的为本学科和人类服务的问题。为了配合每所高校每年运动会的管理工作，加强运动会的各个项目的管理，而开发研究出这项高校运动会管理系统。

## 1.2 运动会管理系统的现状

虽然现在已经有利用信息技术运动会的例子,但大都处于起步阶段,有的仅仅是一些静态的网页设计或单纯的搬一些必需的表格上电脑,缺乏互动性,参与者不能主动参与,无法实现主办方与参与方之间的交流。这些并没有很好的利用信息技术实现真正的运动会管理自动化。

## 1.3 运动会管理系统的功能

本系统用户主要分为三类：（1）学生，（2）教师，（3）管理员。

学生可以登录、报名（分为报名运动员和报名志愿者）、报名运动员的可以查看已报名项目、自己比赛得分、是否晋赛、以及得奖情况。未报名运动员和报名志愿者的同学，可以查看志愿者名单、查看比赛进度。

教师可以登录、查看志愿者名单、项目运动员名单、项目比赛进度。

管理员分为多种管理员：具体如表1.1所示。

表1.1 管理员类别与功能

|  |  |  |
| --- | --- | --- |
| 管理员编号 | 管理员类别 | 管理员职能 |
| 1 | 管理员管理 | 其他管理员进行管理，包括增删改查等 |
| 2 | 注册管理 | 录入学校、学院、班级以及代号、录入注册人员信息 |
| 3 | 现场项目管理 | 按具体情况录入项目信息，包括地点、时间、跑道信息等 |
| 4 | 新闻公告管理 | 其他管理员信息汇总，以及搜集信息。在通告栏通告。所有人可以访问、包括游客 |
| 5 | 报名管理员 | 项目介绍，包括项目信息及竞赛规则。审核报名人员信息。 |
| 6 | 裁判 | 成绩判定、成绩录入 |
| 7 | 成绩管理 | 各项目成绩排名、筛选下一轮运动员名单 |

## 1.4 运动会管理系统的作用

运动会管理系统在现代高校运动会管理中发挥着越来越重要的作用，具体作用如下:

（1）提高运动会管理的效率和质量。

（2）能实现运动会的集中管理、自动核对报名信息情况和统计报名人数等;

（3）提高办事效率;充分利用网络资源。

# 第2章 系统分析

本章主要是对运动会管理系统项目进行合理性的分析，通过问题描述、技术分析、需求分析这三个方面连描述这个项目。

## 2.1问题描述

就目前各个高校运动会管理中出现的诸多问题,总结为以下几点:

（1）人力资源不足;

（2）传统的报名模式效率极低;

（3）报名信息的核对极不方便;

（4）新技术资源(如网络)未能充分利用等。

## 2.2技术分析

运动会管理系统开发工具为MyEclipse和Navicat For Mysql，主要采用MVC设计模式。视图层运用了JSP、HTML、CSS，控制层用的是Servlet。同时还需要MySQL创建数据库,和Java连接数据库的操作

## 2.3需求分析

该高校运动会管理系统所需实现的功能模块图如图2.1所示:



图2.1 系统模块设计图

用户需求如下：

（1）超级管理员：负责系统维护,具有系统的所有权限,包括对其他管理员的增删改查等。

（2）运动会管理员：具有除超级管理员之外的所有权限,包括运动员和运动项目基础信息的录入、运动员的检录,比赛流程管理系统管理员可根据实际需要添加、删除、修改各角色、用户。

（3）运动员:包括学生和教师,具有查看比赛成绩信息、比赛公告消息、修改个人用户密码的权限。

# 第3章 数据库分析与设计

数据库拥有很强的功能，可以建立一个数据库把我们所需要的数据给存储起来，并可以进行增删改查的操作，还可以对数据库进行导入导出等操作。我们采用MySQL数据库来保存运动会管理系统的各项信息。

## 3.1 实体分析

本系统涉及的实体包括学校、学院、班级、教师、学生、管理员等。经过分析，学校实体E-R图如图3.1，学院实体E-R图如图3.2，班级实体E-R图如图3.3，教师实体E-R图如图3.4，管理员实体E-R图如图3.5，学生实体E-R图如图3.6。



图3.1 学校实体E-R图



图3.2 学院实体E-R图



图3.3 班级实体E-R图



图3.4 教师实体E-R图



图3.5 教师实体E-R图

图3.6 学生实体E-R图

## 3.2关系设计

这里有六个实体，分别是：学校、学院、班级、管理员、教师、学生，它们之间的关系分别是：学校包含学院，一对多;学院包含班级，一对多;学校委任管理员，一对多；教师任职于学院，多对一；学生就读于班级，多对一。具体如下图3.7所示。

图3.7 系统E-R图

## 3.3数据表设计

运动会管理系统的数据库设计总共有六个表：学校代码表，学院代码表，班级代码表，管理员表，教师表，学生表。表的具体结构见表3.1，表3.2，表3.3，表3.4，表3.5，表3.6所示。

表3.1 学校代码表(code\_school)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 描述 | 数据类型及长度 | 是否为空 | 主键 |
| id | 学校代码 | int(5) | 否 | 是 |
| name | 学校名称 | varchar(20) | 否 | 否 |

表3.2 学院代码表(code\_college)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 描述 | 数据类型及长度 | 是否为空 | 主键 |
| id | 学院代码 | int(5) | 否 | 是 |
| name | 学院名称 | varchar(20) | 否 | 否 |

表3.3 班级代码表(code\_class)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 描述 | 数据类型及长度 | 是否为空 | 主键 |
| id | 班级代码 | int(5) | 否 | 是 |
| name | 班级名称 | varchar(20) | 否 | 否 |

表3.4 管理员表(tb\_manager)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 描述 | 数据类型及长度 | 是否为空 | 主键 |
| id | 账号 | varchar(15) | 否 | 是 |
| pwd | 密码 | varchar(15) | 否 | 否 |
| name | 姓名 | varchar(20) | 否 | 否 |
| work | 管理员类别 | varchar(20) | 否 | 否 |
| date | 注册日期 | date | 否 | 否 |

表3.5 教师表(tb\_teacher)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 描述 | 数据类型及长度 | 是否为空 | 主键 |
| id | 教师工号 | varchar(15) | 否 | 是 |
| pwd | 密码 | varchar(15) | 否 | 否 |
| name | 姓名 | varchar(20) | 否 | 否 |

表3.6 学生表(tb\_student)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 描述 | 数据类型及长度 | 是否为空 | 主键 |
| id | 学号 | varchar(15) | 否 | 是 |
| pwd | 学生密码 | varchar(15) | 否 | 否 |
| name | 学生姓名 | varchar(20) | 是 | 否 |
| sex | 学生性别 | varchar(20) | 是 | 否 |
| nation | 学生民族 | varchar(20) | 是 | 否 |
| birthday | 学生生日 | date | 是 | 否 |
| schoolid | 学校代码 | int(5) | 是 | 否 |
| collegeid | 学院代码 | int(5) | 是 | 否 |
| classid | 班级代码 | int(5) | 是 | 否 |
| tel | 手机号 | varchar(20) | 是 | 否 |
| email | 邮箱 | varchar(20) | 是 | 否 |

# 第4章 系统实现

软件设计是一个把软件需求变换成软件表示的过程。最初这种表示知识描绘出软件的总的框架,然后进一步细化,把它加工成在程序细节上非常接近于源程序的软件表示。从工程管理的角度来看，软件设计分两步完成。首先做概要设计，将软件需求转化成数据结构和软件的系统结构。然后是详细设计，即过程设计。通过对结构表示进行细化，得到软件的详细的数据结构和算法。

运动会管理系统设计阶段主要目的是将系统分析阶段所提出的反映了用户信息需求的系统逻辑方案转换成可以实施的基于计算机与通信系统的物理(技术) 方案。

## 4.1用户登录流程与权限

高校运动会管理系统主要包括以下要件：管理员模块、教师模块、学生模块。将系统主要三个模块的工作功能展示如下：



图4.1 管理员模块功能图



图4.2 教师模块功能图



图4.3 学生模块功能图

## 4.2功能实现

系统的用户界面作为人机接口起着越来越重要的作用。它的好坏直接影响到系统的使用性。具有友好界面的系统，对于用户来说，无疑是一种享受。

本系统的各模块功能实现过程如下所示。

**1.用户登录功能**

用户登录分为三种身份方式登录，管理员登录，教师登录和学生登录。登录界面如图4.4所示。



图4.4 登陆界面

主要代码如下：

response.setContentType("text/html;charset=UTF-8");//避免浏览器乱码，默认英文编码处理不了中文字符

String id=request.getParameter("id");//接收请求端数据,id

String pwd=request.getParameter("pwd");//接收请求端数据,pwd

String work=request.getParameter("work");//接收请求端数据,work,work表示身份，选项有"学生"、"教师"、"管理员"

PrintWriter out = response.getWriter();

//out.print("test is"+test+"<br>");

**if**(id==**null**||id.equals("")||pwd==**null**||pwd.equals("")){//验证帐号密码输入是否为空

out.println("Please Input id or pwd");

out.close();

**return**;

}

**if**(work==**null**||work.equals("")||work.equals("null")){

out.println("Please select the work(work表示身份，选项有“学生”、“教师”、“管理员”)");

out.close();

**return**;

}

LoginDao login=**new** LoginDao();//实例化java管理员登录验证类

**boolean** flag=login.prepared(id,pwd,work);//调用验证方法

**if**(flag){

out.println("<html><head><title>正在跳转</title></head>");

String path="";

String url=" ";

**switch**(work){

**case** "manager": path="/管理员登录首页";

ManagerDao manager=**new** ManagerDao();

**try** {

entity.*setName*(manager.getname(id));

entity.*setWork*(manager.getwork(id));

} **catch** (SQLException e) {}

**break**;

**case** "teacher": url="<body><script language='javascript'>document.location = 'sportManage/html/教师登录首页.html'</script></body></html>";**break**;

**case** "student": url="<body><script language='javascript'>document.location = 'sportManage/html/学生登录首页.html'</script></body></html>";**break**;

}

RequestDispatcher rd = request.getRequestDispatcher(path);//转发时，路径不要带项目名

rd.forward(request, response);

out.println(url);

}**else**{

out.println("pwd is wrong!");

}

out.close();

**2.管理员管理功能：**

以管理员身份登录进来之后,管理员的功能包括系统用户管理、运动员报名管理、运动会项目管理、比赛项目编排、检录管理、成绩管理、系统公告管理。其中，管理员管理首页如图4.5所示：



图4.5 管理员管理首页

主要代码如下：

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

PrintWriter out = response.getWriter();

out.println("<HTML>");

out.println(" <HEAD>"

+ "<link type='text/css' rel='stylesheet' href='sportManage/css/管理员登录首页.css' />"

+ "<title>管理员登录首页</title>"

+ "<div style='position:absolute;top:20px;left:1100px;color:#FFF'>"

+ "<a href='/sportmanage/sportManage/html/登录.html'><</a>"

+ " <font color='darkblue'>欢迎,"+entity.*getName*()+"<br>您的身份:"+entity.*getWork*()+"</font>"

+ "<form action='"+entity.*getWork*()+"'>"

+ " <a name='test' value='1'></a>"

+ " <input type='submit' value='"+entity.*getWork*()+"'>"

+ "</from>"

+ " <br><a href='sportManage/html/index.html' style='text-decoration: none;'>返回首页</a>"

+ " <a href='#' style='text-decoration: none;' onclick='ShowDiv('MyDiv','fade')'>|个人信息</a>"

+ "</div>"

+ "<img src='sportManage/img/9.PNG' style='width:100%;height: 185px;'/>"

+ "<div style='background-color: #bbbbbb;width: 100%;height: 500px;'>"

+ "</HEAD>");

out.println(" <BODY>");

out.println(" </BODY>");

out.println("</HTML>");

out.flush();

out.close();

其中，数据库操作主要代码如下：

/\*

\* getname(id)方法作用:在管理员表中，根据账号查询姓名

\* 返回值类型:String型，查到的姓名

\* \*/

**public** String getname(String id) **throws** SQLException{

Connection conn = **null**; //声明Connection对象

PreparedStatement perstat =**null**; //声明PreparedStatement对象

ResultSet rs =**null**;

String name = **null**;//根据账号查到的姓名

**try**{

conn=DBConnection.*getConnection*();//获取数据库连接

String sql="select name from tb\_manager where id=?";

perstat=conn.prepareStatement(sql);

perstat.setString(1,id);

rs=perstat.executeQuery();

}**catch**(Exception e){

e.printStackTrace();

}

**if**(rs.next()){

name= rs.getString(1);

}

**return** name;

}

/\*

\* getwork(id)方法作用:在管理员表中，根据账号查询管理员身份

\* 返回值类型:String型，查到的姓名

\* \*/

**public** String getwork(String id) **throws** SQLException{

Connection conn = **null**; //声明Connection对象

PreparedStatement perstat =**null**; //声明PreparedStatement对象

ResultSet rs =**null**;

String work = **null**;//根据账号查到的管理员身份

**try**{

conn=DBConnection.*getConnection*();//获取数据库连接

String sql="select work from tb\_manager where id=?";

perstat=conn.prepareStatement(sql);

perstat.setString(1,id);

rs=perstat.executeQuery();

}**catch**(Exception e){

e.printStackTrace();

}

**if**(rs.next()){

work= rs.getString(1);

}

**return** work;

}

**3.超级管理员功能：**

超级管理员可以对其他的管理员进行管理。包括对其他管理员的增删改和全部显示（管理员用户不多）。超级管理员界面如下图4.6所示：



图4.6 超级管理员界面

主要代码如下：

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

ManagerDao managerall=**new** ManagerDao();

ResultSet rs=managerall.queryall();

PrintWriter out = response.getWriter();

out.println("<HTML>"

+ "<HEAD>"

+ " <meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>"

+ " <TITLE>管理员管理</TITLE>"

+ " <link type='text/css' rel='stylesheet' href='sportManage/css/管理人员.css' />"

+ "</HEAD>"

+ "<BODY>"

+ " <div id='mydiv'>"

+ "<a href='LoginServlet?id=1&pwd=1&work=manager'><</a>"

+ "<form action='管理员管理Servlet'>"

+ " <table align='center'>"

+ " <tr>"

+ " <th>"

+ "账号"

+ " </th>"

+ " <th>"

+ "姓名"

+ " </th>"

+ " <th>"

+ "身份"

+ " </th>"

+ " <th>"

+ "注册日期"

+ " </th>"

+ " </tr>");

**try** {

**while**(rs.next()){

out.println(" <tr>"

+ " <th>"

+rs.getObject(1)

+ " </th>"

+ " <th>"

+ rs.getObject(3)

+ " </th>"

+ " <th>"

+ rs.getObject(4)

+ " </th>"

+ " <th>"

+ rs.getObject(5)

+ " </th>"

+ " <th>"

+ "<input type='radio' name='choice' value='"+rs.getObject(1)+"'>"

+ " </th>"

+ " </tr>");

}

} **catch** (SQLException e) {

e.printStackTrace();

}

out.print( " <tr>"

+ "<th>"

+ ""

+ "</th><th></th>"

+ " <th>"

+ " <select name='exe'>"

+ " <option value='修改'>修改</option>"

+ " <option value='添加'>添加</option>"

+ " <option value='删除'>删除</option>"

+ " </select>"

+ "<input type='submit' value='执行操作'>"

+ " </th>"

+ " <th>"

+ " </th>"

+ " </tr>"

+ " </table>"

+ " </div>"

+ "</form>"

+ "</BODY>"

+"</HTML>");

out.flush();

out.close();

其中，控制层Servlet主要代码如下：

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

PrintWriter out = response.getWriter();

String choice=request.getParameter("choice");//所选管理员的账号

String exe=request.getParameter("exe");//修改、添加、删除

String path="";

**if**(exe.equals("添加")){

path="/添加管理员";

RequestDispatcher rd = request.getRequestDispatcher(path);

rd.forward(request, response);

}**else** **if**(choice==**null**||exe==""){

out.print("请选择一个管理员"

+ "<form action='管理员管理'>"

+ "<input type='submit' value='确认'>"

+ "</form>");

}**else** **if**(exe.equals("删除")){

ManagerDao manager=**new** ManagerDao();

**boolean** havedele=**false**;

havedele=manager.delete(choice);

**if**(havedele){

out.println("<HTML>");

out.println(" <HEAD><TITLE>删除管理员</TITLE></HEAD>");

out.println(" <BODY>"

+ "删除成功！"

+ "<form action='管理员管理'>"

+ "<input type='submit' value='确认'>"

+ "</form>"

+ "");

out.println(" </BODY>");

out.println("</HTML>");

}

}**else** **if**(exe.equals("修改")){

path="/修改管理员";

request.setAttribute("id",choice);

RequestDispatcher rd = request.getRequestDispatcher(path);//转发时，路径不要带项目名

rd.forward(request, response);

}

out.flush();

out.close();

}

其中，数据库操作的主要代码如下：

**public** ManagerDao(){}//无参构造方法

/\*

\* add(id,pwd,name,work,date)方法作用: 往数据库中添加管理员

\* 添加成功返回true

\* 添加失败返回false

\* \*/

**public** **boolean** add(String id,String pwd,String name,String work,String date){

Connection conn = **null**; //声明Connection对象

PreparedStatement perstat =**null**; //声明PreparedStatement对象

**int** rows=0;

**try**{

conn=DBConnection.*getConnection*();

String sql="insert into tb\_manager(id,pwd,name,work,date) values(?,?,?,?,?)";

perstat=conn.prepareStatement(sql);//执行SQL语句

perstat.setString(1,id);

perstat.setString(2,pwd);

perstat.setString(3,name);

perstat.setString(4,work);

perstat.setString(5,date);

rows=perstat.executeUpdate();//rows接收更新结果,rows大于0表示插入成功

}**catch**(Exception e){

System.***out***.println("插入数据时发生异常");

e.printStackTrace();

}**finally**{

DBConnection.*close*(perstat);

DBConnection.*close*(conn);

}

**if**(rows>0){

**return** **true**;

}**else**{

**return** **false**;

}

}

/\*

\* delete(id)方法作用: 在数据库中删除管理员

\* 删除成功返回true

\* 删除失败返回false

\* \*/

**public** **boolean** delete(String id) {

Connection conn = **null**; //连接数据库对象

PreparedStatement perstat = **null**; //预编译对象

**int** rows = 0; //更新语句行数变量

**try** {

conn = DBConnection.*getConnection*(); //获取数据库连接

String sql = "DELETE FROM tb\_manager WHERE id=?";

perstat = conn.prepareStatement(sql); //使用预编译语句进行插入操作

perstat.setString(1, id); //将名称参数设置到SQL语句中

rows = perstat.executeUpdate(); //执行更新操作

}**catch** (Exception e) { //判断是否发生异常

JOptionPane.*showMessageDialog*(**null**,"删除数据时发生异常","Message",

JOptionPane.***INFORMATION\_MESSAGE***); //异常提示

e.printStackTrace();

} **finally** {

DBConnection.*close*(perstat);

DBConnection.*close*(conn); //关闭连接数据库对象资源

}

**if**(rows>0){

**return** **true**;

}**else**{

**return** **false**;

}

}

/\*

\* update(id,pwd,name,work,workid,date)方法作用: 修改数据库中管理员信息

\* 修改成功返回true

\* 修改失败返回false

\*\*/

**public** **boolean** update(String id,String pwd,String name,String work,String date){

Connection conn=**null**;//连接数据库对象

PreparedStatement perstat=**null**;//预编译对象

**int** rows=0;//更新语句行数变量

**try**{

conn=DBConnection.*getConnection*();//获取数据库连接

String sql="update tb\_manager set pwd=?,name=?,work=?,date=? where id=?";

perstat=conn.prepareStatement(sql);//使用预编译语句进行插入操作

perstat.setString(1,pwd);

perstat.setString(2,name);

perstat.setString(3,work);

perstat.setString(4,date);

perstat.setString(5,id);

rows=perstat.executeUpdate();//执行更新操作

}**catch**(Exception e){//判断是否发生异常

JOptionPane.*showMessageDialog*(**null**, "修改数据时发生异常！！！",

"☆★提示信息☆★", JOptionPane.***INFORMATION\_MESSAGE***);//异常提示

e.printStackTrace();

}**finally**{

DBConnection.*close*(perstat);

DBConnection.*close*(conn);//关闭连接数据库对象资源

}

**if**(rows>0){

**return** **true**;

}**else**{

**return** **false**;

}

}

/\*

\* queryall()方法作用: 在数据库中查询所有管理员

\* 返回包含所有管理员的ResultSet集合

\* \*/

**public** ResultSet queryall(){

Connection conn = **null**; //声明Connection对象

PreparedStatement perstat =**null**; //声明PreparedStatement对象

ResultSet rs =**null**;

**try**{

conn=DBConnection.*getConnection*();//获取数据库连接

String sql="select \* from tb\_manager";

perstat=conn.prepareStatement(sql);

rs=perstat.executeQuery();

}**catch**(Exception e){

e.printStackTrace();

}

**return** rs;

}

**4.添加管理员功能：**

超级管理员有添加其他管理员的功能。添加管理员界面如下图4.7所示：



图4.7 添加管理员界面

主要代码如下：

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

Date now =**new** Date();

SimpleDateFormat dateFormat=**new** SimpleDateFormat("yyyy-MM-dd");

String date=dateFormat.format(now);

PrintWriter out = response.getWriter();

out.println("<HTML>");

out.println(" <HEAD><TITLE>添加管理员</TITLE></HEAD>");

out.println(" <BODY>"

+ "<form action='添加管理员Servlet'>"

+ "账号<input type='text' name='id'><br>"

+ "姓名<input type='text' name='name'><br>"

+ "密码<input type='password' name='password1'><br>"

+ "密码确认<input type='password' name='password2'><br>"

+ "身份<select name='work'>"

+ " <option value='管理员管理'>管理员管理</option>"

+ " <option value='注册管理'>注册管理</option>"

+ " <option value='现场项目管理'>现场项目管理</option>"

+ " <option value='新闻公告管理'>新闻公告管理</option>"

+ " <option value='报名管理员'>报名管理员</option>"

+ " <option value='裁判'>裁判</option>"

+ " <option value='成绩管理'>成绩管理</option>"

+ "</select><br>"

+ "注册日期<input type='text' name='date' value='"+date+"'><br>"

+ "<input type='submit' value='添加'>"

+ "</form>");

out.println(" </BODY>");

out.println("</HTML>");

out.flush();

out.close();

**5.修改管理员功能：**

超级管理员有修改其他的管理员信息的功能。修改管理员界面如下图4.8所示：



图4.8 修改管理员界面

主要代码如下：

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

String id=(String) request.getAttribute("id");

ManagerDao manager=**new** ManagerDao();

ResultSet rs=manager.queryid(id);

PrintWriter out = response.getWriter();

out.println("<HTML>");

out.println(" <HEAD><TITLE>修改管理员</TITLE></HEAD>");

**try** {

out.println(" <BODY>"

+ "<form action='修改管理员Servlet'>");

**if**(rs.next()){

out.println("账号<input type='text' name='id' value='"+rs.getObject(1)+"'><br>"

+ "密码<input type='password' name='pwd' value='"+rs.getObject(2)+"'><br>"

+ "姓名<input type='text' name='name' value='"+rs.getObject(3)+"'><br>"

+ "身份<input type='text' name='work' value='"+rs.getObject(4)+"'><br>"

+ "注册日期<input type='text' name='date' value='"+rs.getObject(5)+"'><br>");

}

out.println("<input type='submit' value='修改'>"

+ "</form>");

} **catch** (SQLException e) {

e.printStackTrace();

}

out.println(" </BODY>");

out.println("</HTML>");

out.flush();

out.close();

**6.学生注册功能**

在登录前，没有账号的学生需要先注册学生账号。学生注册界面如图4.9所示：



图4.9 学生注册界面

主要代码如下：

<body style="background-color: *bisque*;">

<div id=*div* >

<a href=*"/sportmanage/sportManage/html/index.html"*><</a>

<br>

<h1 style=";font-size: *28*;"><i>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;欢迎</i>&nbsp;注册高校运动会管理系统(学生)</h1>

<br><br>

<form action=*"../../RegisterServlet"* method=*"post"*>

<h2 style="font-weight: *bold*;">&nbsp;&nbsp;&nbsp;学号：<input type=*"text"* name=*"id"*/></h2>

<h2 style="font-weight: *bold*;">&nbsp;&nbsp;&nbsp;出生年月：<select name=*"year"*>

<option value=*"2007"*>2007</option>

<option value=*"2006"*>2006</option>

<option value=*"2005"*>2005</option>

<option value=*"2004"*>2004</option>

<option value=*"2003"*>2003</option>

<option value=*"2002"*>2002</option>

<option value=*"2001"*>2001</option>

<option value=*"2000"*>2000</option>

<option value=*"1999"*>1999</option>

<option value=*"1998"*>1998</option>

<option value=*"1997"*>1997</option>

<option value=*"1996"*>1996</option>

<option value=*"1995"*>1995</option>

<option value=*"1994"*>1994</option>

<option value=*"1993"*>1993</option>

<option value=*"1992"*>1992</option>

<option value=*"1991"*>1991</option>

</select>年

<select name=*"month"*>

<option value=*"01"*>01</option>

<option value=*"02"*>02</option>

<option value=*"03"*>03</option>

<option value=*"04"*>04</option>

<option value=*"05"*>05</option>

<option value=*"06"*>06</option>

<option value=*"07"*>07</option>

<option value=*"08"*>08</option>

<option value=*"09"*>09</option>

<option value=*"10"*>10</option>

<option value=*"11"*>11</option>

<option value=*"12"*>12</option>

</select>月

<select name=*"day"*>

<option value=*"01"*>01</option>

<option value=*"02"*>02</option>

<option value=*"03"*>03</option>

<option value=*"04"*>04</option>

<option value=*"05"*>05</option>

<option value=*"06"*>06</option>

<option value=*"07"*>07</option>

<option value=*"08"*>08</option>

<option value=*"09"*>09</option>

<option value=*"10"*>10</option>

<option value=*"11"*>11</option>

<option value=*"12"*>12</option>

<option value=*"13"*>13</option>

<option value=*"14"*>14</option>

<option value=*"15"*>15</option>

<option value=*"16"*>16</option>

<option value=*"17"*>17</option>

<option value=*"18"*>18</option>

<option value=*"19"*>19</option>

<option value=*"10"*>20</option>

<option value=*"21"*>21</option>

<option value=*"22"*>22</option>

<option value=*"23"*>23</option>

<option value=*"24"*>24</option>

<option value=*"25"*>25</option>

<option value=*"26"*>26</option>

<option value=*"27"*>27</option>

<option value=*"28"*>28</option>

<option value=*"29"*>29</option>

<option value=*"30"*>30</option>

<option value=*"31"*>31</option>

</select>日

</h2>

<h2 style="font-weight: *bold*;">&nbsp;&nbsp;&nbsp;密码：<input type=*"password"* name=*"pwd1"*/></h2>

<h2 style="font-weight: *bold*;">&nbsp;&nbsp;&nbsp;确认密码：<input type=*"password"* name=*"pwd2"*/></h2>

<h2 style="font-weight: *bold*;">&nbsp;&nbsp;&nbsp;手机号：<input type=*"text"* name=*"tel"*/></h2>

<h2 style="font-weight: *bold*;">&nbsp;&nbsp;&nbsp;邮箱：<input type=*"text"* name=*"email"*/></h2>

<h2>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;

<input type=*"submit"* value=*"确定"* style="width: *60px* ;height:*30px*;font-weight: *bold*;" onclick="javascript:show()"/>&nbsp;&nbsp;&nbsp;

</h2>

</form>

</div>

</body>

## 4.3系统的发布

运动会管理系统的发布，就是将整个项目部署到服务器上的过程。在JavaWeb项目已经写好的前提下，把项目导出到war包，把war包放到Tomcat的Webapps目录下，再打开Tomcat服务，系统就发布了。访问这个已经部署好的项目，只需要在浏览器中输入URL即可。例如：http://39.106.204.205:8080/sportmanage/sportManage/html/index.html

# 第5章 结束语

计算机时代的飞速发展，它以其强大的性能价格优势给人们的工作、学习和生活带来极大的方便。运动会是全国乃至全世界的一个重要的项目，在各所高校也是必不可少的一个工作，体育运动的越来越普及，参与运动会的人越来越多，素质越来越高，但是对于运动会的管理工作确是一件比较庞大而复杂的事情。随着计算机科学与技术、特别是网络技术的飞速发展，各高等院校为了提高工作的效率，适应社会发展的需要，方便的高校运动会综合信息管理系统来代替传统的手工操作方式来管理运动会已经是目前各高校正在考虑与运作的一件大事。

运动会管理系统实现办公自主化、智能化，有利于拓展和补充现有人力资源，提高管理的效率；实现运动会信息的集中管理。

这次课题实践让对所学知识有了更深的利用，学会了应用，实践能力有了很大提高。通过这次课题实践，我更体会到时刻保持学习新知识的精神状态的重要性。计算机的各种技术突飞猛进，更新换代的速度很快。要做一个优秀的IT人才，就不应该墨守成规，滞足不前。

在此非常感谢韩老师的指导、李鑫忠学长的督促指正和魏宁同学的合作与帮助。

# 指导老师评语

成绩：

指导老师签字：

年 月 日