

**学生成绩管理系统**

**课程设计报告**

**设计题目： 学生成绩管理系统**

**学 院： 计算机科学学院**

**专 业： 软件工程1602班**

**小组成员： 彭卫杰 41612196**

**张靖漪 41612263**

**汪 鹏 41612232**

**张 铖 41612196**

**指导教师： 武 杰**

**2018 年 12 月**

目录

[1 概述 5](#_Toc19573)

[2 关键技术分析 6](#_Toc12135)

[2.1 前端 6](#_Toc24468)

[2.1.1 前端界面优化 6](#_Toc14831)

[2.1.2 前端技术 6](#_Toc18536)

[2.2 后端 6](#_Toc27086)

[2.2.1 数据访问方式（DAO） 6](#_Toc28380)

[2.2.2 后端逻辑处理 7](#_Toc29924)

[2.3 数据库 8](#_Toc28774)

[2.3.1 数据库的连接 8](#_Toc11647)

[2.3.2 数据库的设计 9](#_Toc7690)

[2.4 服务器 9](#_Toc28887)

[2.5 文件上传 9](#_Toc30552)

[2.5.1 图片上传 9](#_Toc25228)

[2.5.2 批量导入数据（上传读取Excel） 9](#_Toc29304)

[2.6 安全机制 10](#_Toc9482)

[2.6.1 登录验证 10](#_Toc8580)

[2.6.1 Filter技术 10](#_Toc32243)

[2.7 单元测试 10](#_Toc1982)

[3 系统需求分析 11](#_Toc3847)

[3.1 可行性分析 11](#_Toc17762)

[3.1.1经济可行性 11](#_Toc20257)

[3.1.2 技术可行性 11](#_Toc14118)

[3.1.3 操作可行性 12](#_Toc3135)

[3.1.4 法律可行性 12](#_Toc20138)

[3.2 功能模块需求分析 12](#_Toc18100)

[3.2.1 主要功能模块划分 12](#_Toc21344)

[3.2.2 系统用例图 13](#_Toc30449)

[3.3 非功能性需求分析 14](#_Toc30642)

[3.3.1系统的安全性 14](#_Toc6789)

[3.3.2 系统的可靠性 14](#_Toc19575)

[3.3.3 系统的易实用性和可扩展性 14](#_Toc2637)

[4 系统概要设计 14](#_Toc11511)

[4.1 系统概述 14](#_Toc27366)

[4.2 系统结构图 15](#_Toc20484)

[4.3 系统结构分析 15](#_Toc20954)

[4.4 数据库的设计与实现 16](#_Toc20217)

[4.4.1 数据表的概述 16](#_Toc19752)

[4.4.2 数据表的结构 16](#_Toc14754)

[4.4.3 ER图（实体-联系图） 19](#_Toc10261)

[4.5 系统活动图 24](#_Toc17997)

[4.5.1 学生活动图 24](#_Toc24417)

[4.5.2 教师活动图 25](#_Toc26529)

[4.5.3 管理员活动图 25](#_Toc2139)

[5 详细设计与实现 25](#_Toc7661)

[5.1 代码约定 26](#_Toc18795)

[5.1.1 前端代码约定 26](#_Toc18431)

[5.1.2 后端代码约定 26](#_Toc14344)

[5.2 登录模块 27](#_Toc6098)

[5.2.1 登录模块功能和技术实现介绍 27](#_Toc20272)

[5.3 学生模块 29](#_Toc12691)

[5.3.1 查询和修改个人资料 29](#_Toc2785)

[5.3.2 查询成绩 31](#_Toc28613)

[5.4 教师模块 32](#_Toc22565)

[5.4.1 查询和修改个人资料 32](#_Toc23948)

[5.4.2 录入学生成绩 32](#_Toc18095)

[5.4.3 查询和修改成绩 33](#_Toc16498)

[5.4.4 统计分析学生成绩 34](#_Toc4117)

[5.5 管理员模块 38](#_Toc5402)

[5.5.1 学生信息管理 38](#_Toc27181)

[5.5.2 教师信息管理 39](#_Toc16062)

[5.5.3 基本信息管理 40](#_Toc1312)

[5.5.4 管理系统用户 41](#_Toc26475)

[6 项目总结 42](#_Toc23332)

1 概述

**1.1 开发背景**

随着学生数量的日渐增多，学生教务系统的数据量也不断增加，这无疑大大增加了教务系统的负担。如果能把负责学生成绩管理的模块独立出来形成一个独立的系统，便可以有效降低教务系统的数据量，不仅可以方便管理员对于所有学生的信息进行系统的管理，而且便于教师对学生成绩进行查询和修改，学生也可以查询自己的成绩。一个好的学生管理系统，正好可以满足当前的市场需求，提高工作效率，并取得一定的经济效益。本系统的创建就是为了减少人力、物力、时间的耗费。这些优点能够极大地提高学生成绩管理的效率。因此，开发一套合适的、兼容性好的系统是很有必要的。

**1.2 开发目标**

在规定的时间和经济范围内，交付一个基于JSP/SEVLET的成绩管理系统，该系统具备一下要求:

1. 有良好的前端展示，符合商业化的要求，同时便于用户操作，可以轻松上手；
2. 具有可靠性，系统不同角色分配的权限合理，不泄露用户信息；
3. 可以批量录入信息，实现对信息的增删改查；
4. 反应迅速，能及时给出反馈；

**1.3 设计原则**

为了保持良好的用户体验，设计系统的时候需要遵循以下通用原则：

1、设计适当的表示导引。在用户浏览网站的时候，随时告知访问者他们身处何处，想要找的东西在哪里，如何找到想要找的东西。

2、设定期望并做出反馈。当用户提交表单，点击链接的时候，成功与否都需要网页及时给出反馈，保证用户不会因为久久得不到回复而对网站不信任。

3、设计要基于人机工程学。无论是字体、字号、字色、字间距的选择还是按钮的放置位置，按钮的数量等，这些都要考虑到用户的身体，即手、目、耳等。

4、考虑标准，保持一致。设计网站切不可因为如何好看如何设计，一定要保持前后风格一致。

5、考虑辨识。用户不喜记忆，所以网站设计一定要具有记忆，访问前后不能靠用户去辨识记忆。

6、考虑用户水平不一样。我们遵循的原则是做最简单的网站，然后再在其基础上添加特殊的功能和属性。

2 关键技术分析

## 2.1 前端

### 2.1.1 前端界面优化

为了保证良好的用户体验，我们以HTML语言与CSS样式表为基础制作页面，js进行页面与组件间的逻辑实现。在界面美化上，我们保证所有页面风格一致，并具有自适应性的功能，可以随着分辨率的大小和设备的大小而变化，同时，也注意了不同代码在浏览器兼容性上的问题，所以在相应CSS文件里添加了不同浏览器所需的备用样式。在配色和布局上参考了一些优秀网站的设计，保证良好的交互性。

### 2.1.2 前端技术

前端主要使用html、css、js等基本技术，此外还用到bootstrap、jQuery、jstl等技术使得前端交互性增强。页面布局多采用相对布局，放大或缩小页面，页面内容分仍保持相对位置。

**Frame框架**：用于实现页面的局部加载和刷新。

**jQuery技术**：为了可以直接在某条记录上动态修改数据（即直接在当前页面修改而不是重新打开一个修改数据的页面），采用了jQuery技术，实现了在记录原位置直接修改数据。

**Echarts**：为了使数据可视化，我们采用了百度的Echart图表技术http://echarts.baidu.com/，将数据如学生的成绩、课程的及格率等以条形统计图、动态水球图的形式展现，使得教师可以方便直观地分析学生的成绩情况（各分数段的人数，及格率等）。

**Ajax**：AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。AJAX 通过在后台与服务器进行少量数据交换，使网页实现异步更新。这意味着可以在不重载整个页面的情况下，对网页的某些部分进行更新。传统的网页（不使用 AJAX）如果需要更新内容，必须重载整个页面。

**JSON:**JavaScript Object Notation(JavaScript 对象表示法)

JSON 是存储和交换文本信息的语法。类似 XML，比 XML 更小、更快，更易解析，是一种轻量级的数据传输方式。

## 2.2 后端

### 2.2.1 数据访问方式（DAO）

DAO (DataAccessObjects 数据存取对象)是指位于业务逻辑和持久化数据之间实现对持久化数据的访问。通俗来讲，就是将数据库操作都封装起来。

DAO 模式提供了访问关系型数据库系统所需操作的接口，将数据访问和业务逻辑分离对上层提供面向对象的数据访问接口。

DAO 模式的优势就在于它实现了两次隔离。

1、隔离了数据访问代码和业务逻辑代码。业务逻辑代码直接调用DAO方法即可，完全感觉不到数据库表的存在。分工明确，数据访问层代码变化不影响业务逻辑代码,这符合单一职能原则，降低了藕合性，提高了可复用性。

2、隔离了不同数据库实现。采用面向接口编程，如果底层数据库变化，如由 MySQL 变成 Oracle 只要增加 DAO 接口的新实现类即可，原有 MySQL 实现不用修改。这符合"开-闭"原则。该原则降低了代码的藕合性，提高了代码扩展性和系统的可移植性。

### 2.2.2 后端逻辑处理

使用了三层架构（web层、service层和dao层）

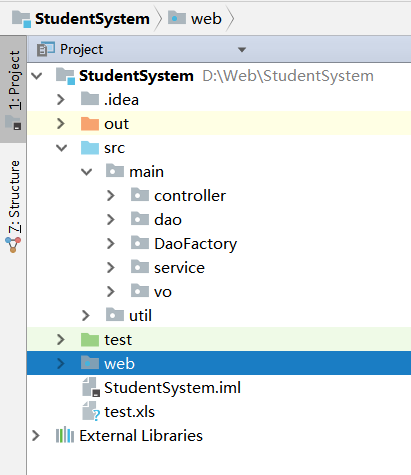


图2-1 项目结构图

**Controller层**:（也叫web层）Controller层负责具体的业务模块流程的控制，在此层里面要调用Serice层的接口来控制业务流程。

**Service层**：Service层主要负责业务模块的逻辑应用设计。调用DAO层存取数据。

**DAO层**：DAO层主要是做数据持久层的工作，负责与数据库进行联络的一些任务都封装在此，DAO层的设计首先是设计DAO的接口，然后定义此接口的实现类，然后就可在模块中调用此接口来进行数据业务的处理，而不用关心此接口的具体实现类是哪个类，显得结构非常清晰。

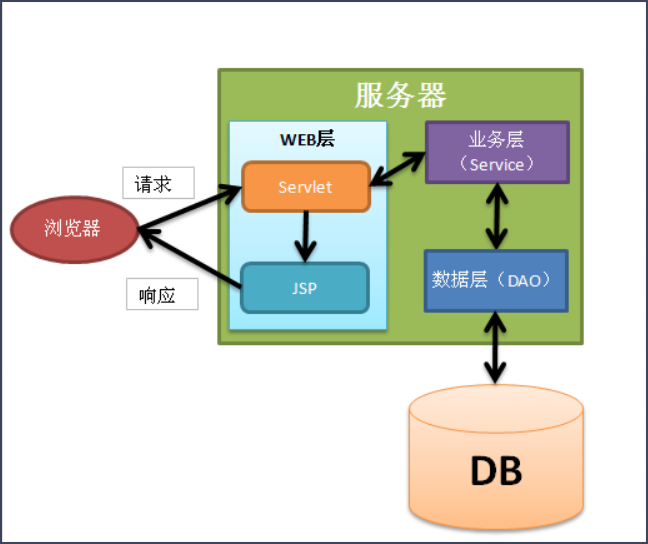


图2-2 处理请求的业务逻辑流程图

该三层架构的优点：

1、开发人员可以只关注整个结构中的其中某一层；

2、可以很容易的用新的实现来替换原有层次的实现；

3、可以降低层与层之间的依赖；

4、有利于标准化；

5、利于各层逻辑的复用。

6、结构更加的明确

7、在后期维护的时候，极大地降低了维护成本和维护时间

## 2.3 数据库

### 2.3.1 数据库的连接

该系统采用mysql数据库，连接数据库用到jdbc数据库驱动。当系统比较庞大时，可以使用数据库连接池如c3p0、DBCP等。数据库连接池负责分配、管理和释放数据库连接，它允许应用程序重复使用一个现有的数据库连接，而不是再重新建立一个；释放空闲时间超过最大空闲时间的数据库连接来避免因为没有释放数据库连接而引起的数据库连接遗漏。这项技术能明显提高对数据库操作的性能。

### 2.3.2 数据库的设计

该系统使用关系型数据库，建表时采用关系模型的数据结构，每个实体对应一张表，联系若有属性，也可对应一张表，如学生实体对应一张学生表，课程对应一张课程表，课程和学生之间是学习的关系，他们会产生一个属性-分数，因此也应该有一张学生课程表。

设置表的主键、外键以及表之间的级联关系。如：成绩表中的学生必须是学生表中存在的学生，当删除学生表中某一个学生时，若两张表是级联删除关系时，成绩表中该学生的信息自动删除。更新时也一样。这样避免了后端逻辑实现的复杂性。

## 2.4 服务器

使用Tomcat服务器。Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8" \t "https://baike.baidu.com/item/tomcat/_blank)，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。

## 2.5 文件上传

### 2.5.1 图片上传

用到的工具：



对于文件上传，浏览器在上传的过程中是将文件以流的形式提交到服务器端的，如果直接使用Servlet获取上传文件的输入流然后再解析里面的请求参数是比较麻烦，所以一般选择采用apache的开源工具common-fileupload这个文件上传组件。这个common-fileupload上传组件的jar包可以去apache官网上面下载common-fileupload是依赖于common-io这个包的。

图片的储存（存取图片的相对路径而不是图片）：

如果直接将图片以二进制流的形式存到数据库中，既浪费时间有浪费空间，因而我们采用的方式是将图片的相对路径存到数据库中，这样极大地提高了访问数据库的速度，同时提高了数据库的空间利用率。

### 2.5.2 批量导入数据（上传读取Excel）

用到的工具：





Apache POI是Apache软件基金会的开放源码函式库，POI提供API给Java程序对Microsoft Office格式档案读和写的功能。

JXL只能对Excel进行操作,属于比较老的框架，它只支持到Excel 95-2000的版本。现在已经停止更新和维护。  
 POI是apache的项目,可对微软的Word,Excel,ppt进行操作,包括office2003和2007,Excl2003和2007。poi现在一直有更新。所以现在主流使用POI。

## 2.6 安全机制

### 2.6.1 登录验证

该系统有学生、教师和管理员三种不同身份的用户，系统分别为设置权限等级，当某用户登录时，系统会检索他们的等级，让他们访问对应的操作页面，学生进入学生系统模块，教师进入教师系统模块，管理员进入管理员系统模块。

### 2.6.1 Filter技术

用户可以通过url访问系统资源，对于一些敏感保密的资源需要有保护机制，因此采用filter技术。

Filter也称之为过滤器，它是Servlet技术中最激动人心的技术，WEB开发人员通过Filter技术，对web服务器管理的所有web资源：例如Jsp, Servlet, 静态图片文件或静态 html 文件等进行拦截，从而实现一些特殊的功能。例如实现URL级别的权限访问控制、过滤敏感词汇、压缩响应信息等一些高级功能。

同时，为了防止页面出现乱码，该系统中添加了编码过滤器。

在该系统中，使用了过滤器，防止用户直接通过URL地址栏直接进入该系统，实现原理是，在过滤器中添加一个判断，判断用户的身份信息是否在session中，只有通过登录身份验证正确时，用户的身份才会添加到session，否则session中无该用户信息，过滤器将阻止非法访问。

## 2.7 Junit4单元测试

开发过程中采用Junit4单元测试技术，分别测试各个子功能模块的功能。

在项目开发中，主要来测试DAO层（数据持久层）的数据的查询增删改功能，确保和数据库实现正确的数据存取功能。

**public class** UserDaoTest {  
 @Test  
 **public void** selectUser() **throws** Exception {  
 UserDao userDao = DaoFactory.*getUserDaoInstance*();  
 System.***out***.println(userDao.selectUser(**"123"**, **"124"**));  
 }  
  
 @Test  
 **public void** insertUser() **throws** Exception {  
 User user = **new** User(**"41612197"**, **"123"**, **"3"**);  
 UserDao userDao = DaoFactory.*getUserDaoInstance*();  
 userDao.insertUser(user);  
 }

3 系统需求分析

## 3.1 可行性分析

本节主要从经济可行性、技术可行性、操作可行性和法律可行性进行分析。

### 3.1.1 经济可行性

从经济可行性分析来讲，环境拥有建立该系统所需的网络设备及软件，具备了开发Web平台系统的基本条件；系统投资成本低，回报率高；人力资源方面消耗量少；而且，系统的生存周期短、系统工作负荷量不是很大、处理速度要求快。无论从横向还是纵向相比，本系统都是可以实行的。在线购物系统的设计和开发不仅是电子商务领域的重要组成部分，也是管理信息系统应用的主要方向之一。

经以上分析，该系统可行。

### 3.1.2 技术可行性

从技术可行性来讲，本系统是基于jsp的Web开发的，jsp 是在前段和后端分离下产生的框架技术，已经被大家公认为经典的框架。而且，Java 语言技术也具有卓越的通用性、高效性、平台移植性和安全性，比较适合这个系统的开发和应用。

在软件方面，考虑到系统实施的可行性，选择了现今比较流行的开发工具intelijidea进行开发管理平台的设计，使用MySQL数据库存储数据。

在硬件方面，系统只需要一台内存4G，硬盘在500.0G的品牌机或兼容机。

经以上分析，该系统可行。

### 3.1.3 操作可行性

本系统界面友好、整洁，操作简易，并配有相关说明，学习方便，管理人员对开发此应用系统的态度确定。

经过以上分析，该系统可行。

### 3.1.4 法律可行性

考虑到使用到的软件产品的版权问题，软件服务器所使用的软件，应该尽量使用正版软件。如果正版软件价格高得难以接受，可以采用一些能满足系统运行的免费软件来替代。开发阶段可以使用试用版软件来替代，但正式发布时建议操作系统采用正版软件。

该平台是作为课程设计，与商业无关，又因为是自主开发设计，因此不会构成侵权，在法律上可行。

## 3.2 功能模块需求分析

本节主要通过系统的主功能模块划分以及系统用例图来分析系统需求。

### 3.2.1 主要功能模块划分

学生成绩管理系统分为三个模块：学生模块、教师模块和管理员模块。

学生模块的功能：个人信息的查询，部分信息的修改，本人成绩的查询

教师模块的功能：个人信息的查询，成绩的录入、修改、查看和分析

管理员模块的功能：对学生、教师、班级、课程的增删改查，课程的安排，系统用户的添加和删除，为学生分配班级等。

对于各功能的详细介绍，如表3-1所示：

表3-1 学生成绩管理系统功能模块介绍

|  |  |  |  |
| --- | --- | --- | --- |
| **权限等级** | **身份** | **功能菜单** | **功能说明** |
| 3 | 管理员 | 学生信息管理 | 包括学生信息的查询、添加、修改和删除 |
| 教师信息管理 | 包括教师信息的查询、添加、修改和删除 |
| 班级管理 | 先进行录入班级的信息，班级号，班级名称，班级院系。 |
| 基本信息管理 | 包括课程的管理、班级管理和课程安排 |
| 个人管理 | 管理员有自己的个人信息，可以查看并修改自己的个人信息。 |
| 系统用户管理 | 管理员添加或删除使用该系统的学生、教师以及管理员的账号密码权限的信息 |
| 1 | 学生 | 登录 | 登录系统界面 |
| 成绩查询 | 通过输入自己的课程号，学期，查询自己的某个课程的相关成绩及其他信息。 |
| 个人信息 | 个人信息就是学生姓名，学生的性别年龄等个人相关信息。可以上传图片 |
| 2 | 教师 | 登录 | 登录到教师模块 |
| 成绩录入 | 1. 教师选择要录入分数的课程，直接在输入每个人的成绩，点击保存即可。 2. 通过Excel导入成绩。 |
| 成绩查看 | 1. 查看某门课程的所有学生的成绩 2. 查看某一个学生的成绩 |
| 成绩分析 | 教师可以查看某门课的各个分数段的人数分布，课程的及格率等，还可看最高最低分、平均分。 |
| 个人信息 | 教师可以查看自己的个人信息，修改自己的个人信息，可以更换头像 |

### 3.2.2 系统用例图

根据3.2.1的主功能模块分析，学生成绩管理系统包括三个参与者：学生、教师、管理员。根据不同身份的用户和系统的交互活动不同，可以画出如下用例

图：

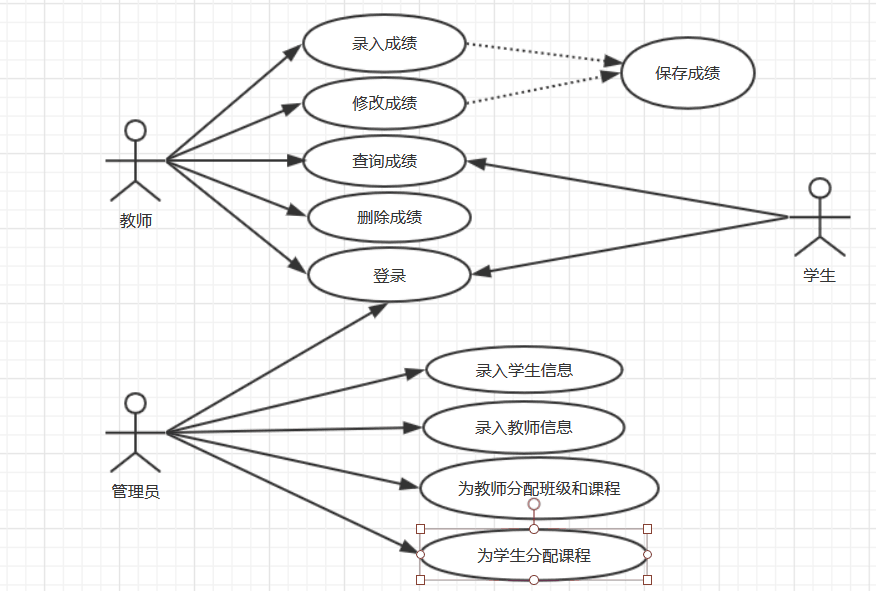


图3-1 学生信息管理系统用例图

## 3.3 非功能性需求分析

### 3.3.1系统的安全性

通常在描述学生成绩管理系统安全性时，越是薄弱的部分，则往往越会受到攻击的影响。因为在攻击者在对系统进行攻击行为时，无论其出于何种目的，往往会找出该系统最易攻击的环节，从而沿阻力最小的路径行动。

通常我们在使用系统的时候，都会经登录，输入所需的用户名及对应密码。为防止偷窥造成的密码泄露，而在对外显示时都会以“\*”作为输入字符的代号。用户在登录时所使用的账号必须是数据库中已存的，且所输入的账号密码必须与数据库中存储的账号密码一致，否则都会造成登陆失败。无论是管理员还是学生或教师，为了报障学生信息系统的安全性，都必须要经过登陆方能进入系统。

### 3.3.2 系统的可靠性

为了防止用户的非法操作，系统通常会对不同用户提供不同的服务，并对其设置不同的操作权限。管理员可以使用基于Web的学生成绩管理系统甚至所有功能，性能可靠稳定。

### 3.3.3 系统的易实用性和可扩展性

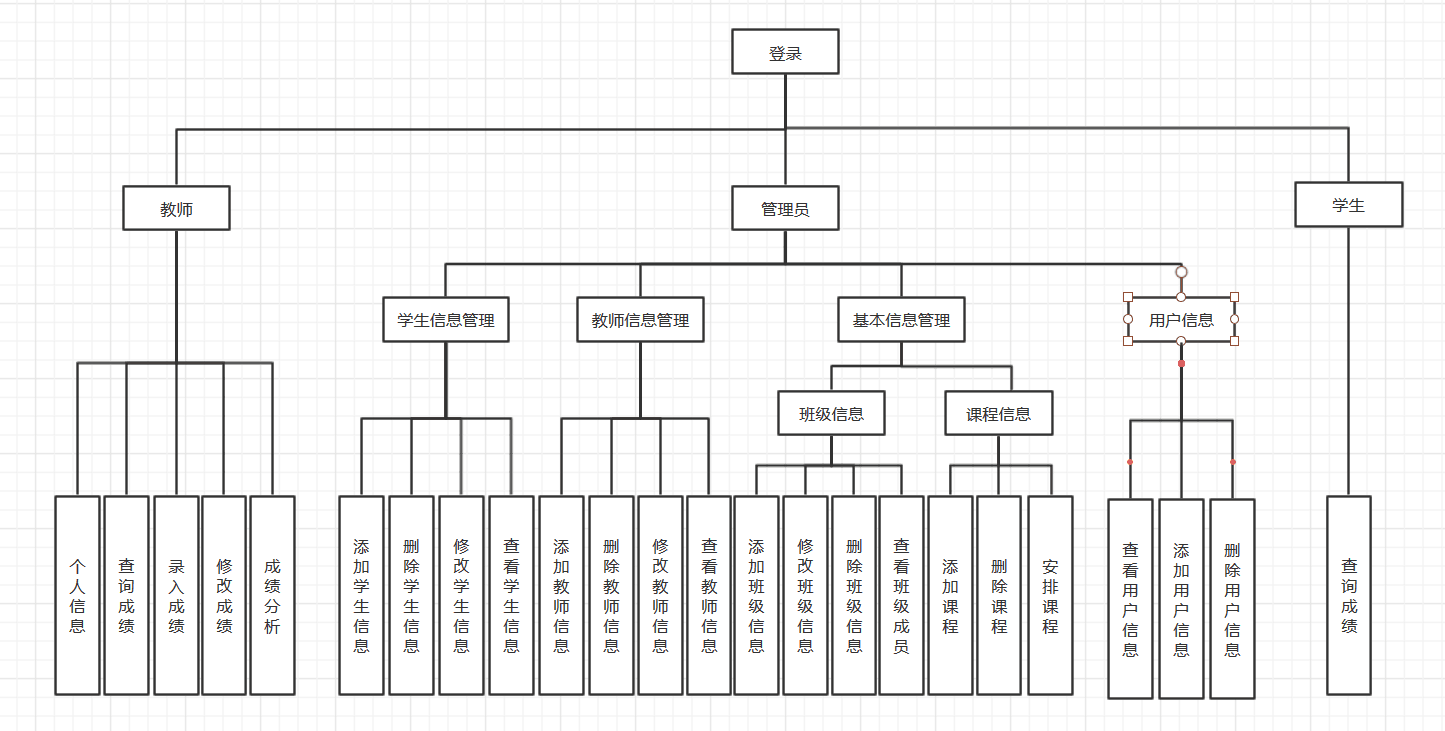
一个完善的操作系统应具备以下特点：为保证系统界面具有操作简单，界面的设计以美观大方为主；系统还应该具备展示学生信息的功能，从而方便用户进行学生课程信息的浏览或比较；为方便教师分类查找，系统应该对学生的课程分类采取规范管理；为方便教师进行修改确认行为，我们队每个用户的使用权限加以完善；让学生及教师对系统内的信息修改具有自己可以操作的部分。除此之外，还有管理员的信息查询、访问次数的信息查询、成绩排行公示等功能，此外，为了方便管理者管理与教师进行成绩更新学生信息，还设置并完善的后台管理功能。

4 系统概要设计

## 4.1 系统概述

学生信息管理系统主要包括三大模块：学生模块、教师模块和管理员模块。学生模块的功能有：查看和修改个人信息，查看成绩。教师模块的功能有：查看和修改个人资料，录入学生成绩、查看学生成绩、成绩的统计与分析。管理员模块的功能有：学生信息管理、教师信息管理、课程信息管理、班级信息管理（班级的查询增删）、安排课程、系统用户管理等。

## 4.2 系统结构图

****

**图4-1 学生成绩管理系统结构图**

## 4.3 系统结构分析

学生成绩管理系统分为学生、教师和管理员三大模块。每个模块有分为几个子功能模块。这些子模块分别实现了不同的功能，学生模块的查询成绩功能，教师模块的个人信息管理、成绩的查询录入功能、系统统计分析成绩功能，管理员的学生信息管理、教师信息管理、基本信息管理和系统用户信息管理。

*学生登录后可进行的操作：*

1. 查看个人资料（包括姓名、学号、照片、年级、联系方式和住址等）
2. 修改个人资料
3. 查询分数

*教师登陆后可进行的操作：*

1. 查看个人资料（包括姓名、工号、照片、学位、职位等）
2. 修改个人资料
3. 录入学生成绩（可以批量导入）
4. 修改成绩
5. 查询学生成绩（可以根据课程名查询，也可以根据学号查询）
6. 成绩分析统计查询（系统统计分析每门课各个成绩段分布的人数，课程的及格率等，以条形图）

*管理员登陆后可进行的操作：*

1. 管理学生信息（学生的添加删除等，可以批量导入）
2. 管理教师信息（教师的添加删除）
3. 基本信息管理（课程添加删除、班级添加删除、课程安排）
4. 系统用户信息管理（添加或删除系统用户，信息包括用户名、密码和权限等）

## 4.4 数据库的设计与实现

本系统采用mysql数据库，系统数据库名为ccs.数据库中有10张表。下面分别介绍不同的表。

### 4.4.1 数据表的概述

系统中包括：学生表（student）、教师表（teacher）、班级表（class）、课程表（course）、成绩表（score）、学生班级（student\_class）、教师课程（teacher\_class）、教师班级（teacher\_class）、课程安排表（course\_arrange）、系统用户表（user）

### 4.4.2 数据表的结构

1. student（学生表）

学生表主要存取学生的学号、姓名、性别、年龄、专业、入学年份、电话、地址、照片，表结构如表4-1所示：

表4-1 student的结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | studentId | int | 11 | 学生编号 |
| 2 | studentName | varchar | 255 | 学生姓名 |
| 3 | sex | varchar | 255 | 性别 |
| 4 | age | int | 20 | 年龄 |
| 5 | major | varchar | 255 | 专业 |
| 6 | yearSchool | int | 255 | 入学年份 |
| 7 | telephone | varchar | 255 | 电话号码 |
| 8 | address | varchar | 255 | 地址 |
| 9 | photo | varchar | 255 | 照片 |

1. teacher（教师表）

教师表主要存取教师的教师号、姓名、性别、学院、职位、学位、照片

表结构如表4-2所示：

表4-2 teacher表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | teacherId | int | 11 | 教师编号 |
| 2 | teacherName | varchar | 255 | 教师姓名 |
| 3 | sex | varchar | 255 | 性别 |
| 4 | college | varchar | 255 | 学院 |
| 5 | professionalTitle | varchar | 255 | 职称 |
| 6 | degree | varchar | 255 | 学位 |
| 7 | photo | varchar | 255 | 照片 |

1. class（班级表）

班级表主要存取班级号、班级名称、学院，表结构如表4-3所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | classId | int | 11 | 班级编号 |
| 2 | className | varchar | 255 | 班级名称 |
| 3 | ofCollege | varchar | 255 | 学院 |

1. score（成绩表）

成绩表主要存取学生的学号、姓名、课程号、课程名、成绩。表结构如表4-4所示：

表4-4 score表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | studentId | int | 11 | 学生编号 |
| 2 | courseId | int | 11 | 课程编号 |
| 3 | studentName | varchar | 255 | 学生姓名 |
| 4 | courseName | varchar | 255 | 课程名称 |
| 5 | grade | int | 5 | 分数 |

1. student\_class（学生\_班级表）

学生班级班级表主要存取学生学号、学生姓名、班级号、班级名，表结构如表4-5所示：

表4-5 student\_class表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | studentId | int | 20 | 学生编号 |
| 2 | studentName | varchar | 255 | 学生姓名 |
| 3 | classId | int | 11 | 班级编号 |
| 4 | className | varchar | 255 | 班级名称 |

1. teacher\_class（教师\_班级表）

教师班级表主要存取教师号、教师姓名、班级号、班级名称，表结构如表4-6所示：

表4-6 teacher\_class表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | teacherId | int | 11 | 教师编号 |
| 2 | teacherName | varchar | 255 | 教师姓名 |
| 3 | classId | int | 11 | 班级编号 |
| 4 | className | varchar | 255 | 班级名称 |

1. teacher\_course（教师课程表）

教师课程表主要存取教师号、教师名称、课程号、课程名，结构如表4-7所示：

表4-7 teacher\_course表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | teacherId | int | 11 | 教师编号 |
| 2 | teacherName | varchar | 255 | 教师姓名 |
| 3 | courseId | int | 11 | 课程编号 |
| 4 | courseName | varchar | 255 | 课程名称 |

1. course\_arrange（课程安排表）

课程安排表是存取课程安排信息。主要存取教师号、教师名、课程号、课程名、班级号、班级名。结构如表4-8所示：

表4-8 course\_arrange表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | classId | int | 11 | 班级编号 |
| 2 | className | varchar | 255 | 班级名称 |
| 3 | courseId | int | 11 | 课程编号 |
| 4 | courseName | varchar | 255 | 课程名称 |
| 5 | teacherId | int | 11 | 教师编号 |
| 6 | teacherName | varchar | 255 | 教师姓名 |

1. course（课程表）

课程表存取的是课程信息，包括课程号、课程名、学分，结构如表4-9所示：

表4-9 course表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | courseId | int | 11 | 课程编号 |
| 2 | courseName | varchar | 255 | 课程名称 |
| 3 | credit | float | 10 | 课程学分 |

1. user（系统用户表）

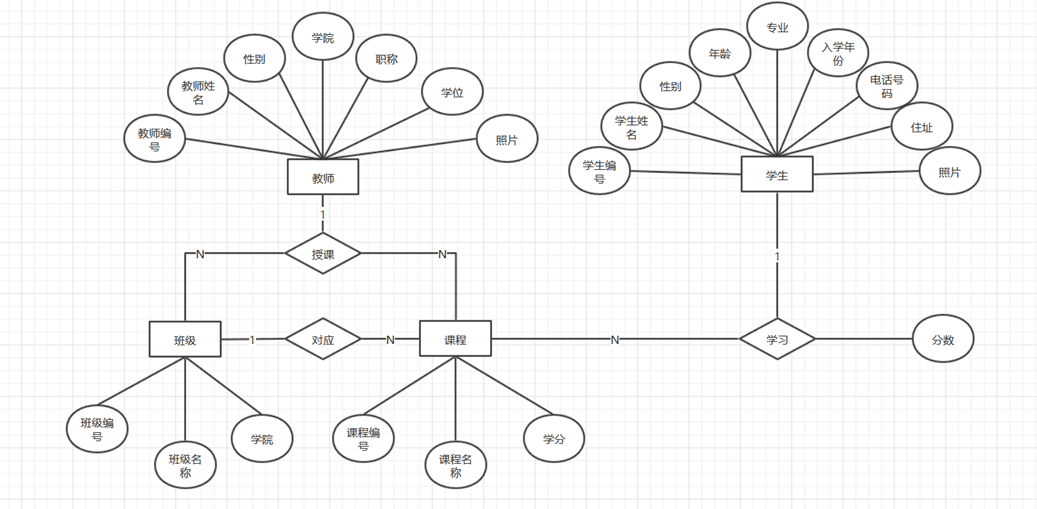
系统用户表存取的是该系统的用户的账号密码信息，结构如表4-10所示：

表4-10 user表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名** | **类型** | **长度（字符）** | **描述** |
| 1 | username | varchar | 255 | 用户名 |
| 2 | password | varchar | 255 | 密码 |
| 3 | superuser | varchar | 255 | 超级用户 |

### 4.4.3 ER图（实体-联系图）

系统整体的实体关系图：



1. 班级信息实体E-R图，如图4-2所示：

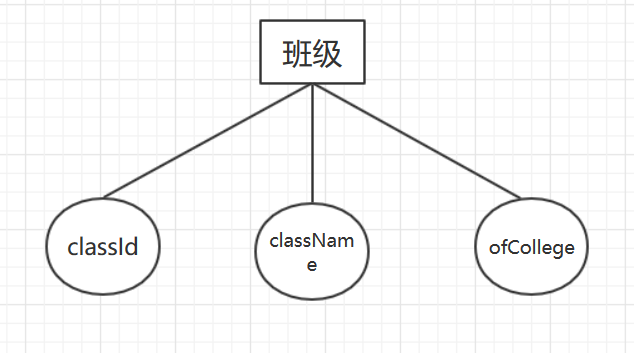


图4-2 班级实体属性图

1. 学生信息实体E-R图，如图4-3所示：

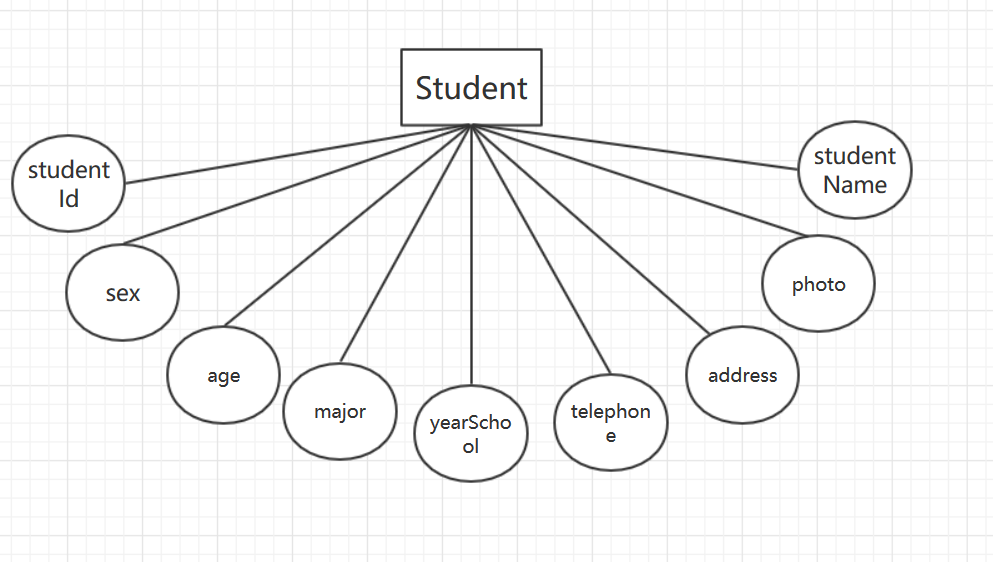


图4-3 学生信息实体属性图

1. 教师信息实体E-R图，如图4-4所示：

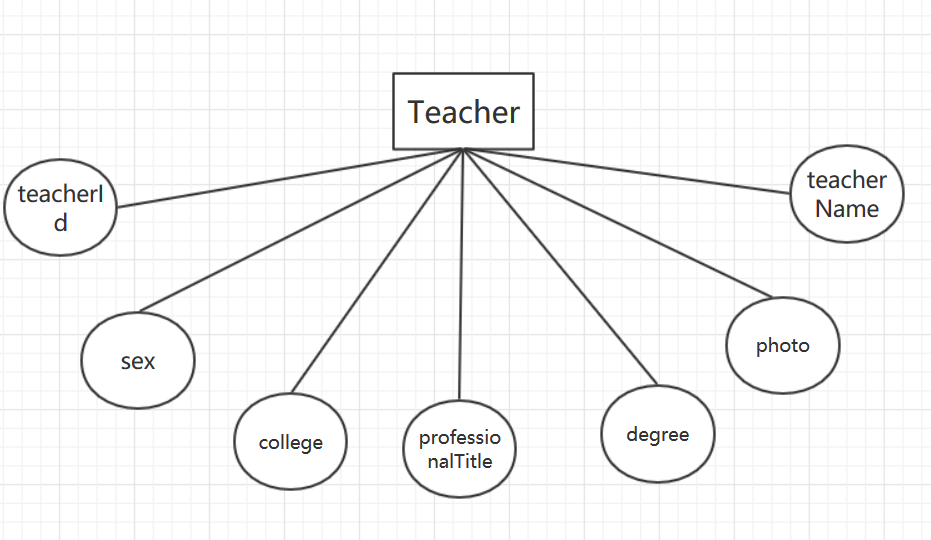


图4-4 教师信息实体属性图

1. 课程信息实体E-R图，如图4-5所示：

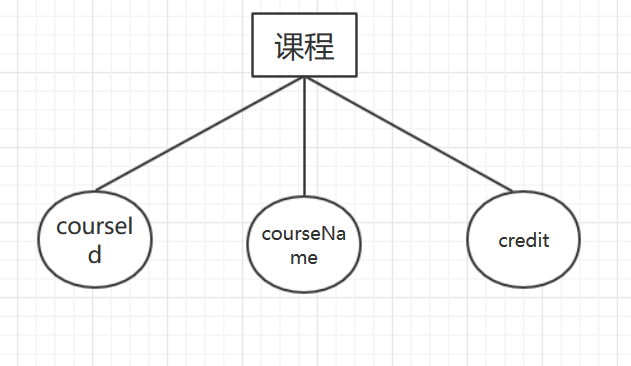


图4-5 课程信息实体属性图

1. 成绩信息实体E-R图，如图4-6所示：

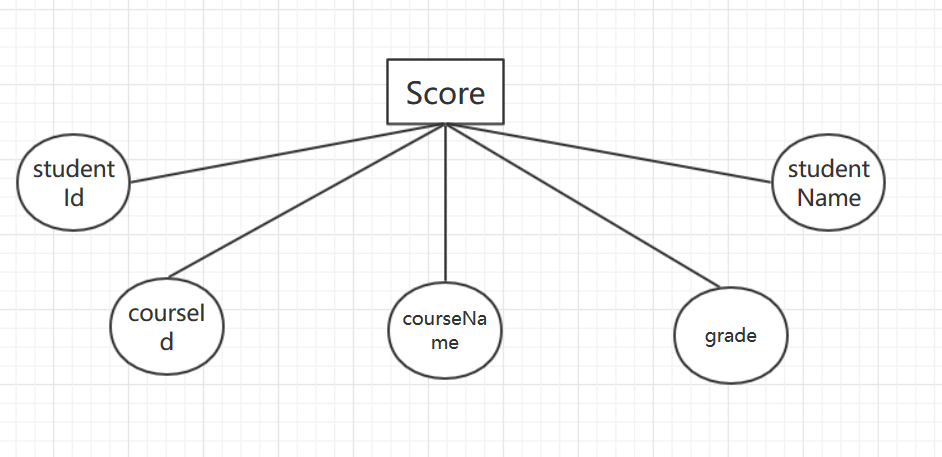


图4-6 成绩信息实体属性图

1. 课程安排信息实体E-R图，如图4-7所示：

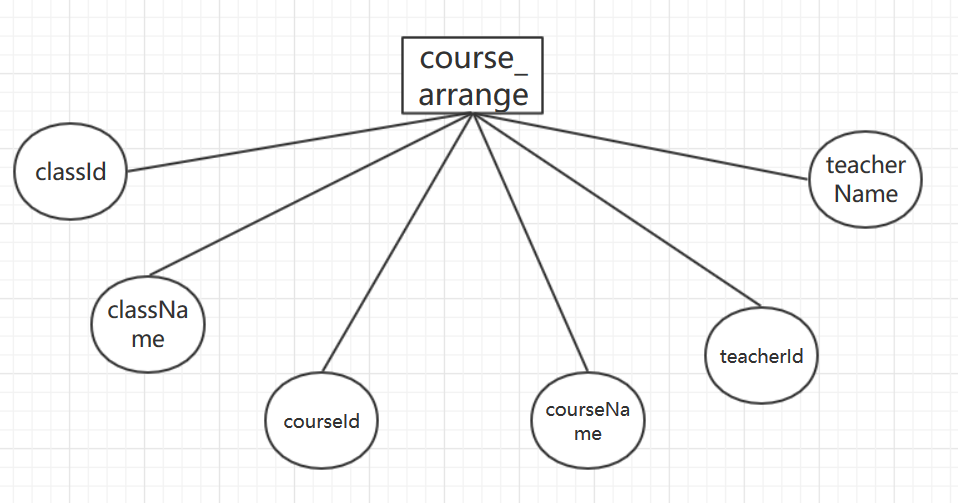


图4-7 课程安排信息实体属性图

1. 学生班级信息实体E-R图，如图4-8所示：

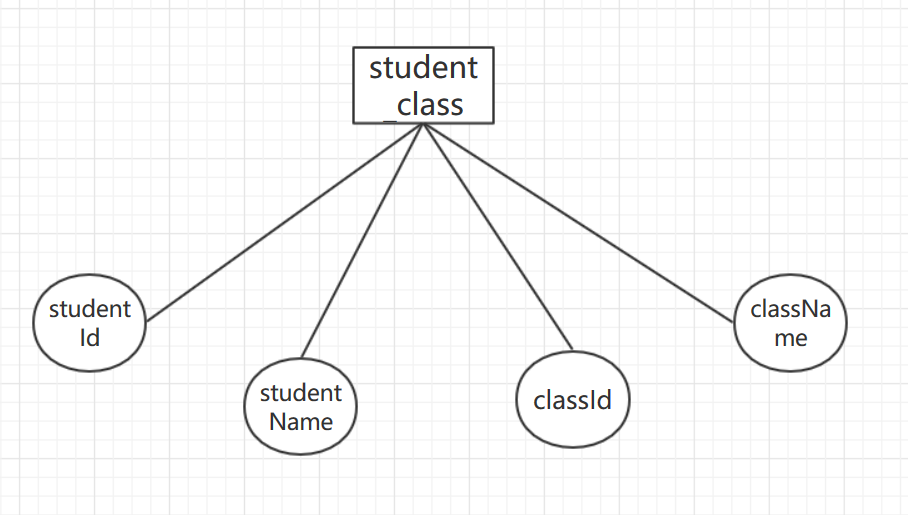


图4-8 学生班级信息实体属性图

1. 教师班级信息实体E-R图，如图4-9所示：

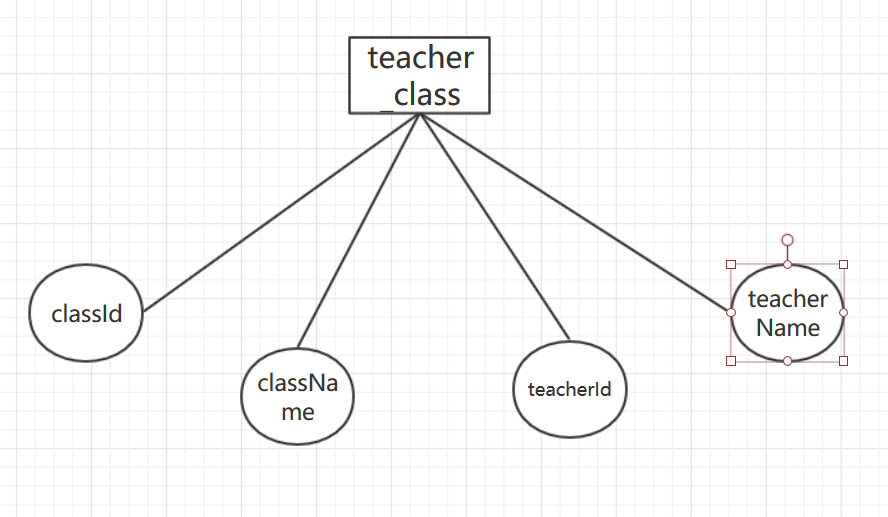


图4-9 教师班级联合实体属性图

1. 教师课程实体E-R图，如图4-10所示：

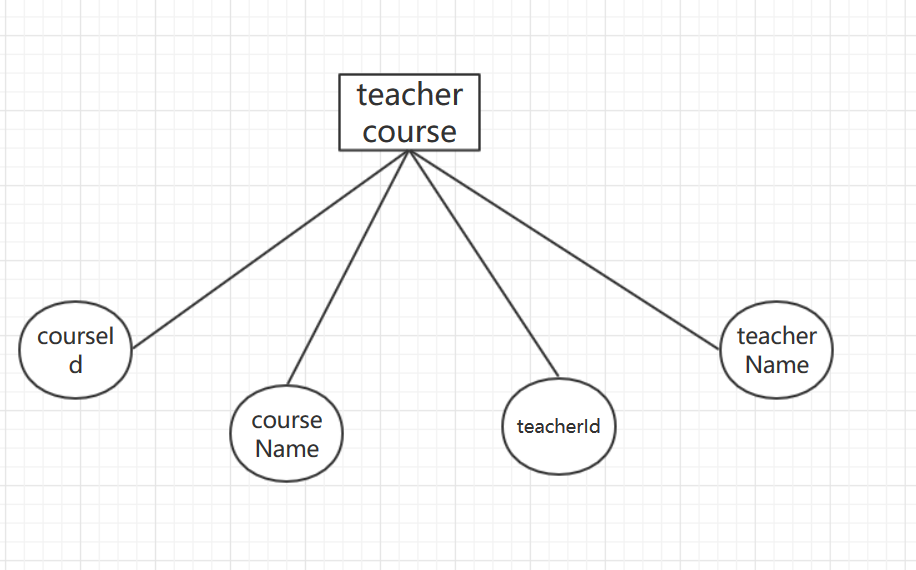


图4-10 教师课程实体关系图

1. 系统用户信息实体E-R图，如图4-11所示：

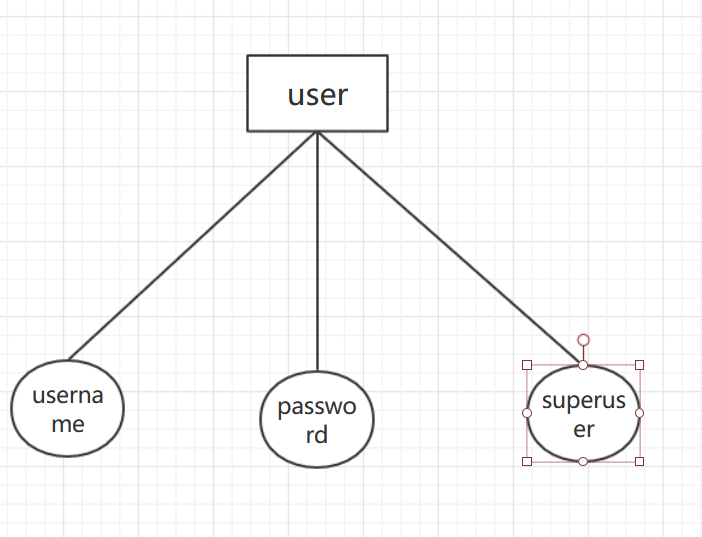


图4-11 系统用户实体属性图

## 4.5 系统活动图

活动图（activity diagram，动态图）是阐明了业务[用例](https://baike.baidu.com/item/%E7%94%A8%E4%BE%8B/163511" \t "https://baike.baidu.com/item/%E6%B4%BB%E5%8A%A8%E5%9B%BE/_blank)实现的工作流程。业务工作流程说明了业务为向所服务的业务主角提供其所需的价值而必须完成的工作。业务用例由一系列活动组成，它们共同为业务主角生成某些工件。工作[流程](https://baike.baidu.com/item/%E6%B5%81%E7%A8%8B/31013" \t "https://baike.baidu.com/item/%E6%B4%BB%E5%8A%A8%E5%9B%BE/_blank)通常包括一个基本工作流程和一个或多个备选工作流程。工作流程的结构使用活动图来进行说明。

### 4.5.1 学生活动图

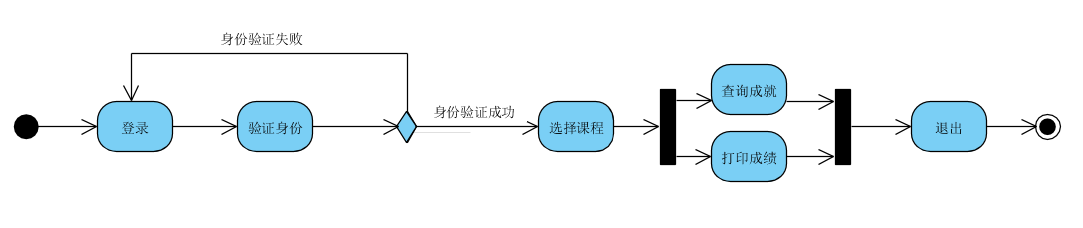


图4-12 学生活动图

### 4.5.2 教师活动图

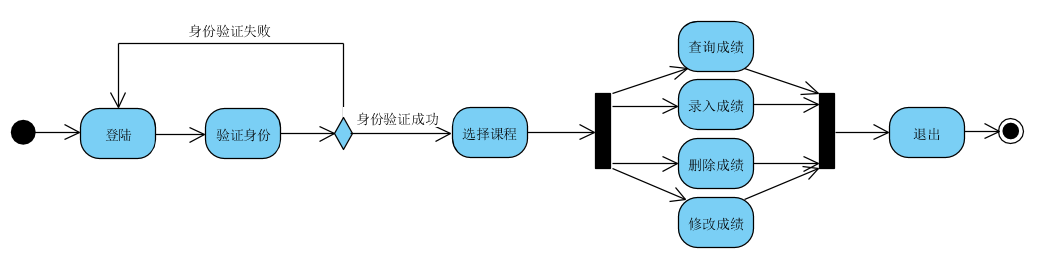


图4-14 教师活动图

### 4.5.3 管理员活动图

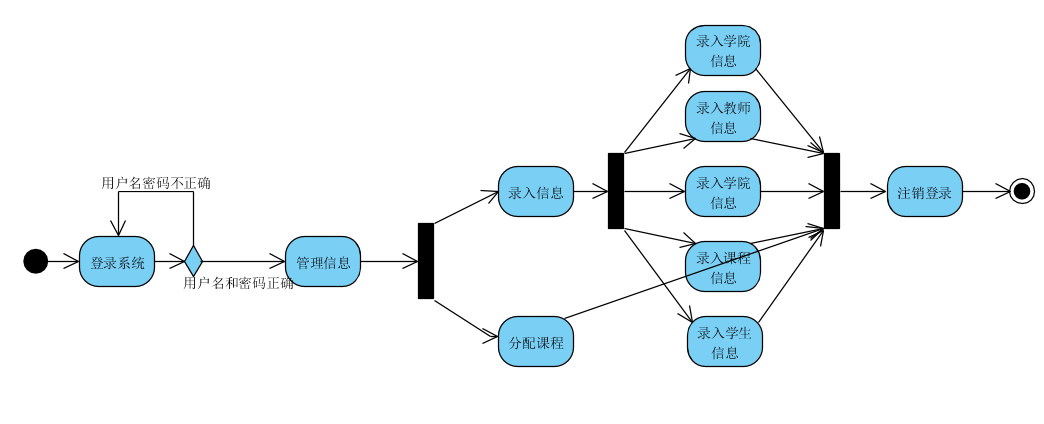


图4-15 管理员活动图

5 详细设计与实现

完成系统分析与概要设计后，系统的主要功能已经确定，最后通过编写代码来实现系统的设计。本章主要介绍学生成绩管理系统的实现，从而对系统有了视觉上的了解。

## 5.1 代码约定

### 5.1.1 前端代码约定

1. 使用正确的文档类型：始终在文档的首行声明文档类型：<!DOCTYPE html>
2. 使用小写元素名：混合大小写名称容易混乱；
3. 关闭所有 HTML 元素，空的也要关闭
4. 使用小写属性名
5. 属性值加引号（如果属性值包含值，则必须使用引号）
6. 始终对图像使用alt属性
7. 始终定义图像尺寸，浏览器会在图像加载之前为图像预留空间，可减少闪烁，
8. 避免长代码行，关键代码处添加注释
9. 用简单的语法链接样式表与JavaScript，尽量不要写在一个文件里
10. 文件命名使用小写，避免大小写混用

### 5.1.2 后端代码约定

1. 文件编码：源文件编码格式为UTF-8。
2. 包名使用小写字母
3. import不要使用通配符。即，不要出现类似这样的import语句：import java.util.\*;
4. 列长限制：一个项目可以选择一行120个字符的列限制，超出列长限制时换行，换行时缩进至少4个空格，缩进不要用tab。
5. 注释：注释应少而精，代码的关键处应该有注释, 注释不能误导读者。
6. 变量声明：每次只声明一个变量，不要使用组合声明，比如int a, b;需要变量时才声明，并尽快进行初始化。
7. 命名约定：命名应该见名知意、简洁，避免拼音与英文混用。
8. 命名风格：类名以UpperCamelCase风格编写；非常量字段名、方法名、参数名、局部变量名以lowerCamelCase风格编写；常量名以CONSTANT\_CASE风格编写
9. 类成员顺序：每个类应该以某种逻辑去排序它的成员，维护者应该要能解释这种排序逻辑。比如，新的方法不能总是习惯性地添加到类的结尾。
10. 使用log而不是System.out.println()
11. 大括号与if, else, for, do, while语句一起使用，即使只有一条语句(或是空)，也应该把大括号写上。
12. 减少代码嵌套： (1)合并条件；(2)利用return以省略后面的else；(3)使用子方法
13. 异常处理：捕获的异常不能忽视，典型的响应方式是打印日志。

## 5.2 登录模块

### 5.2.1 登录模块功能和技术实现介绍

该模块是进入系统的统一入口，用户需要输入用户名和密码信息，验证通过后才能进入主系统界面。在这里，系统可以根据用户的账号识别出用户的身份，进而导航到不同的主操作界面。

技术：判断用户输入的信息与数据库中的用户信息进行对比，判断该用户是否输入正确的验证信息，如果身份信息正确，则根据其权限等级导航到对应的主界面。

此外用到filter技术，防止通过在URL地址栏中直接输入统一资源定位符获取到敏感页面。

5.2.2 截图展示



图5-1 登录界面

5.2.3 关键代码

//验证用户身份信息并匹配不同主页面

**public void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  
 String username = request.getParameter(**"username"**);  
 String password = request.getParameter(**"password"**);  
 System.***out***.println(username);  
 System.***out***.println(password);  
   
 String str = **loginService**.login(username, password);  
 HttpSession session = request.getSession();  
 session.setAttribute(**"username"**, username);  
 session.setAttribute(**"superuser"**, str);  
 **if** (str != **null**) {  
 String s=**null**;  
 **switch** (str){  
 **case "1"**:  
 s = **"student"**;  
 **break**;  
 **case "2"**:  
 s = **"teacher"**;  
 **break**;  
 **case "3"**:  
 s = **"admin"**;  
 **break**;  
  
 }  
 String path =**"/jsp/"**+s+**"/framework"** + s + **".jsp"**;  
 request.getRequestDispatcher(request.getContextPath() +path).forward(request,response);  
 } **else** {  
  
 response.sendRedirect(request.getContextPath() + **"/index.jsp"**);  
  
 }

//过滤器：

**public void** doFilter(ServletRequest servletRequest, ServletResponse servletResponse, FilterChain filterChain) **throws** IOException, ServletException {  
 HttpServletRequest httpServletRequest = (HttpServletRequest) servletRequest;  
 HttpSession session = httpServletRequest.getSession();  
 **if** (session.getAttribute(**"username"**) == **null** || session.getAttribute(**"username"**).equals(**""**)) {  
 httpServletRequest.getRequestDispatcher(**"index.jsp"**).forward(servletRequest, servletResponse);  
 }**else** {  
 filterChain.doFilter(servletRequest, servletResponse);  
 }  
}

## 5.3 学生模块

### 5.3.1 查询和修改个人资料

（1）功能介绍

学生可以点击个人资料查看自己的个人信息，同时可以更新自己的照片，同时还有动态时钟显示。点击不同的指针可以查看月份、星期、时、分、秒

1. 截图展示

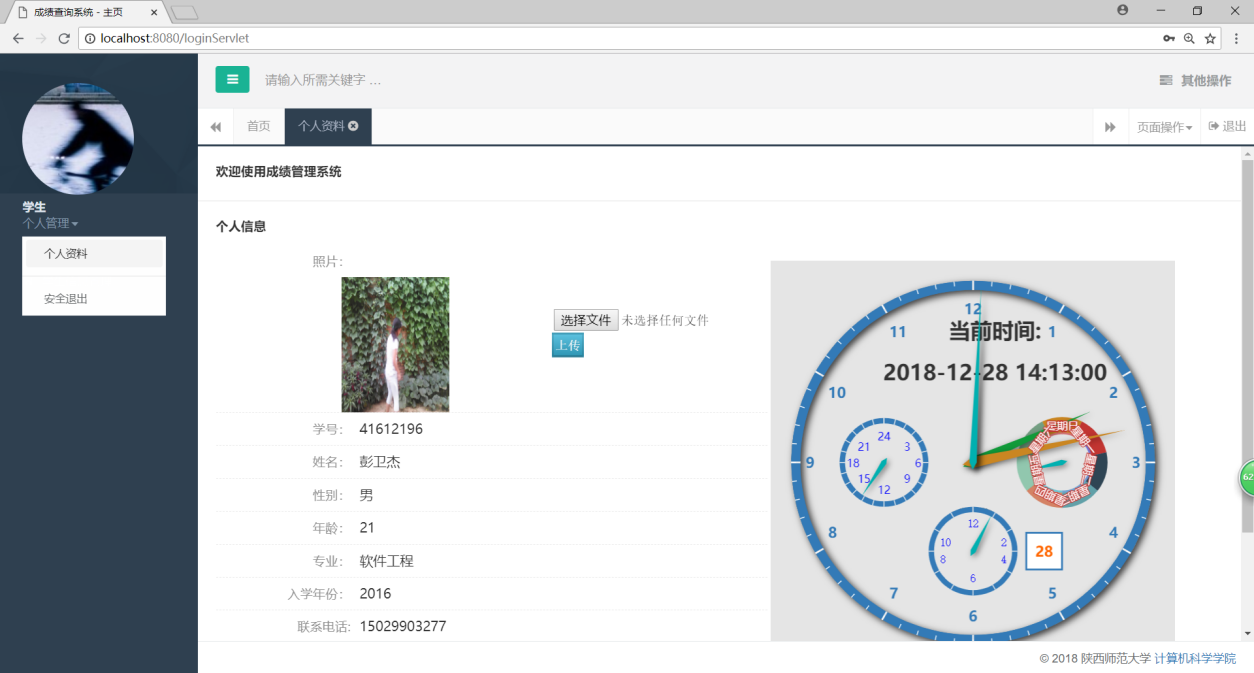


图5-2 学生个人信息查看界面

1. 关键代码

//图片上传（图片上传是上传到服务器，然后将图片的相对路径存到数据库）

**protected void** doPost(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  
 **if** (!ServletFileUpload.*isMultipartContent*(req)) {  
 PrintWriter writer = resp.getWriter();  
 writer.println(**"Error: 表单必须包含 enctype=multipart/form-data"**);  
 writer.flush();  
 **return**;  
 }  
  
 *// 配置上传参数* DiskFileItemFactory factory = **new** DiskFileItemFactory();  
 *// 设置内存临界值 - 超过后将产生临时文件并存储于临时目录中* factory.setSizeThreshold(***MEMORY\_THRESHOLD***);  
 *// 设置临时存储目录* factory.setRepository(**new** File(System.*getProperty*(**"java.io.tmpdir"**)));  
  
 ServletFileUpload upload = **new** ServletFileUpload(factory);  
  
 *// 设置最大文件上传值* upload.setFileSizeMax(***MAX\_FILE\_SIZE***);  
  
 *// 设置最大请求值 (包含文件和表单数据)* upload.setSizeMax(***MAX\_REQUEST\_SIZE***);  
  
 *// 中文处理* upload.setHeaderEncoding(**"UTF-8"**);  
  
 *// 构造临时路径来存储上传的文件  
 // 这个路径相对当前应用的目录* String uploadPath = getServletContext().getRealPath(**"/"**) + File.***separator*** + ***UPLOAD\_DIRECTORY***;  
  
 *// 如果目录不存在则创建* File uploadDir = **new** File(uploadPath);  
 **if** (!uploadDir.exists()) {  
 uploadDir.mkdir();  
 }

//时钟显示

（调用Echart图表，加载到该jsp文件中）

部分代码：

**option3** = {  
 **tooltip**: {  
 *// formatter: "{a}：{c}"* **backgroundColor**: **'#fff'**,  
 **borderColor**: **'#f60'**,  
 **borderWidth**: **'1px'**,  
 **textStyle**: {  
 **color**: **'#333'** },  
 formatter: **function**(param) {  
 **var** time = ***Math***.floor(param.**value**);  
 **if** (param.**seriesIndex** === 0) {  
 **return '<em style="color:'** + param.**color** + **';">当前小时:'** + time + **'</em>'** }  
 **if** (param.**seriesIndex** === 1) {  
 **return '<em style="color:'** + param.**color** + **';">当前星期:'** + time + **'</em>'** }  
 **if** (param.**seriesIndex** === 2) {  
 **return '<em style="color:'** + param.**color** + **';">当前月份:'** + time + **'</em>'** }

### 5.3.2 查询成绩

（1）功能介绍

学生点击查询成绩可以看到自己每门课的成绩

1. 截图



图5-3 学生查询成绩界面

1. 关键代码

从数据库中获取到成绩，在jsp页面中通过jstl中的遍历，将信息逐条显示。

**<%** String username = (String) session.getAttribute(**"username"**);  
 List<Score> scoreList = (List) session.getAttribute(**"scoreList"**);  
 session.setAttribute(**"scoreList"**, scoreList);  
**%>**<**c:forEach var="sl" items="${**scoreList**}"**>  
 <**tr**>  
 <**td**>**${**sl.studentId**}**</**td**>  
 <**td**>**${**sl.courseId**}**</**td**>  
 <**td**>**${**sl.studentName**}**</**td**>  
 <**td**>**${**sl.courseName**}**</**td**>  
 <**td**>**${**sl.grade**}**</**td**>  
 </**tr**>  
</**c:forEach**>

## 5.4 教师模块

### 5.4.1 查询和修改个人资料

（1）功能介绍

教师可以查看个人信息，可以更新个人照片

1. 功能截图

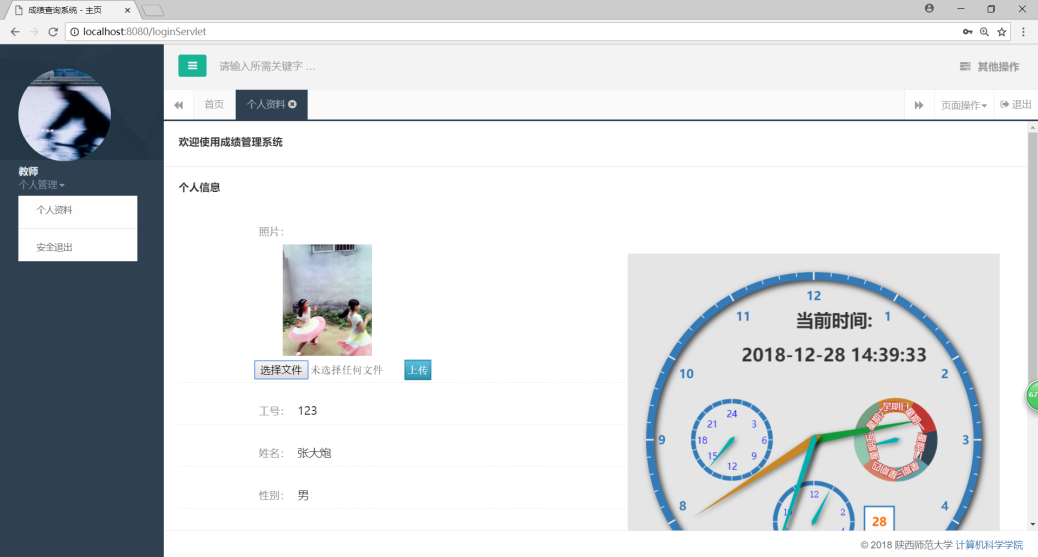


图5-4 教师个人信息界面

### 5.4.2 录入学生成绩

（1）功能介绍

教师可以通过两种方式录入学生成绩：

手动一个一个输入和导入Excel成绩单

1. 运行截图

手动输入截图：



图5-6 教师录入成绩界面

Excel导入成绩结果：



图5-7成绩录入完成界面

### 5.4.3 查询和修改成绩

（1）功能介绍

教师可以输入课程名查询这门课的成绩，也可以输入学号查询某个学生的成绩，还可以直接修改成绩

1. 关键技术

修改成绩时，用到了jQuery技术，遍历选中的某条记录，将其table标签变为input标签，并将原来的值保存并显示到input标签中。另外按钮的状态也在改变，有修改变为确定，点击“确定”，按钮重新变为“修改”。

Ajax传值和json：将修改后的数据传到后端时，采用了ajax传值方式，将修改后的数据以json的形式传到后端，在后端通过解析，然后保存到数据库，将修改是否成功信息返回前端。

（3）运行截图



图5-8教师查询某门课的学生成绩



图5-9教师查询某个学生的成绩



图5-10 教师修改成绩

5.4.4 统计分析学生成绩

（1）功能介绍

教师在该功能中可以查看不同课程的各分数段人数以及该门课程的及格率。具体信息通过条形图和水球图展示。还可以同时分析多门课程的最高分、最低分、平均分等，可以查看表格数据、折线统计分析、条形堆叠统计分析、条形图统计分析等多种方式浏览分析结果，并且可以将分析结果下载为图片。

在多门课成绩统一分析时，可以点击不同按钮，分别显示最高分、最低分、平均分等

通过点击图表右侧的按钮，可以查看表格数据、折线统计分析、条形堆叠统计分析、条形图统计分析等多种方式浏览分析结果，并且可以将分析结果下载为图片。

（2）关键技术

该功能用到了Echart技术，ajax技术，加载页面时通过ajax请求学生的成绩信息，将获取的值，赋给有JavaScript写好变量中。

1. 运行截图



图5-11 某门课的成绩分析图

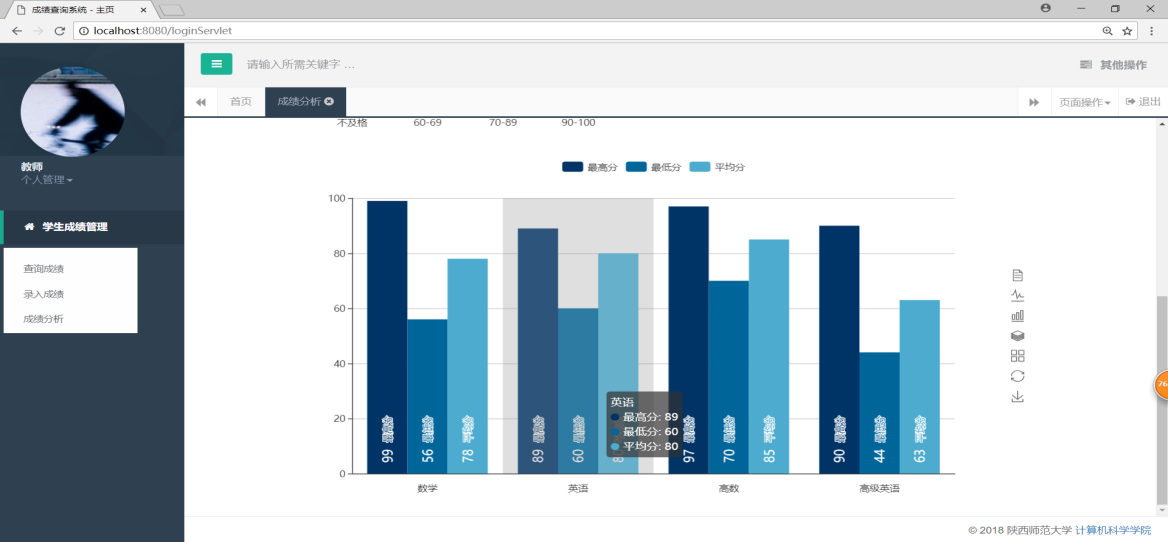


图5-12 该教师多门课成绩统一分析图

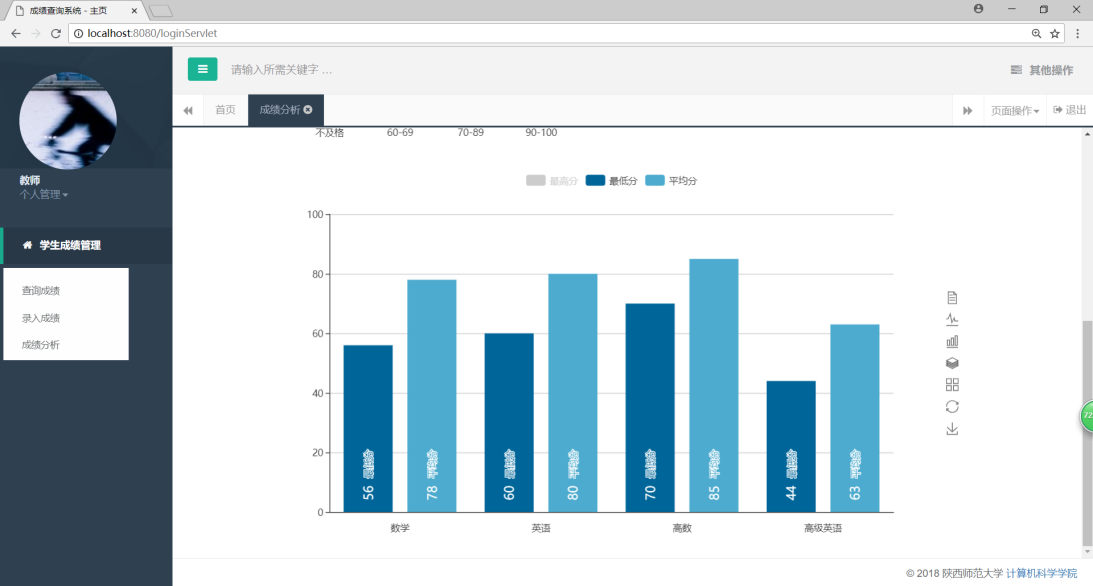


图5-13 最高分和平均分统计图

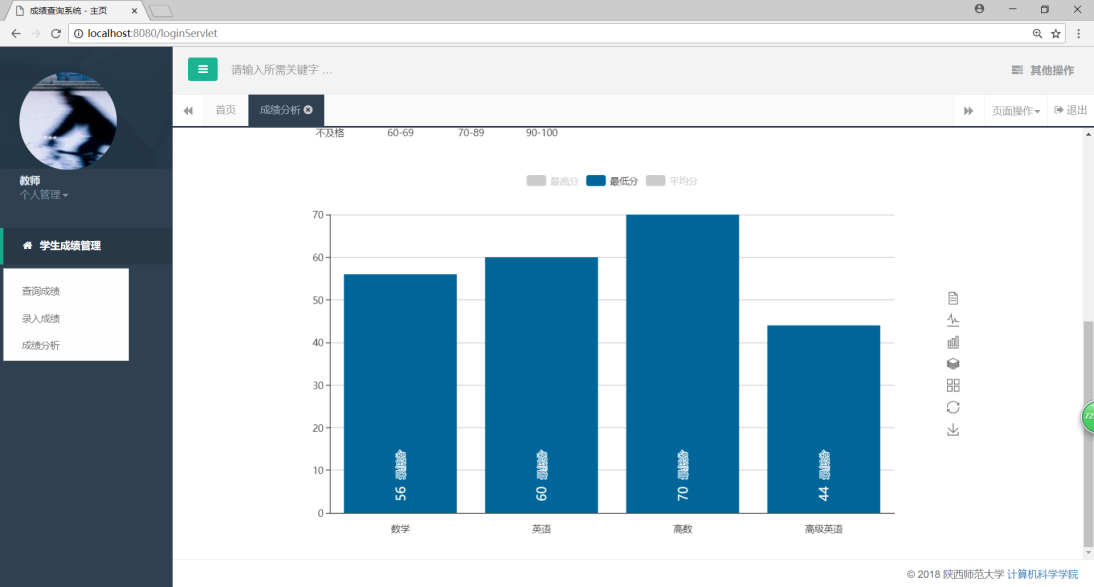


图5-14最低分的统计图



图5-15表格视图



图5-16 折线统计视图

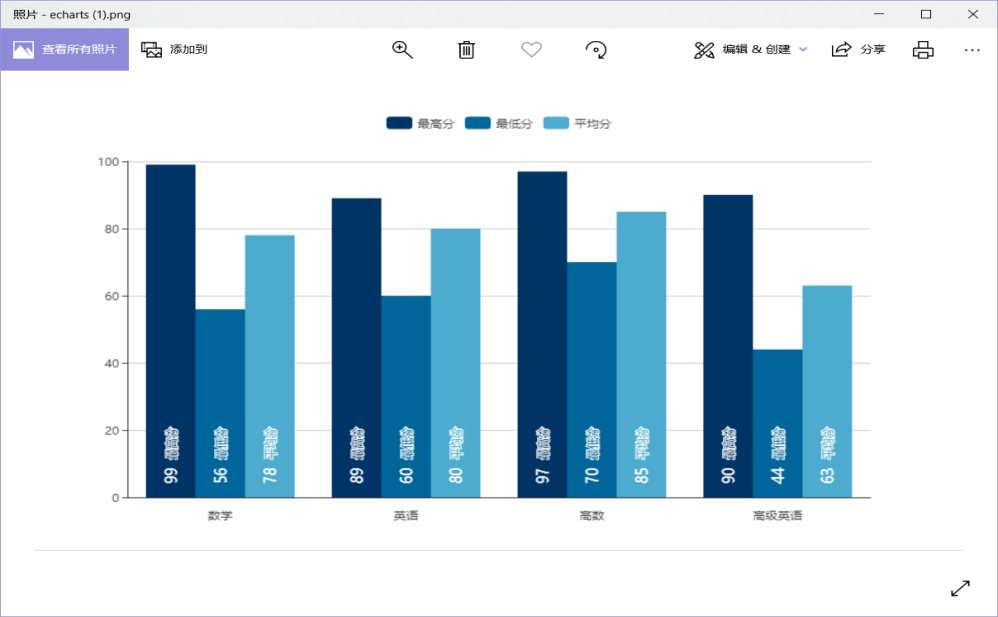


图5-17 下载为图片

1. 关键代码

//分数统计

passrate=(**float**) (num2+num3+num4)/(num1+num2+num3+num4);  
System.***out***.println(**"及格率为"** + passrate);  
  
list.add(**new** ScorePeople(**"不及格"**, num1, passrate));  
list.add(**new** ScorePeople(**"60-69"**, num2));  
list.add(**new** ScorePeople(**"70-89"**, num3));  
list.add(**new** ScorePeople(**"90-100"**, num4));  
ObjectMapper mapper = **new** ObjectMapper();*//提供java-json相互装换功能*

*//综合分析***var *posList*** = [  
 **'left'**, **'right'**, **'top'**, **'bottom'**,  
 **'inside'**,  
 **'insideTop'**, **'insideLeft'**, **'insideRight'**, **'insideBottom'**,  
 **'insideTopLeft'**, **'insideTopRight'**, **'insideBottomLeft'**, **'insideBottomRight'**];

## 5.5 管理员模块

主要有四大功能木块：学生信息管理、教师信息管理、基本信息管理（班级、课程、课程安排）、系统用户信息管理

### 5.5.1 学生信息管理

（1）功能介绍

管理员可以查看、添加和删除学生信息，有两种添加方式：添加单个学生和批量导入学生信息。

1. 关键技术

多级联动——在单个添加学生信息使，在选择学院、专业和班级时，使用了多级联动技术，即选择某个学院、专业的选项显示不同。班级也一样，比如选择了计算机科学学院，那么专业选项中只包含“计算机科学与技术（师范），计算机科学与技术（创新实验班），软件工程，信息系统与信息管理”4个选项。

批量上传——管理员可以通过导入Excel文件，批量导入学生信息。

1. 运行截图



图5-18 管理员添加学生

1. 关键代码

//读取Excel文件导入学生信息

**public class** ReadStudentsExcel {  
  
 **public** List readStudentExcel(File file) {  
 List<Student> studentList = **new** ArrayList<Student>();  
 **try** {  
 InputStream inputStream = **new** FileInputStream(file.getAbsolutePath());  
 Workbook workbook = Workbook.*getWorkbook*(inputStream);  
 **int** sheet\_size = workbook.getNumberOfSheets();  
  
 List<String> eachStudent = **new** ArrayList<>();  
 Student student = **null**;  
 **for** (**int** index = 0; index < sheet\_size; index++) {  
 Sheet sheet = workbook.getSheet(index);  
 **for** (**int** i =1 ; i < sheet.getRows(); i++) {  
 **for** (**int** j = 0; j < sheet.getColumns(); j++) {  
 String cellinfo = sheet.getCell(j, i).getContents();  
 System.***out***.print(cellinfo+**'\t'**);  
 eachStudent.add(j,cellinfo);  
 }

### 5.5.2 教师信息管理

（1）功能介绍

管理员可以查看、添加和删除教师信息。

1. 运行截图

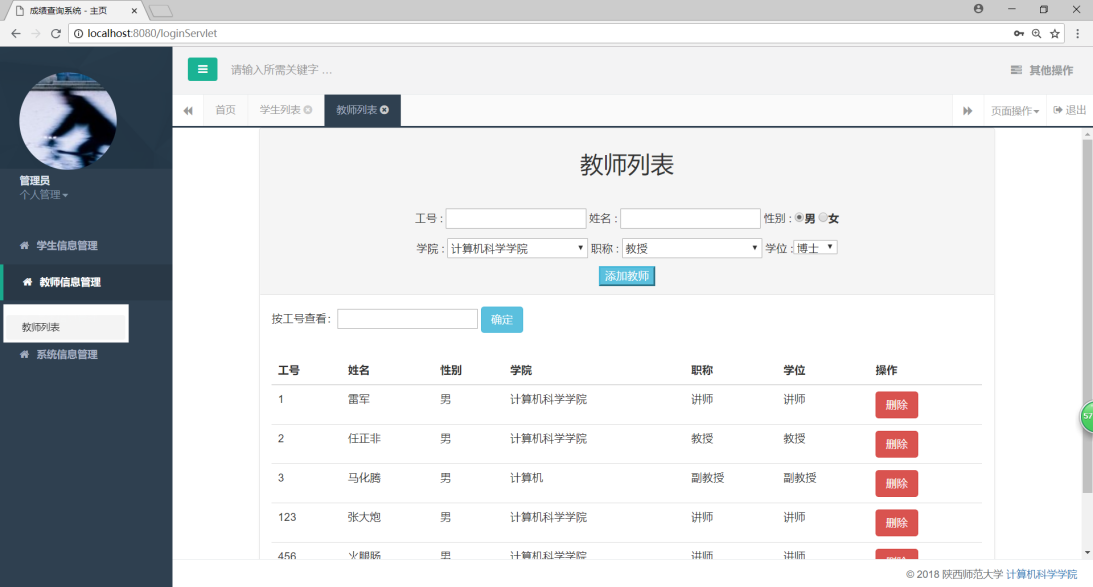


图5-19 管理员管理教师信息

### 5.5.3 基本信息管理

（1）班级信息管理

功能介绍：查询、添加、删除班级

运行截图：

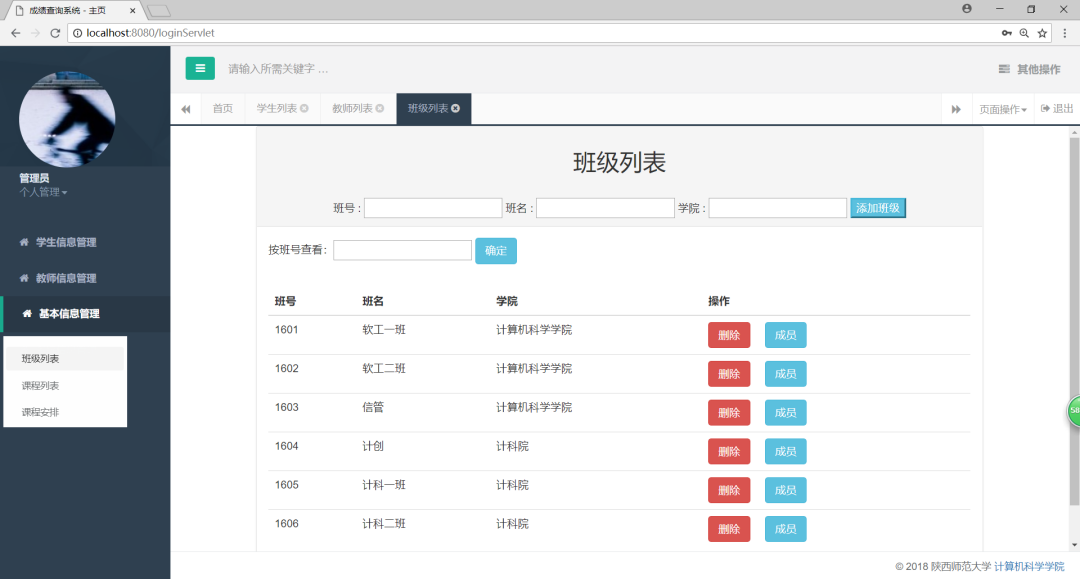


图5-20管理员管理班级信息

（2）课程信息管理

功能介绍：查询、添加、删除课程

运行截图：

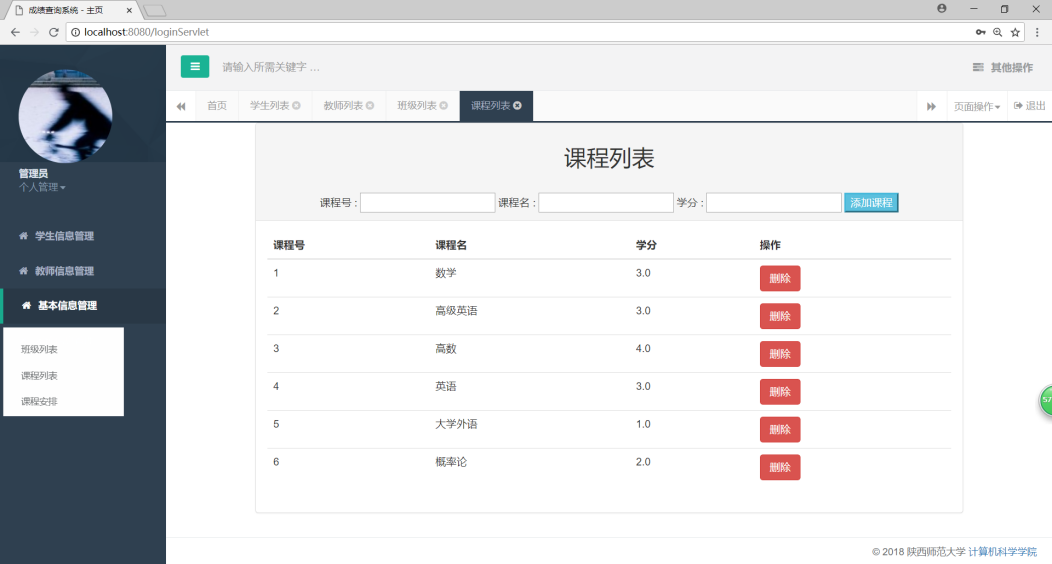


图5-21 管理员管理课程信息

1. 安排课程

功能介绍：管理员指定班级、课程、教师来安排课程

运行截图：

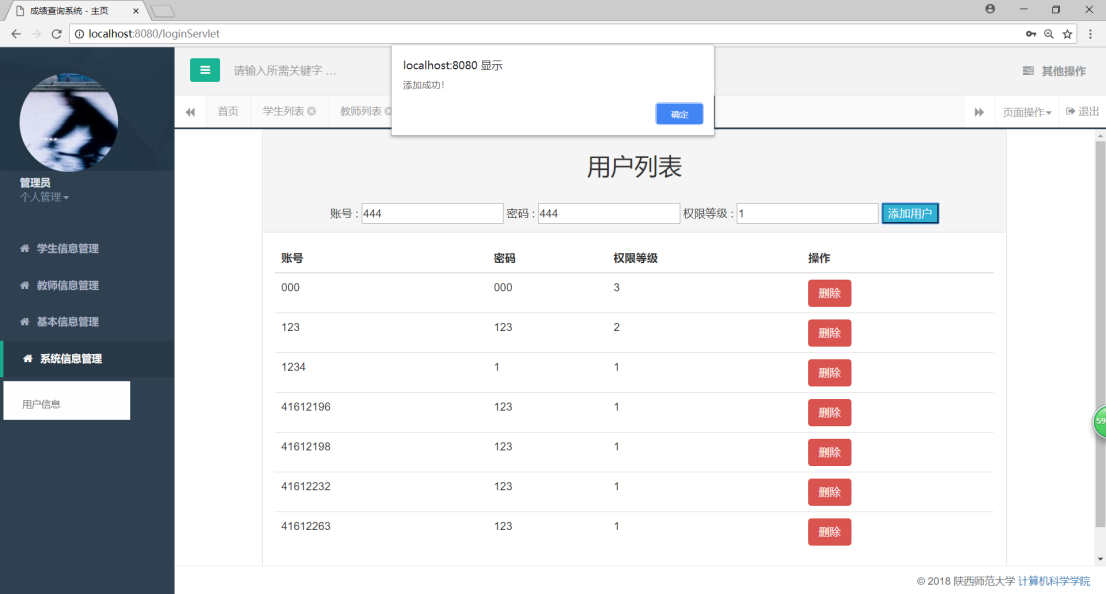


图5-22 管理员安排课程

### 5.5.4 管理系统用户

功能介绍：添加、删除系统用户。

运行截图：



6 项目总结

项目名称：学生成绩管理系统

开发周期：2018年11月4日——2018年12月28日星期五

开发人员：彭卫杰 41612196

张靖漪 41612263

汪 鹏 41612232

张 铖 41612196

小组分工：需求分析：全组人员共同参与

后端设计：教师功能模块由彭卫杰设计开发

管理员功能模块由汪鹏设计开发

学生功能模块由张铖设计开发

前端设计：主要由张靖漪负责设计开发

黑盒测试：小组成员参与测试，另外邀请其他同学参与测试

项目文档：张靖漪、汪鹏、张铖、彭卫杰共同完成。

缺陷与不足：

该系统基本实现了成绩管理系统的常用功能，但是还未测试高并发状态系统运行的流畅性及健壮性，另外成绩分析统计方面部分功能还未实现，但这只是时间问题，技术上是可以实现的。

感谢：感谢武杰老师、张汇泉老师和祁超老师的指点和建议，感谢小组中的每一个人！加油