

# 西安电子科技大学

## 软件工程专业大作业

--学生选课系统设计

### 小组成员

学号：16020510019 姓名：周明祖 成绩：85

学号：16020510014 姓名：曾杨克 成绩：85

学号：16020510047 姓名：肖梦龙 成绩：85

指导教师：林杰

---

## 第一章 绪论

### 1.1 课题背景和意义

随着科学技术的不断提高, 计算机科学日渐成熟, 其强大的功能已为人们深刻认识, 它已进入人类社会的各个领域并发挥着越来越重要的作用. 随着计算机网络技术发展, Web 数据库技术已经成为应用最为广泛的网站架构基础技术在应用系统中, Web 提供了与用户进行通信联络的有效手段, 利用 Web 技术, 实现 Web 服务器与数据库系统的连接, 完成对数据的处理与查询, 用户可以通过操作简单易学的浏览器来查询处理所需要的各种数据.

学生信息管理系统是一个教育单位不可缺少的部分, 它的内容对于学校的决策者和管理者来说都至关重要, 所以学生信息管理系统应该能够为用户提供充足的信息和快捷的查询手段. 但一直以来人们使用传统人工的方式管理文件档案, 这种管理方式存在着许多缺点, 如: 效率低、保密性差, 另外时间一长, 将产生大量的文件和数据, 这对于查找、更新和维护都带来了不少的困难.

随着科学技术的不断提高, 计算机科学日渐成熟, 其强大的功能已为人们深刻认识, 它已进入人类社会的各个领域并发挥着越来越重要的作用. 作为计算机应用的一部分, 使用计算机对学生信息进行管理, 具有手工管理所无法比拟的优点. 例如: 检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等. 这些优点能够极大地提高学生信息管理的效率, 也是企业的科学化、正规化管理, 与世界接轨的重要条件.

### 1.2 本系统主要功能模块

本系统的主要任务就是负责对学生信息管理. 主要用户为老师、学生, 其中, 学生可对自己的信息进行查询, 可以进行选课, 也可以进行删除已选课程, 教师可对学生和课程的信息进行查询, 教师拥有所有的权限, 可以添加删除学生信息,, 课程. 系统提供界面, 操作简单.

为实现这些功能, 系统一个包含如下几个基本模块:

1. 用户登陆模块. 用户可以选择自己的登陆身份.

- 
2. 添加信息模块. 教师可以实现对学生信息和课程信息的添加.
  3. 修改信息模块. 教师可以实现对学生信息和课程信息的修改.
  4. 查询信息模块. 学生和教师可以在此界面进行操作.
  5. 删除信息模块. 教师可以实现对学生和课程信息的删除.

---

## 第二章 学生选课系统需求分析

### 2.1 需求概要

设计内容：

设计开发一个简单的“学生选课系统”，并实现基本的选课功能，查询功能。

主要功能及要求：

- (1) 能够实现学生基本信息的录入、修改、删除等操作，其中学生信息包括班号、学号、性别等；
- (2) 能够实现课程信息的录入、修改、删除等功能，其中课程信息包括课程号、课程名、学分等；
- (3) 能实现选课功能，每个学生有选修学分限制，超出可选学分限制会提示；
- (4) 能方便的对学生选课情况进行查询，可以根据学号、姓名、班级、课程名等多种方式查询；
- (5) 有汇总功能，能对每个学生的选修学分汇总，并对选修学分不足的学生进行筛选。

### 2.2 详细需求分析

对系统功能分析，应包含以下功能：

- 1、身份验证，用以用户登录，分别有教师用户与学生用户；
- 2、教师用户可以录入、修改、删除学生信息、课程信息；
- 3、教师用户可以按编号、名称、等多种方式查询学生信息、课程信息；
- 4、教师用户可以统计学分，并可筛选出选修学分不够的同学；
- 5、学生用户可以查看可选科目、并选课；
- 6、当学分超出一个学生学分上限时，发出提示，并且不能继续选课；
- 7、学生选课结束，可以查看已选课程；

---

## 2.3 该系统实现的大致功能:

1. 用户登陆界面. 该界面可以选择使用者的身份, “, 教师, 学生”. 不同的身份有不同的操作界面和功能权限. 学号和密码输入正确即可登录. 学生管理界面. 提供了学生选课的功能和查询已选课程的功能 2. 教师管理界面. 拥有最高的权限. 允许添加学生信息和课程信息, 提供了对学生学籍信息的查询, 添加, 修改, 删除; 提供了对教师信息的查询, 添加, 修改, 删除; 修改密码等功能.

## 2.4 系统的数据性能需求

学生选课系统主要提供信息查询, 方便管理的网上的信息查阅平台, 学生可以通过该系统查阅与自己所选的课程, 教师可以通过该系统对学生的信息进行添加、删除、修改和查询, 为了保证系统能够长期、安全、稳定、可靠、高效的运行, 学生信息管理系统应该满足以下的性能需求:

### 1. 系统处理的准确性和及时性

系统处理的准确性和及时性是系统的必要性能. 在系统设计和开发过程中, 要充分考虑系统当前和将来可能承受的工作量, 使系统的处理能力和响应时间能够满足各使用者对信息处理的需求.

由于信息查询系统的查询功能对于整个系统的功能和性能完成举足轻重. 作为系统的很多数据来源, 而学生的信息的准确性很大程度上决定了学生选课管理系统的成败. 在系统开发过程中, 必须采用一定的方法保证系统的准确性.

### 2. 系统的开放性和系统的可扩充性

学生选课管理系统在开发过程中, 应该充分考虑以后的可扩充性. 例如科目的改变和学生基本信息的改变, 学生查询的需求也会不断的更新和完善. 所有这些, 都要求系统提供足够的手段进行功能的调整和扩充. 而要实现这一点, 应通过系统的开放性来完成, 既系统应是一个开放系统, 只要符合一定的规范, 可以简单的加入和减少系统的模块, 配置系统的硬件. 通过软件的修补、替换完成系统的升级和更新换代.

---

### 3. 系统的易用性和易维护性

学生信息管理系统是直接面对使用人员的,而使用人员往往对计算机并不是非常熟悉.这就要求系统能够提供良好的用户接口,易用的人机交互界面.要实现这一点,就要求系统应该尽量使用用户熟悉的术语和中文信息的界面;针对用户可能出现的使用问题,要提供足够的在线帮助,缩短用户对系统熟悉的过程.

学生信息管理系统中涉及到的数据是学生的重要的信息,系统要提供方便的手段供系统维护人员进行数据的备份,日常的安全管理,系统意外崩溃时数据的恢复等工作.

### 4. 系统的标准性

系统在设计开发使用过程中都要涉及到很多计算机硬件、软件.所有这些都要符合主流国际、国家和行业标准.例如在开发中使用的操作系统、网络系统、开发工具都必须符合通用标准.如规范的数据库操纵界面、作为业界标准 TCP/IP 网络协议及 ISO9002 标准所要求的质量规范等;同时,在自主开发本系统时,要进行良好的设计工作,制订行之有效的软件工程规范,保证代码的易读性、可操作性和可移植性.

### 5. 系统的响应速度

学生信息管理系统系统在日常处理中的响应速度为秒级,达到实时要求,以及时反馈信息.在进行统计分析时,根据所需数据量的不同而从秒级到分钟级,原则是保证操作人员不会因为速度问题而影响工作效率.

## 2.5 系统功能需求分析

综上所述本系统具体需要完成功能如下:

1.全面展示学生选课管理系统所包含的信息.

2.登陆窗口模块区分教师(管理员)和学生身份.

3.教师登陆模块:教师登陆系统可以录入学生信息和课程信息以及管理课程和学生信息查看学生的学分情况

---

4.学生登陆模块：学生登陆系统后可以选课，并查看自己所选的课程。

该系统使用了先进的数据库技术与数据管理技术,使数据的准确性与安全性得到了很大的提高,且在用户的并行操作与用户管理方面也有了极大地改善.

教师登录模块功能需求

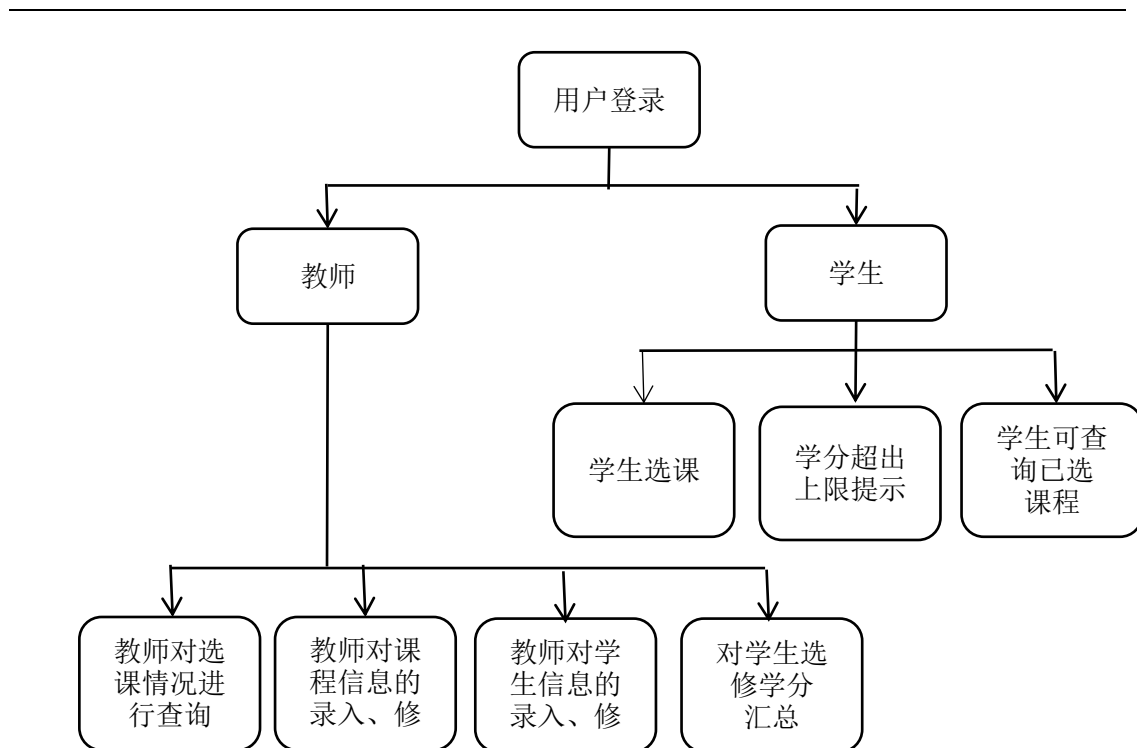
教师登陆系统后可以对学生的个人信息进行查询操作,然后教师可以对学生和课程信息进行增删改查操作。

学生模块功能需求

通过调研得出学生信息管理系统中需要在学生登录或验证身分后,可以自行完成学生的选课以及学生个人信息的查询,也可以进行选课和删除已选课程等操作,方便用户使用.

## 2.6 系统数据流图

根据该系统需求进行分析,使用数据流图示描述数据处理过程.数据流图从数据传递和加工的角度,以图形的方式刻画数据流从输入到输出的传输变换过程.数据流图是结构化系统分析的主要工具,它表示了系统内部信息的流向,并表现系统的逻辑处理功能.



## 2.7 功能模块划分

### 用户验证登录模块

用户根据自己身份，选择相应的登录选项（教师或学生）登录，与数据库中数据相验证，若用户存在，则跳转到相应页面；若用户不存在，则提示密码或用户名错误，跳回登录页面。

### 教师对课程信息的操作模块

教师可以添加课程、删除课程、修改课程，查询课程。修改、删除、添加模块应有修改成功的提示，查询页面应有分页显示功能。

### 教师对学生信息的操作模块

教师可以添加、修改，删除、查询学生信息。对学生信息的修改、删除、添加操作应该有修改成功的提示，对学生的查询应该可以分页显示。

### 模糊查询功能

教师用户可以根据某一个数据项查询学生或课程信息，或是查询含有某一字符的相关元组。

### 学分控制模块

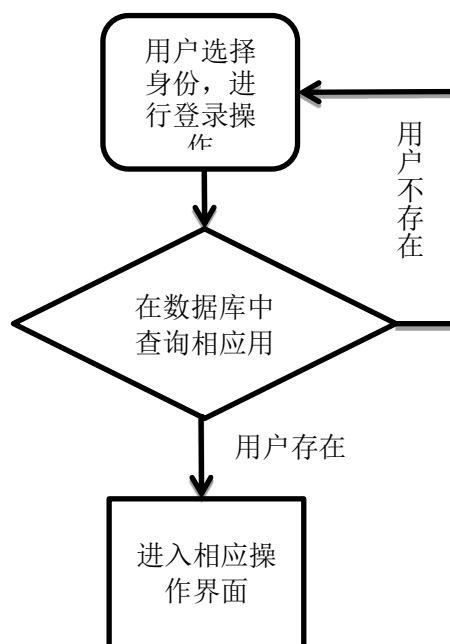


---

本模块可统计选课学分不足或是学分超出上限的学生。

### 学生选课模块

学生可查看所有课程，可选择选课人数未滿课程。选课前要有确认选项。选课不能更改。成功选课后，选课对该学生关闭。该学生只能看到已选课程以及课程信息。



## 第三章 总体设计

### 3.1 系统功能层次图

学生信息管理系统是一个典型的数据库开发应用程序,系统是和数据库相结合.,进入系统,有三个用户,分别是学生、教师和管理员,

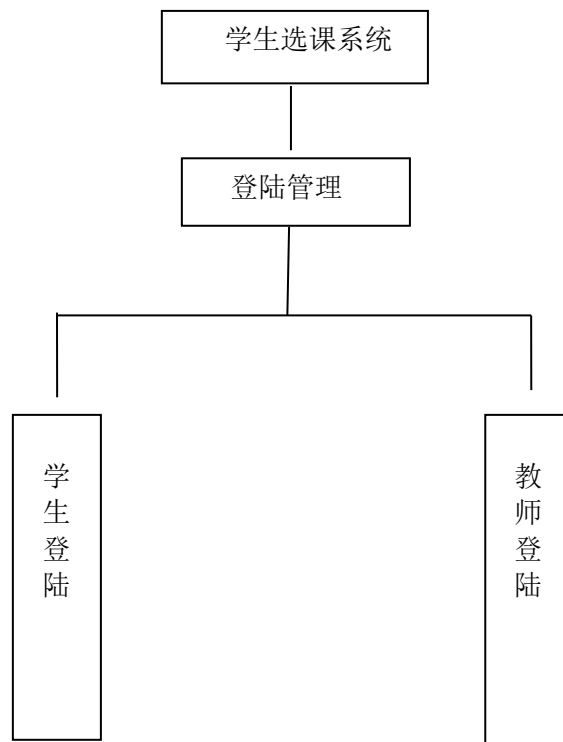


图 3.1 登陆层次图

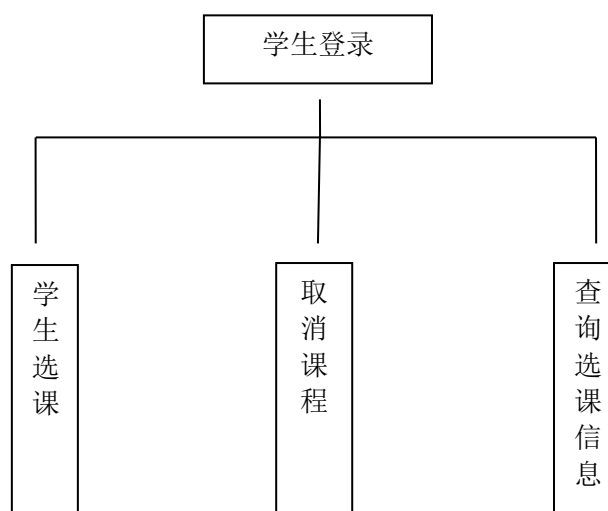


图 3-2 学生管理层次图

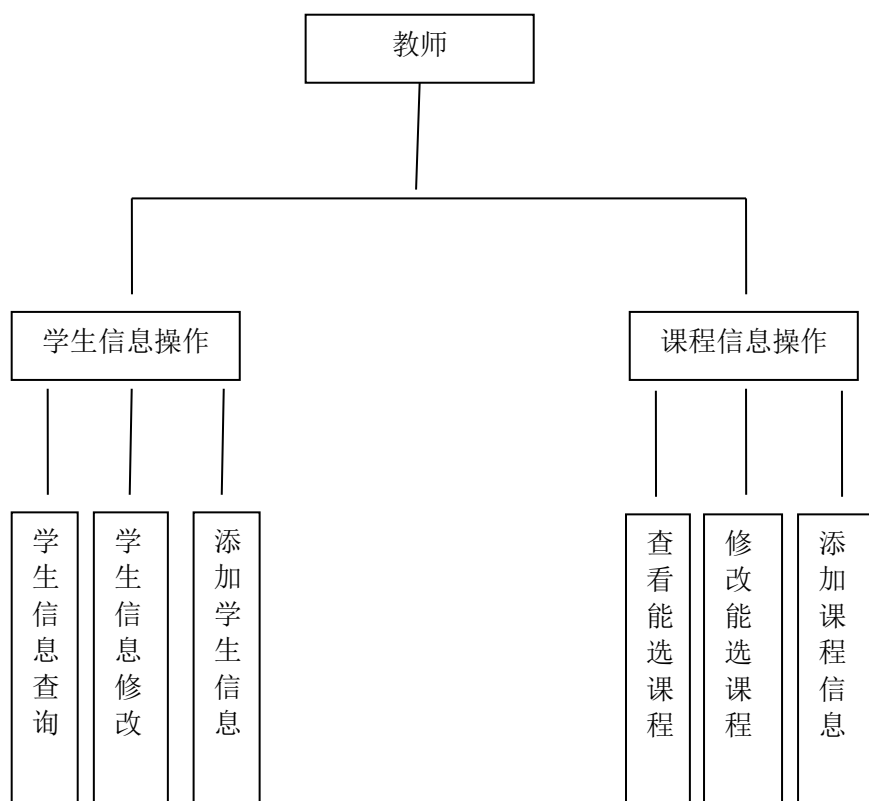


图 3-3 教师管理层次图

## 3.2 设计目标

学生信息管理系统为用户提供信息交流的交互平台,让老师、学生间建立良好联系,主要实现如下目标.

### 1 教师管理模块

1. 基本信息录入, 对系统的所有基本信息进行管理操作.
2. 学生信息操作, 对学生的信息进行管理操作.
3. 教师管理操作, 对所教课程和听课通知进行查询操作

### 2 学生登录管理模块

1. 学生管理操作, 对学生自己选课和删除已选课程进行管理操作.
2. 系统中数据库起着非常重要的作用, 不能随意存取或改动, 因此必须对数

---

数据库的存取和改动进行严格的控制,对系统数据进行有效的保护,防止对数据库的非法操作,防止计算机病毒对数据的破坏,定期对数据库进行备份.

## 第四章 数据库设计

### 4.1. 数据库分析

#### 1. 实体的分析

根据学生信息管理系统的总体设计,我们将这个系统分为三大模块.由此我们可以首先确定二大实体:教师和学生,而由于用户可以进行选课、查询等操作.所以确定该系统共有以下实体:

- 1.教师实体.
- 2.学生实体.
- 3.课程实体.
- 4.选课实体.

#### 2. 实体的属性

该系统涉及到的实体分别为教师实体,学生实体.根据系统需求,我们将以下实体的属性确定如下:

数据结构	组 成
教 师	教师编号、密码、姓名
学 生	学号、姓名、密码、班级、性别。
课 程	课程编号、课程名称、授课教师、课程学分
选课关系	课程编号、学号、课程学分、上课地点、成绩

表 4-1 实体属性表

---

1.教师实体,包含教师的信息,详细属性如下:

教师实体, 包含教师的基本信息, 详细属性如下:

- (1) 教师号.
- (2) 教师姓名.
- (3) 登录密码.

3.学生实体,包含学生的基本信息,详细属性如下:

- (1) 学生号.
- (2) 学生姓名.
- (3) 密码.
- (4) 班级.
- (5) 性别.

4 课程实体,包含课程的信息,详细属性如下:

- (1) 课程号.
- (2) 课程名称.
- (3) 课程学分.

### 3. 数据表设计

逻辑结构是独立于任何一种数据模型的信息结构.逻辑结构的任务是把概念结构设计阶段设计好的基本 E-R 图转化为宜选用的 DBMS 所支持的数据模型相符合的逻辑结构,并对其进行优化.

定义每个表的数据类型以及字段限制,使数据库达到一定的完整性.每个表的物理设计如下:

数据项	含义	别名	类型	长度	取值范围
-----	----	----	----	----	------

教师编号	唯一表示教师	TeaNo	Varchar	11	——
学号	唯一表示学生	StuNo	Varchar	11	——
教师密码	教师用户登录验证	Teapass word	Varchar	20	———
学生密码	学生用户登录验证	Stupass word	Varchar	20	——
姓名	老师、学生的属性	Name	Varchar	11	——
班级	学生所属班级	class	Varchar	11	——
课程学分	课程属性，学生要修够一定学分	Creait	Float	5	0-5
教室	学生上课地点	Classro om	Varchar	11	——
授课教师	课程由老师教授	Teacher	Varchar	11	——

#### 4. 2. SQL 语句实现

(1) 教师信息表创建：

```
CREATE TABLE `teacher` (
  `TNo` int(11) NOT NULL AUTO_INCREMENT,
  `Tname` varchar(30) DEFAULT NULL,
  `Tpassword` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`TNo`)
)
```

---

(2) 课程信息表创建:

```
CREATE TABLE `course` (  
    `CNo` int(11) NOT NULL AUTO_INCREMENT,  
    `Cname` varchar(30) NOT NULL,  
    `Ccredit` float(5,1) DEFAULT '0.0',  
    PRIMARY KEY (`CNo`)  
)
```

(3) 学生信息表创建:

```
CREATE TABLE `student` (  
    `SNo` int(11) NOT NULL AUTO_INCREMENT,  
    `Sname` varchar(30) NOT NULL,  
    `Spassword` varchar(30) NOT NULL,  
    `Sclass` varchar(30) DEFAULT "",  
    `Ssex` varchar(10) DEFAULT "",  
    PRIMARY KEY (`SNo`)  
)
```



## 第五章 系统模块详细设计

本学生信息管理系统主要模块,即用户登录模块、用户操作模块,用户操作模块又分为学生操作模块、教师操作模块和管理员操作模块.下面将显示系统的主要功能模块.

### 5.1 教师模块设计

学生信息管理系统前台登陆模块只有一个登陆窗口分为学生、老师、两个级别登陆,系统在后台程序有自动限制设置,可以自动识别登陆者的限制.其系统登陆模块算法如下:

- 1 输入用户名和密码;
- 2.判断用户名和密码是否匹配
  - (1) 若提示输入信息错误,则返回登陆界面重新登陆;
  - (2) 否则进入系统.

#### 5.1.1 添加学生信息功能

学生信息管理系统对学生信息的管理非常重要,其中对信息的录入是系统最为关键的地方,以往我们管理学生信息的时候都是手工操作,而随社会不断发展的今天计算机的应用已全部取代了手工操作的方法,利用计算机可以方便的录入各类信息,进行高效的管理.学生信息的录入就是利用计算机通过程序读入数据库,录入学生信息模块算法描述如下:

- 1.教师登陆后,录入学生信息模块;
- 2.没有错后,学生信息添加成功.

#### 5.1.2 添加新课程功能

学生选课模块主要是给学生选择课程的功能,本功能主要由教师在后台发布自己的课程信息进入数据库,然后学生登录系统,可以看见所有发布的课程信息,然后学生再进行选择,学生选择的课程信息会自动进入后台数据库,其学生选课模块算法描述如下:

- 
1. 教师成功登陆;
  2. 点击添加新课程;
  3. 输入课程号;
  4. 输入课程名;
  5. 点击提交按钮,完成添加新课程操作.

#### 5.1.4 修改学生信息功能

此信息修改模块主要是给学生和教师建立信息库,方便教师查询操作,信息修改模块算法描述如下:

1. 教师成功登陆;
2. 点击修改学生信息;
3. 输入学生的编号;
4. 输入正确,进入修改页面进行相关修改;

#### 5.1.6 修改课程信息功能

此信息修改模块主要是给课程和教师建立信息库,方便教师查询操作,信息修改模块算法描述如下:

1. 教师成功登陆;
2. 点击修改课程信息;
3. 输入课程的编号;
4. 输入正确,进入修改页面进行相关修改.

/\*  
\*\*\*\*\*  
\*/

```
public class DaoQuery {
```

```
    /*
```

```
        * 根据学号查询选课情况
```

---

```
    */

    public List<QueryResult> selectSno(int sno){

        List<QueryResult> list = null;

        try{

            list = new ArrayList<QueryResult>();

            String          sql          =          "select
student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +
                                "where  student.sno=sc.sno  and  sc.cno=course.cno  and
student.sno=?";

            Connection conn = new Conn().getConn();

            PreparedStatement pst = conn.prepareStatement(sql);

            pst.setInt(1, sno);

            ResultSet rs = pst.executeQuery();

            while(rs.next()){

                QueryResult qrs = new QueryResult();

                qrs.setSNo(rs.getInt("sno"));

                qrs.setCNo(rs.getInt("cno"));

                qrs.setSname(rs.getString("sname"));

                qrs.setCname(rs.getString("cname"));

                qrs.setSclass(rs.getString("sclass"));

                qrs.setClassroom(rs.getString("classroom"));

                list.add(qrs);

            }

        } catch(Exception e){e.printStackTrace();}
```

---

```
        return list;

    }

    /*
     * 根据学生姓名查询选课情况
     */

    public List<QueryResult> selectSname(String name){

        List<QueryResult> list = null;

        try{

            list = new ArrayList<QueryResult>();

            String sql = "select
student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +
                        "where student.sno=sc.sno and sc.cno=course.cno and
student.sname=?";

            Connection conn = new Conn().getConn();

            PreparedStatement pst = conn.prepareStatement(sql);

            pst.setString(1, name);

            ResultSet rs = pst.executeQuery();

            while(rs.next()){

                QueryResult qrs = new QueryResult();

                qrs.setSNo(rs.getInt("sno"));

                qrs.setCNo(rs.getInt("cno"));

                qrs.setSname(rs.getString("sname"));

                qrs.setCname(rs.getString("cname"));

                qrs.setSclass(rs.getString("sclass"));

                qrs.setClassroom(rs.getString("classroom"));
```

---

```

        list.add(qrs);

    }

    }catch(Exception e){e.printStackTrace();}

    return list;

}

/*
 * 根据班级查询选课情况
 * */

public List<QueryResult> selectSclass(String _class){

    List<QueryResult> list = null;

    try{

        list = new ArrayList<QueryResult>();

        String sql = "select
student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +
        "where student.sno=sc.sno and sc.cno=course.cno and
student.sclass=?;";

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql);

        pst.setString(1, _class);

        ResultSet rs = pst.executeQuery();

        while(rs.next()){

            QueryResult qrs = new QueryResult();

            qrs.setSNo(rs.getInt("sno"));

            qrs.setCNo(rs.getInt("cno"));

            qrs.setSname(rs.getString("sname"));

```

---

```

        qrs.setName(rs.getString("cname"));

        qrs.setSclass(rs.getString("sclass"));

        qrs.setClassroom(rs.getString("classroom"));

        list.add(qrs);
    }

} catch (Exception e) {e.printStackTrace();}

return list;
}

/*
 * 根据课程号查询选课情况
 */

public List<QueryResult> selectCno(int cno){
    List<QueryResult> list = null;

    try{
        list = new ArrayList<QueryResult>();

        String sql = "select
student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +
        "where student.sno=sc.sno and sc.cno=course.cno and
course.cno=?";

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql);

        pst.setInt(1, cno);

        ResultSet rs = pst.executeQuery();

        while(rs.next()){

```

---

```

        QueryResult qrs = new QueryResult();

        qrs.setSNo(rs.getInt("sno"));

        qrs.setCNo(rs.getInt("cno"));

        qrs.setSname(rs.getString("sname"));

        qrs.setCname(rs.getString("cname"));

        qrs.setSclass(rs.getString("sclass"));

        qrs.setClassroom(rs.getString("classroom"));

        list.add(qrs);

    }

} catch(Exception e){e.printStackTrace();}

return list;

}

/*
 * 根据课程名查询选课情况
 */

public List<QueryResult> selectCname(String cname){

    List<QueryResult> list = null;

    try{

        list = new ArrayList<QueryResult>();

        String sql = "select
student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

        "where student.sno=sc.sno and sc.cno=course.cno and
course.cname=?";

        Connection conn = new Conn().getConn();

```

---

```

        PreparedStatement pst = conn.prepareStatement(sql);

        pst.setString(1, cname);

        ResultSet rs = pst.executeQuery();

        while(rs.next()){

            QueryResult qrs = new QueryResult();

            qrs.setSNo(rs.getInt("sno"));

            qrs.setCNo(rs.getInt("cno"));

            qrs.setName(rs.getString("sname"));

            qrs.setCname(rs.getString("cname"));

            qrs.setSclass(rs.getString("sclass"));

            qrs.setClassroom(rs.getString("classroom"));

            list.add(qrs);

        }

    } catch (Exception e) {e.printStackTrace();}

    return list;

}

/*
 * 根据上课教室查询选课情况
 */

public List<QueryResult> selectRoom(String room){

    List<QueryResult> list = null;

    try{

        list = new ArrayList<QueryResult>();

        String          sql          =          "select

```



---

```

student.sno,sname,sclass,course.cno,cname,classroom from student,sc,course " +

        "where student.sno=sc.sno and sc.cno=course.cno and
classroom=?;";

Connection conn = new Conn().getConn();

PreparedStatement pst = conn.prepareStatement(sql);

pst.setString(1, room);

ResultSet rs = pst.executeQuery();

while(rs.next()){

    QueryResult qrs = new QueryResult();

    qrs.setSNo(rs.getInt("sno"));

    qrs.setCNo(rs.getInt("cno"));

    qrs.setSname(rs.getString("sname"));

    qrs.setCname(rs.getString("cname"));

    qrs.setSclass(rs.getString("sclass"));

    qrs.setClassroom(rs.getString("classroom"));

    list.add(qrs);

}

} catch(Exception e){e.printStackTrace();}

return list;

}

}

/*****/

```

---

## 5.2 学生模块设计

### 5.2.1 学生选课功能

学生可以进行网上选课,在每学期快要结束时,学校都会把下学期要修的课程放到学校网站上,供同学们进行选课,这样节省了大家的宝贵时间,也方便了老师的管理,方便又实用.

此学生选课功能主要是方便学生操作,信息模块算法描述如下:

- 1 学生成功登陆;
2. 点击选定课程按钮,进入选定课程的页面;
3. 在页面中点击添加课程按钮;
4. 点击想要选择的课程的复选框,进行选择;
5. 然后点击提交按钮;
6. 课程选择成功.

### 5.2.2 查看学生陆;

2. 点击查询学生信息;

### 3. 输入学生的编信息功能

学生信息管理系统方便了学生查看自己的信息,如果学校由于疏忽输入了错误信息,自己可以查看并且及时通知学校教师,及时修改,减少不必要的麻烦.

此信息查询模块主要是方便学生查询操作,信息查询模块算法描述如下:

- 1 学生成功登号;
4. 输入正确,进入信息查询页面进行相关修改.

### 5.2.3 查看已选课程功能

学生可以查看本学期已选课程,并且可以方便的知道所选课程的教课老师和上课地点等,这样方便了学校专员,方便了老师和我们学生自己,需要的时候从网上下载课程表,随时看都可以.

---

此信息查询模块主要是方便学生查询操作,选课信息查询模块算法描述如下:

1. 学生成功登陆;
2. 点击选定课程;
3. 右边显示出本学期已选定的课程信息;
4. 如果想继续选课,点击添加课程即可;

#### 5.2.4 删除已选课程功能

学生可以进行网上选课,就可以删除已选课程,比如说不小心选错了某一门课程,就必须在规定的时间内删除掉,然后重新选择要修的课程,因为有时粗心是不可避免的,所以有删除的功能是必须,这样既方便了老师的管理,又方便了自己.

1. 学生成功登陆;
2. 点击选定课程;
3. 右边显示出本学期已选定的课程信息,选择想要删除的课程;
4. 单击课程后面的删除按钮即可;

```
/**/
```

```
public class DaoStu {
```

```
    /**
```

```
     * 学生登录
```

```
     */
```

```
    public Student loginStu(Student stu){
```

```
        Student stu1 = null;
```

```
        String sql_loginS="select * from student where SNo=? and Spassword=?";
```

```
        try{
```

```
            Connection conn = new Conn().getConn();
```

```
            PreparedStatement pst = conn.prepareStatement(sql_loginS);
```

```
            pst.setInt(1, stu.getSNo());
```

---

```

        pst.setString(2,stu.getSpassword());

        ResultSet rs = pst.executeQuery();

        if(rs.next()){

            stu1 = new
Student(rs.getInt(1),rs.getString(2),rs.getString(3),rs.getString(4),rs.getString(5),0);

            System.out.println("student login select over");

        }

    } catch(Exception e){e.printStackTrace();}

    return stu1;

}

/*
 * 查询所有学生
 */

public List<Student> selectStu(){

    List<Student> list = null;

    try{

        String sql_select = "select SNo,Sname,Spassword,Sclass,Ssex from student;";//
        查询全部学生信息+学分数;查询所有的 sql 语句

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql_select);

        ResultSet rs = pst.executeQuery();

        list = new ArrayList<Student>();

        DaoCou scredit = new DaoCou();

        while(rs.next()){

            Student stu = new
Student(rs.getInt("SNo"),rs.getString("Sname"),rs.getString("Spassword"),rs.getString("Sclas
s")

            ,rs.getString("Ssex"), scredit.selectScre(rs.getInt("SNo")));

```

---

```

        list.add(stu);

    }

} catch(Exception e){e.printStackTrace();}

return list;

}

/*
 * 查询学生总学分_某位学生
 * */

public Student selectScre(Student stu){

    ResultSet rs = null;

    try{

        String sql_insert = "select student.sno as sno,sum(course.ccredit) as scredit
from student,sc,course" +

            "    where student.sno=sc.sno" +

            "        and sc.cno=course.cno" +

            "        and student.sno=?";//查询某位同学学分

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql_insert);

        pst.setInt(1,stu.getSNo());

        rs = pst.executeQuery();

        if(rs.next()){

            System.out.println("Student_id"+stu.getSNo()+"select one over!");

            stu                =                new

Student(rs.getInt("sno"),null,null,null,null,rs.getFloat("scredit"));

        }

    } catch(Exception e){e.printStackTrace();}

    return stu;

}

```

---

```

/*
 * 查询某个学生
 */

public Student selectStu(int id){
    Student Stu = null;

    try{
        String sql_selectone = "select * from student where SNo=?";//查询某一条课程

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql_selectone);

        pst.setInt(1, id);

        ResultSet rs = pst.executeQuery();

        if(rs.next()){
            Stu = new
Student(rs.getInt("SNo"),rs.getString("Sname"),rs.getString("Spassword"),rs.getString("Sclas
s")

            ,rs.getString("Ssex"),0);
        }

    } catch(Exception e){e.printStackTrace();}

    return Stu;
}

/*
 * 修改某个学生信息
 */

public int updateStu(Student stu){
    int rs = 0;

    try{
        String sql_update = "update Student set Sname=?,Spassword=?,Sclass=?,Ssex=?
where SNo=?";//修改一条学生信息

```

---

```

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql_update);

        pst.setString(1, stu.getSname());

        pst.setString(2, stu.getSpassword());

        pst.setString(3, stu.getSclass());

        pst.setString(4, stu.getSsex());

        pst.setInt(5, stu.getSNo());

        rs = pst.executeUpdate();

        if(rs!=0){

            System.out.println("Student_id"+stu.getSNo()+"update over!");

        }

    } catch (Exception e) {e.printStackTrace();}

    return rs;

}

/*

* 删除某个学生信息

* */

public int deleteStu(int id){

    int rs = 0;

    try{

        String sql_delete = "delete from student where SNo=?";//删除一条课程信息

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql_delete);

        pst.setInt(1, id);

        rs = pst.executeUpdate();

        if(rs!=0){

            System.out.println("Student_id"+id+"delete over!");

        }

    } catch (Exception e) {e.printStackTrace();}

    return rs;

}

}

```

---

```

    }

    } catch (Exception e) { e.printStackTrace(); }

    return rs;

}

/*
 * 添加一位学生信息
 * */

public int insertStu(Student stu) {

    int rs = 0;

    try {

        String sql_insert = "insert into student(Sname,Spasword,Sclass,Ssex)
values(?,?,?,?);"; //添加一条课程信息

        Connection conn = new Conn().getConn();

        PreparedStatement pst = conn.prepareStatement(sql_insert);

        pst.setString(1, stu.getSname());

        pst.setString(2, stu.getSpasword());

        pst.setString(3, stu.getSclass());

        pst.setString(4, stu.getSsex());

        rs = pst.executeUpdate();

        if (rs != 0) {

            System.out.println("Student_id" + stu.getSNo() + "insert over!");

        }

    } catch (Exception e) { e.printStackTrace(); }

    return rs;

}

}

```



## 第六章 系统演示

在完成了上面的所有工作之后,本系统的各种功能已经全部确定,下面要完成的工作就是在详细设计的基础上,建立各功能界面,实现系统的各个功能.

### 6.1 系统登录界面

本系统有三个用户,分别是学生、教师和管理员,非配给他们三个不同的权限.



图 6.1 用户登录界面

## 6.2 录入学生信息界面



图 6.2 录入学生信息界面

## 6.3 修改学生信息界面



图 6.3 修改学生信息界面

## 6.4 学生选课界面



图 6.4 学生选课界面

---

## 总结

本系统是采用 Java 技术开发的, 想要将系统做的完美无缺还是很困难的, 由于在这个阶段中, 基本是处于边学习边实践的状态, 基础知识还不是很牢靠, 加之自身水平原因, 程序还是有许多不足之处:

界面模块做的不够好.

实现的功能不够完整.

界面需要更精确的美化.

经过这一段时间的努力, 收获了很多, 在收获的同时也发现了很多的不足, 回首论文的写作过程, 得到了许多人的帮助, 尤其是我的指导老师. 在我论文的完成过程中给予了我许多的教诲、帮助和鼓励. 每当我在写作中遇到困难的时候, 是他们从方法上和精神上给予我支持, 令我克服每个困难, 我在心底里备受鼓舞. 借此机会, 向他们表示我由衷的感激之情!

在系统开发的两个月里, 我受到了指导老师的细心指导, 董老师老师严谨的指导态度与深厚的理论知识都让我受益非浅, 从她身上我学到了很多的东西, 无论是理论还是实践都使我的知识有很大的提高. 借此我特提出感谢.

通过本次课程设计实训, 我掌握了 Java 编程技能, 对 Oracle 数据库有了较深的理解. 比较全面的巩固了编程思想. 使我懂得的谨慎在编程工作中的重要性. 程序员的工作态度以及良好的习惯也决定了自己工作的质量.

---

## 参考文献

- [1] Eric. Java 编程思想[M]. 第 4 版.机械工业出版社.
- [2] 陈刚. MyEclipse 从入门到精通 [M]. 北京: 清华大学出版社, 2005 年 4 月.
- [3] 谭怀远,让 Oracle 跑得更快优化思路[M].北京: 电子工业出版社,2011.
- [4] 孙卫琴. Tomcat 与 Java Web 开发技术详解[M]. 第 2 版. 电子工业出版社, 2009 年 01 月.
- [5] 刘京华 等. Java Web 整合开发王者归来[M]. 清华大学出版社, 2010 年 01 月.
- [6] 萨师煊, 王珊.《数据库系统概论》.高等教育出版社,2002 年 2 月.
- [7] 温谦. HTML+CSS 网页设计与布局从入门到精通[M].人民邮电出版社, 2008 年 08 月.
- [8] 张银鹤. 点石成金: JSP+Ajax 网站开发典型实例[M].电子工业出版社, 2009 年 01 月.
- [9] 马晓虎,基于 B / S 结构的通用学生管理系统的开发[J].电脑知识与技术,2006.
- [10] 张学群、楼克明,库存管理,运筹学基础,经济出版社,1996.3.
- [11] 黄梯云,管理信息系统导论,机械工业出版社.
- [12] 李昭原,刘又诚数据库系统原理与技术北京航空航天大学出版社.
- [13] 甘仞初,信息系统开发,北京: 经济科学出版社,1996.
- [14] 王珊,数据组织与结构,经济科学出版社.
- [15] 孙卫琴. 基于 MVC 的 JavaWeb 设计与开发[M]. 北京: 电子工业出版社, 2004.
- [16]BruceEckel.Java 编程思想[M]. 机械工业出版社,2003.
- [17]FLANAGAN.Java 技术手册[M]. 中国电力出版社,2002.
- [18]孙一林,彭波.Java 数据库编程实例[M]. 清华大学出版社,2002.

---

[19]LEE ANNE PHILLIPS.巧学活用 HTML4[M].电子工业出版社,2004.

[20]飞思科技产品研发中心.JAVA 应用开发详解[M].电子工业出版社,2003.