## 选题目的和意义

目前国家的教育体制正处在不断改革、创新的阶段，我国教育部门充分吸取国外优秀的教学模式，结合国内多年的办学经验，逐步探索出适合中国特色的教学模式，因此国家教育部面向各级各类学校开展了全面学分制改革。同时随着国内大学生基数大幅度增加，因此，各院校迫切要对自己的现有教务管理系统进行改进和提高，根据国内大学的现有管理模式，在校园网络环境下建设先进的信息化管理平台。提高学校管理工作的现代化水平，使之成为学校公共信息服务体系的重要组成部分。在这样的大环境下，我们选择了“教务管理系统”这个设计题目，符合实际需求。

## 需求分析

通过对各院校教学工作流程做初步了解，初步确定需求如下：

学生：选择选修课、查询固定分配的必修课成绩、选修的选修课成绩

教师：为自己教授的所有课程的所有班级的所有学生录入成绩

管理员：课程、学生、教师基本信息管理

根据需求分析结果得高级用例图如下：

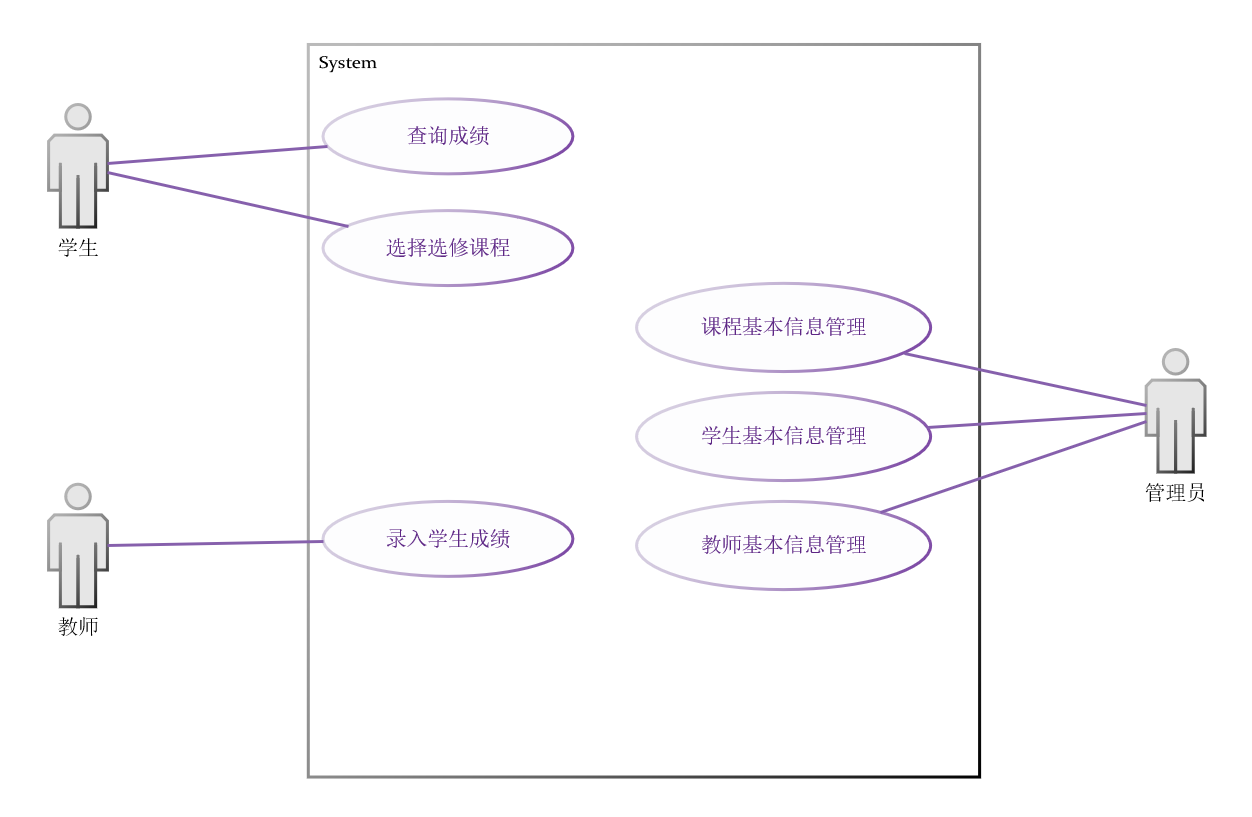


图2.1-用例图

## 系统设计

#### 3.1功能模块设计

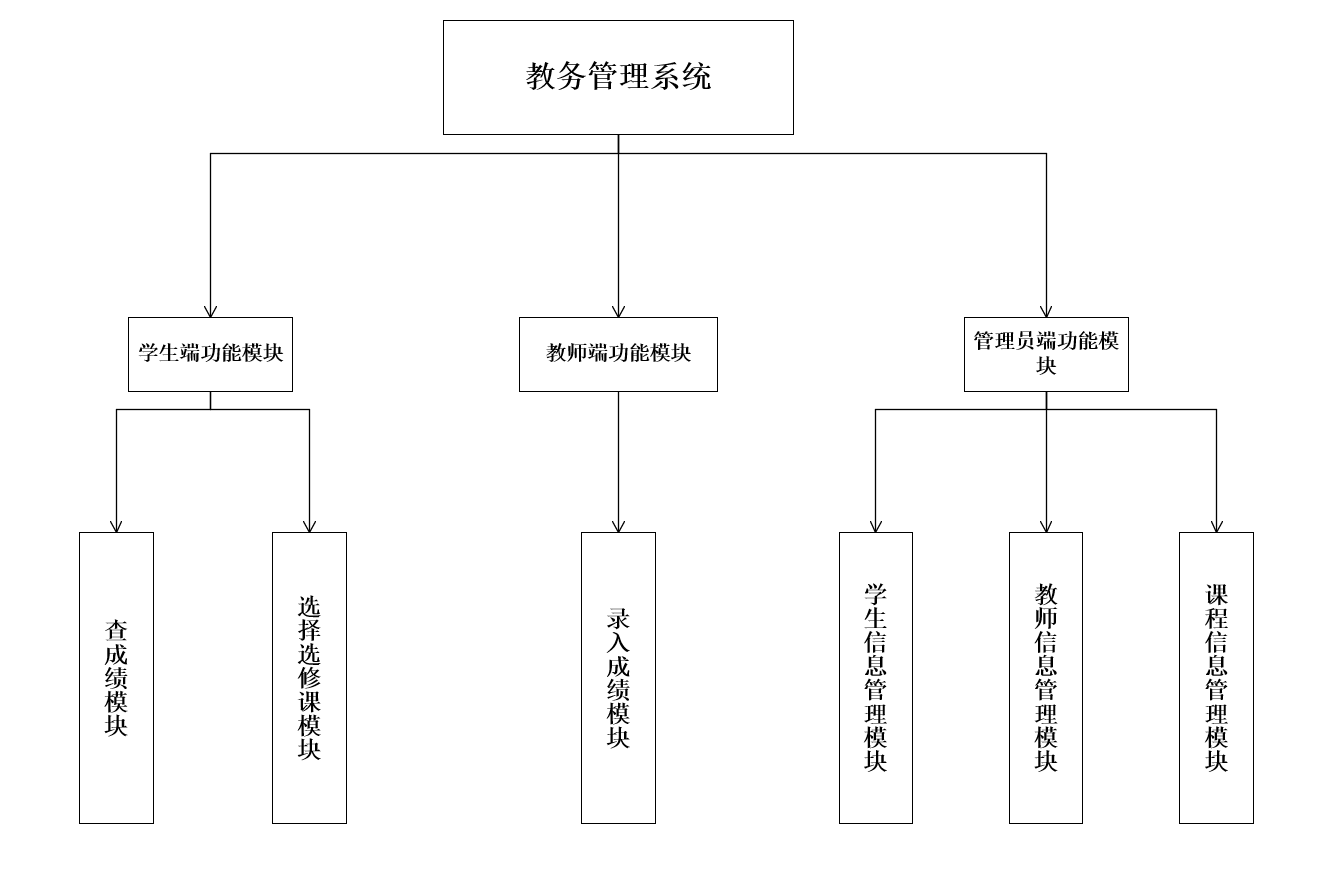


图3.1.1-系统功能图

##### 3.1.1学生端功能模块：

1. 学生端登录到主页之后，在系统上可以查询自己的所有课程成绩包含排好的必修课成绩和自己选择的选修课成绩，成绩列表内包含课程名和教师名以及对应的成绩，若成绩还未录入则显示为空。
2. 学生端登录到主页之后，在系统上可在选课开放期间进行选修课的选择，主页会显示所有的当前可选选修课列表，若选择了某选修课，则会在该页面下方显示已选择的选修课列表，并且可以在此时对已选择的课程退课。

##### 3.1.2教师端功能模块

教师端登录到主页之后，可以查看自己所教的所有课程，在每个课程项还可查询所教该课程的所有班级，定位到班级后，还可对该班级的所有学生录入对应课程的成绩。

##### 3.1.3管理员端功能模块

1. 登录到主页之后，可选择对看到课程基本信息进行查询、增加、删除、修改等操作。
2. 也可对教师基本信息进行查询、增加、删除、修改等操作。
3. 对学生基本信息进行查询、增加、删除、修改等操作。

#### 3.2数据库设计

##### 3.2.1概念模型设计：

该系统主要涉及到教师、学生、班级、课程这四个实体类，其中一门教师可以教多门课，一门课可以由多门老师所教，故教师与课程之间为多对多关系；一门学生可以选择多门选修课，一门选修课也可以由多个教师所教，故课程与学生之间为多对多关系；一个教师可以教多个班级，一个班级也可以被多个教师所教，故教师与班级之间为多对多关系；一个班级可以包含多个学生，故学生与班级之间为多对一关系。但同时考虑到实际情况中学生选择选修课以及为学生排必修课，排课的同时得为其分配对应的教师，因此考虑加上课程-学生-教师的多对多对多三元联系集以便存储学生选某门课同时对应的教师信息，并且在此时课程与学生之间的多对多联系冗余去除即可。考虑到教师录入成绩的方便，加入了班级这一实体集，方便教师在所教课程列表中直接找到班级列表同时为班级内的所有学生录入该科的成绩，因此加入了教师-课程-班级这样的多对多对多的三元联系集以便存储教师所加课程对应的班级信息。同时考虑到一位教师教某门课程可能暂时未排班级因此需要教师-课程的二元联系集，并且教师也需要直接查询自己所教课程的一个过程，因此对该联系集进行保留。

经处理得ER图如下所示：

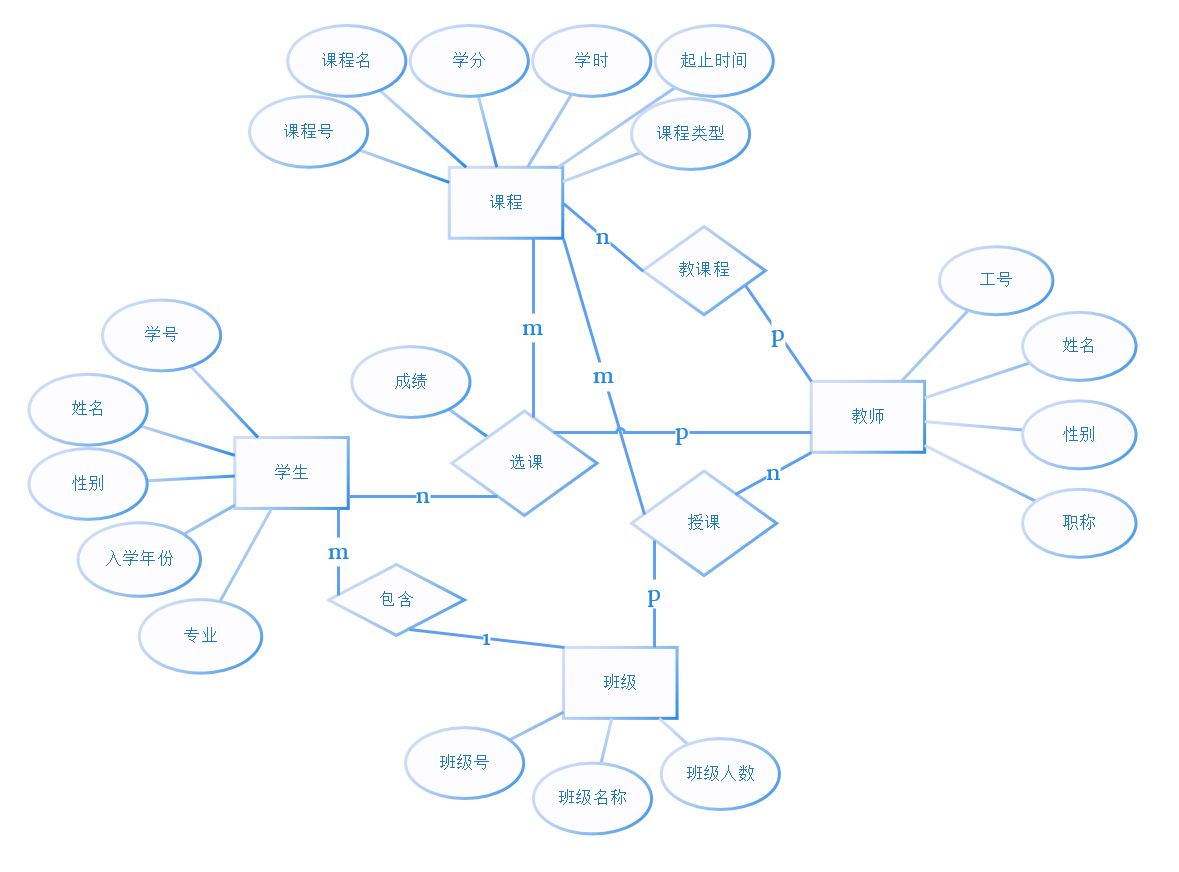


图3.2.1-系统实体-联系图

得CDM（数据库概念数据模型如下）：

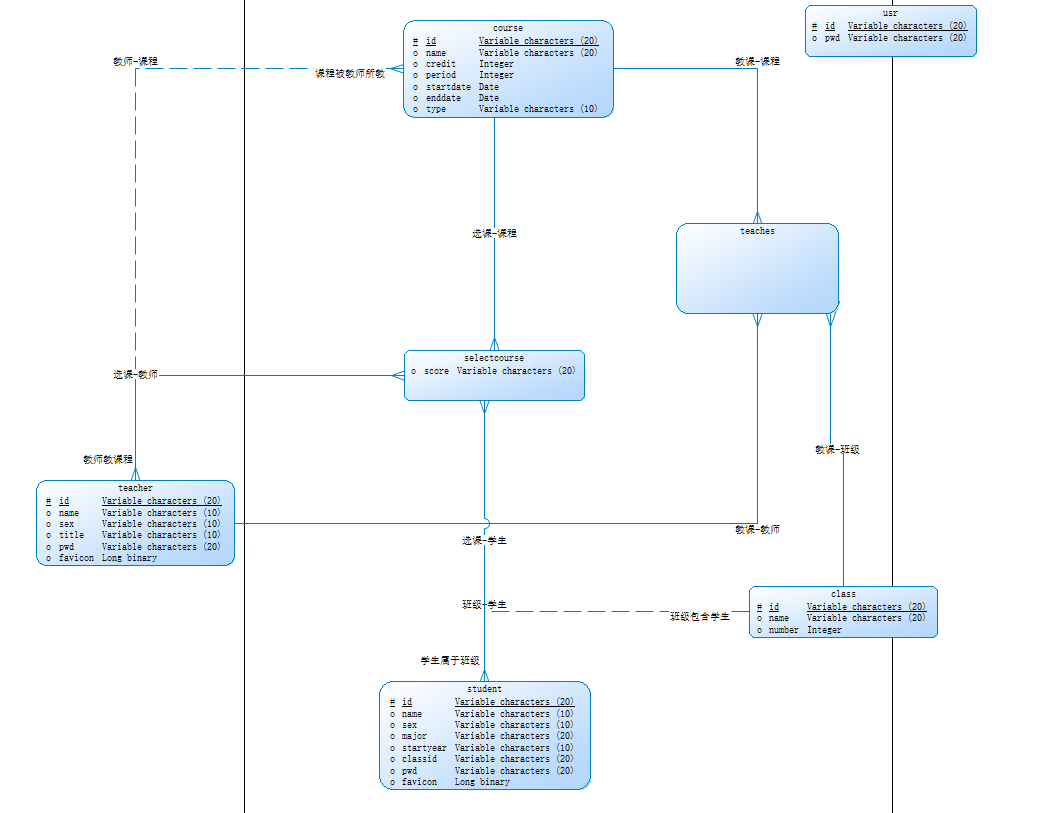


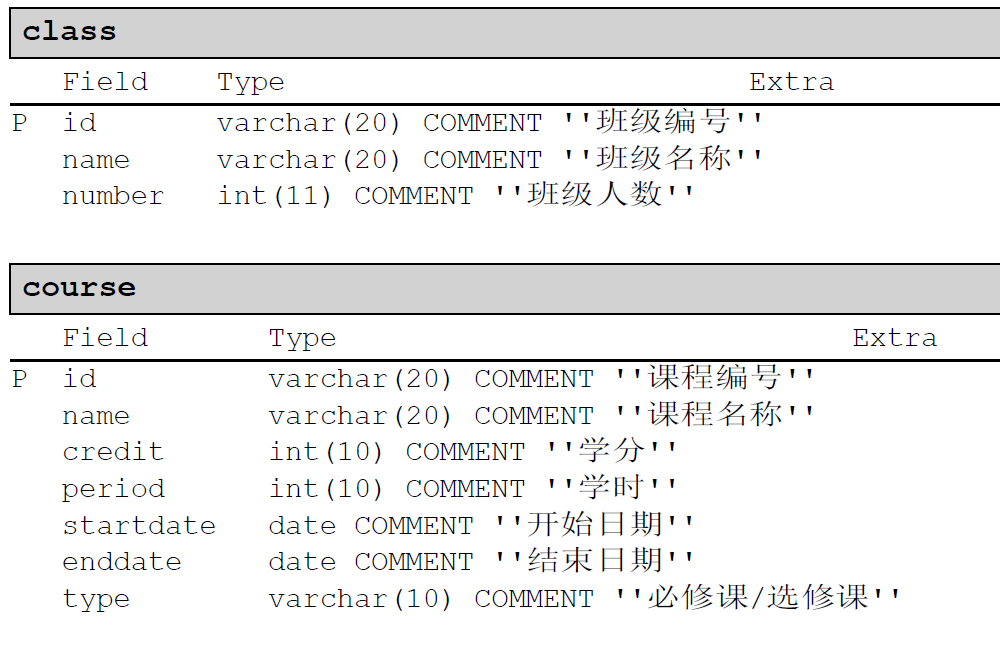
图3.2.2-概念数据模型

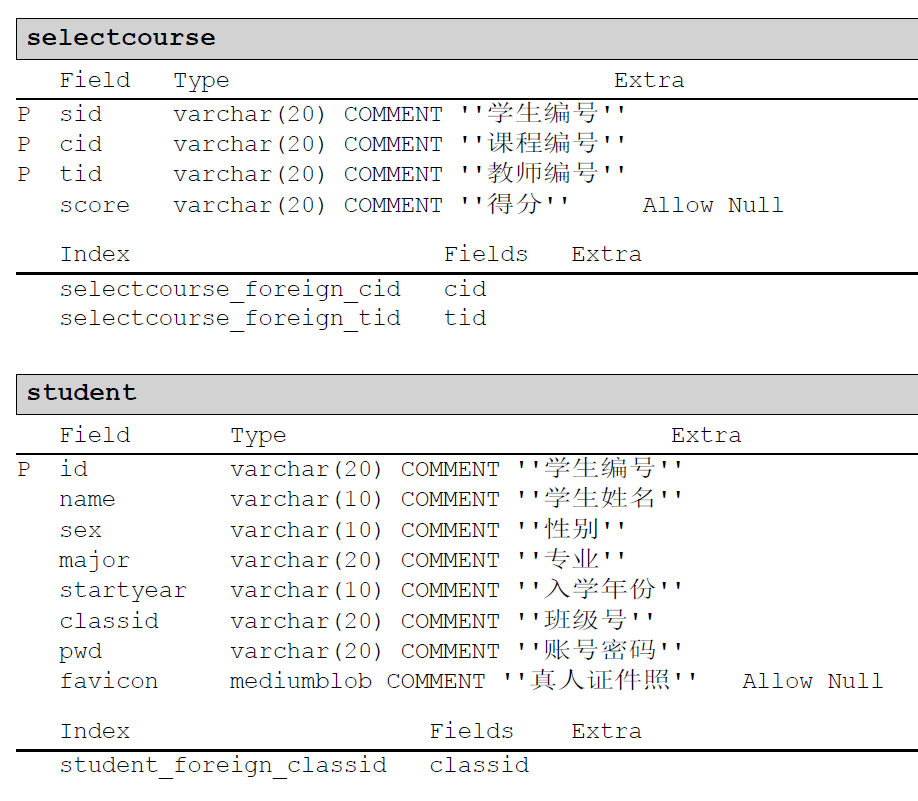
##### 3.2.2物理模型设计

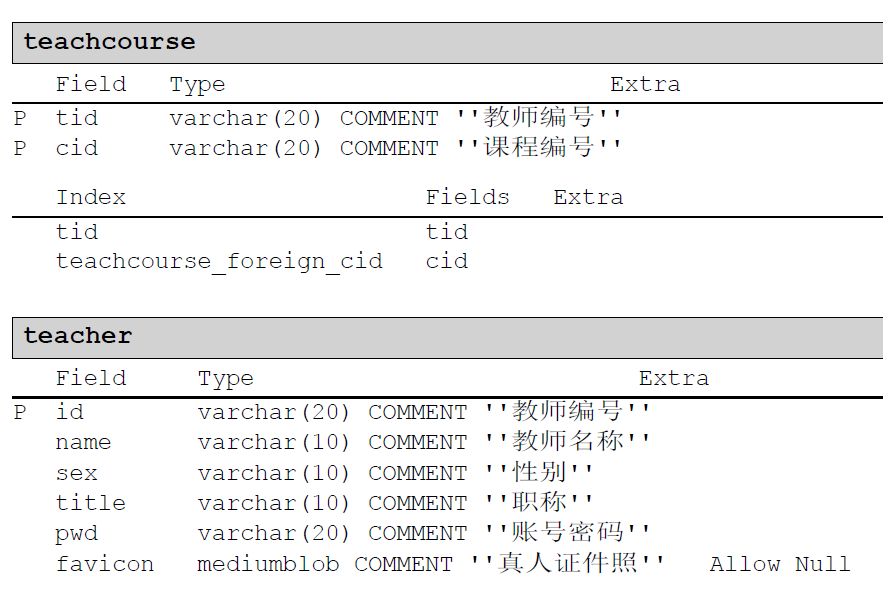
经概念模型分析与设计后得到学生、班级、课程、教师四个实体集，联系集有学生-课程-教师的多对多对多三元联系集、教师与课程之间的二元联系集、教师-课程-班级的三元联系集。考虑到班级与学生之间为一对多关系，因此这一联系集所对应的关系模式可以归到学生这一实体集内的班级编号，因此建立关系模式时不考虑该联系集。为上述每个实体集与联系集建立相应的关系模式并填充相应的属性，实体集对应主键均为相应的实体ID，联系集要考虑外键依赖于相应的实体集。

同时由于单独的联系集仅有各编号的联系，不含有各实体的具体联系，因此要把各联系集和联系起来的实体集进行自然连接得到各个视图以便后续信息的查询

最终得关系模式用数据字典表示如下图所示：







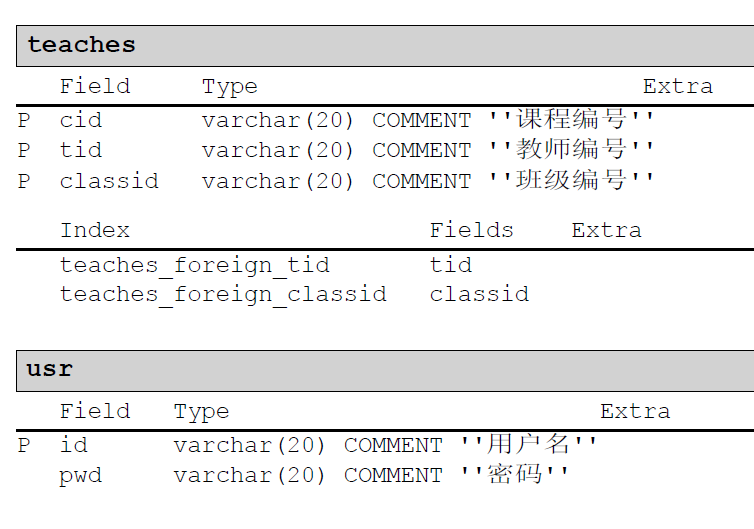


图3.2.3-数据字典

得PDM（物理数据模型）如下图所示：

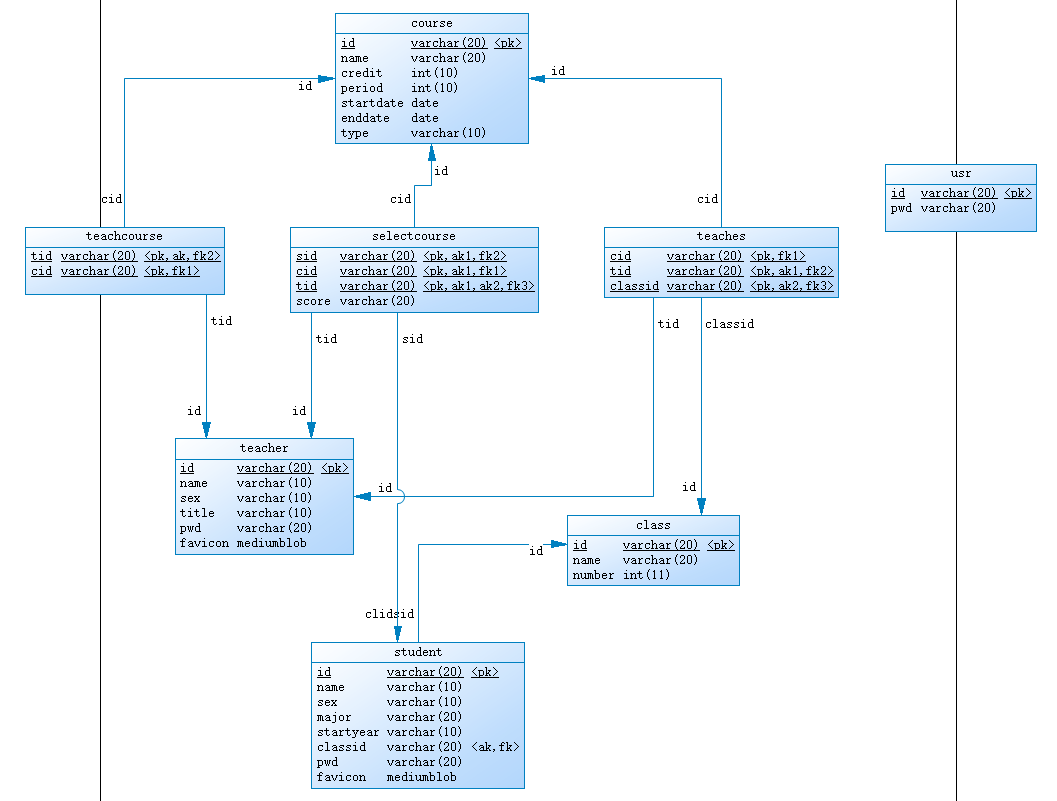


图3.2.4-物理数据模型

## 系统实现

该系统是通过JavaWeb相关技术进行开发，整个系统架构为：前端JSP(LayUI框架的使用)+Servlet服务器端应用程序+DAO数据访问层+VO模型层（实体类），数据库采用MySQL实现，服务器采用本地Tomcat搭建。采用了B/S网络架构下的简单的MVC的Web端开发模式进行开发。基本的架构图如下：



图4.1-系统架构图

#### 4.1MySQL数据库实现

首先在MySQL下建立一个EAMS数据库用于存储各个关系表，根据数据库设计阶段的物理概念模型，在MySQL数据库中根据各个关系模式建立对应的关系表，并生成数据对各个关系表进行填充，同时建立三个视图如下以便查询：

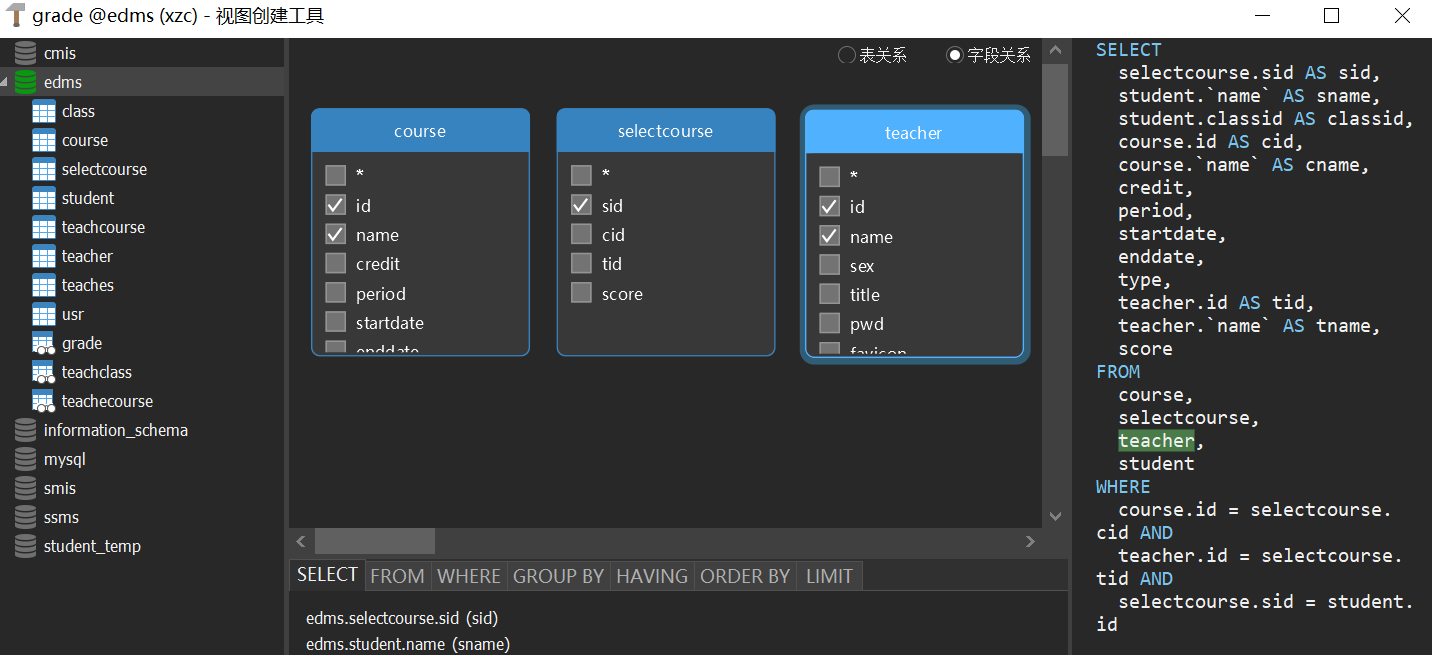


图4.1.1-grade（成绩）视图

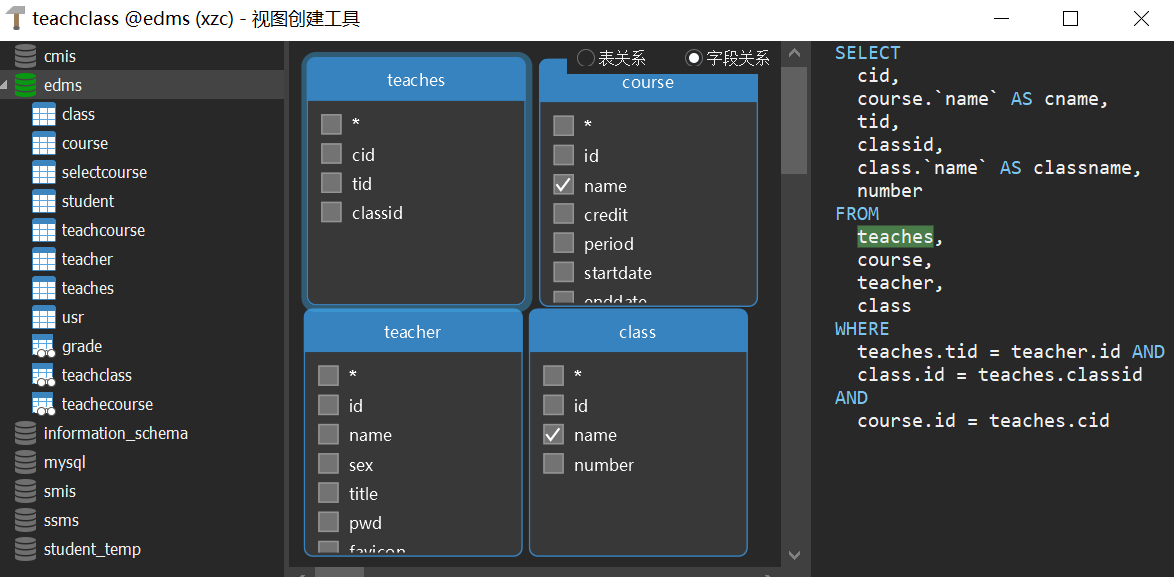


图4.1.2-teachclass（教师-班级-课程）视图

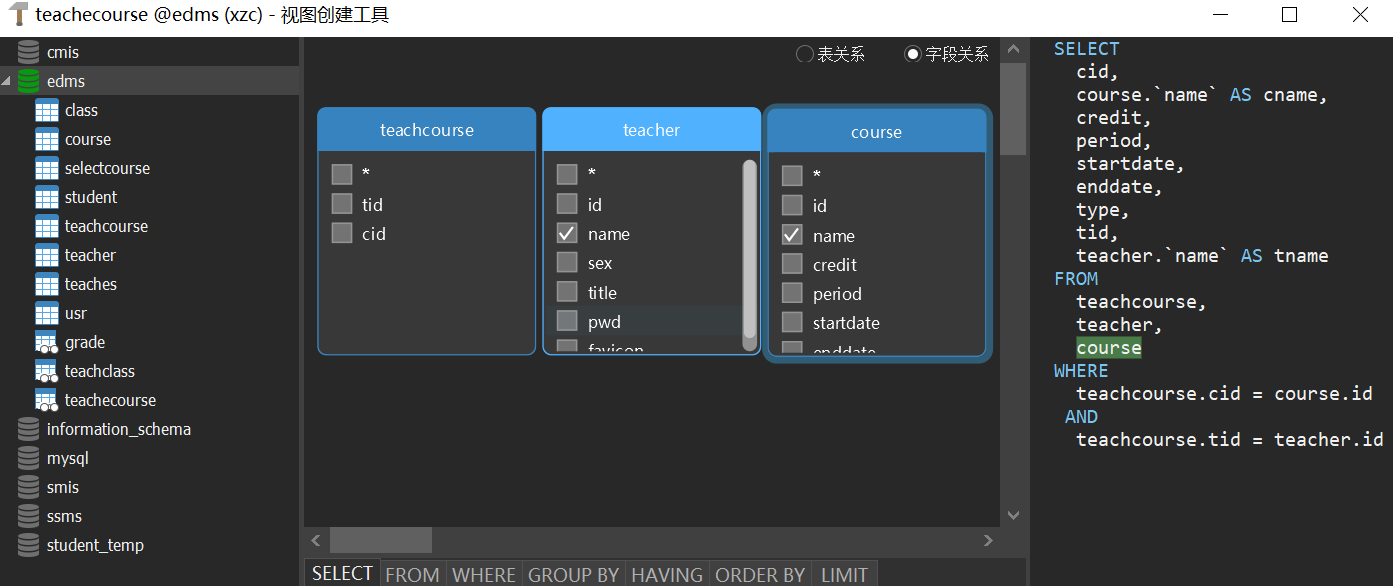


图4.1.3-teachcourse（教师-课程）视图

#### 4.2模型层（实体类）建立

根据MySQL数据库中的各个关系表建立相应的实体类，以便数据访问层和servlet控制层进行传整个数据对象的值。

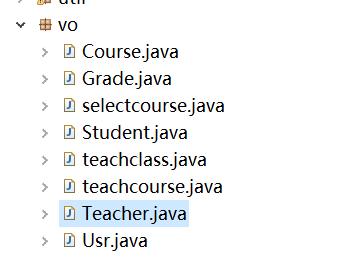


图4.2.1-模型层建立

建立相应的实体类之后，根据关系表各字段确定实体类中的各私有属性名称和类型，然后通过Eclipse中的source的get Setter and Getter来获取对应的共有的属性对应方法。

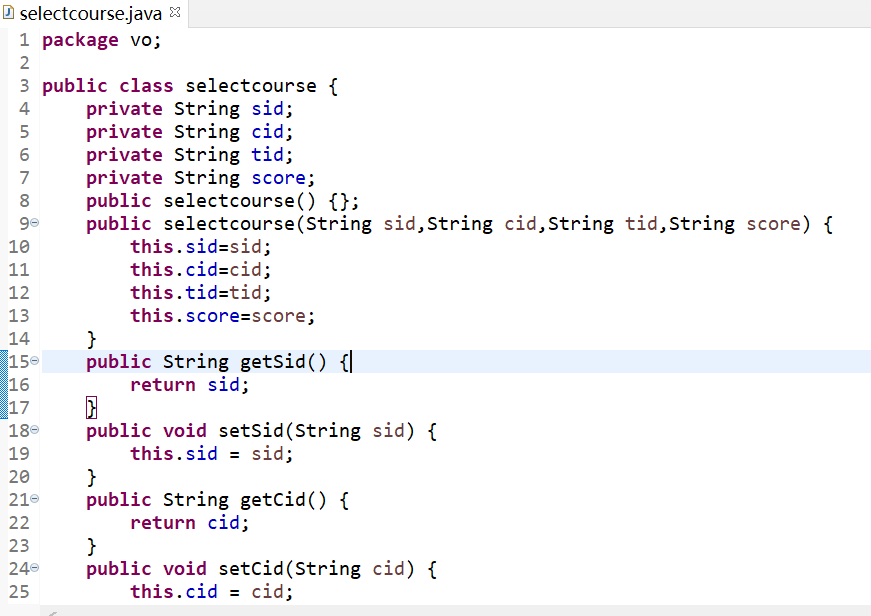


图4.2.2-selectcourse选课实体类示例（对应grade视图）

#### 4.3数据访问层的建立

数据访问层（DAO）的建立是为了做一个中间层的代码复用，方便servlet控制层的系统性的调用，是直接对数据库中的各个关系表数据进行增删改查的中间层，因此需要对每个实体类建立相应的数据访问接口。

同时为了后续代码复用性和可维护性，为每个实体类建立数据访问类的同时还要建立相应的接口（Interface）。

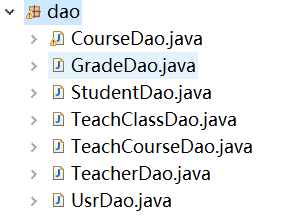
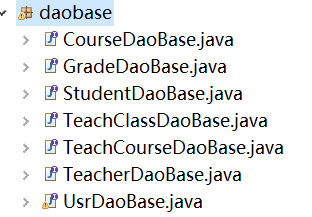
 

图4.3.1-建立数据访问层

同时也要根据控制层的需要，为每个实体类的数据访问类建立不同的查询函数：



图4.3.2-GradeDao数据访问类示例

为了更方便数据访问层每次对数据库连接、查询更新、关闭等操作，创建了数据库公共工具类DBUtil。

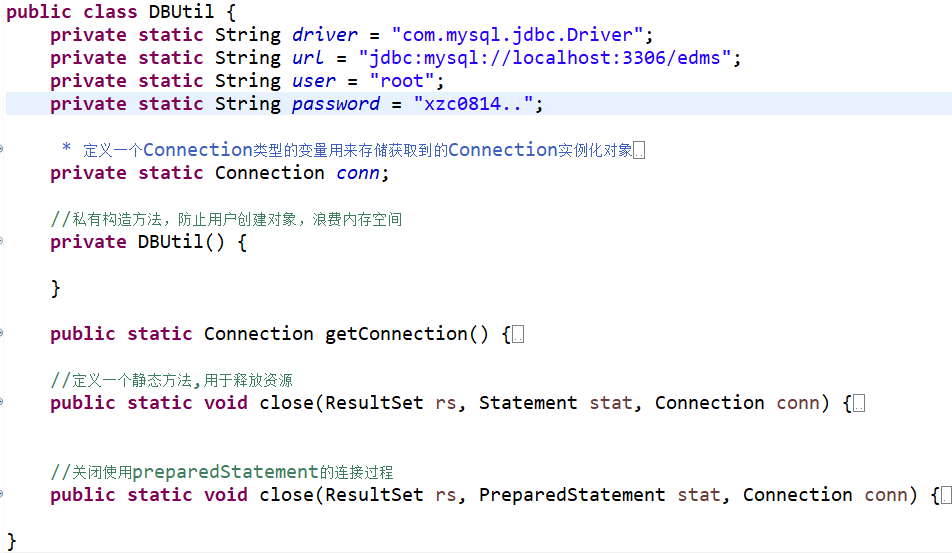


图4.3.3-数据库连接关闭工具类-DBUtil

#### 4.4控制层(Servlet)建立

控制层是直接与视图层进行交互的，用户在视图层进行一些参数设定，然后发送请求，此时控制层接受各个参数，进行处理，然后通过数据访问层对数据库数据进行增删改查等操作。

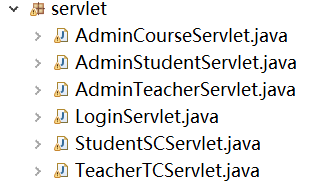


图4.4.1-控制层的建立

以下说明各个Servlet的实际作用：

**LoginServlet:**

学生、教师、管理员端登陆时均使用该Servlet进行用户名密码验证，当控制器接收到为学生请求时会根据传来的用户名通过StudentDao数据访问类访问数据库中Student表查询密码，若查询不到或者密码不匹配则登录失败重新返回主页，反之登录到相应主页；接收到为教师请求时则使用TeacherDao查询数据库Teacher表；接收到为管理员请求时则使用UsrDao查询数据库中Usr表。

**StudentSCServlet:**

该Servlet主要用在学生端查询成绩、查询选修课列表、选择选修课、退课选修课，教师端更新录入成绩。

由于在视图层查询和对数据库数据进行增删改等操作时均涉及的是selectcourse表，因此考虑到代码复用性，无论视图层在进行何种操作，均使用method这一参数指定操作类型然后通过URL传值传给Servlet，便于Servlet进行统一处理，后续Servlet处理同此处。

**对于查询成绩操作**：指定method为showGrade，由于视图层采用的是JSP+LayUI前端框架，其中数据表格是采用访问URL得到指定格式的JSON数据进行填充数据的，在Servlet通过数据访问类查询到结果后要转化为JSON数据并通过Writer进行输出，以便视图层数据表格的数据绑定。

**对于查询选修课列表操作**：指定method为showGrade\_option，并通过于查询成绩类似的过程，只是在数据访问层选择select函数有所不同。

**对于选课、退课操作**：指定method为addSC、deleteSC，并直接通过GradeDao的add和delete函数根据给定的编号进行添加和删除操作。

**对于录入修改成绩**：指定method为updateSC，直接通过GradeDao并根据给定的编号和成绩信息封装一个GradeVO来调用update方法进行数据库中的更新。



图4.4.2-StudentSCServlet的说明

**TeacherTCServlet:**

该Servlet主要用在教师端先查询自己所教课程、再查询某课程下的所教班级、再查询某班级下所有学生的该科成绩。

**对于查询所教课程**：指定method为showCourse\_byTid，然后通过TeachCourseDao查询输出结果。

**对于查询某课程下所教班级**：指定method为showClass，然后通过TeachClassDao查询输出结果。

**对于查询某班级下的所有成绩**：指定method为showClassGrade，然后通过GradeDao查询给定编号下的成绩。

**对于查询所有老师教的所有选修课**：指定method为showCourseAll，通过TeachCourseDao查询所有选修课课程。

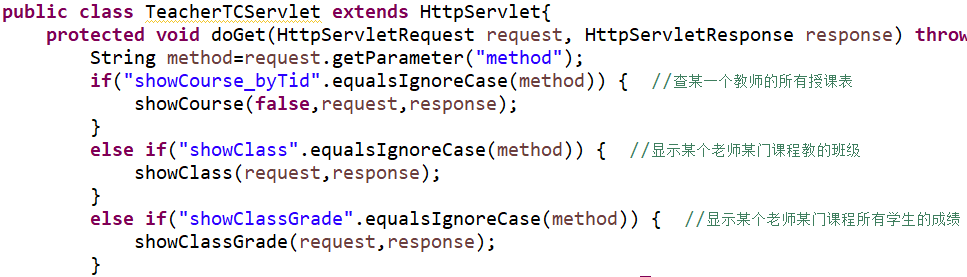


图4.4.3-TeacherTCServlet的说明

**AdminCourseServlet/AdminTeacherServlet/AdminStudentServlet:**

这三个Servlet用途为Course、Teacher、Student三个实体基本信息的增删改查，因此实现一致且与上述Servlet类似，以下仅对AdminCourseServlet进行简要的图片说明。



图4.4.4-AdminCourseServlet的说明

#### 4.5视图层（前端页面）的建立

前端JSP页面结构如下：

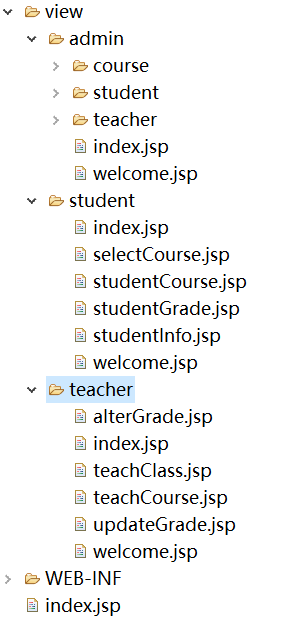
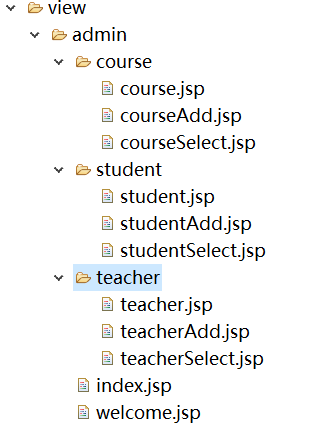
 

图4.5.1-视图层全局结构 图4.5.2-管理员界面结构

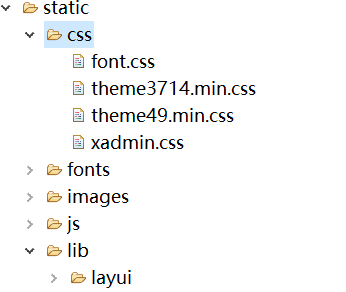


图4.5.3-前端框架CSS、JS、静态图片资源引入

## 系统测试

#### 5.1登录测试



图5.1.1-登录页面

输入账号和密码201801然后选择学生身份登录成功后页面如下：

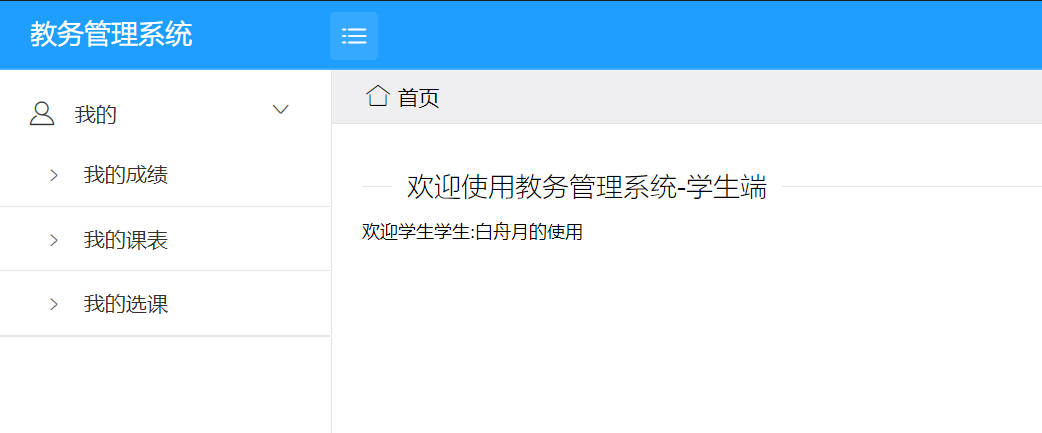


图5.1.2-学生登录成功首页

#### 5.2学生端测试

点击左侧菜单栏的我的成绩得成绩查询页面如下：

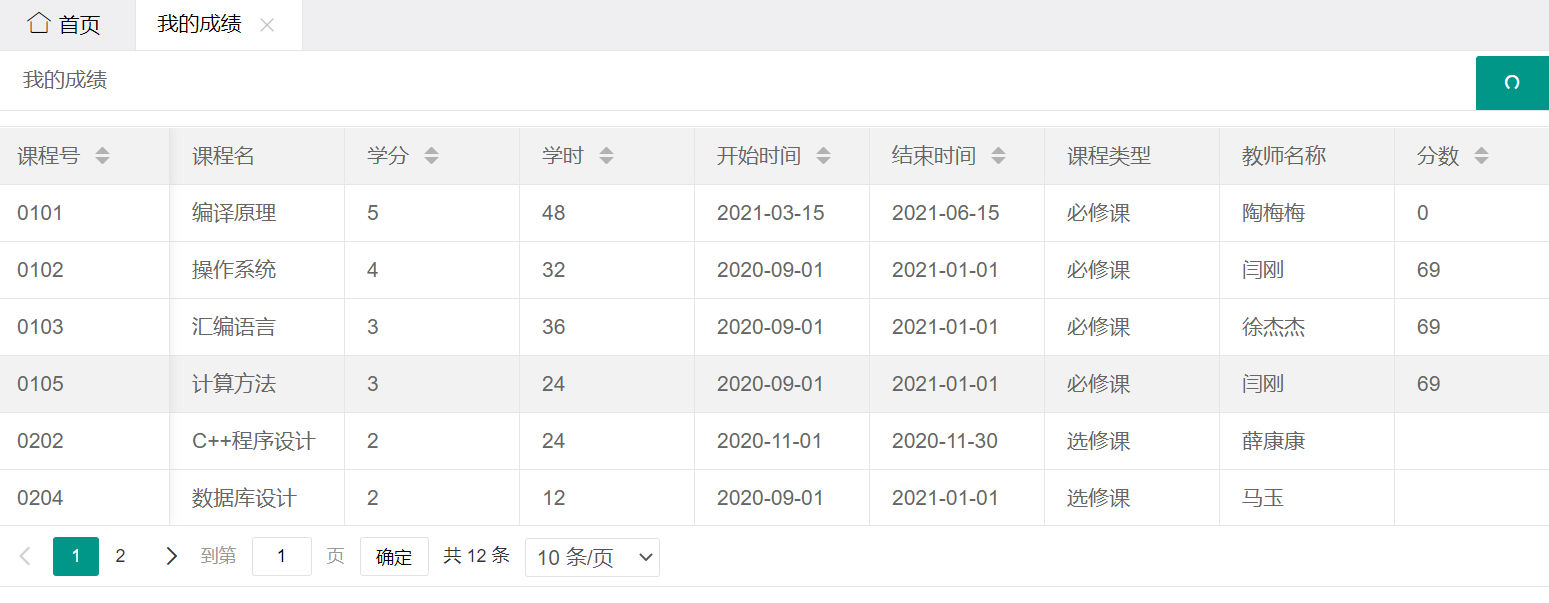


图5.2.1-成绩查询页面

点击左侧菜单栏的我的选课得选课页面如下：

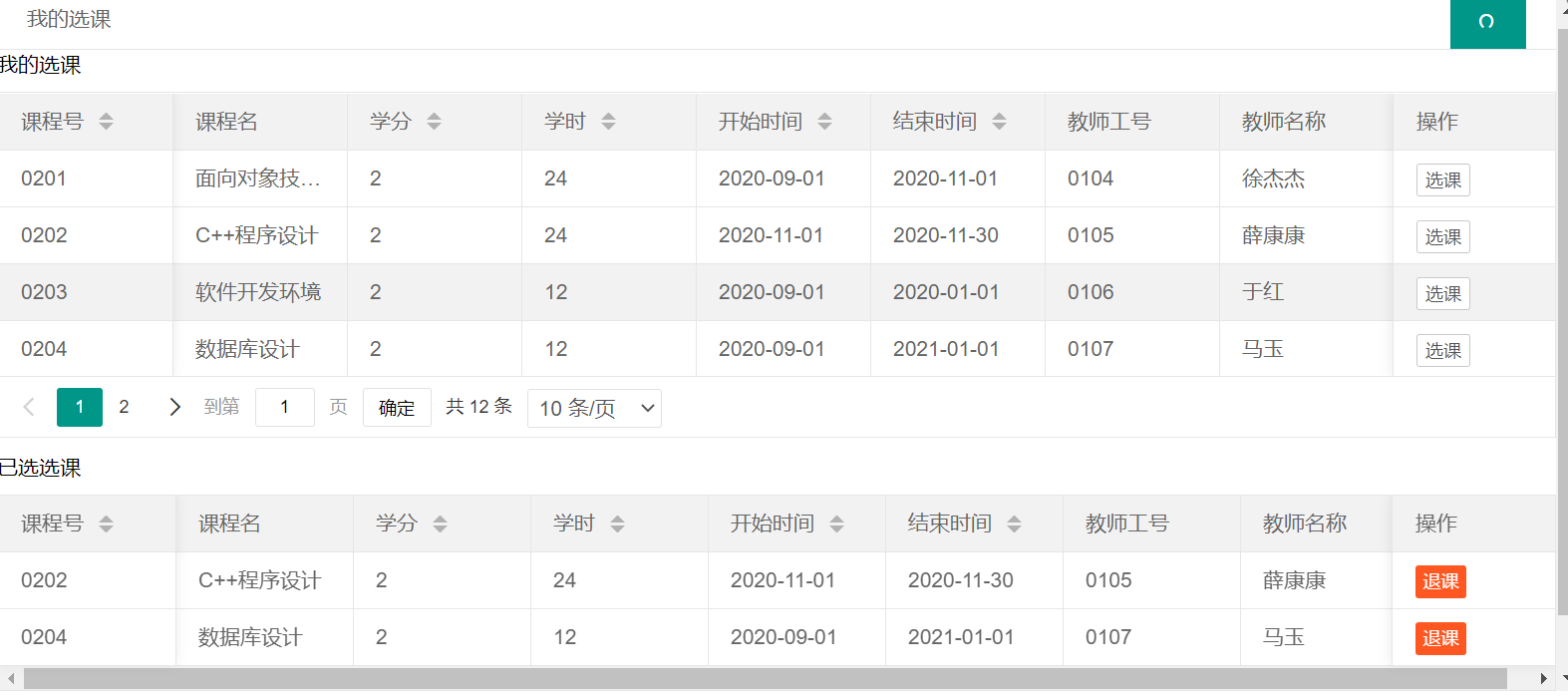


图5.2.2-选课退课页面

点击选课软件开发环境这一门课后，并点击退课数据库设计这一门课后页面显示分别如下：

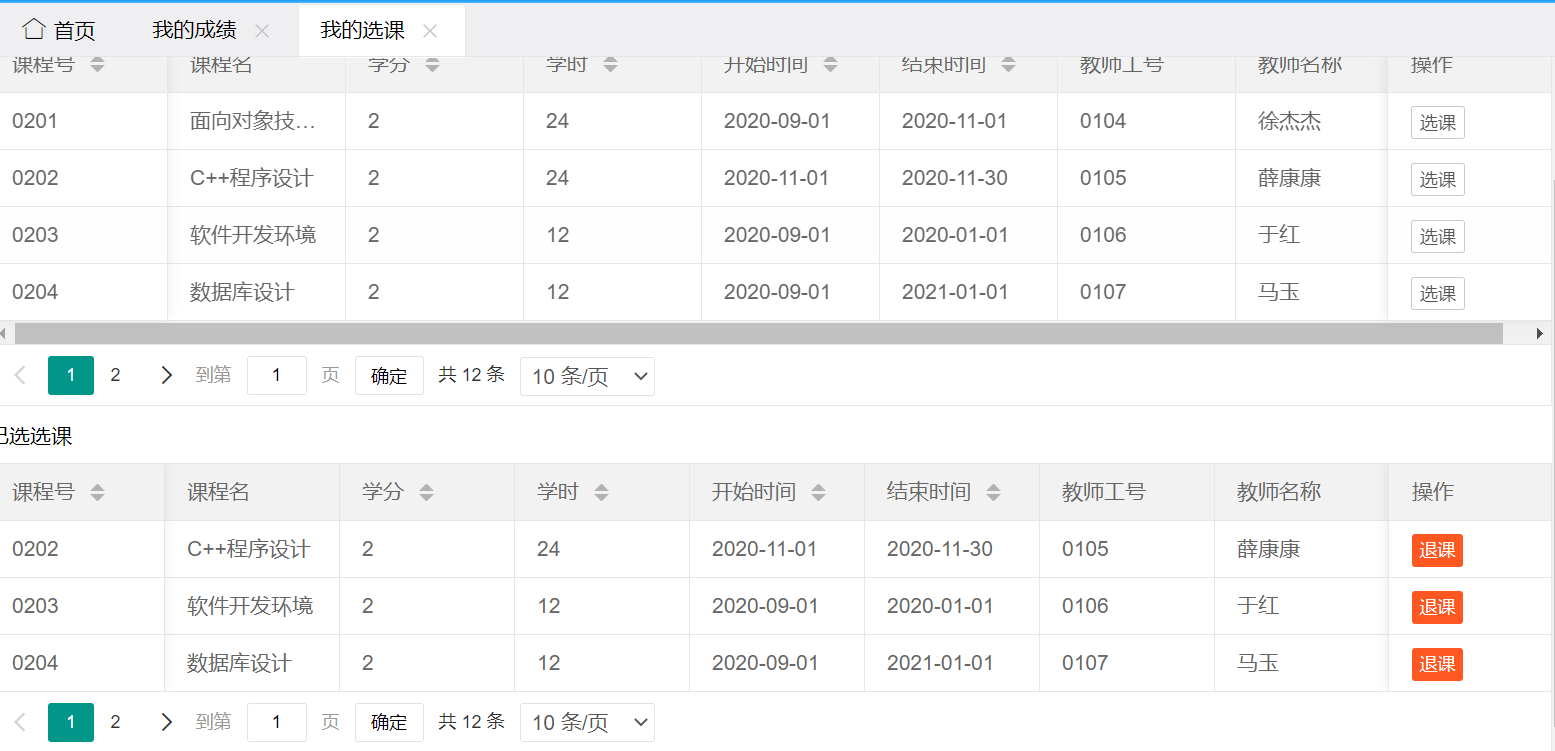


图5.2.3-选课软件开发环境后页面

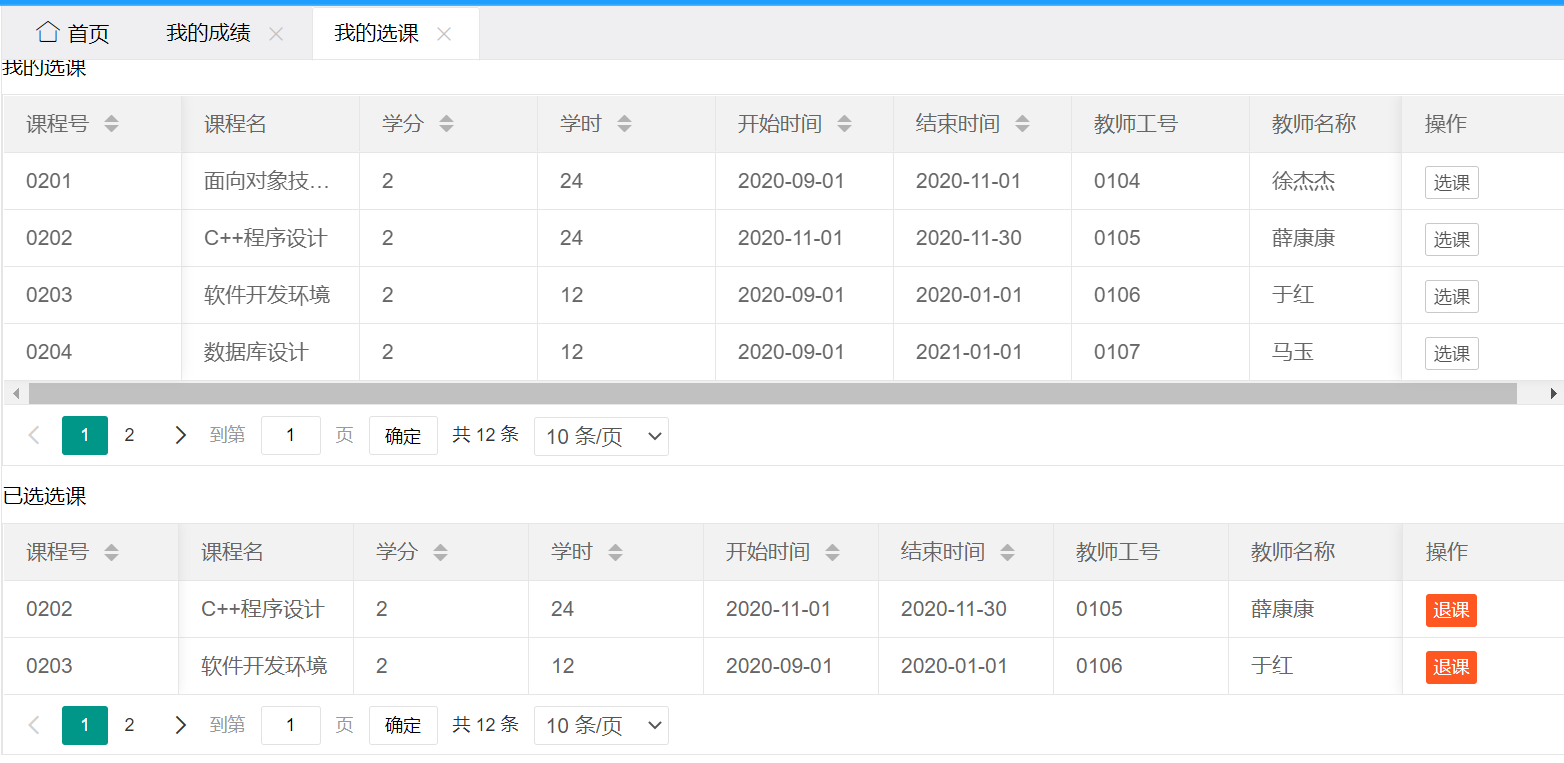


图5.2.4-退课数据库课设后页面

#### 5.3教师端测试

输入用户名和密码0101后，登录成功页面如下：

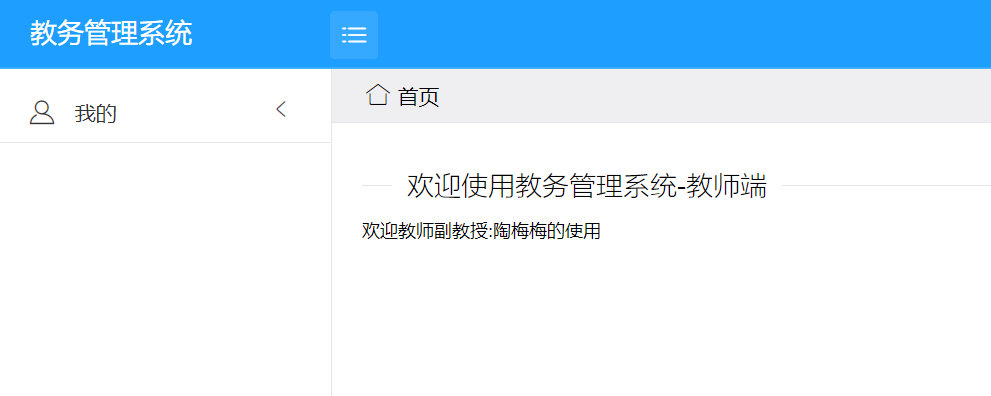


图5.3.1-教师端登录成功页面

点击录入成绩，首先出现所交的所有课



图5.3.2-我的授课页面

然后点击编译原理这门课程，显示我的班级页面：

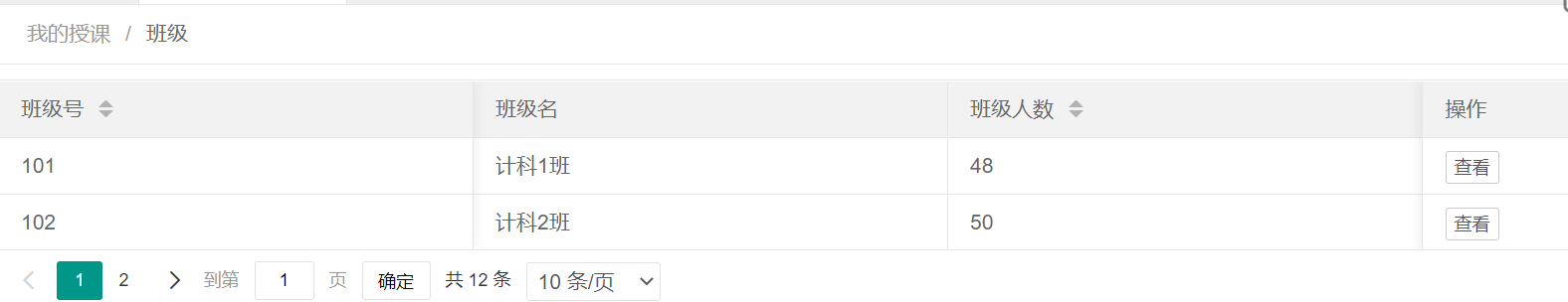


图5.3.3-我的班级页面

然后点击计科一班的查看即可到录入成绩页面



图5.3.4-录入成绩显示页面

原白舟月成绩为0，现录入80，然后点击修改



图5.3.5-录入成绩80修改页面

录入后登录学生端查看对应成绩是否修改



图5.3.6-录入后学生端验证页面

可以看到对应编译原理成绩从0改为80，录入成功。

#### 5.4管理员端验证

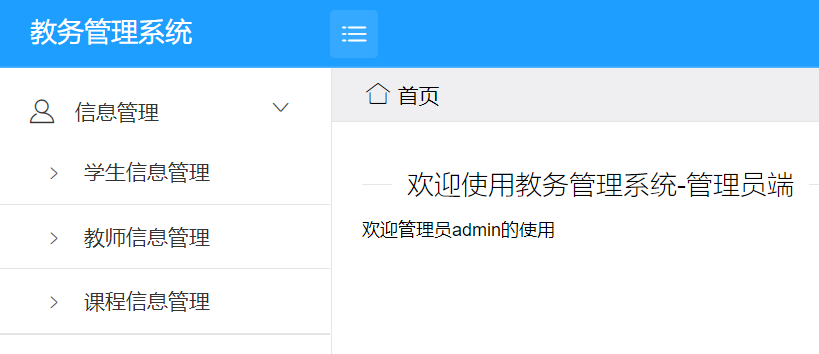


图5.4.1-管理员端登录成功页面

以下仅通过课程信息管理这一模块来测试功能，点击左侧菜单栏的课程信息管理：

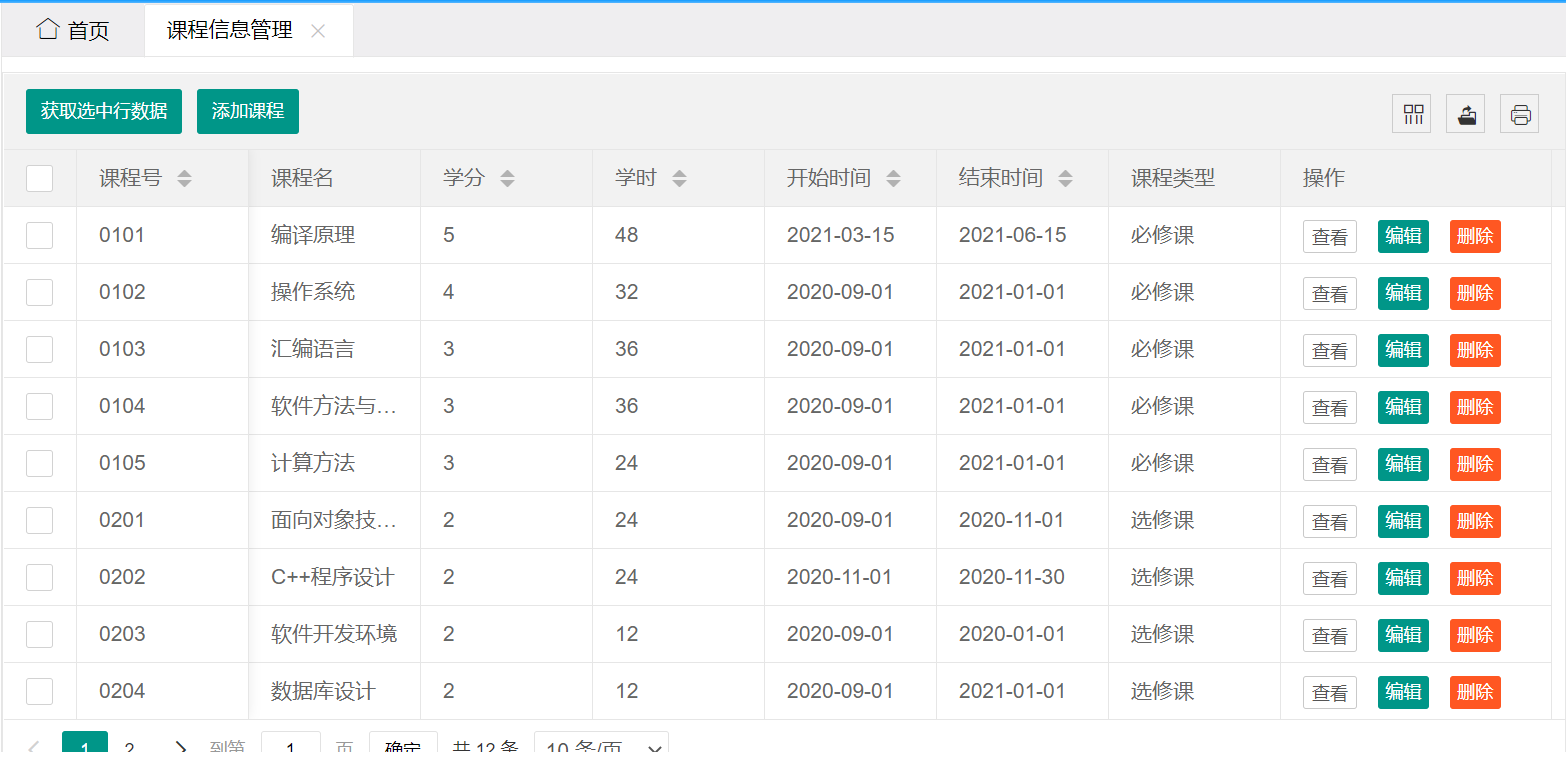


图5.4.2-课程信息管理

通过点击右侧每行的操作栏编辑可修改课程信息，删除可删除课程信息

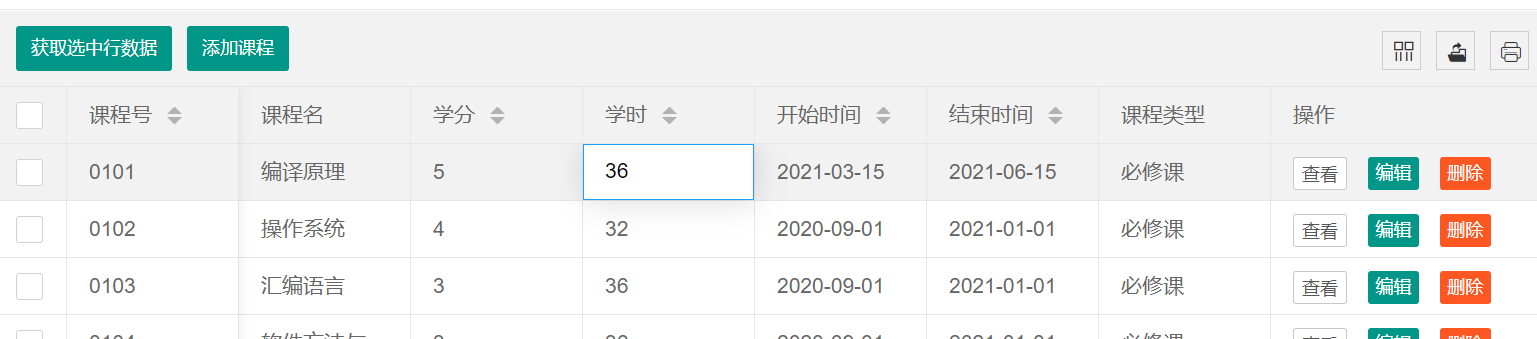


图5.4.3-修改课程前页面



图5.4.4-点击修改后修改成功

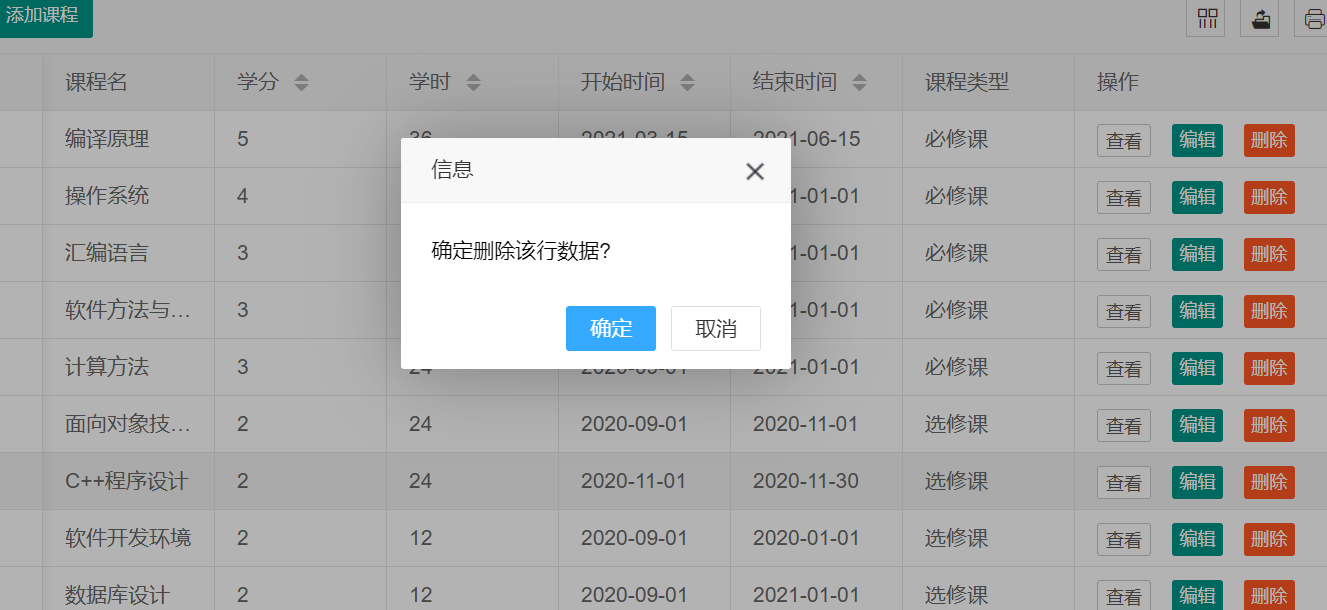


图5.4.5-删除C++程序设计这门课操作



图5.4.6-删除后课程列表

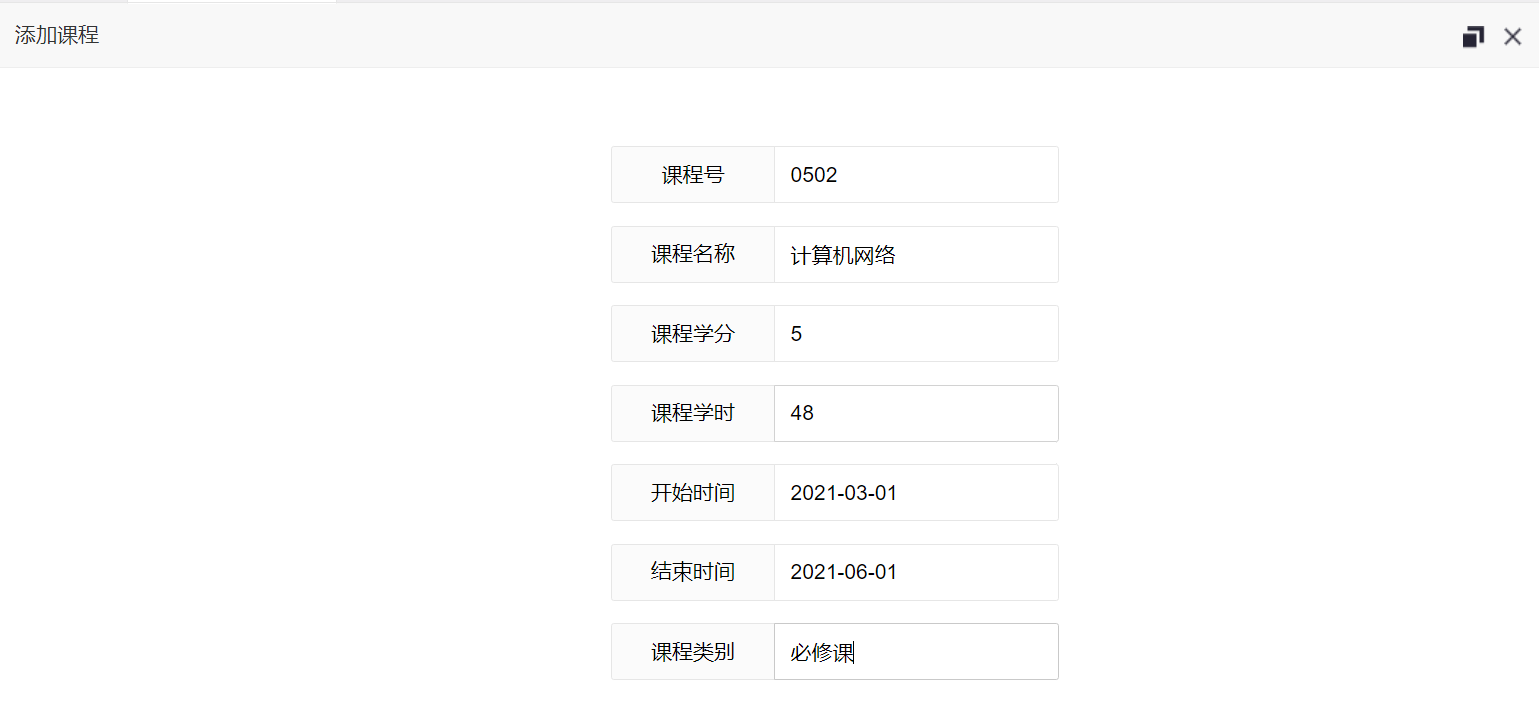


图5.4.7-添加课程页面

点击立即提交后成功添加课程计算机网络

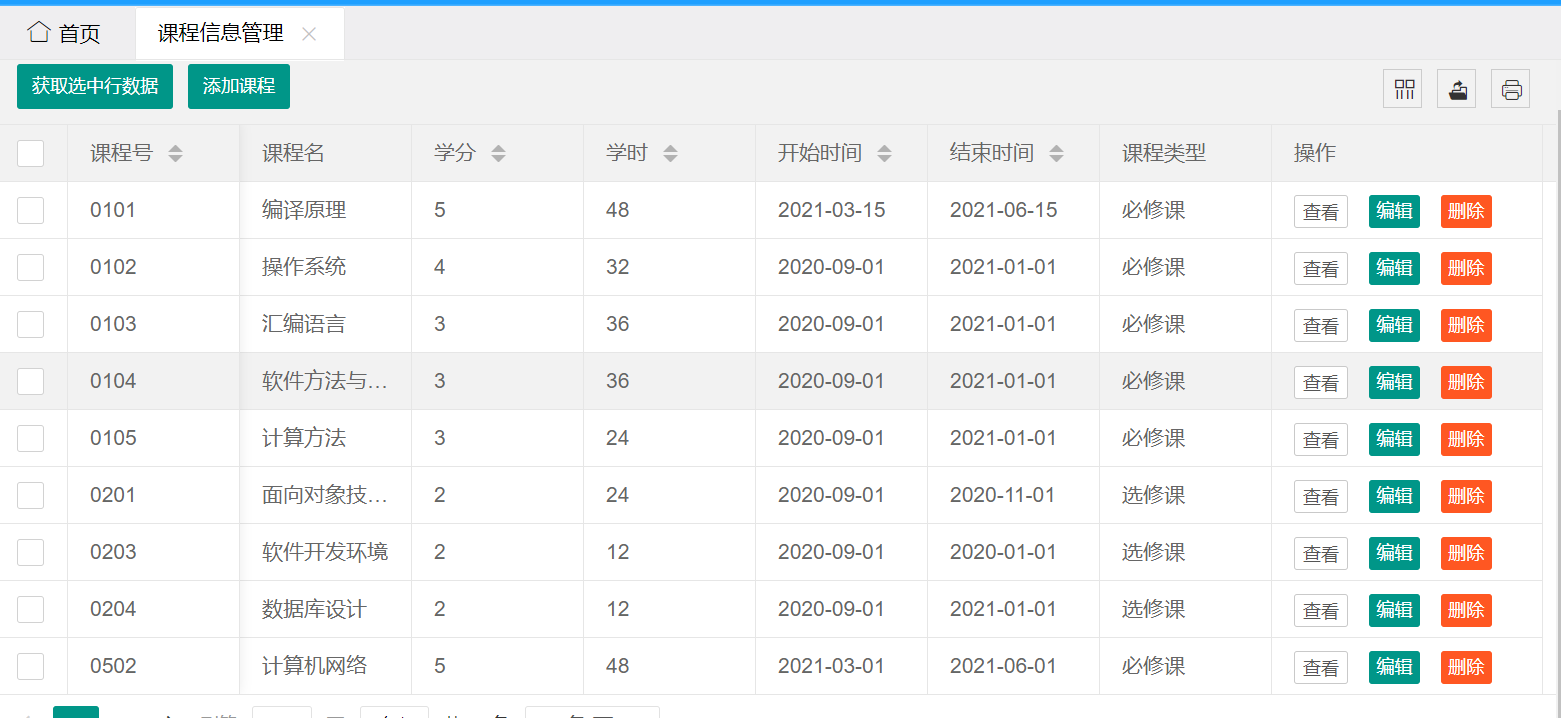


图5.4.8-成功添加后页面

至此，各功能模块测试均正常。

## 六、总结

通过为期8周的JavaWeb课程学习和12周的教务管理系统设计，主要学到了完整系统设计的流程以及Web前端的一些设计思路和架构。针对一个系统来讲，首先要进行需求分析，明确系统要有哪些功能，然后通过系统设计最主要是数据库的设计，只有数据库关系模式建立的合理才能保证后续业务功能的完成不会出差错，否则会导致系统实现阶段发现问题不好回该的问题。对于JavaWeb技术，最核心的是学到了MVC的这样一个简单而且好用的Web端设计结构，M即模型层与数据库关系表相对应，C即控制器服务器端的Servlet业务逻辑层，然后通过V视图层将M和C联系起来，同时搭配DAO这样一个数据访问中间缓冲层， 便于控制层简单直接的对数据库数据进行增删改查等操作，同时提高了代码的复用性和可维护性，这样便形成了一个相对完整的Web端开发架构。同时学到了B/S这样一个网络架构的实现原理，它不同于C/S，用户不需要下载专门的客户端，而只需要通过浏览器访问服务器发送请求等待服务器端程序返回文档的方式来实现页面展示。除了学到了JSP页面如何编写和Servlet程序如何设计之外，还学到了HTML、CSS、JS的基本使用和一些前端框架使用。