教务系统方案设计

1 系统目的

此项目以教务系统为依托，目的在于将所有相关操作规范化，流程化。对用户而言，涉及到的只有开始和结果，无需关心中间的每一个具体环节；对于程序而言，将所有的操作抽象为触发动作以及控制条件称为微动做和微条件，所有的动作和条件构成了整个系统的工作流程；动作触发系统的工作，条件控制系统的流程。

2 主要模块

系统第一部分是课程管理。此部分的功能是以学生为主要关注对象，以学生的选课为核心操作，方便学生，导师，任课教师，教务科随时掌握学生的课程情况。以下将该系统的用户按照角色分为以上四种进行功能详细描述。几种角色如图2.1所示。

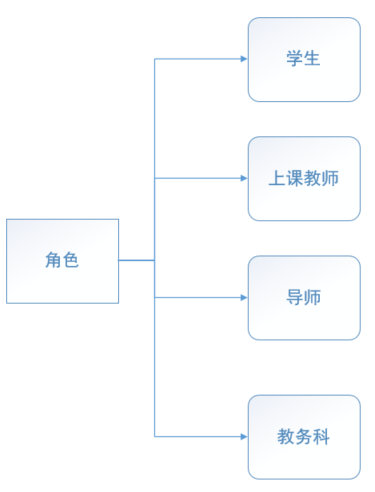


图2.1 角色示意图

根据角色的划分，其实有重叠的部分，普通教师既是任课教师，也是学生的导师，因此可以考虑将两者合并到同一类，使用权限控制来实现。

2.1 学生

学生的动作是选课，条件是此课程可选，课程可选由几个条件构成

（1）课程开课

（2）上课时间不冲突

在此条件下，需要一张课程信息表，需要一张学生信息表，课程表信息可以看到该课程的具体情况，学生信息是自己的信息。从页面上来看，学生部分主要由三个页面构成，如图2.2所示。

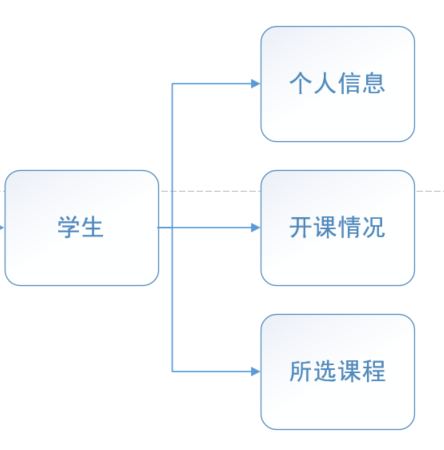


图2.2 学生角色下的前端页面

学生的所选课程展示有两种方式，一种是直接以列表的形式展示，另外一种是以课程表的形式展示。

2.2 普通教师

普通教师的动作主要是查看自己开课的选课情况

查看自己的学生的选课情况

其进入系统之后前端页面如图2.3所示。

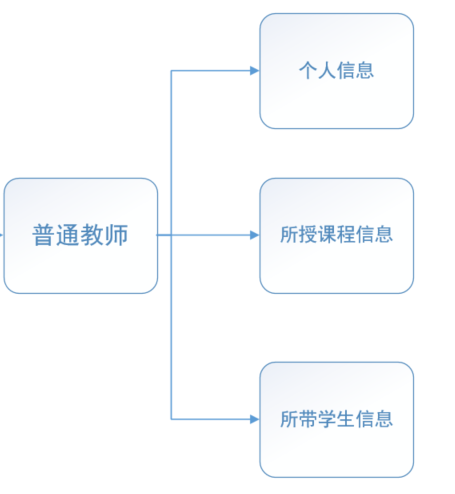


图2.3 普通教师前端页面

教师所带的学生信息里面展示了自己名下研究生的选课情况以及所有的学生可以公开的信息。

2.3 教务科

系教务科是系统管理员，从权限上而言是超超级用户，就具备对所有信息的编辑权利。

所有的学生信息（添加，编辑学生信息，删除，包括选课情况（超链接跳到其他页面），）

所有的课程信息（添加，编辑课程信息，排课（自动/手动），）

所有的教员信息（添加，编辑，删除）

其前端页面如图2.3所示。

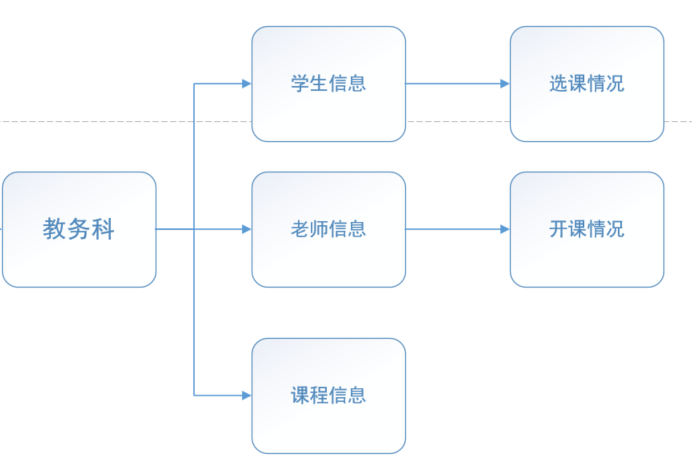


图2.3 系教务科页面

第二层页面表示可以通过链接跳转到相应的页面，只有教务科拥有对所有用户信息的增删改权限，尤其是增加和修改，学生和老师不得增加和修改用户信息。

3 数据模型

用户信息表

用户id 用户编号（学号/工号）用户类别（0，1，2，3）

表3.1 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户信息表-存储可以登录系统的用户 | | | | |
| 字段名 | name | type | length | note |
| id | id | auto | auto | primary key |
| 用户名 | userName | string | 20 | 证件号（学号或者工号） |
| 密码 | password | string | 200 | hash(password) |
| 用户类型 | userType | int | 3 | 0超级用户，1学生，2教师 |

3.1 学生信息表

姓名 学号 专业 年级 导师

表3.2 学生信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生信息表-存储学生自己的信息 | | | | |
| 字段名 | name | type | length | note |
| id | id | auto | auto | primary key |
| 姓名 | name | string | 20 |  |
| 学号 | userID | string | 11 | unique = True |
| 学院 | academy | string | 50 |  |
| 专业 | prefession | string | 50 |  |
| 入学年份 | enterTime | date | auto | 日期格式 |
| 学制 | needTime | int | auto | 整形数字 |
| 学年 | state | int | auto | 表示学生现在的年级 |
| 导师 | tutor | string | 20 | 存放导师姓名 |

使用整数来表示学生现在的年级，同一个学生的年级不会大于该学生的state。

3.2 课程信息表

课程名称 课程编号 课程类别 上课老师 开课季节（春/秋） 上课时间 上课地点 人数上限

表3.3 课程信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 课程信息表 | | | | |
| 字段名 | name | type | length | note |
| id | id | auto | auto | primary key |
| 课程名称 | name | string | 50 |  |
| 课程编号 | number | string | 20 | 固定生成 |
| 上课老师 | teacher | foreingkey |  | 外键链接教师信息 |
| 授课对象 | student | int |  | 和学生的state对应 |
| 上课时间 | teachTime | int |  | 把每周可用时间排列 |
| 上课地点 | teachPlace | string | 20 |  |
| 开课季节 | season | bool | 1 | True = 秋 |

3.3 教员信息

姓名 工号 课程 学生（外键）

表3.4 教师信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 教师信息表-存储老师自己的信息 | | | | |
| id | id | auto | auto | primary key |
| 姓名 | name | string | 20 |  |
| 工号 | userID | string | 11 | unique = True |
| 职称 | position | string | 20 | 不可编辑 |
| 学院 | academy | string | 50 |  |
| 专业 | prefession | string | 50 |  |

3.4 学生选课情况

需要控制权限，不同的人看到的情况不一样，教务科人员可以看到所有学生的选课情况，

表3.5 选课信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 学生选课情况表 | | | | |
| 字段名 | name | type | length | note |
| id | id | auto | auto | primary key |
| 学生 | student | foreignKey | auto | 外键关联到学生 |
| 所选课程 | lesson | foreignKey | auto | 外键关联到课程信息 |

从模型层面上来说，选课就是在学生信息表核课程信息表之间建立了一个链接，这个连接通过一张将学生信息核课程信息连接在一起的一对多表格来实现，这张表记录了学生对应的课程；从表现层面来说，选课就是学生出发的一个动作，动作可选是条件，动作也是一个条件，此动作的结果是在学生的已选课程中多了一份课程信息；从控制层面上而言，选课的过程就是将选择的课程作为参数进行查询，建立模型，将这一新增的对应关系保存。

选课和退课是两个相反的动作，出发的动作不相同，内部控制的条件也不相同。

4 项目技术

服务端使用java spring，spring MVC 框架，访问数据库Java Hibernate.

前端使用bootstrap显示页面，angularJS实现数据交互。

5 页面设计

5.1 登陆验证

验证不同的角色，返回不同的页面，记录登录时间

5.2 学生

个人信息

可选课程

所选课程

5.3 普通教师

个人信息

开课信息（教师的课程以及这门课程的选课情况）

所带学生选课情况

5.4 教务科

学生信息（可跳到学生详细信息）

教师信息（可跳转学生详细信息）

课程信息（每门课的详细信息）