

Project1: 选课系统

小组成员：朱瑞媛，卢颖，王丹青

【项目介绍】

该项目网页主要实现选课系统的基本功能，包括用户的登录及密码修改，学生课程查询，选课和退课，以及教师开设课程，删除课程和查看选课学生信息。

【小组分工】

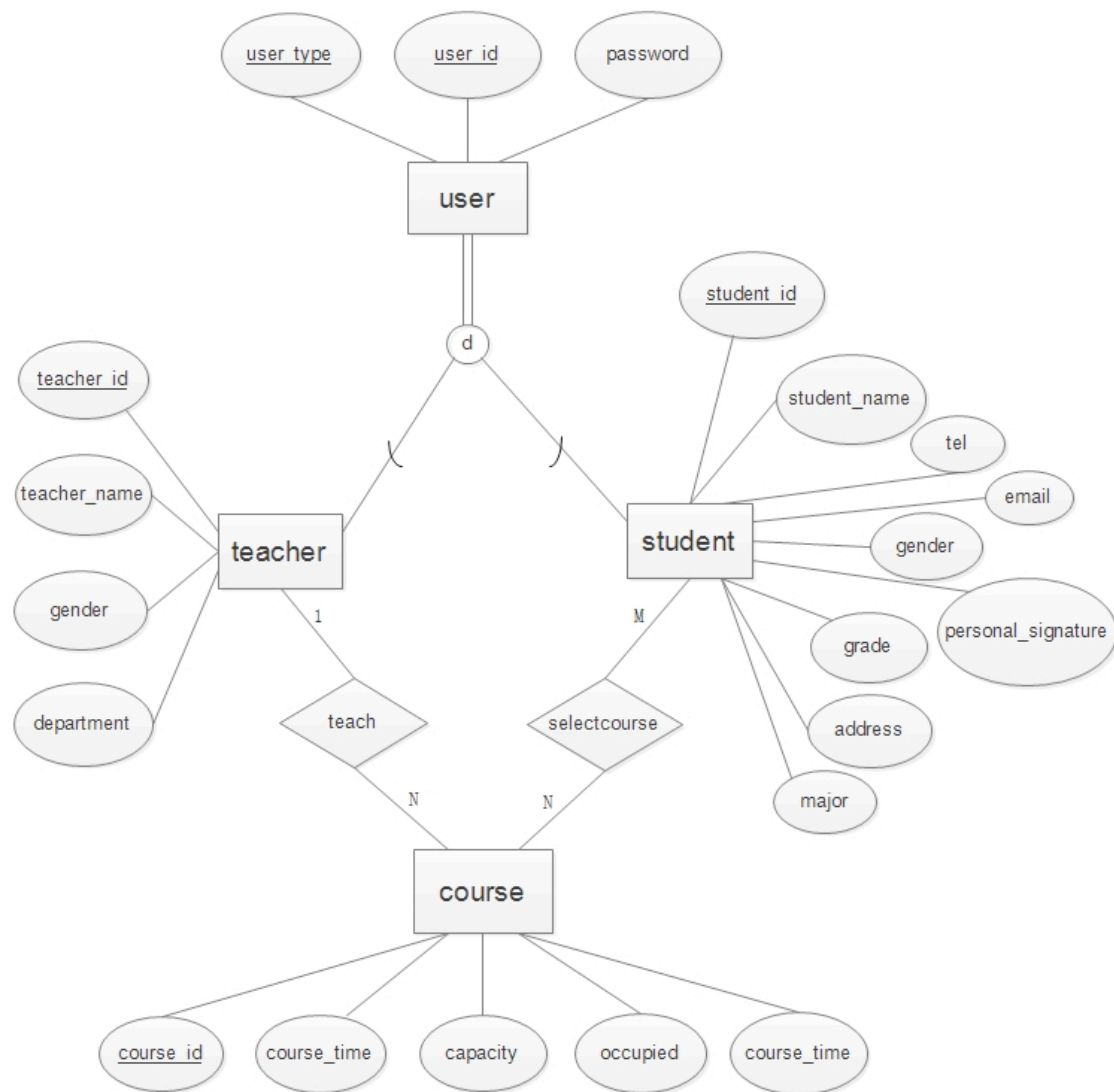
朱瑞媛：功能规划，网页制作，选课系统功能的实现。

卢颖：底端数据库的数据生成及插入，优化了部分功能。

王丹青：数据库结构以及 er 图设计，编写报告。

【项目内容】

（一）ER 图设计



（二）关系模型设计

用户实体 user(user_id, password, user_type)

学生实体 student(student_id, student_name, gender, grade, address, tel., email, personal_signature, major)

教师实体 teacher(teacher_id, teacher_name, gender, department)

课程实体 course(course_id, course_time, capacity, occupied, course_name, teacher_id)

选课关系 selectcourse(course_id, student_id)

//这两个键是这个表的主键同时也是外键，但是由于不知道如何同时添加直下划

//线和波浪下滑线，特此注明。

1.用户实体

登录网站的用户全体，主要用于登录和密码修改。它是 **student** 和 **teacher** 的超类实体，其整体特化的子类实体 **teacher** 和 **student** 不相交。

user_type:用户类型，0 表示教师，1 表示学生

user_id:唯一标示用户身份的 id，教师为 1-18，学生为 1001-1050.

password: 用户密码，至少 4 位，在数据库中以 md5hash 加密

样例:

USER_ID	PASSWORD	USER_TYPE
1	e10adc3949ba59abbe56e057f20f883e	0
2	ccffb0bb993eeb79059b31e1611ec353	0
3	82470256ea4b80343b27afccbca1015b	0

数据总数: 70

2.学生实体

student_id: 用来唯一标示学生的学号

student_name: 学生姓名

gender: 性别

grade: 年级

address: 住址

tel.: 电话

email: 邮件地址

personal_signature: 个性签名

major: 专业

样例:

STUDENT_ID	STUDENT_NAME	GENDER	GRADE	ADDRESS	TEL	EMAIL	
1001	Antwon Tillman	M	4	6882 Hegmann Cliffs Apt. 528 Beiershire, AR 46342-6459	13941039256	std1001@kwschool.edu.cn	Eaque exerc
1002	Yoshio Schumm	M	1	5143 Eugenia Pine Apt. 313 Port Lacyview, IA 68555	13019218948	std1002@kwschool.edu.cn	Aperiam ser
1003	Viva Kunze PhD	M	3	724 Damaris Plains Batzhaven, MN 22912-0619	13188231328	std1003@kwschool.edu.cn	Quia a quas

数据总数: 50

3.教师实体

teacher_id:用来唯一标示教师的教师号

teacher_name:教师姓名

gender:性别

department: 所属部门

样例:

TEACHER_ID	TEACHER_NAME	GENDER	DEPARTMENT
1	Prof. Kevin Wang	M	COMP
2	Dr. Brenden Poulos	F	CHIN
3	Dr. Lucky Kertzmnn	M	CHIN

数据总数: 20

4.课程实体

course_id: 唯一标示课程的课程代码

course_time: 课程时间, 由两位数字组成, 第 1 位表示一星期中第几天 (范围 0 到 4), 第 2 位表示一天中的第几节课 (范围 1 到 6)

capacity: 课程最多能够容纳的学生数

occupied: 当前选课人数, 随着选课动态变化

course_name: 课程名称, 不同的课程 (不同课程号) 可以有相同的名称

teacher_id: 教师号。规定一门课程 (同一课程号) 只能由一名老师开设, 一名老师可以教授多门课程

样例:

COURSE_ID	TEACHER_ID	COURSE_TIME	CAPACITY	OCCUPIED	COURSE_NAME
1	16	42	31	4	Graph theory
2	11	11	29	9	Ancient Chinese
3	9	24	29	4	Selections of Ancient poems

数据总数: 70

5.选课关系

course_id: 课程代码

student_id: 学号

样例:

STUDENT_ID	COURSE_ID
1001	2
1001	3
1001	5

数据总数: 294

（三）开发设计流程

1. 项目环境

- 语言：Python
- 编程环境：Python 2.7
- 数据库连接：本地连接，使用 JDBC，JDK 和 jaydebeapi
- 项目建立：借助 flask 框架和 jinja2 模板引擎
- 可移植性：配置完本地环境之后，将程序 ZRy-PJ.py 中

```
conn = jaydebeapi.connect('com.informix.jdbc.IfxDriver',
                        'jdbc:informix-sqli://crl.ptopenlab.com:9088/d_1460372203599252:INFORMIXSERVER=ifxserver1;USER=yixyfluu;PASSWORD=4e',
                        "D:\\Database\\ifxjdbc.jar", )
```

中"D:\\Database\\ifxjdbc.jar"修改为本地 ifxjdbc.jar 所在目录即可

2. 项目目录

```
DatabasePJ
├── static                # 静态资源，如图片、CSS、Js 等，本项目中为空
├── templates            # 模板引擎中用到的模板
│   ├── partials        # 输出信息部件
│   │   ├── error_msg.html    # 错误信息输出
│   │   └── success_msg.html  # 成功信息输出
│   ├── base.html        # 所有网页公用模板
│   ├── Change_password.html # 密码修改界面
│   ├── StudentIndex.html  # 学生界面
│   ├── StudentInformation.html # 学生信息界面
│   ├── TeacherIndex.html  # 教师界面
│   └── Welcome.html      # 登录界面
└── ZRy-PJ.py            #主程序
```

3. 实现功能

①用户登录：在 welcome 界面中选择身份（学生或老师），输入 id 和对应密码
代码片段：

```
if password_hash(password) != result[0][1]:
    flash("Wrong password.", 'error')
    return redirect('/')

session['user_id'] = result[0][0]
#print session['user_id']
if user_type == 'student' and result[0][2] == 1:
    print "====="
    flash("Log in successfully.", 'success')
    return redirect('/StudentIndex')
```

②修改密码：在 student 或 teacher 界面中点击 Change Password 按钮，进入修改密码界面。首先确认旧密码，再输入两次新密码，当旧密码正确且两次新密码相同（长度大于 4）时修改密码成功

代码片段：

```
if result[0][0] != password_hash(request.form['current_password']):
    flash("Wrong password", 'error')
    return redirect('/passwordchange')
else:
    if len(request.form['new_password']) < 4:
        flash("Too short password.", 'error')
        return redirect('/passwordchange')
    if request.form['new_password'] != request.form['repeat_password']:
        flash("Two password are not the same", 'error')
        return redirect('/passwordchange')
    else:
        flash("Change password successfully.", 'success')
        cursor.execute('update user set password =? where user_id =?', [password_hash(request.form['new_password']), session['user_id']])
```

③用户登出：在 student 或 teacher 界面中点击 log out 按钮，即可退出，返回 welcome 界面

代码片段：

```
flash('Logout Successfully!', 'success')
del session['user_id']
return redirect('/')
```

③学生操作：

(i) 查询课程：根据课程代码或者课程名称查询课程

代码片段：

```
cursor.execute(
    'select course_id, course_name, course_time, teacher_name, capacity, occupied from course, teacher where course.teacher_id = teacher_id and (' + course_name + '))
```

(ii) 选课：当输入课程名称唯一，或输入课程代码时，可以进行选课。在选课时检查时间冲突和人数是否上限。选课成功增加 selectcourse 记录以及更新 course 的 occupied 属性

代码片段：

```

if course_info[0][4] == course_info[0][5]:
    flash("The number of student reaches the limit.", 'error')
    return redirect('/StudentIndex')
elif len(conflict) != 0:
    flash("Time conflicts.", 'error')
    return redirect('/StudentIndex')
else:
    cursor.execute('insert into selectcourse (student_id, course_id) values (?, ?)',
                  [session['user_id'], course_info[0][0]])
    cursor.execute('update course set occupied = occupied + 1 where course_id = ?',
                  [course_info[0][0]])
    flash("Select successfully.", 'success')

```

(iii) 退课：在该学生已选课程中唯一，或者输入课程代码时，可以进行退课。在退课时检查该学生是否已选此课程。退课成功删除对应 selectcourse 记录以及更新 course 的 occupied 属性

代码片段：

```

if len(selected) == 0:
    flash("The course has not been selected.", 'error')
    return redirect('/StudentIndex')
if len(selected) > 1:
    flash("You have more than 1 course with the course name.", 'error')
    return redirect('/StudentIndex')
# print "delete successfully"
cursor.execute('delete from selectcourse where student_id = ? and course_id = ?',
               [session['user_id'], selected[0][0]])
cursor.execute('update course set occupied = occupied - 1 where course_id = ?',
               [course_info[0][0]])
flash("Delete successfully.", 'success')
return redirect('/StudentIndex')

```

④教师操作

(i) 开设课程：在输入信息完整时，可以进行开设课程。在开课时检查时间冲突和该门课程是否已开（课程代码重复）。开课成功后增加 course 记录

代码片段：

```

if len(course_id) != 0 and len(course_name) != 0 and len(course_time) != 0 and len(capacity) != 0:
    #courseinfo = get_course(course_id)
    #print "-----"
    cursor.execute('select course_id from course where course_time = ? and teacher_id = ?', [course_time,
                                                                                             session['user_id']])
    conflict = cursor.fetchall()
    cursor.execute('select teacher_id from course where course_id = ?', [course_id])
    exist = cursor.fetchall()
    if len(conflict) != 0:
        flash("Time conflicts.", 'error')
        return redirect('/TeacherIndex')
    #modified
    if len(exist) != 0:
        flash("The course exists.", 'error')
        return redirect('/TeacherIndex')
    #add operation
    cursor.execute('insert into course (course_id, course_name, course_time, teacher_id, capacity, occupied)')
    flash("Add successfully.", 'success')
    return redirect('/TeacherIndex')

```

(ii) 删除课程：输入课程代码，可以删除课程。在删除课程时检查开门课

程是否开设。删除课程成功后删除对应的 course 和 selectcourse 记录
代码片段:

```
if operate_type == 'remove':
    #print "==="
    if len(course_id) != 0:
        cursor.execute('select course_id from course where course_id = ?', [course_id])
        course_exist = cursor.fetchall()
        if len(course_exist) != 0:
            cursor.execute('delete from selectcourse where course_id = ?; delete from course where course_id = ?;',
                           [course_id, course_id])
            flash("Remove successfully.", 'success')
            return redirect('/TeacherIndex')
        else:
            flash("The course does not exist.", 'error')
            return redirect('/TeacherIndex')
    else:
        flash("No input.", 'error')
        return redirect('/TeacherIndex')
```

(iii) 查看选课学生信息: 在教师界面的开课信息中, 点击 occupied 显示的数字, 可以查看选择该课程学生的个人信息。

代码片段:

```
cursor.execute(
    'select student.student_id, student_name, gender, major, grade, tel, email, address from student, selectcourse where student.stu'
    [course_id])
student_info = cursor.fetchall()
cursor.execute('select course_id, course_name from course where course_id = ?', [course_id])
query_course = cursor.fetchall()
```