桂林电子科技大学2019-2020学年 第2学期

**数据库系统原理综合 实验报告**

**课号： 1922159 任课老师：唐麟**

|  |  |  |  |
| --- | --- | --- | --- |
| 项目名称 | **课程选择系统** | | |
| 院 系 | **计算机与信息安全学院** | 专业 | **软件工程** |
| 实验日期 | **2021年6月20日** | | |

团队任务分工表

|  |  |  |  |
| --- | --- | --- | --- |
| 题目 | 课程选择系统 | | |
| 负责人 | 1900301236 谢浚霖 | | |
| 合作  方式 | 使用git和github进行多人远程协作 | | |
| 序号 | 学号 | 姓名 | 个人任务描述 |
| 1 | 1900301236 | 谢浚霖 |  |
| 2 | 1900300819 | 卢林军 |  |

# 1、引言

## 1.1、社会背景

在传统的选课模式中，选课记录大多数都由人工手动完成，这样既限制了信息流通，又浪费了时间，甚至可能造成一些错误的发生，不能很好地使用户快速解决自己也业务。随着科学技术的发展，网络web技术得到不断发展，现在该项技术已经成熟，人们视线了帮助学生和教师高效，快速地完成选课排课需求的平台——课程选择系统。

先如今的课程选择系统开启了新的模式，摆脱了手工操作的过程，优化了用户的操作，通过用户友好型的人机交互，使用简洁地操作完成复杂的功能，同时由于线程同步的出现，避免了“脏数据”的出现。根据不同的用户不同的需求，可以有针对性得改变着平台提供的服务，轻松便捷地引导用户完成业务，即教师和学生根据现有课程进行合理排课和选课。但是在信息飞速发展的今天海量信息每天都在扑面而来，不断进行更新和迭代，为了是用户快速了解业务的状态，并高效地完成任务，选课系统仍然需要完善相关技术。因此，在切实提高用户业务质量的基础上改善，加强用户信息化条件的建设，利用现今流行和高速发展的web技术将选课系统逐渐完善是当前的首要任务。综上所述，课程选择系统的开发和发展是有必要的。

## 1.2、用户需求

1.2.1、学生端

学生能够在课程选择系统中，查看自己的个人信息，能够查看所有的课程计划安排。提供可以选择的课程列表，在点击选课后会展示出该课程在对应学期的所有课程计划。学生可以通过点击选课进行课程选择。在选择课程结束后还可以对已选课程进行退课。学生还可点击查询自己的已选课程。

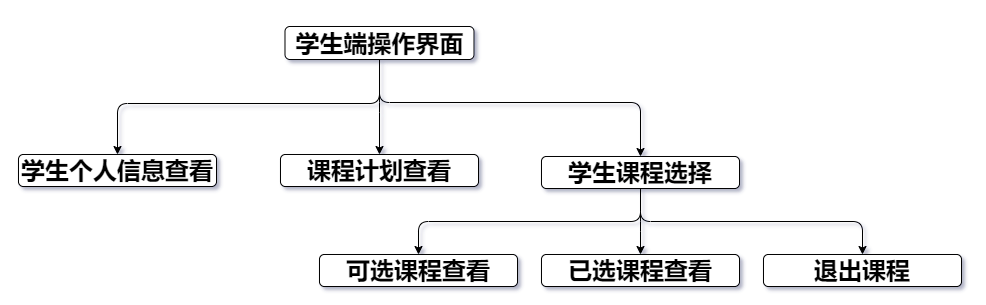
1.2.2、教师端

教师能够在课程选择系统中，查看自己的个人信息，可以查看自己的课程计划，能够添加自己的课程计划，根据自己添加的课程计划，生成自己已经选择要教的课程信息，通过获取到的教师课程计划，在对应的课程计划中会显示出学生当前课程已经选择的人数，教师可以有选择地删除自己的课程计划。

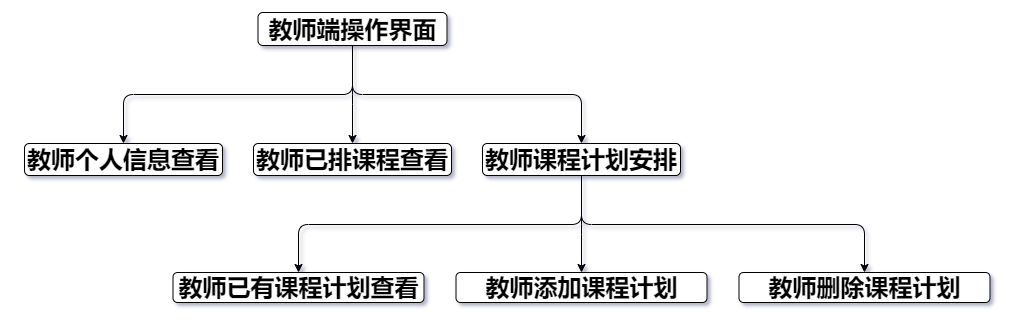
# 2、功能分析

根据需求可以得知，课程选择系统应该提供两个不同的操作端，一个为学生端，为学生提供课程选择服务，另一个为教师端，为教师提供排课服务。为了鉴别用户是否为学生与教师，还需要一个登陆界面，通过单选框的方式确认用户是登陆学生端还是教师端，根据不同的选择，分别到不同的数据库表中进行验证。

对于登陆界面，应该提供登陆功能模块，通过用户对客户端的不同选择到不同的数据库中进行验证，并通过不同的登陆行为进入不同的客户端。

 对于学生端的设计，学生端应该具备的功能如下图2-1所示：

**图2-1 学生端界面功能图**

 对于教师端的设计，教师端应该具备的功能如下图2-2所示：

**图2-2 教师端界面功能图**

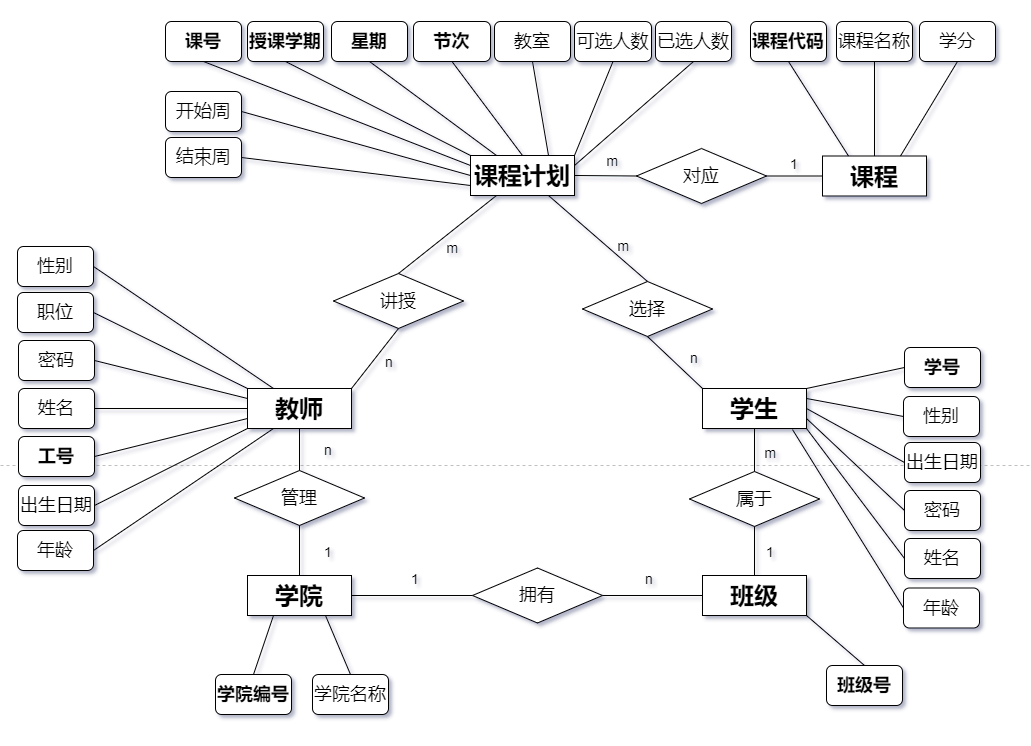
# 3、数据库关系

## 3.1、数据库关系分析

根据上述需求调研，对构建的数据库的关系模式抽象描述如下：

一个学院管理多个教师，管理多个班级；一位老师只能属于一个学院，一个班级也只能属于一个学院。学院属性为：学院编号，学院名称，班级属性为：班级号，教师的属性为：工号，姓名，性别，职位，年龄，出生日期以及教师端的登陆密码。一个学生属于一个班级，一个班级可以有多名学生。学生的属性为：学号，姓名，性别，年龄，出生日期以及学生端的登陆密码。一个门课程因为课号的不同可能对应多个课程计划，一个课程计划只能对应一门课程。课程计划的属性为：课号，授课学期，星期，节次，教师，可选人数，已选人数，开始周以及结束周。课程的属性为：课程代码，课程名称，学分。一位老师可以讲授多门课程，一门课程也可以由多个老师讲授。一位学生可以选择多门课程，一门课程也可以被多个学生选择。

## 3.2、概念结构与逻辑结构设计

 根据关系模式的抽象描述，可以绘制出E-R图，如下图3-1所示：

**图3-1 课程选择系统E-R图**

根据上述E-R图的设计以及E-R图之间的转化关系模型的规范，可以将关系模型转化成如下表3-1所示的关系，根据BC范式的定义，该关系模式为BC范式。

**表3-1 课程选择系统的关系模式**

|  |
| --- |
| 学生 (学号，姓名，性别，年龄，出生日期，班级号，登录密码)  教师 (工号，姓名，性别，出生日期，年龄，职位，学院编号，登录密码)  学院 (学院编号，学院名称)  班级（班级号，学院编号）  课程计划 (课号，授课学期，星期，节次，教室，可选人数，已选人数，开始周，结束周，课程代码)  课程 (课程代码，课程名称，学分)  选择 (学号，课号)  讲授(课号，工号) |

## 3.3、 物理结构的设计与实现

依据上述的关系模式，使用OpenGauss数据库建立了八张关系表，他们的表明以及对应的名称如表3-2所示：

**表3-2 运用OpenGauss数据库建立的八张关系表**

|  |  |
| --- | --- |
| 表名 | 备注 |
| student | **学生表** |
| teacher | **教师表** |
| college | **学院表** |
| class | **班级表** |
| schedule | **课程计划表** |
| course | **课程表** |
| teach | **讲授关系表** |
| selection | **选课关系表** |

3.3.1、学生表(student)

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 学号 | sno | varchar(50) | 主码 |
| 姓名 | name | varchar(50) |  |
| 密码 | passwd | varchar(50) |  |
| 性别 | sex | char(1) |  |
| 生日 | birthday | smalldatetime |  |
| 年龄 | age | int |  |
| 班级号 | classnum | varchar(50) | 班级表的外码 |

在学生表用来存储学生的个人信息，学生的学号sno能唯一标识一个学生，因此将学生的学号作为学生表的主码，根据完整性约束原则，通过设置班级号classnum的外码用来获取学生对应的班级信息。如表3-3所示：

**表3-3 学生表**

3.3.2、教师表(teacher)

在教师表中工号能够唯一标识教师的身份，所以将教师的工号tno作为教师表的主码，同时根据完整性约束原则，设置学院编号collegenum作为学院表的外码，用来获取教师所处学院的信息。如表3-4所示：

**表3-4 教师表**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 工号 | tno | varchar(50) | 主码 |
| 姓名 | name | varchar(50) |  |
| 密码 | passwd | varchar(50) |  |
| 性别 | sex | char(1) |  |
| 生日 | birthday | Smalldatetime |  |
| 年龄 | age | int |  |
| 职位 | position | varchar(50) |  |
| 学院编号 | collegenum | int | 学院表的外码 |

3.3.3、学院表(college)

学院的编号能够唯一标识学院的信息，因此设置学院编号num为主码。学院表构成如图表3-5所示：

**表3-5 学院表**

|  |
| --- |
|  |
| **名称** | **字段** | **类型** | **说明** |
| 学院编号 | num | int | 主码 |
| 学院名称 | name | varchar(50) |  |

3.3.4、班级表(class)

班级表只有班级号num一个属性，设置其为主码，并且根据参照完整性，设置班级表中的学院编号collegenum为对于学院表的外码，如表3-6所示：

**表3-6 班级表**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| ID | id | int | 自增 |
| 班级号 | num | varchar(50) | 主码 |
| 学院编号 | collegenum | int | 学院表的外码 |

3.3.5、课程计划表(schedule)

课程计划表的属性较多，经过我们的分析，课程计划表记录的是一次课的记录，比如说一节课可能在第一周星期一上午上一次，这时候在课程计划表中有一个记录，同一个课号还可能在第一周的星期四上午第二节上一次课，因此我们会发现课号显然无法作为课程的唯一标识符，因此需要引入联合主码。我们将课号cno，授课学期semester，星期day，节次index作为主码。我们看看这样能否唯一标识一条记录，如果有了授课学期，我们就可以确定具体是哪一个学期开始，但是这个学期仍然有很多课；通过课号，可以确定是哪一门课，但是他可能在不同的星期有不同的一次课；通过星期，可以确定该课具体开设在哪一个星期，但是可能在这个星期里面的四节里都有可能；最后通过节次，就可以唯一确认每一次上课的具体记录了。同时根据参照完整性将，课程代码coursecode作为课程表和课程表之间的外码。如表3-7所示：

**表3-7 课程计划表**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 课号 | cno | varchar(50) | 主码 |
| 授课学期 | semester | varchar(50) | 主码 |
| 星期 | day | int | 主码 |
| 节次 | index | int | 主码 |
| 教室 | classroom | varchar(50) |  |
| 可选人数 | optional | int |  |
| 已选人数 | selected | int |  |
| 开始周 | startweek | int |  |
| 结束周 | endweek | int |  |
| 课程代码 | coursecode | varchar(50) | 课程表的外码 |

3.3.6、课程表(course)

在课程表中记录每一门课程的课程信息，课程代码coursecode能唯一标识一门课因此将课程代码设置为主码。课程表如表3-8所示：

**表3-8课程表**

|  |
| --- |
|  |
| **名称** | **字段** | **类型** | **说明** |
| 课程代码 | coursecode | varchar(50) | 主码 |
| 课程名称 | name | varchar(50) |  |
| 学分 | credit | float |  |

3.3.7、讲授关系表(teach)

讲授关系需要结合课程计划表和教师表的主码，联合作为一个新的关系表，主码为教师工号tno和课号cno的联合主码，但是该关系并没有其他的附加属性，因此得到如表3-9所示的讲授关系表。

**表3-9讲授关系表**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 教师工号 | tno | varchar(50) | tno与cno的联合主码 |
| 课号 | cno | varchar(50) | tno与cno的联合主码 |

3.3.8、选课关系表(selection)

选课关系需要结合课程计划表和学生的主码，联合作为一个新的关系表，主码为学号sno和课号cno的联合主码，并且该关系也没有其他多余的附加属性，因此可以得到如表3-10所示的选课关系表。

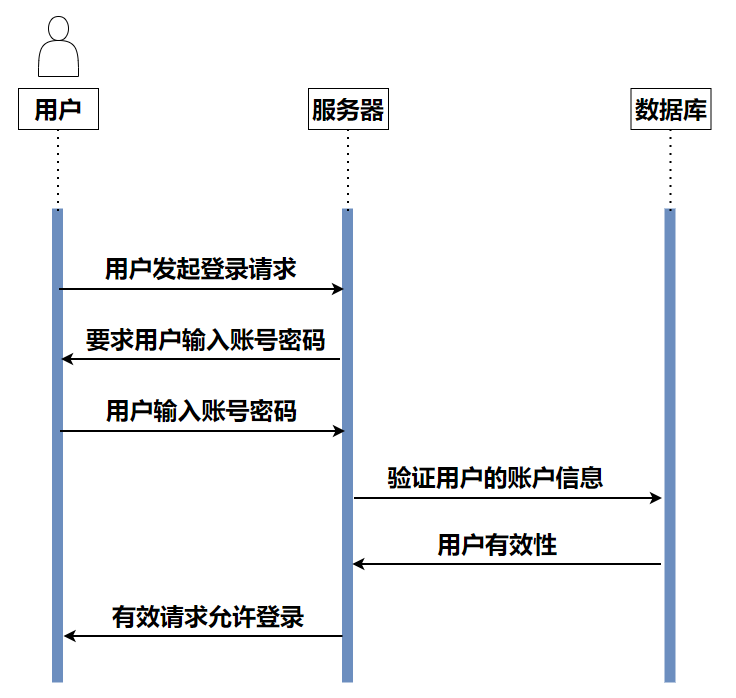
**表3-10选课关系表**

| **名称** | **字段** | **类型** | **说明** |
| --- | --- | --- | --- |
| 课号 | cno | varchar(50) | sno与cno的联合主码 |
| 学号 | sno | varchar(50) | sno与cno的联合主码 |

# 4、详细设计

## 4.1、登录界面

4.1.1、登录模块的设计与实现

登录界面是进入学生端和教师端所必须经过的一扇们。为了防止不良人士对数据进行篡改以及机器自动修改后台信息，登录模块是非常必要的。模块的时序图如图4-1所示。

**图4-1登录模块时序图**

由于用户分别存放在student表(学生表)和teacher表（教师表）中，用户的信息包含了密码字段(passwd)，因此用户在输入时会根据输入的内容调用登录接口，同时根据选择的登录端不同向不同的表中进行查询。由于考虑到前端发送请求时发送的是原密码，因此在数据库中存储的密码是经过MD5加密过的密码，使得登录的过程中能够保持较高的安全性。先获取对应账号的密码(已经经过MD5加密)，然后对输入的密码进行加密，判断两个MD5码是否相同，若相同则允许登录，否则会提示密码错误。

由于学生端和教师端的登录接口基本一致，因此只展示学生端的关键代码：

m = md5()

userid = None

db\_passwd = None

# 确认输入类型

if not isinstance(sno, str):

    sno = str(sno)

if not isinstance(passwd, str):

    passwd = str(passwd)

try:

    # 链接数据库(密码不方便展示)

    conn = psycopg2.connect(database="CourseSelectionSystem", user="\*\*\*\*\*\*\*\*",

                            password="\*\*\*\*\*\*\*\*", host="\*\*\*\*\*\*\*\*", port="\*\*\*\*\*\*\*\*")

    cursor = conn.cursor()

    # 获取输入学号的密码

    cursor.execute(f"select passwd, userid from student where sno={sno}")

    rows = cursor.fetchall()

    for row in rows:

        db\_passwd = row[0]

        userid = row[1]

    cursor.close()

    conn.close()

    # 对输入的密码进行加密

    m.update(passwd.encode("utf-8"))  # md5加密密码

    md\_passwd = m.hexdigest()

    print(db\_passwd, md\_passwd)

    # 判断两个密码是否相同

    if db\_passwd == md\_passwd:

        return {"status": "success", "data": str(userid)}

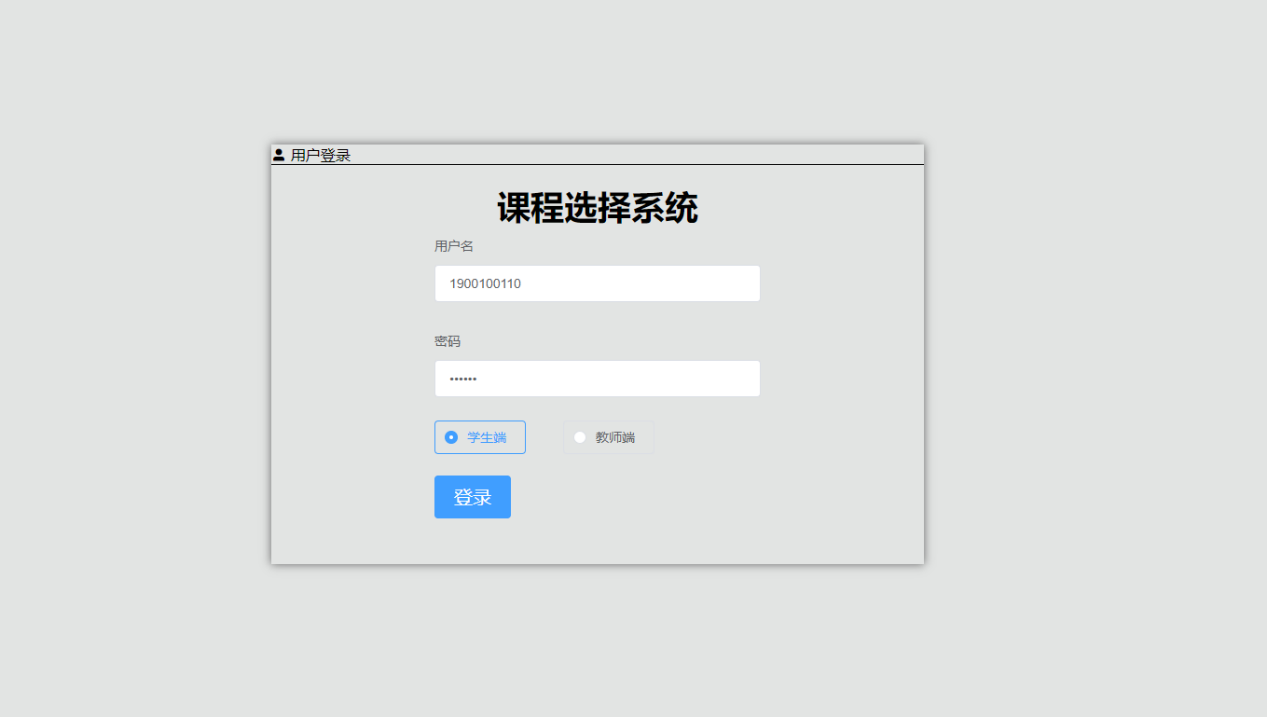
    else:

        return {"status": "failure", "data": []}

except Exception as e:

    traceback.print\_exc()

    return {"status": "error", "data": []}

录界面设计如图4-2所示：

**图4-2登录界面**

## 4.2、学生端

4.2.1、查看个人信息的模块

通过查看个人信息模块，学生可以在前端发出请求，后端根据学生的身份信息到后端中寻找该学生的信息，然后将数据返回到前端，前端通过合适的方式展示给学生，界面设计如图4-3所示：

**图4-3学生查看个人信息**

用户在登录后会产生对应的userid，并将其存储在浏览器缓存中，使用stu/getStuInfo/userid=<userid>的接口可以获得用户用户的个人信息，并将其存储在js变量中，在vue的data函数将js变量返回，使得这个存储着用户信息的变量将变成相应式变量在vue创建页面的视图上使用，对于信息的展示我们选择以表格的方式展示，涉及的vue代码如下所示：

data() {

var tableData;

// 通过axios发出请求，并通过对应的接口获取对应的数据信息

    this.$axios

      .get("stu/getStuInfo/userid=" + window.localStorage.getItem("userid"))

      .then((res) => {

        var temp = res.data.data[0];

        this.tableData = [

          {

            date: "姓名",

            name: temp.name,

          },

          {

            date: "工号",

            name: temp.sno,

          },

          {

            date: "性别",

            name: temp.sex,

          },

          {

            date: "年龄",

            name: temp.age,

          },

          {

            date: "出生日期",

            name: String(temp.birthday.split(" ")[0]),

          },

          {

            date: "班级",

            name: temp.classnum,

          },

        ];

      });

    return {

      tableData,

    };

  },

## 4.3、教师端

# 5、测试

# 6、关键技术

# 7、存在的问题及改进思路

# 8、总结