**第30次作业**

# 需求分析

**1.1目标**

该项目主要实现简单的学生信息管理系统，学生登录成功后可以查询班级内同学的信息以及所学课程的信息，老师则不仅能查看学生信息还能对其进行增删改的操作。

**1.2运行环境**

MySql数据库管理系统，Eclipse以及MySql开发平台，FireFox、Chrome等jsp页面显示显示浏览器，B/S模式。

**1.3业务描述**

**1.3.1登录验证**

如果登录时输入信息为空或者错误则会进行相应的错误提示，并且只有老师登录成功以后才能对数据库里面的内容进行更新操作，学生只能查看。

**1.3.2不同用户操作**

1.学生用户：

（1）查询学生信息：有两种方式，一是按照学生学号查询，二是按照学生姓名查询，按照学生姓名查询可以精准查询以及模糊查询；查询结果显示学生的学号、姓名、班级以及性别等个人信息。

（2）查询课程信息：按照课程名称进行查询，可以显示课程的id以及选择该课程的学生id。

2.老师用户：

（1）查询学生信息：有两种方式，一是按照学生学号查询，二是按照学生姓名查询，按照学生姓名查询可以精准查询以及模糊查询；查询结果显示学生的学号、姓名、班级以及性别等个人信息。

（2）查询课程信息：按照课程名称进行查询，可以显示课程的id以及选择该课程的学生id。

（3）新建学生信息：通过输入新同学的学号、姓名、班级性别等进行新同学注册并更新进入数据库。

**1.3.3分页功能**

分页功能：在查询总的记录数之后对其进行分页显示，例如可以分为一页显示3个用户，或者5个用户等。

**1.4性能需求**

操作界面：要求简洁明了，布局合理，对于用户来说简单易操作。

安全保密性：只有输入正确的用户名和密码才能进行相应的界面操作。

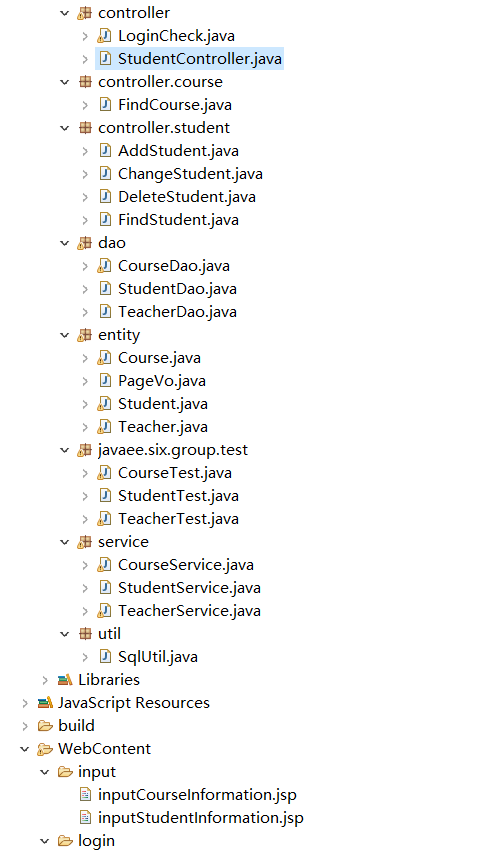
扩展性需求：可以根据用户实际需要，在不修改源代码的基础上进行功能增加。

信息的完整性：输入的新信息应该是合法的，如果输入信息不符合系统要求，应该给出警告。

稳定性：在一般情况下，项目是能够正常运行的，为了避免单一界面信息量爆炸的情况，采用分页查看是十分重要的。

数据处理能力：能够快速响应用户请求，并且运行效果稳定。

# 模块划分



**2.1 Service**

对不同的业务逻辑进行跳转调用：

（1）对Course的所有操作进行跳转调用；

（2）对Student的所有操作进行跳转调用；

（3）对Teacher的所有操作进行跳转调用。

**2.2 entity**

对所有需要使用到的数据库中存在的表格实体属性进行用于接收和显示的设置，具体有四种设置：

（1）Course属性设置

（2）分页实体属性设置

（3）Student属性设置

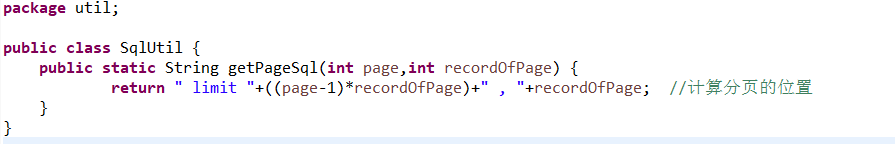
（4）Teacher属性设置

**2.3 controller**

用于响应View层的不同输入并跳转调用响应方法。如响应用户登录，验证并跳转、响应教师对学生信息的操作并跳转、响应Course查询并跳转、响应教师用户增加学生信息操作并跳转、响应查询学生信息操作并跳转、响应教师用户修改学生信息操作并跳转、响应教师用户删除学生信息操作并跳转。

**2.4计算分页的util层**

用于对显示的结果进行分页。



**2.5实现方法的dao层**

对Course、Student、Teacher的所有操作进行实现。

**2.6用于测试的Test层**

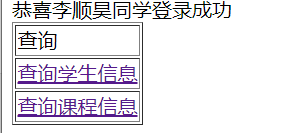
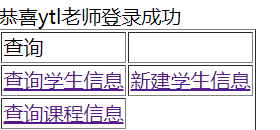
对Course、Student、Teacher中包含的所有操作进行测试。

**2.7显示查询结果的View层**

用于进行查询的交互，引导用户输入。

# 操作设计

我们小组做的是一个老师管理学生的简单系统，系统的功能是老师和学生都可以通过同一个登陆界面登陆到不同类别的用户界面，因为每个用户的权限不同，所以每个用户可以执行的操作也是不同的。



**3.1 数据库的设计**

因为有学生和老师，所以这两个是主表，而学生和老师通过课程来连接，所以课程是学生和老师之间的桥梁。学生的基本信息就是，学号，姓名，性别，班级，和密码。而老师是工号，姓名，课程号，和密码。而中间的桥梁课程表，就包含着两个外键，学号和工号，再包含一个课程号，所以基本的数据库就设计完成了。

**3.2 用户类的设计**

基于数据库的表，用户类的设计也就是和表里面一样，有着相同的属性，而且都是private，防止有人私自更改属性，只能通过公共的方法去设置和获取信息，每个用户类还有一个toString()的方法，主要是显示对象的各种信息，方便测试进行。

**3.2.1用户的操作**

每个用户类都有着自己私有的操作，而这些操作也是基于数据库上的增删改查，数据库的连接也是固定的六步。所以每个用户都有着基本的数据库操作。

**3.4功能实现**

我们是使用Dao来实现对数据的库的增删改查，还有分页的功能。然后将这个功能封装为Service使程序员在调用这些方法的时候比较方便和简洁，然后利用Controller。Contronller里面创建用户的类和Service服务，每个具体的Contronller实现不同的Service服务，然后实现服务将取得的用户放进用户的类里面， 有的用户是多个所以要用到泛型List进行保存，之后使用setAttribute的方法，将取得的用户信息放到域里面，使JSP能够访问到这些信息，并且显示这些信息。

**3.5 JSP页面**

我们的首页是登陆界面，登陆界面老师和学生都可以进行登陆，之后进入Contronller进行数据库的访问，判断是老师还是学生。之后往域里面放一个usertype来告诉程序现在登陆的用户类型。

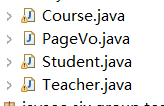


之后，跳转到不同用户的页面上面。

不同的用户可实现的功能也不相同，所以界面也不同。

# 操作实现说明

首先声明了这些实体用来暂存从数据库中得到的信息。



**4.1登陆功能**

登陆功能基于查找功能，主要是验证用户在登陆界面输入的信息是否可以与数据库中的密码进行配对或者是否能在数据库中找到对应的信息，在这样的前提下，登陆功能主要分配成以下的几点：

（1）登陆界面输入相关信息，跳转到Controll中的单独列出来的LoginCheck。



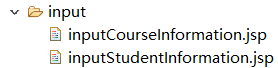
（2）检查登录信息LoginChink,分别调用TeacherService、TeacherDao以及StudentService、StudentDao来查询是否有这个人物，如果有的话是老师还是学生。

（3）如果这个人物存在，如果是老师，跳转到老师的操作界面，如果是学生，跳转到学生的登陆界面学生的界面只可以查找，老师的界面既可以查找也可以对信息进行修改。



**4.2查找功能**

查找功能主要是根据输入的信息来查询数据库中的信息，主要由以下几个部分组成（查找课程就用查找课程的相关代码，查找学生就用查找学生的相关代码）。

（1）Input界面：用户在这个界面中得到数据，然后转发给Controller。 

（2）Controlle：由Controller读取request的参数判断需要进行什么操作。 

（3）查找功能：Controller转发给FindStudent，由FindStudent调用相关类StudentService、StudentDao在数据库中查找相关信息并返回。



（4）将查找之后的结果显示到界面上。



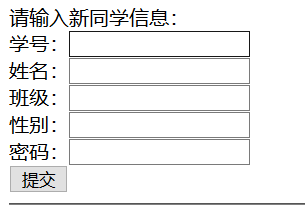
**4.3创建新学生信息功能**

（1）首先，有一个界面CreateStudent.jsp，在这个界面中进行信息的录入。 

（2）在进行录入之后发送到StudentController进行转发。



（3）创建后转发到AddStudent，在AddStudent处对数据库进行操作。

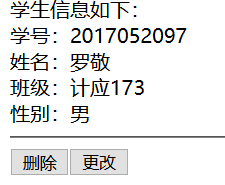


**4.4删除学生信息功能：**

（1）先调用查找模块，查找到想要删除的学生信息。

（2）在查找到所需要删除的学生之后，将要删除的学生信息转发到Controller

（3）由Controller转发到DeleteStudent，调用相关数据库操作函数进行操作。



**4.5更改学生信息功能：**

（1）先查找需要修改的学生信息。

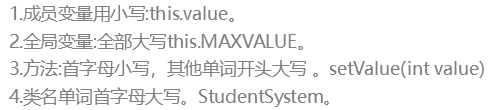
（2）在查找到之后点击更改按键，将需要修改的信息转发给Controller

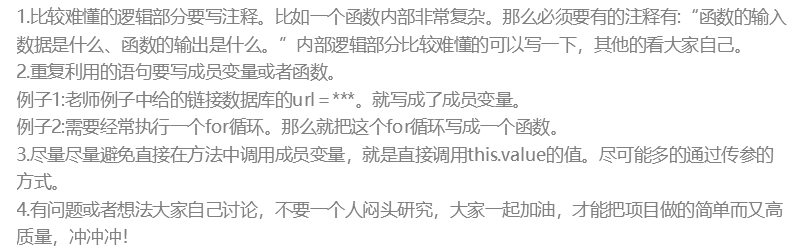
（3）由Controller将需要修改的学生信息转发给ChangeStudent，由ChangeStudent进行对数据库的修改。



**5.心得体会**

1. 这次的项目是一个团队项目。作为团队项目。最难的就是分工，要如何将小组成员的力气拧在一起，就是最难的地方。首先，分配任务要考虑到个人的能力，能力比较强的同学可以分配多一点人物，少一点的同学可以少分配一点任务。不然会造成能力强的同学事情做完了没有事做。能力比较弱的同学不能很好的完成任务。
2. 第二点，要有一个比较好的编码规范：





上面是我写给组员的编码规范，但还是有很大缺陷。

缺陷在于：没有规定如何去进行对接，于是在组员们做好自己的部分的时候，由于没有规定，所以写的就比较随意。这就导致了在最后我试图将每个人完成的部分拼装起来的时候发现这几乎是一个不可能完成的事。在之后的学习工作中我会更加注重这件事。