# 学生管理系统SSM开发文档

2018104132 陈耿彬

1. 技术准备：
2. Java1.8
3. 前端：HTML，CSS，Javascript，jQuery，主要用到的前端框架：Easy UI
4. JavaWeb：jsp，servlet，
5. 服务器：tomcat7.0
6. 开发框架：Spring，Spring MVC，Mybatis
7. 数据库：Mysql5.7，navicat
8. 开发工具：idea

二、需求分析：首先确定要做哪些功能

1. 使用MySQL5.7来保存数据
2. 两种用户：管理员，学生

学生功能分析：可以查询年级信息，班级信息，学生自己个人信息

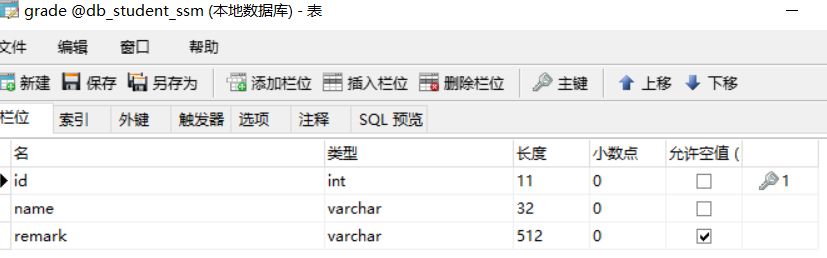
1. 学生列表（查，改）：
   1. 数据表格只能显示自己的个人信息
   2. 只能对自己的信息进行修改
2. 班级列表（查）：
   1. 数据表格能显示所有班级的信息，
   2. 并且可以通过班级名字模糊查询，所属年级进行搜索查询
3. 年级列表（查）：
   1. 数据表格能显示所有年级的信息，
   2. 并且可以通过年级名进行模糊查询

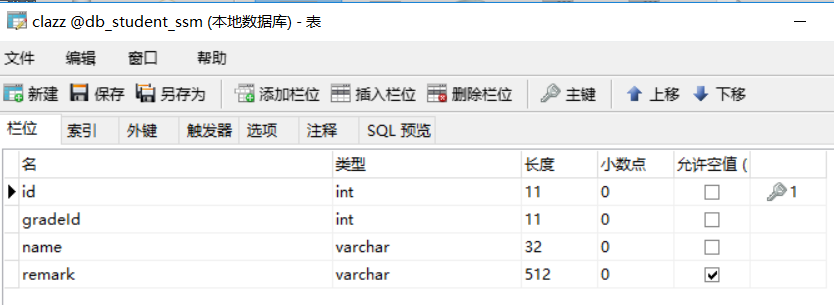
管理员功能分析：

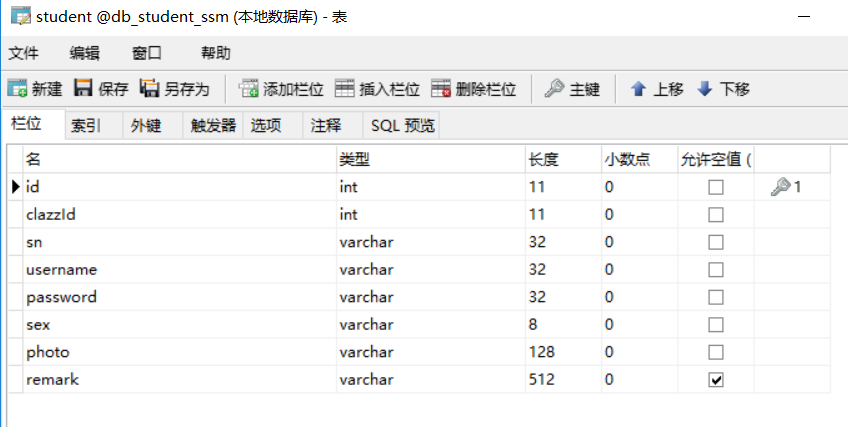
1. 用户（管理员）列表：
   1. 可以增删改查管理员用户，
   2. 可以通过用户名进行模糊查询用户
2. 年级列表：
   1. 可以进行增删改年级
   2. 可以通过年级名进行模糊查询
3. 班级列表：
   1. 可以进行增删改查班级
   2. 可以通过班级名模糊查询，所属年级名查询
4. 学生列表：
   1. 可以对学生进行增删改查
   2. 可以通过学生名进行模糊查询
   3. 可以通过所属班级进行查询所有该班级所有学生
5. 表的结构设计

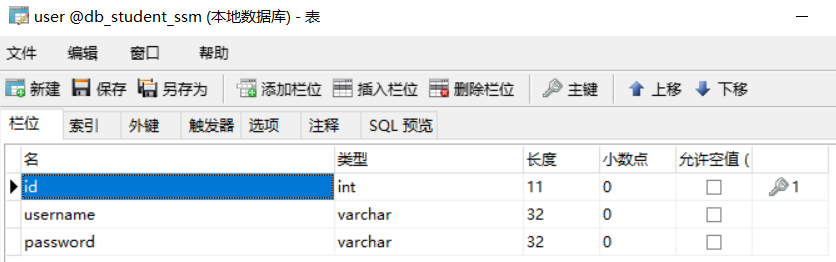
根据需求，需要管理员表格user，学生表格student，班级表格clazz，年级表格grade

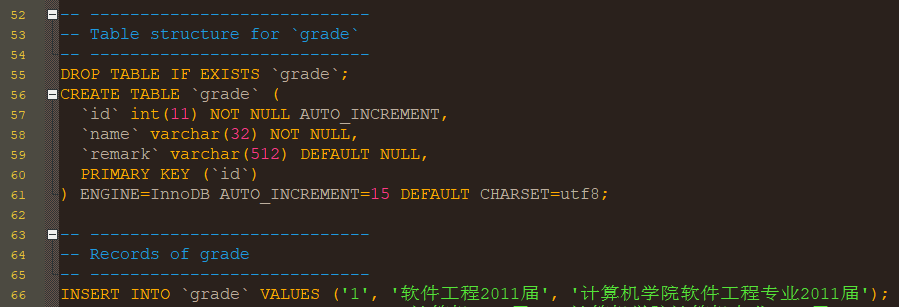
将数据库编码格式设置为UTF-8，便于存取中文数据

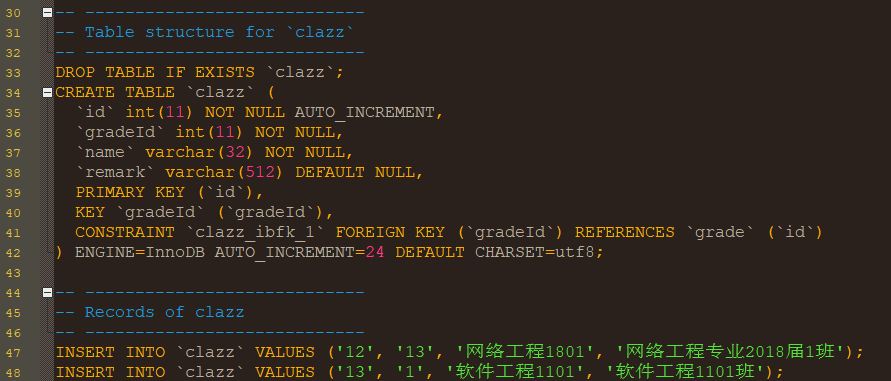


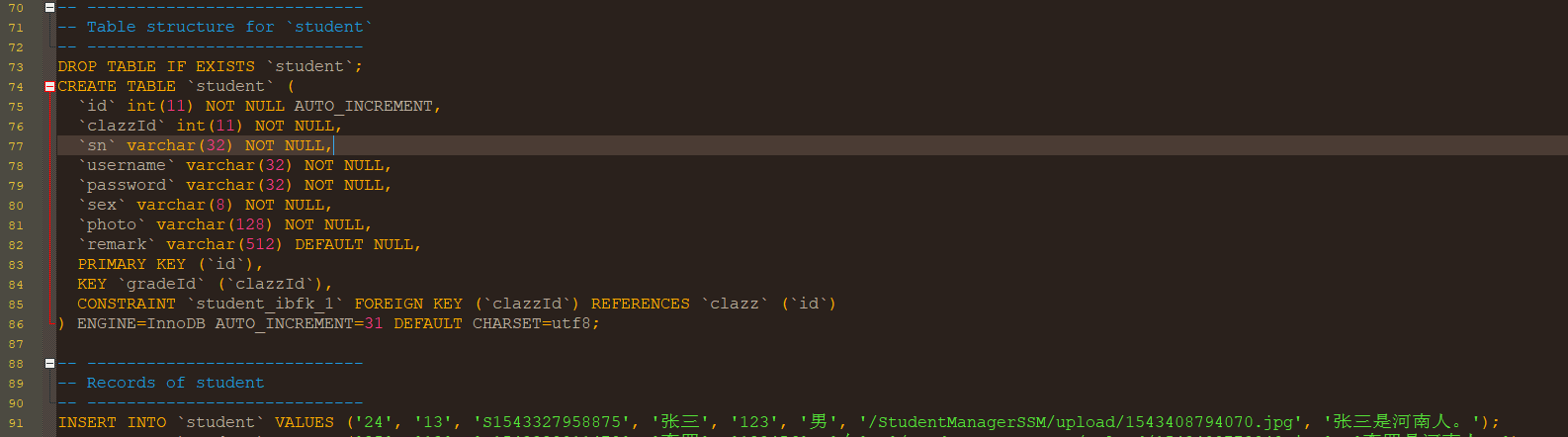


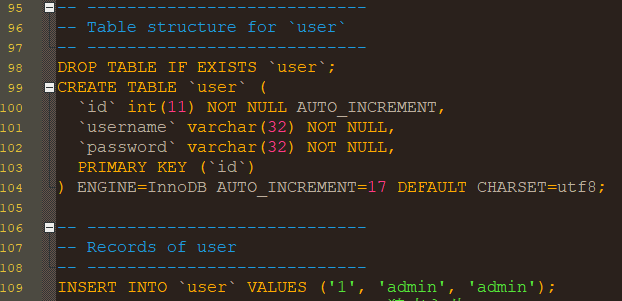










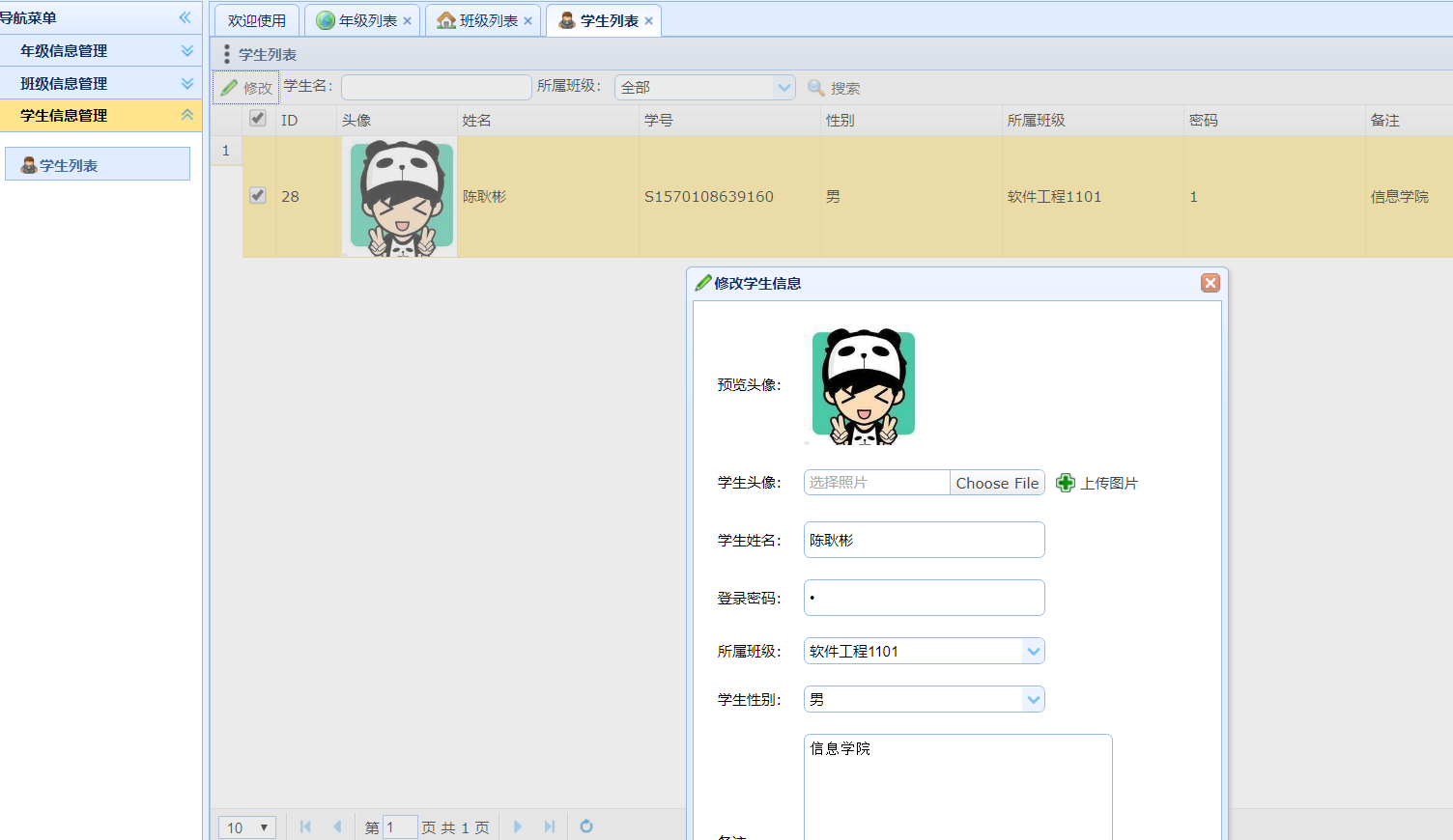


1. 原型设计

原型设计，就是设计界面

学生：

个人编辑页面

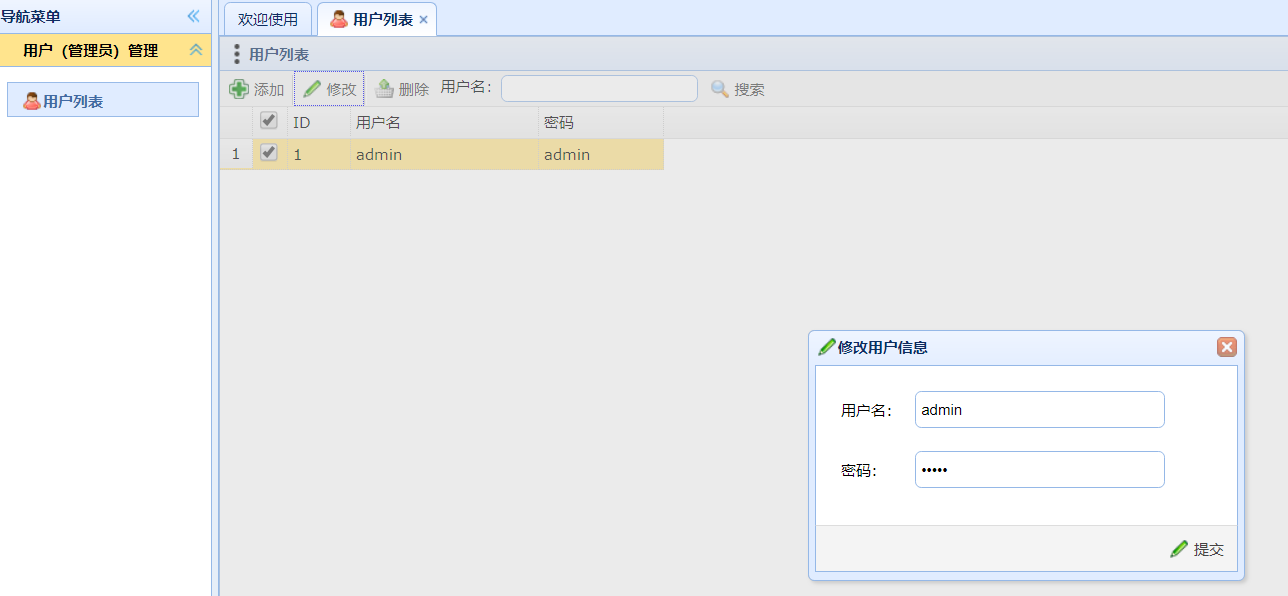


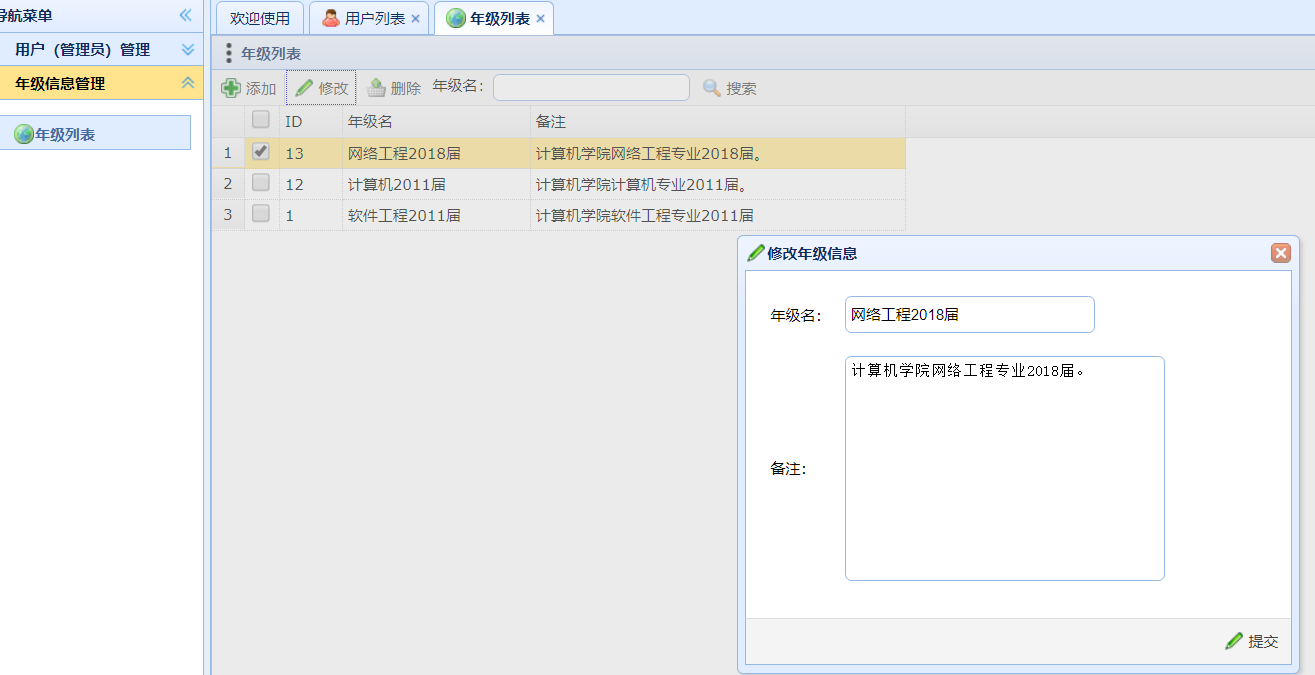
班级页面：

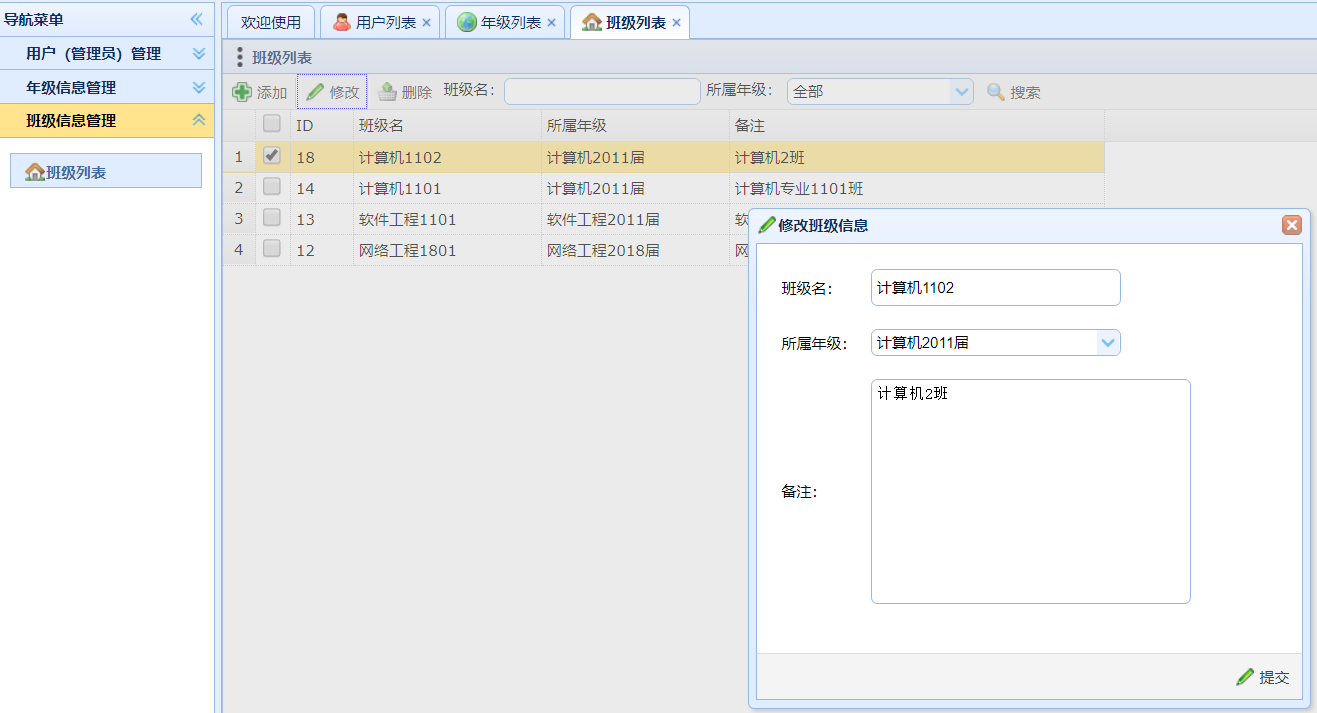
年级页面：

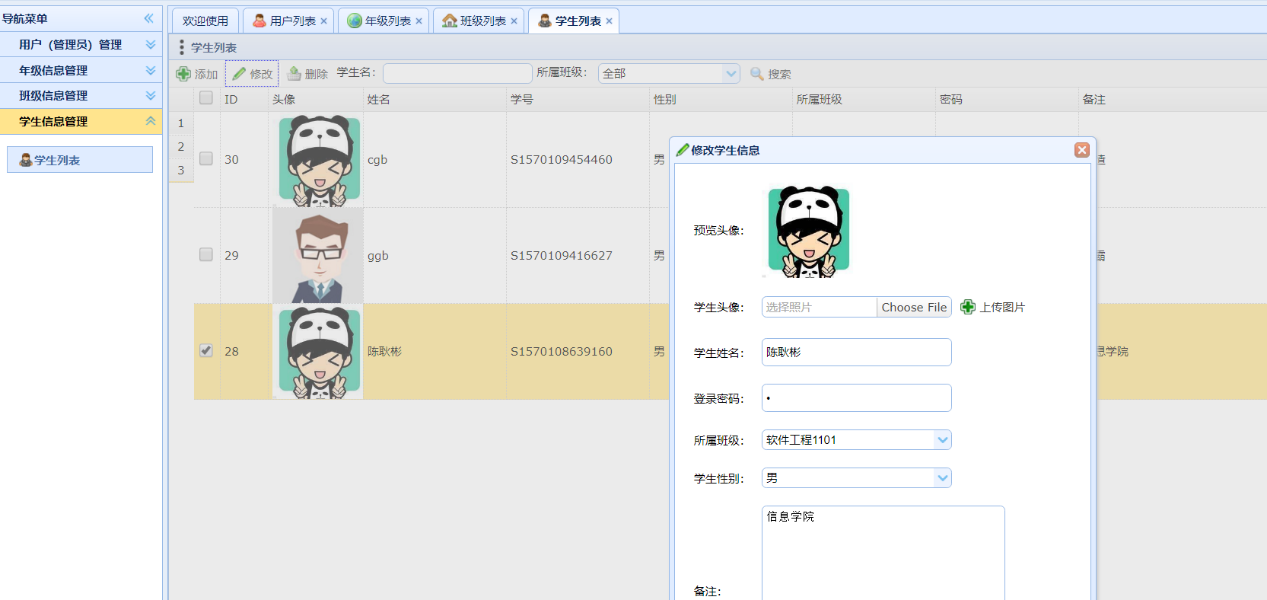
管理员：

欢迎页面：

管理员页面：

年级页面：

班级页面：

学生页面：

1. 开发环境的搭建（SSM环境的搭建）：

在真正开始编写代码之前，我们首先需要搭建好我们的开发环境。

**第一步：**导入相关的jar包

**第二步：**搭建项目结构：

Control层：进行页面的逻辑控制，转向展示的jsp页面，每一个control文件也会对应至少一个jsp文件

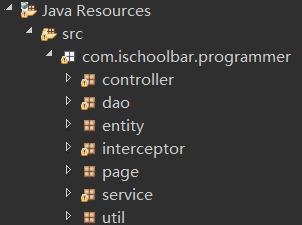
service层：service对每一个操作进一步的封装

dao层：每个数据表对应一个xxxDao.java，对数据库文件的基本操作

entities：每个数据表对应一个实体xxx.java

utils：封装常用的方法

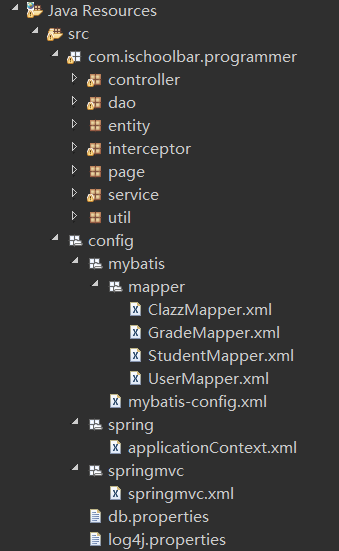
过滤器：对未登陆的用户进行拦截，过滤，不让操作数据库文件



**第三步**：配置文件

配置

1. web.xml
2. SSM框架的配置：spring.xml,springmvc.xml,mybatis.xml
3. 数据库和日志的配置：db.properties,log4j.xml



1. 实体类设计

实体类仅仅是对数据库表的一一映射（表中字段名字应该和实体类中的名称一一对应）

1. DAO类的设计

1.接口文件：DAO，即 Date Access Object，数据库访问对象，就是对数据库相关操作的封装，让其他地方看不到 JDBC 的代码

2.XML实现类文件：然后在【resources/mapper】下创建好对应的映射文件

【StudengDao.xml】：

七、业务类设计：

1.问题： 为什么不直接使用 Dao 类而是还要在上面封装一层 Service 层呢？

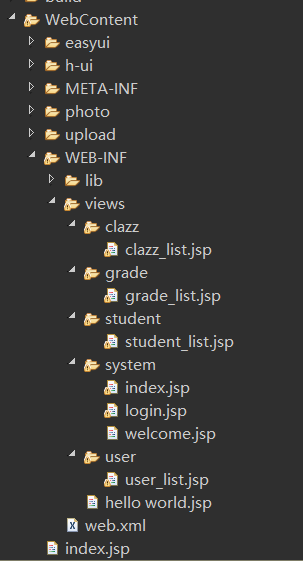
回答：基于责任分离的原则，Dao 层就应该专注于对数据库的操作，而在 Service 层我们可以增加一些非 CRUD 的方法去更好的完成本身抽离出来的 service 服务（业务处理）。2.接口文件，实现类

1. 功能开发

1.Control类的开发，页面的逻辑控制

2.jsp文件：页面的展示

注意： 所有的学号都用 student\_id 表示，为了契合在数据库中的字段名（包括下面的 JSP 文件）



1. 项目总结：
2. 由于之前的项目代码都有，所以在重构的时候，基本上没有花什么时间就完成了项目的搭建，能够体会到代码分离的重要性，这在很大程度上保证了我们的代码复用。
3. 相较于之前用 Servlet + JSP 来完成，很明显的感觉是DAO层的编写方便了很多，仅仅需要编写一个 xml 映射文件和一个 Dao 层接口就可以完成功能，而不用自己再去重复的去在每一个 CRUD 方法中去处理结果集，重复而且繁琐。
4. 注解真的很方便，这不仅仅提升了我们自身开发的效率，写起来也很带劲儿。
5. 开发效率快，而且低耦合，我们基于 xml 配置了大部分的工作，在基于 SSM 框架开发时，我们可以把专注点集中在逻辑处理上。
6. 在 SSM 框架中能方便的对项目进行修改，这不仅仅得益于框架的约定，使得代码分离并且可复用
7. 我们能够通过一个 Controller 来完成五个 Servlet 的功能，并且通过注解来完成配置。
8. 项目改进：

增加其他功能。