《软件开发环境与工具》课程报告

**课题名称：大学生信息管理系统**

龙行超 2017141463145

（四川大学，成都 610000）

摘要：本系统主要设计、描述大学生信息管理系统的功能和构造，通过对学校现有教务处的分析，对这个系统进行了详细的描述和分析，设计并实现了一个简要的教务处系统，用来对大学生的信息进行管理。开发这个系统，主要使用了用例图、类图、时序图、状态图、活动图、组件图和部署图，从而较好的分析了系统主要功能和实现流程，并做出了一个简易的系统。

关键词：大学生；信息管理；UML图

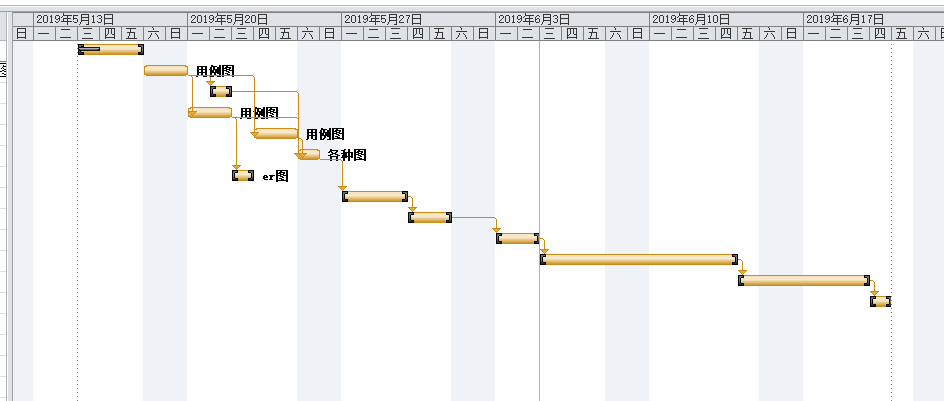
1. 引言

在现在，信息是相当重要的，谁能更好地管理信息，谁将是这个世界的王者。信息对于这个世界是不可或缺的，当然，大学也不例外。大学的学生人数，比高中、初中、小学都要多得多，所要存储的信息方面也要比它们多得多，所以大学生的信息量是相当庞大的，那么学校是如何对这些信息进行管理的呢？本系统基于对学校教务处系统的分析，严格按照软件工程开发步骤，运用各种方法，开发了一个简易的教学系统。

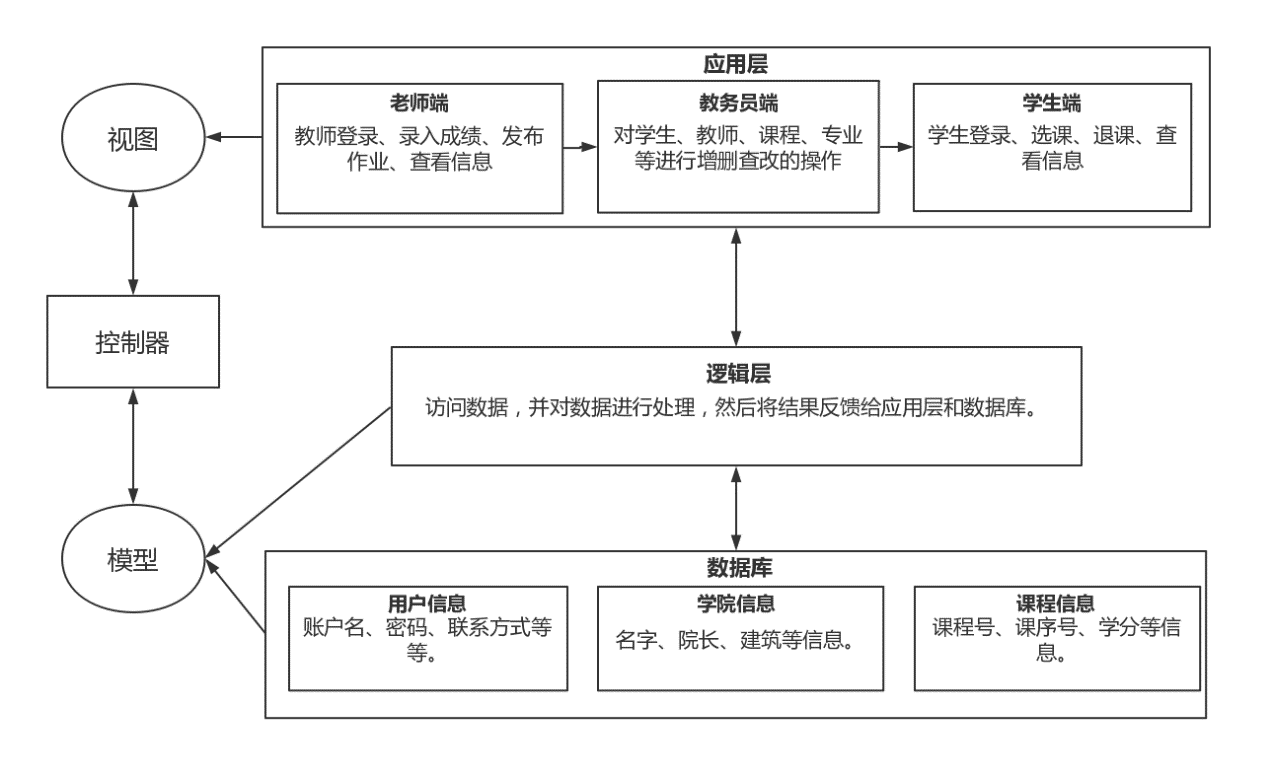
1. 项目计划

项目计划，是完成项目所必需的，它能让你的开发井然有序，按时完成。以下截图是用project项目管理软件指定的项目计划：



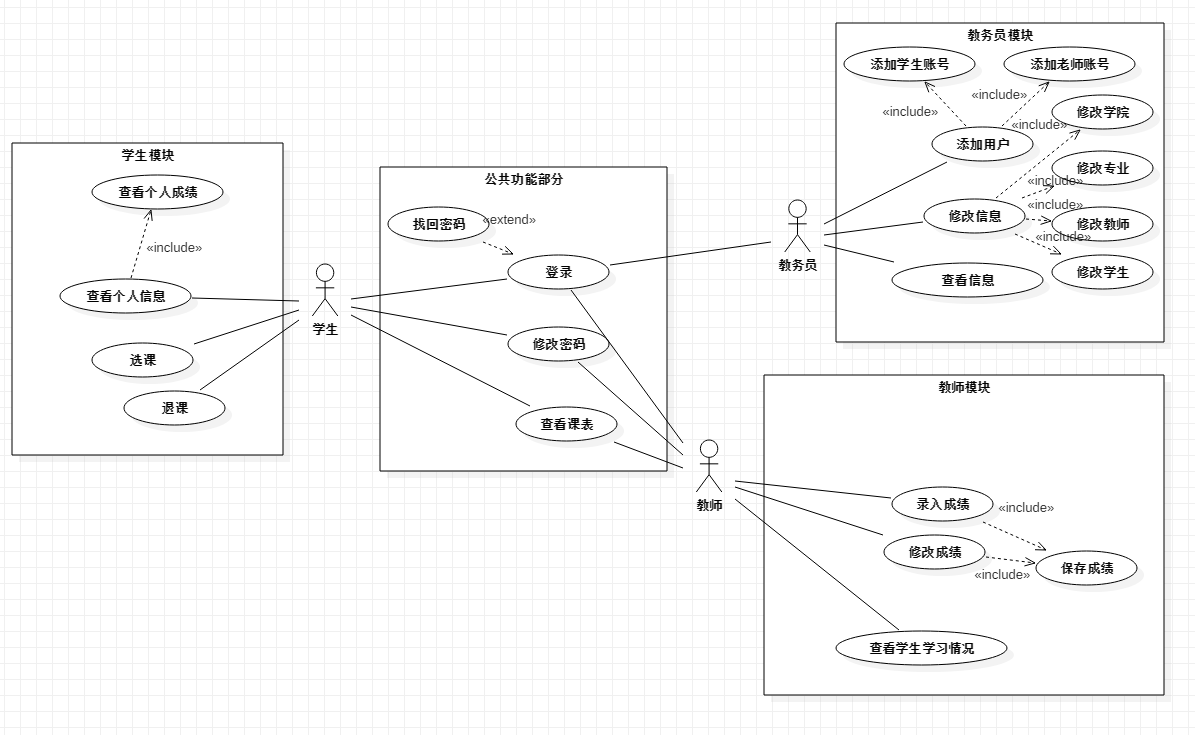


1. 系统总体设计框架图



系统采用MVC模式进行开发，前端应用程用经典的jsp进行编写，jsp是一种动态的网页技术，比较古老，就只是在html中插入Java程序片段，以便实现网页动态；中间的控制器使用servlet来实现，servlet指的是任何实现了servlet接口的类，可用于逻辑调用、处理转发；模型采用Java来编写，数据库使用的是MySQL数据库，两者通过JDBC来进行连接。

1. 需求分析——用例图



这个系统主要面向三个不同的使用者，教师端和学生端共同的功能如下：

* 登录：输入账号名和密码，若正确即可，登入系统。
* 修改密码：输入账号，会自动发送验证码一封邮件到注册时填入邮箱，填写验证码之后即可修改。
* 查看课表：登入系统之后，可查看本学期课表，和以往学期的课表。

教务员，拥有最高权限，能直接对数据库进行处理，并且是内置的账号、密码，不能通过教务员端进行修改，所以教务员端与它们的共同功能只有登录。接下来说说，各端所独有的功能。

学生端：

* 选课：登入系统，点击选课，即可看见课程信息，目前只实现了按学院查课。
* 退课：可删除已选课程。
* 查看个人信息：这里，能看见你的一切个人信息，包括：住址、学号、班级等等。

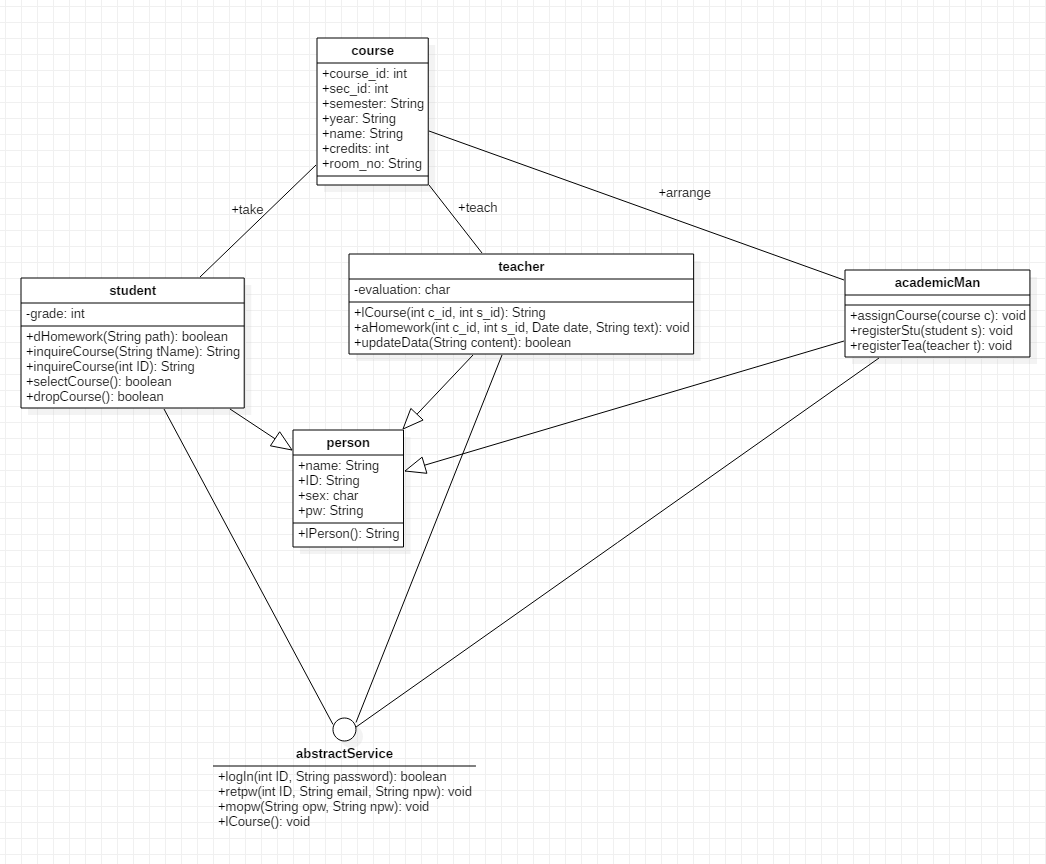
教师端：

* 录入成绩：给学生的课程输入一个成绩。
* 修改成绩：只能修改平时成绩，期末不允许修改。
* 查看信息：了解班上同学的学习情况，包括作业完成情况、半期考试分数、小测分数等等。

教务员端：

* 修改信息：主要指的删、改两个操作，直接针对学院、专业、教师、学生进行修改。
* 查看信息：指的是查这个操作，能查看一切信息。
* 添加信息：指的是增这个操作，主要能添加学院、专业、教师、学生等信息。

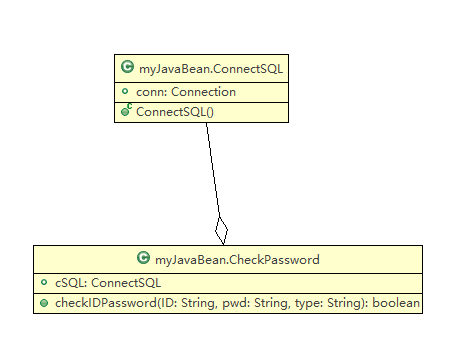
1. 类图

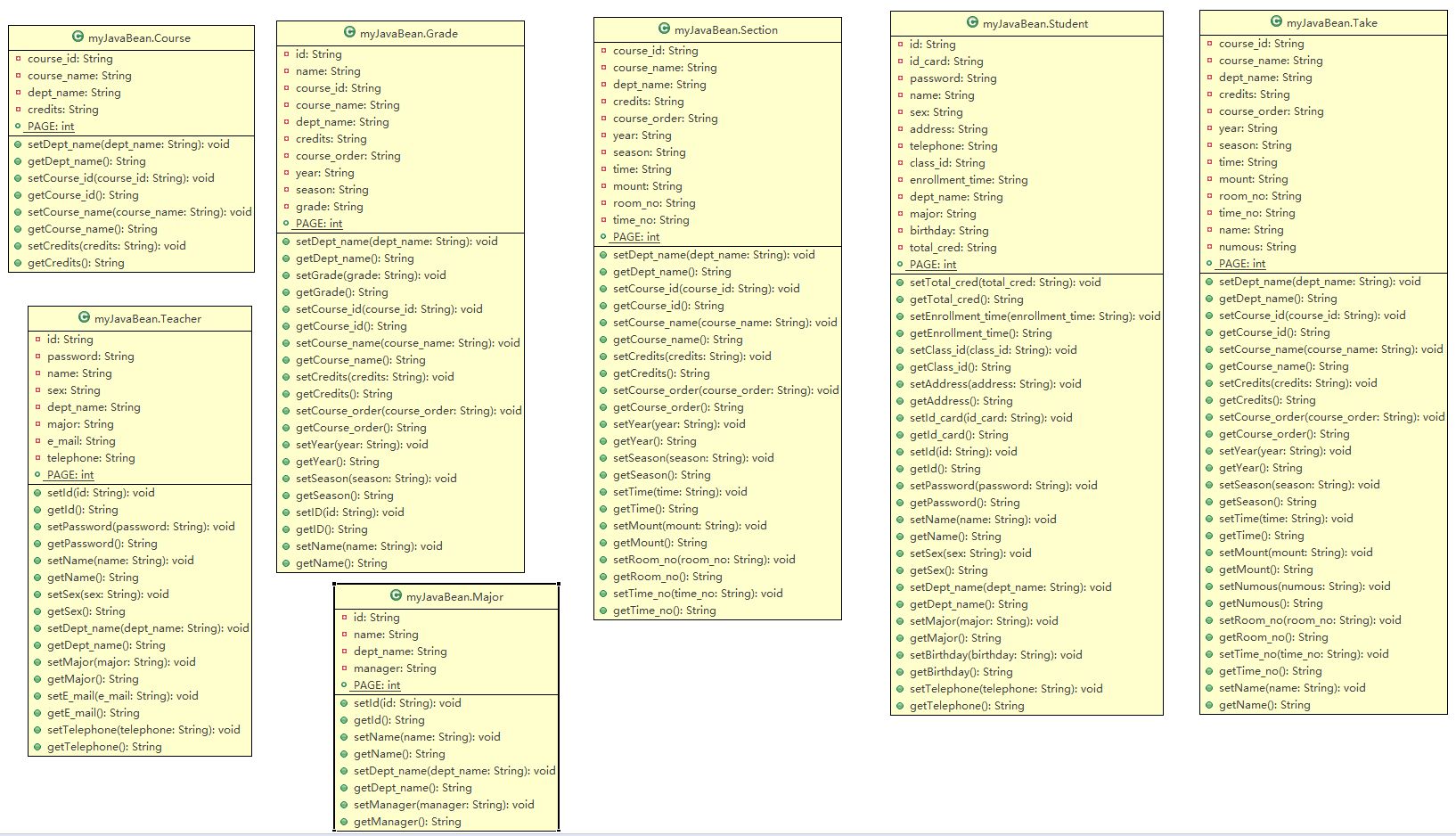


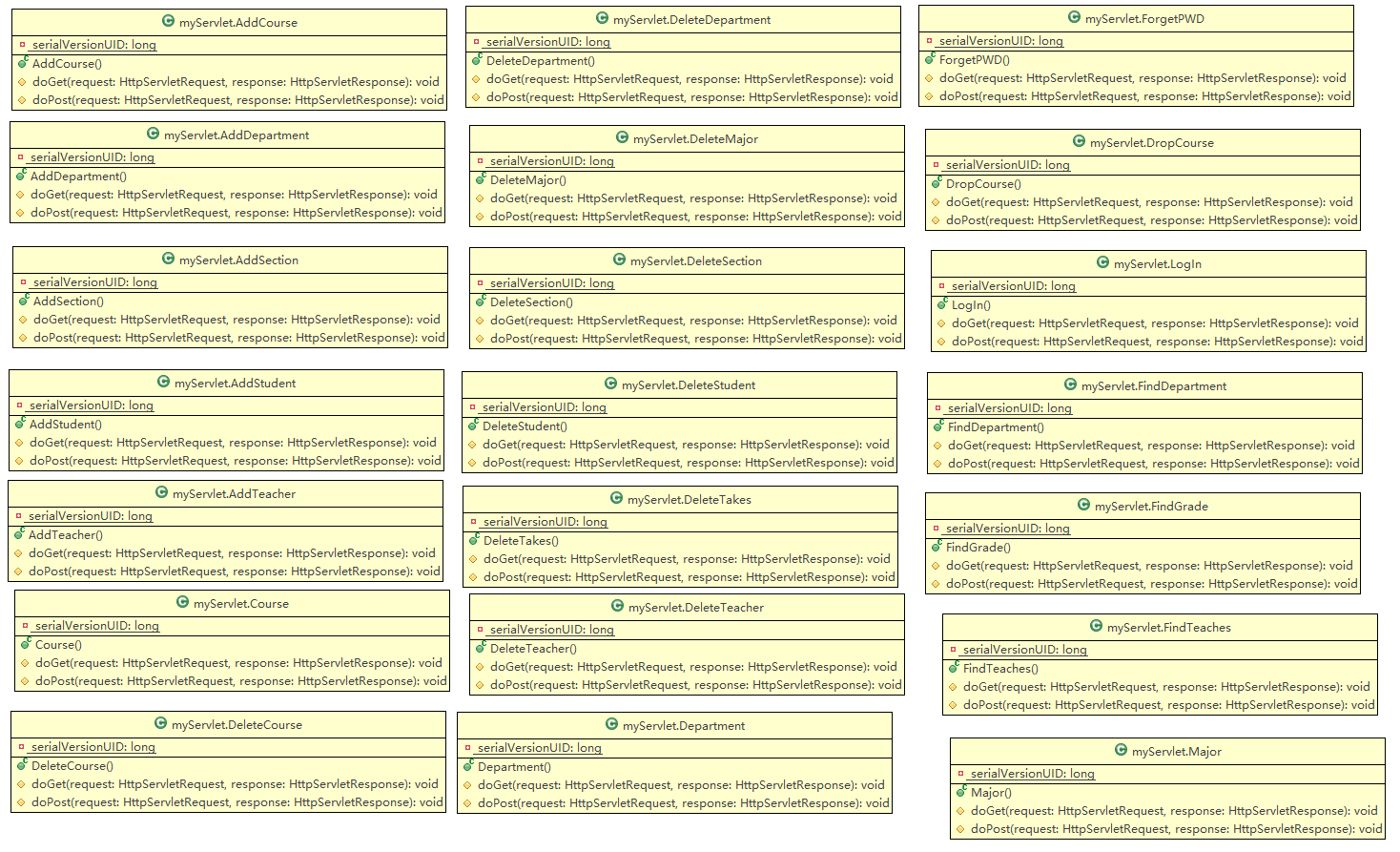
上图为主要存储实物的类图：

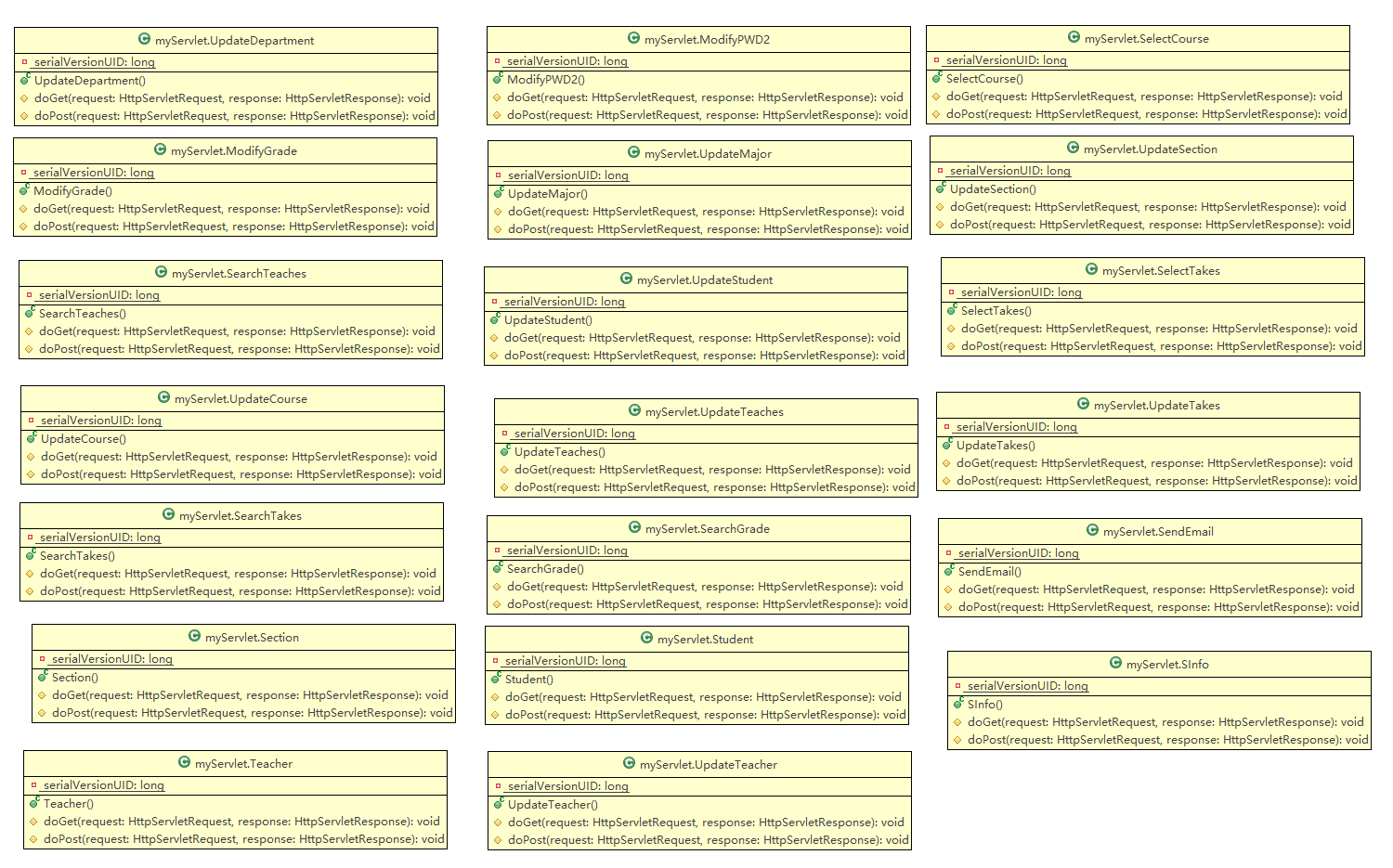
三个主要角色都继承于人，三者与课程之间均是多对多的关系，一个老师讲授多门课程，一个课程被多个老师讲授，其它就省去。Student类主要用来存储学生的信息，teacher类用来存储老师信息，academicMan类用来存储教务员信息，course类用来存储课程信息，最下面的abstractService接口用来封装了一系列的逻辑和对数据库的操作，图上只是一小部分，这部分，比较多，而且类主要是用来实现操作，都只与它是实现关系，故为了简便，用了一个抽象公共服务的类来代替。

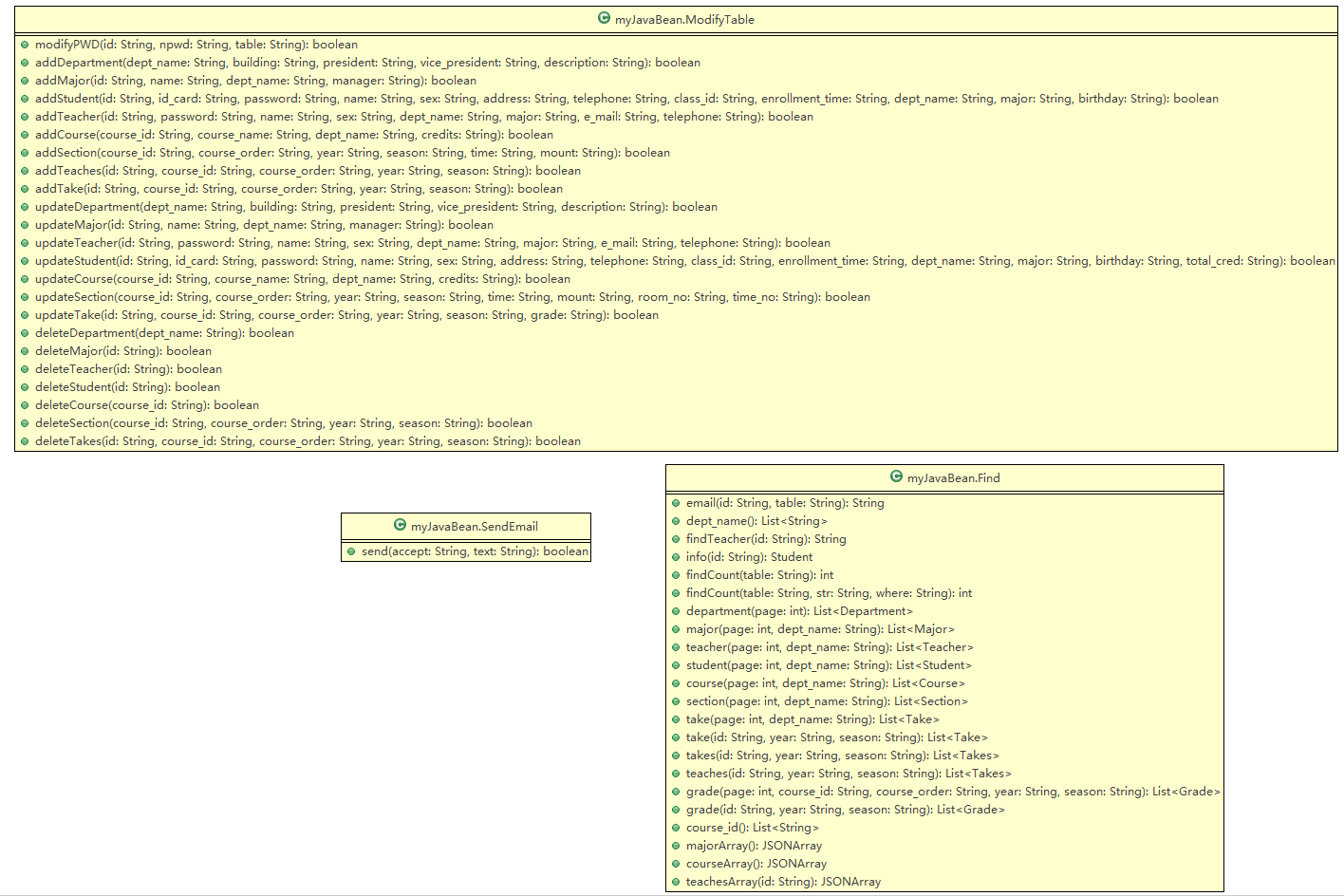
下面一组图为eclipse自动产生的类图：









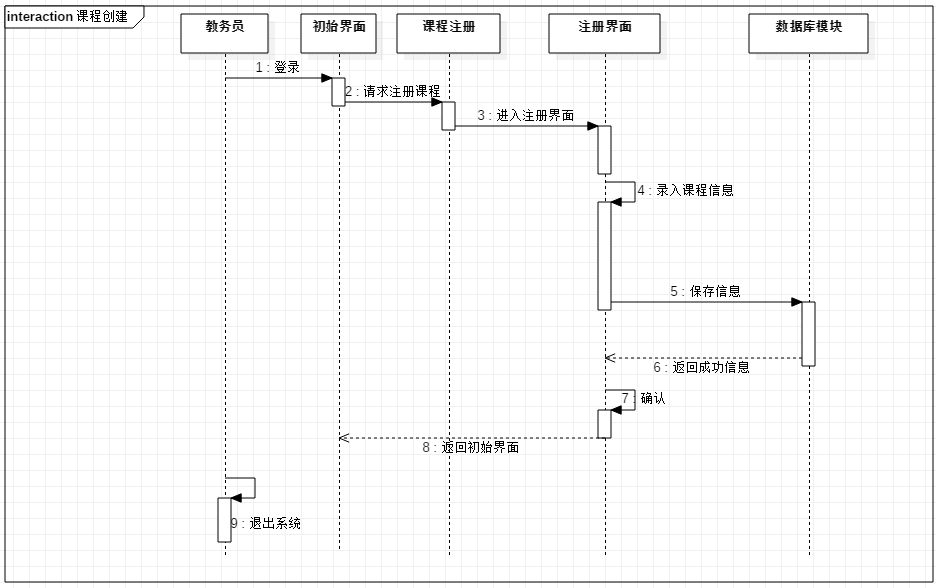


1. 时序图

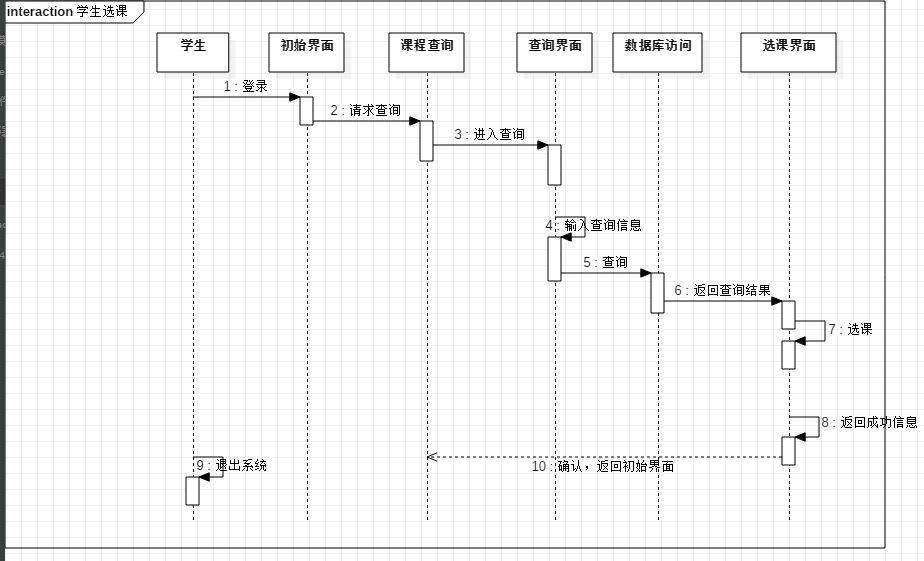
时序图描述了对象之间传递信息的时间顺序，它用来表示用例中的行为顺序。

在大学生信息管理系统中，每个用例都可以建立一个时序图，将用例执行中的各个参与的对象之间的消息传递过程表现出来。由于要求限制，这里选取四个主要的用例，构建时序图，如下：

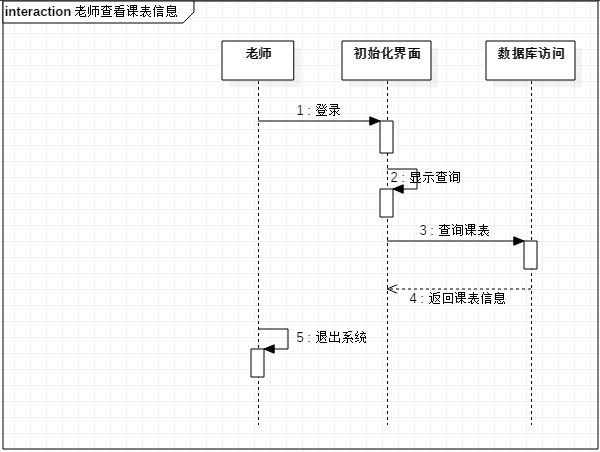
* 1. 课程创建时序图



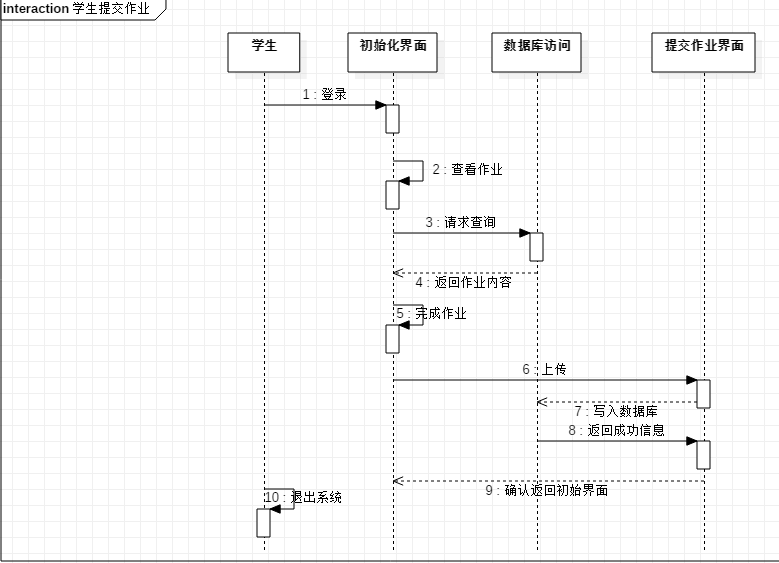
* 1. 学生选课时序图



* 1. 教师查看课表时序图



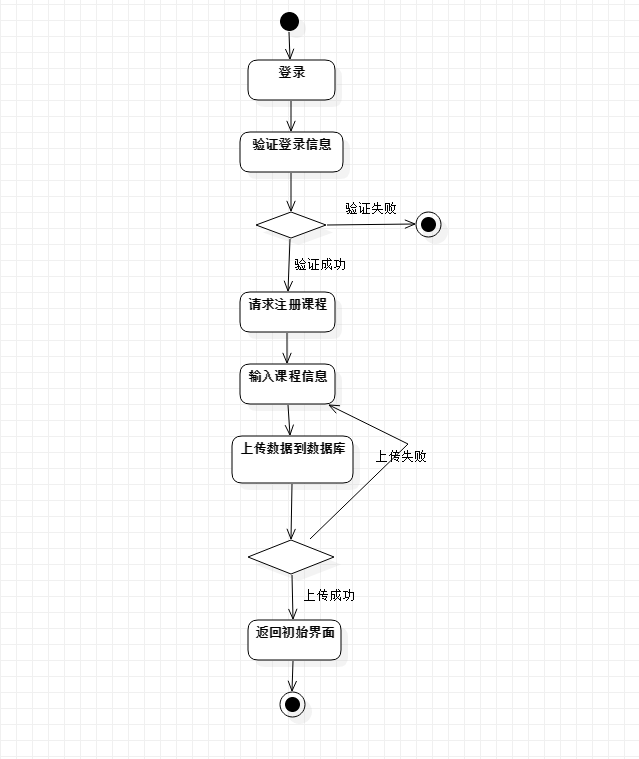
* 1. 学生提交作业时序图



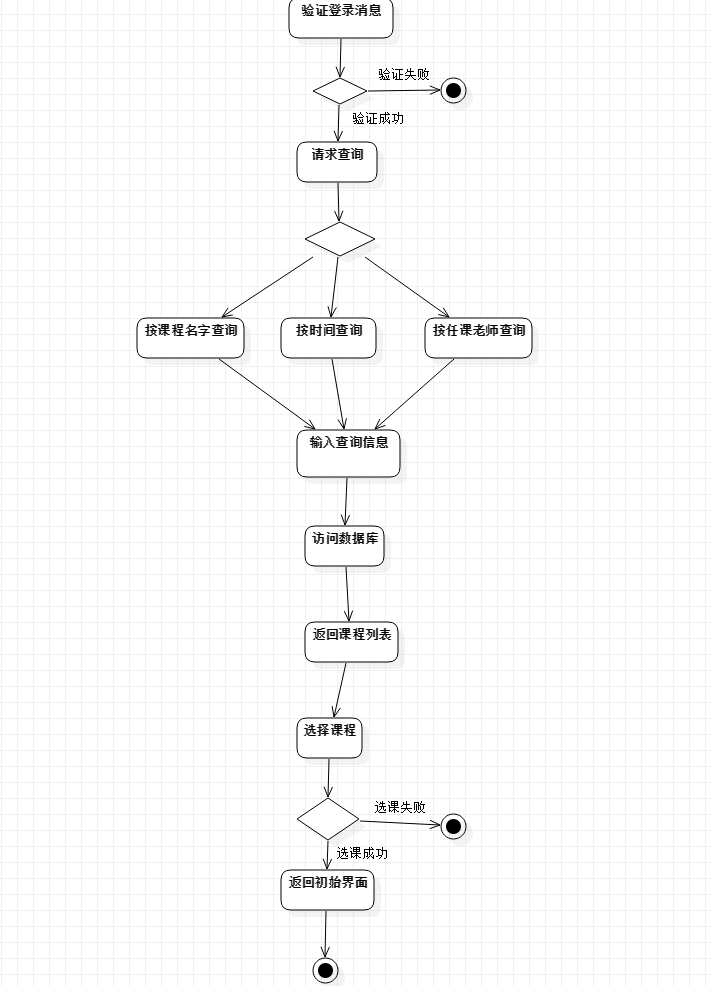
1. 活动图

活动是某个事情正在进行的状态，既可以是实现现实生活中正在进行的某一项工作，也可以是软件系统中某个类对象的一个操作。一个用例也可以做一个活动图，现对应上述时序图，也做出四个活动图。

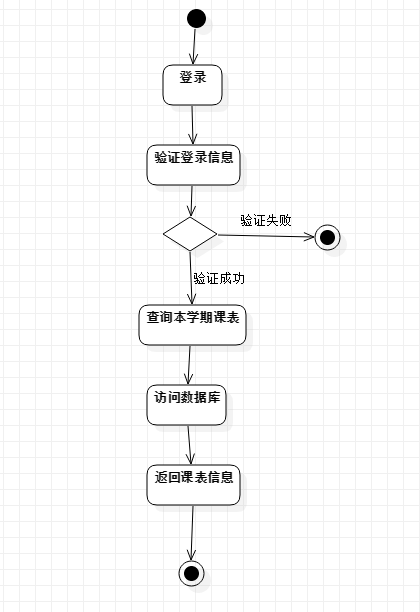
* 1. 课程创建活动图、



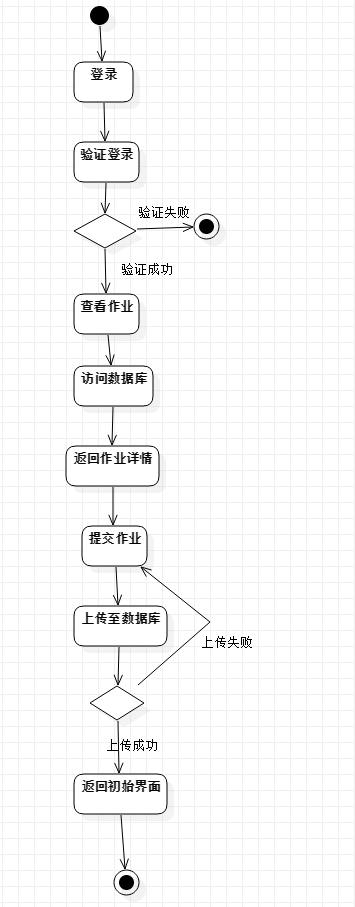
* 1. 学生选课活动图



* 1. 教师查看课表活动图

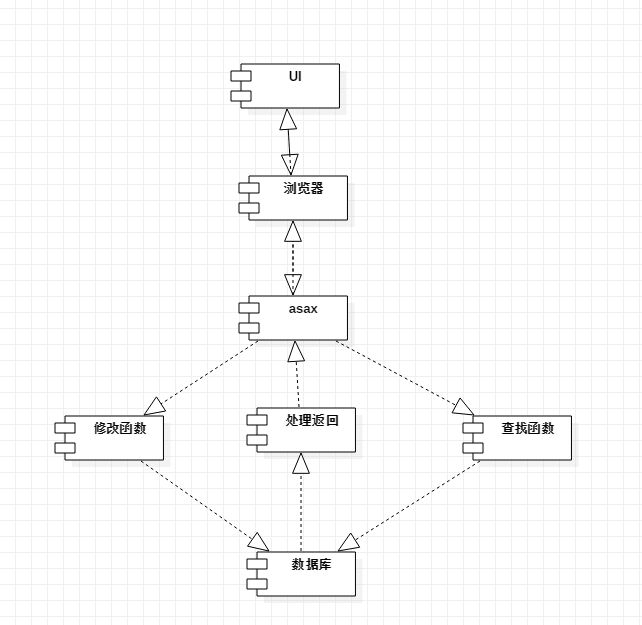


* 1. 学生提交作业活动图



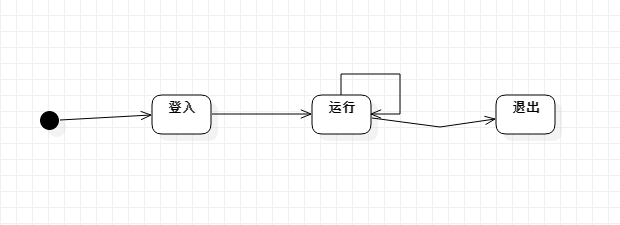
1. 组件图

组件图用来反映代码物理结构的图，组件图中可以了解软件组件之间的关系，如下：



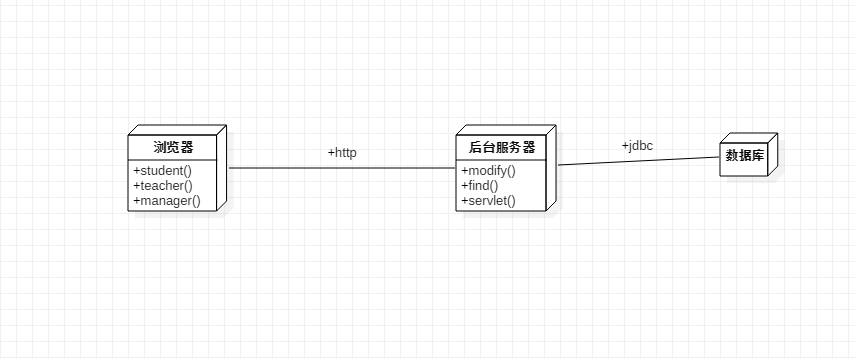
1. 状态图

状态图描述一个实体基于事件反应的动态行为，显示了该实体如何根据当前所处状态对不同的事件做出反应，本系统的状态很少，有三个：登入状态、运行状态、登出状态，状态图如下：

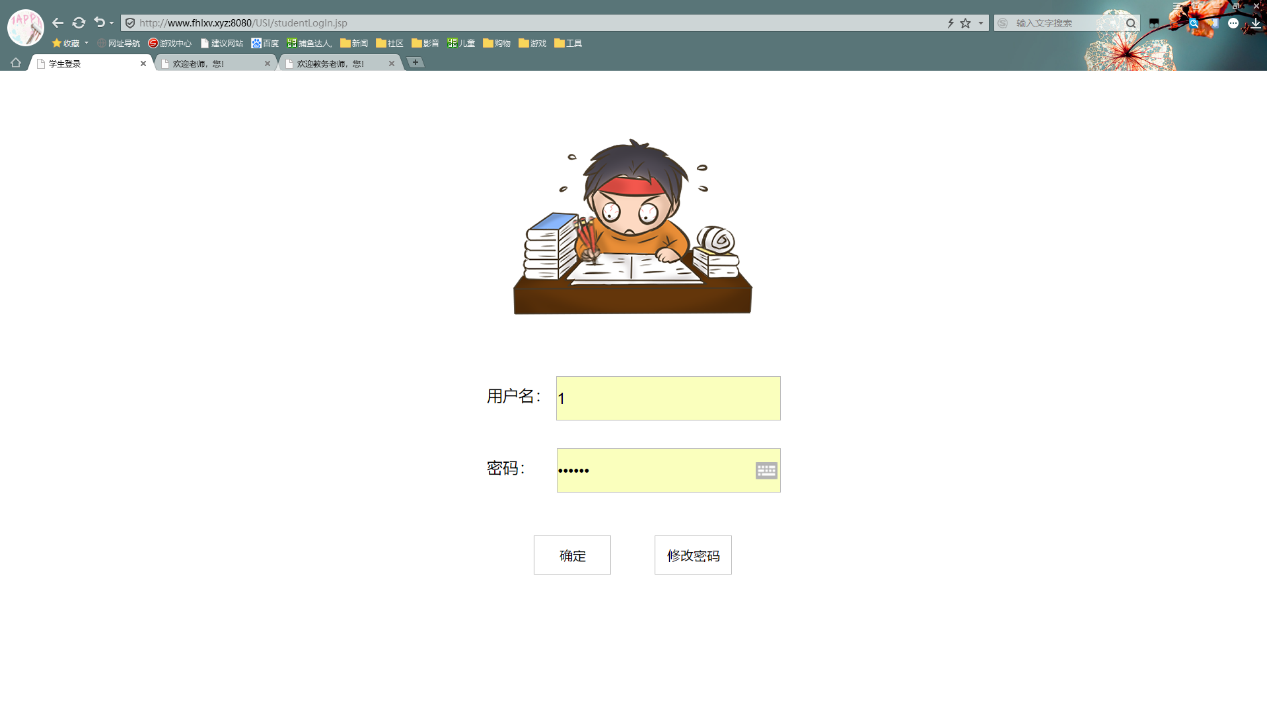


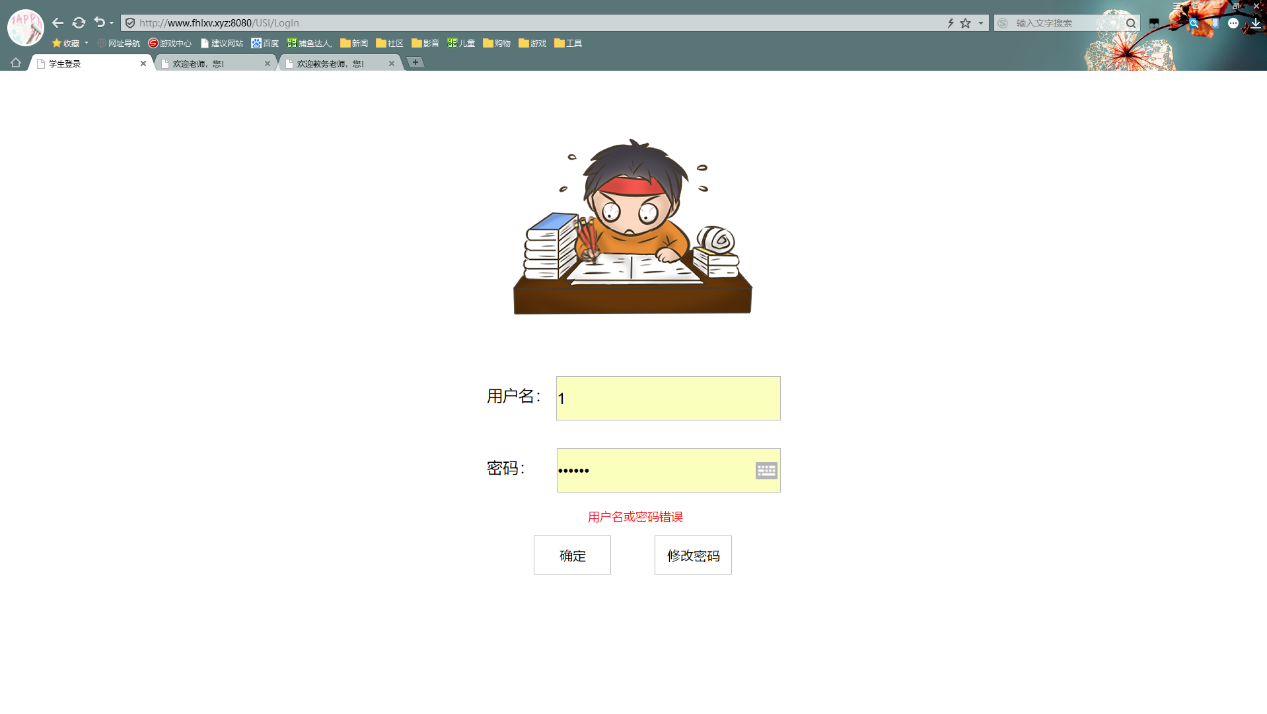
1. 部署图

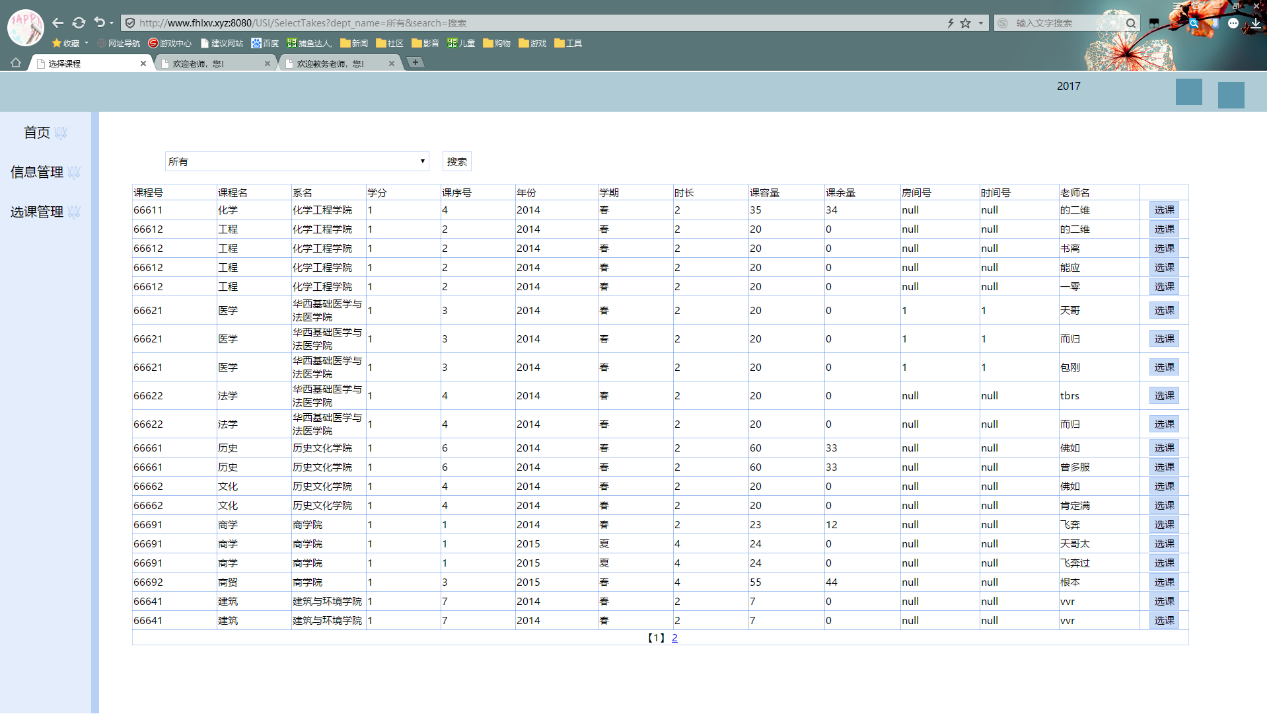
部署图描述的是系统中软件和硬件的物理结构，从这个图中能了解到软件和硬件之间的物理关系以及处理节点的组件分布状况，本系统的部署图如下：



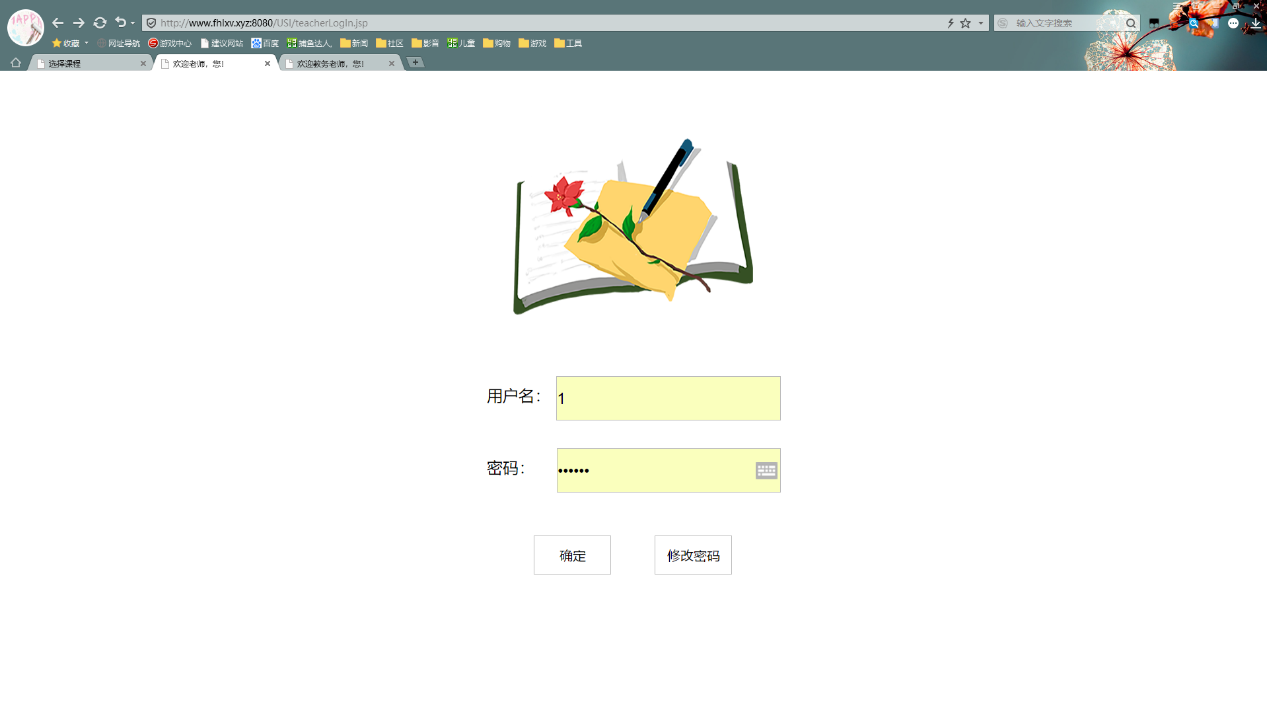
1. 项目运行截图



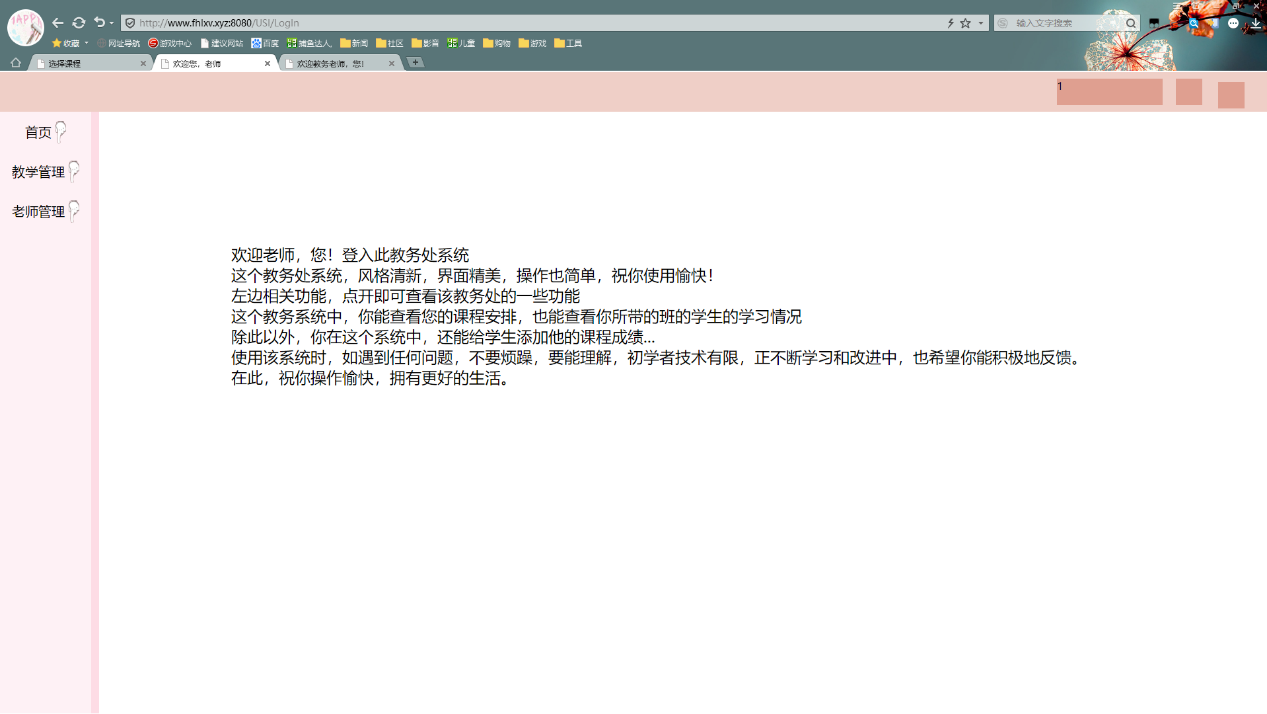




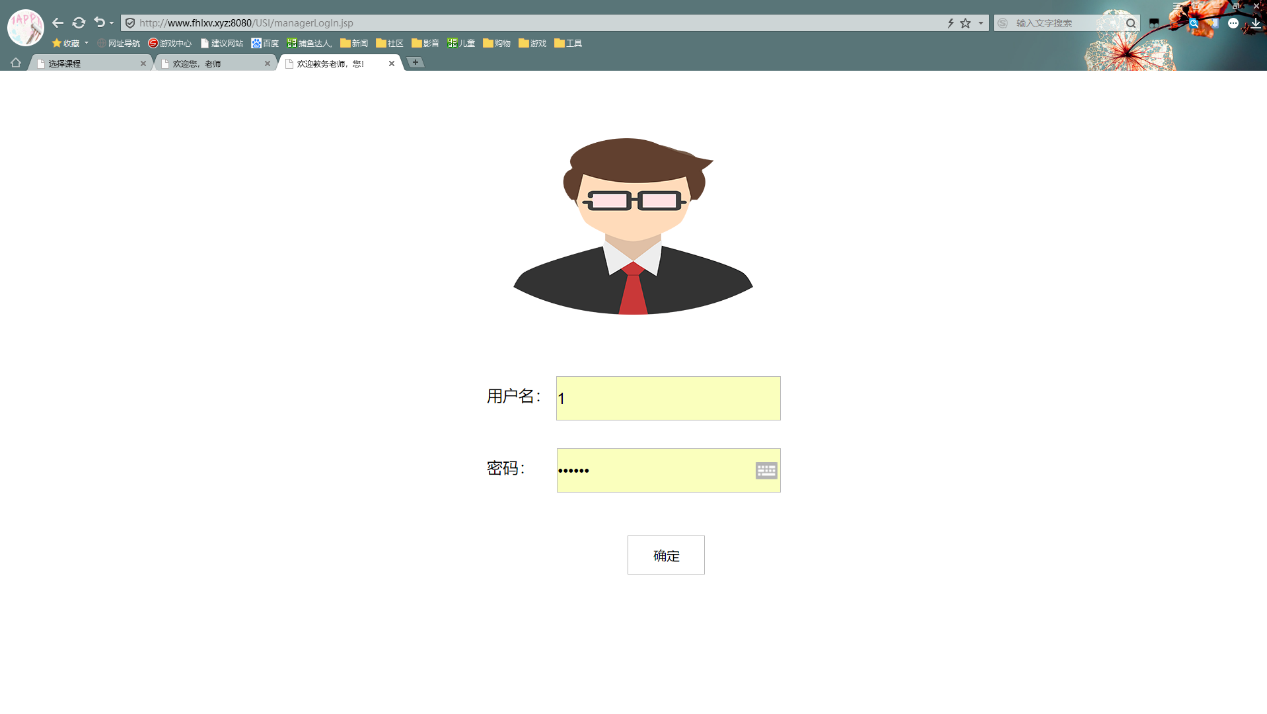
1. 教师端

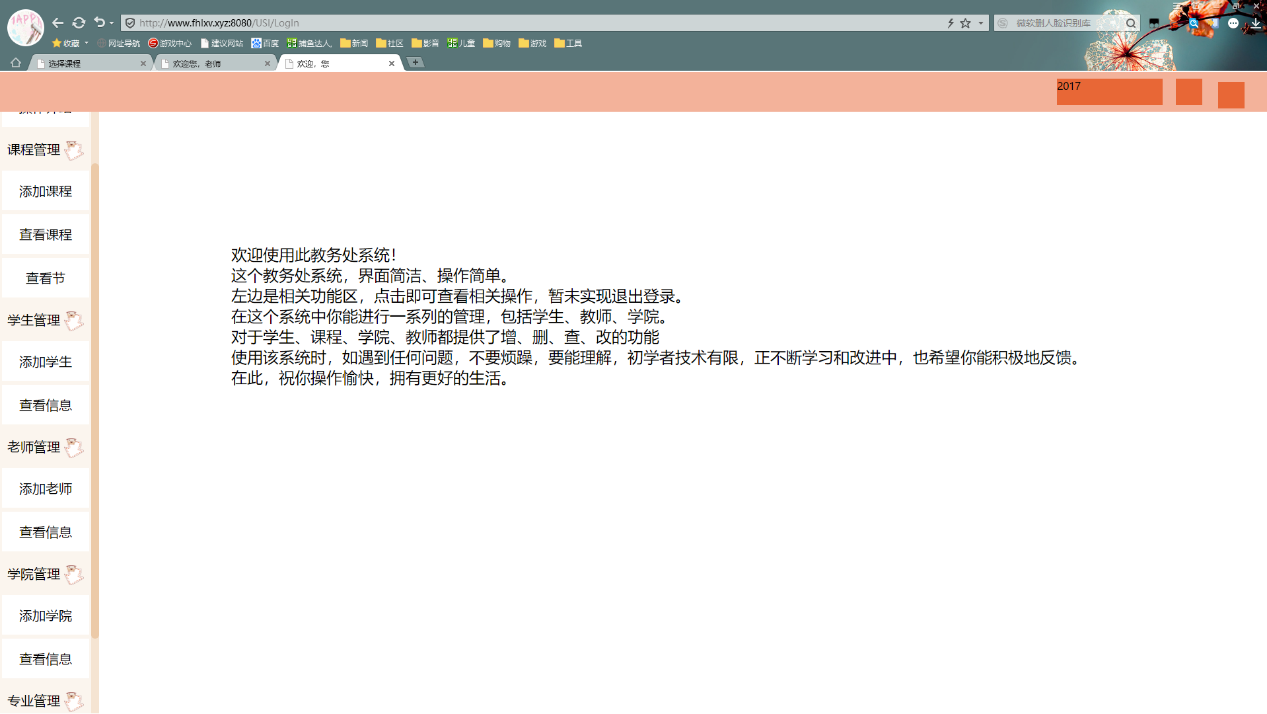




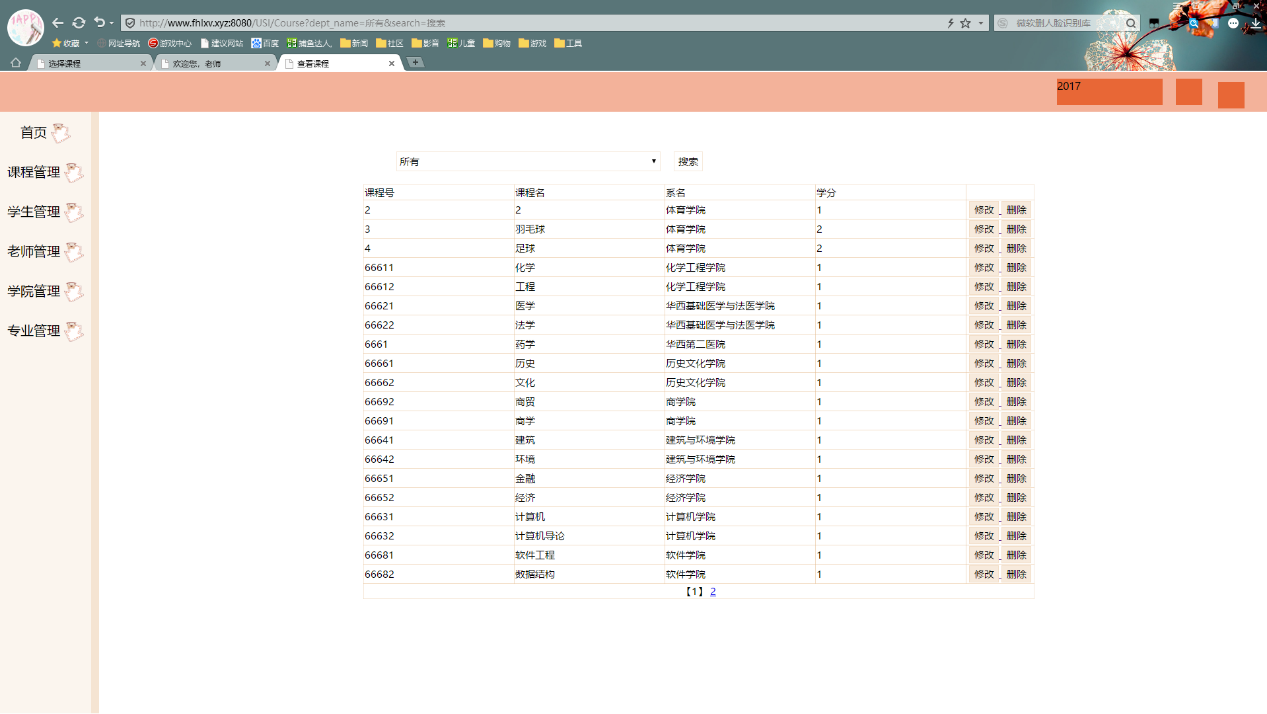


1. 教务端

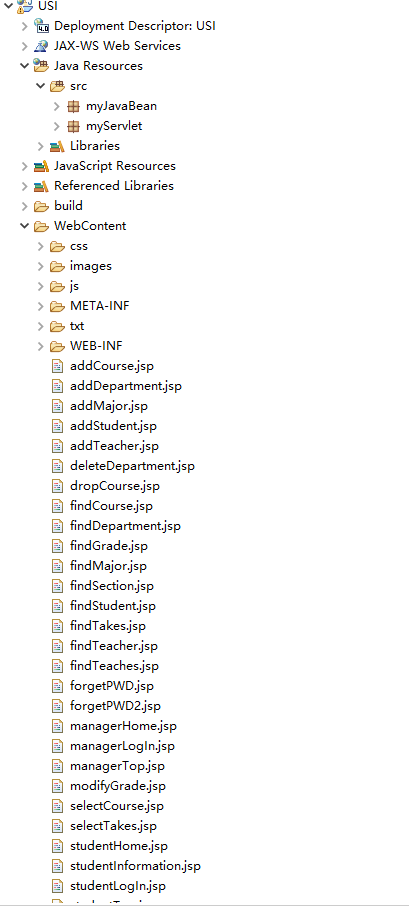


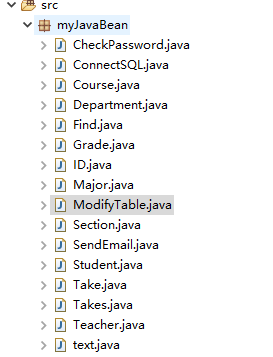


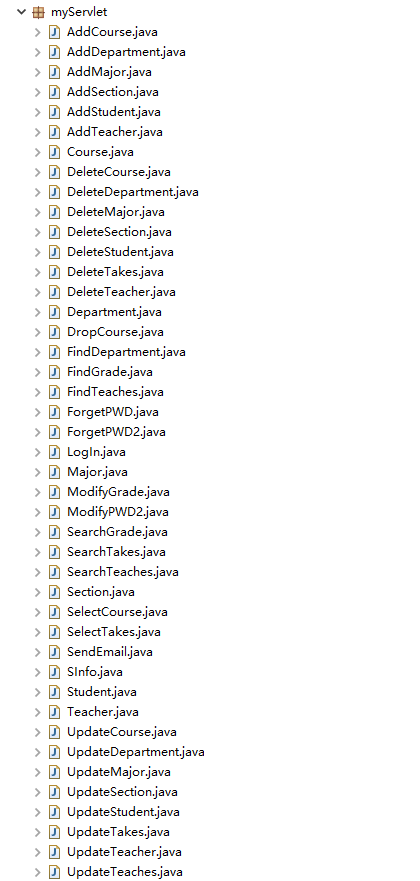


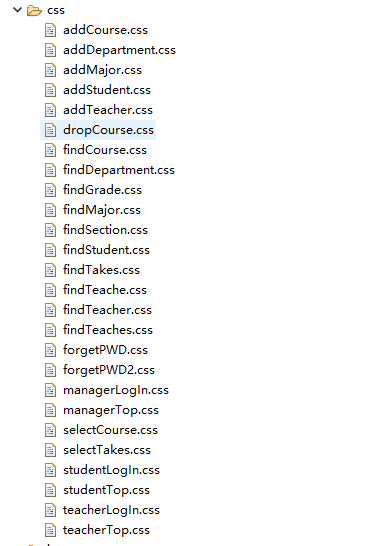


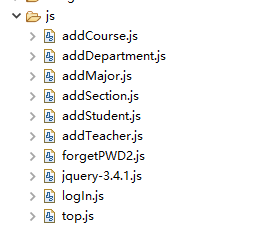
1. 关键源码截图

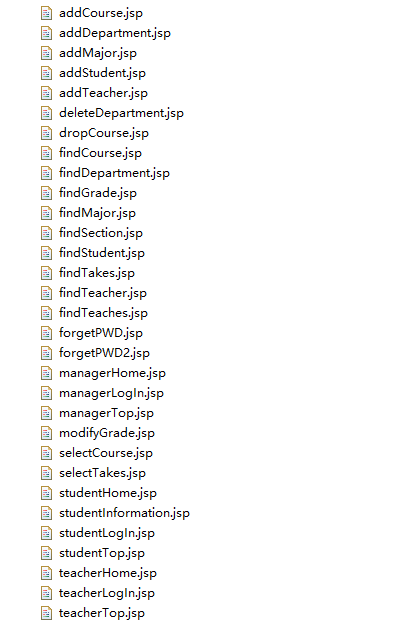


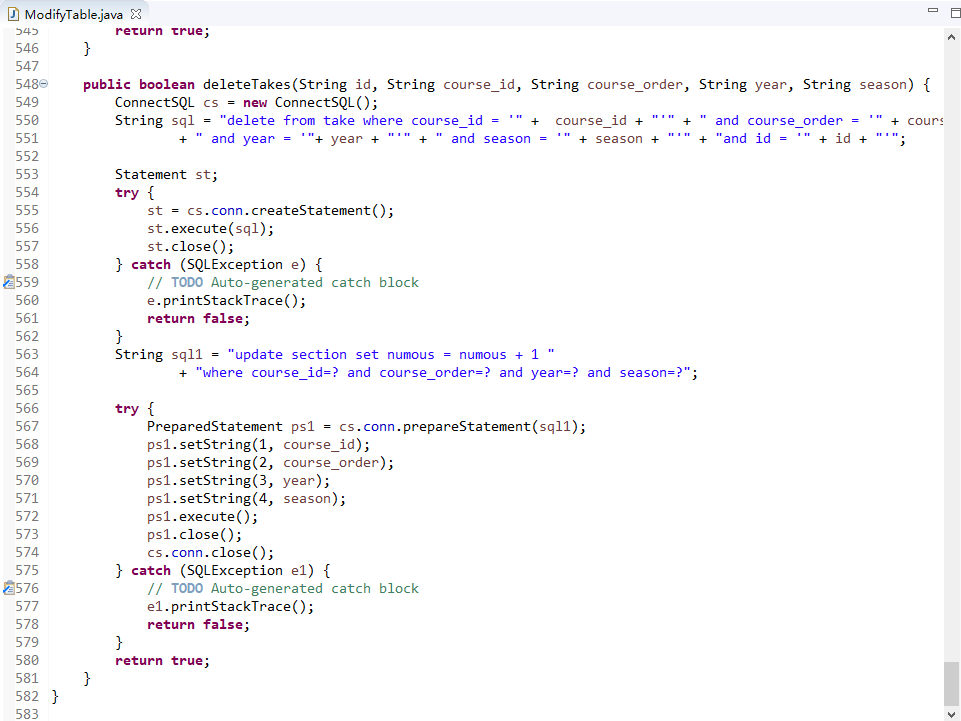


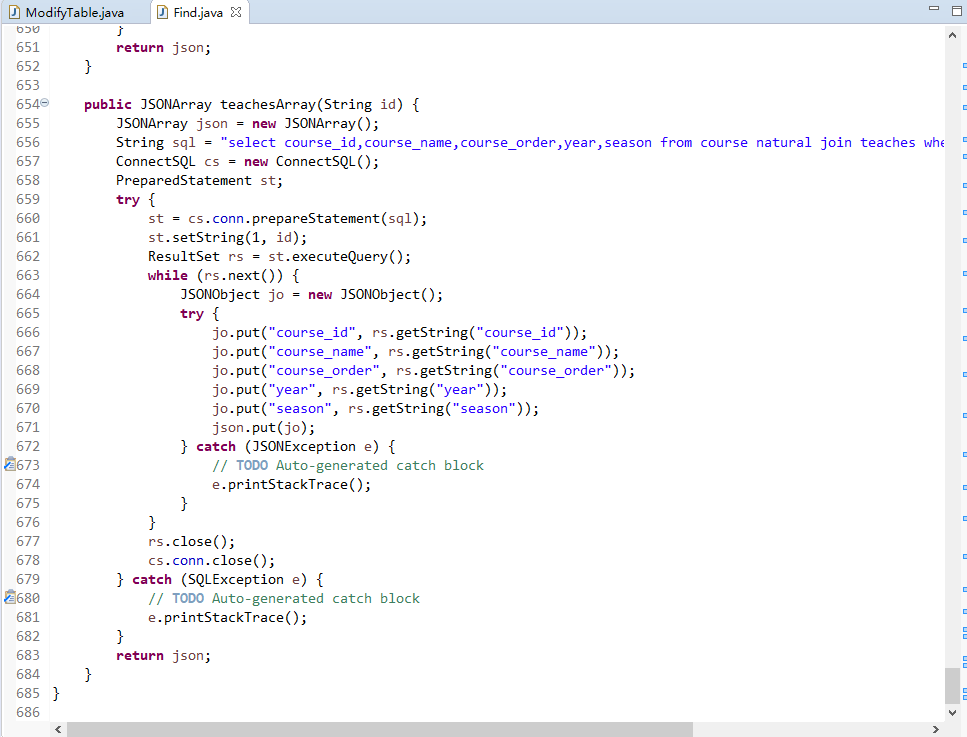


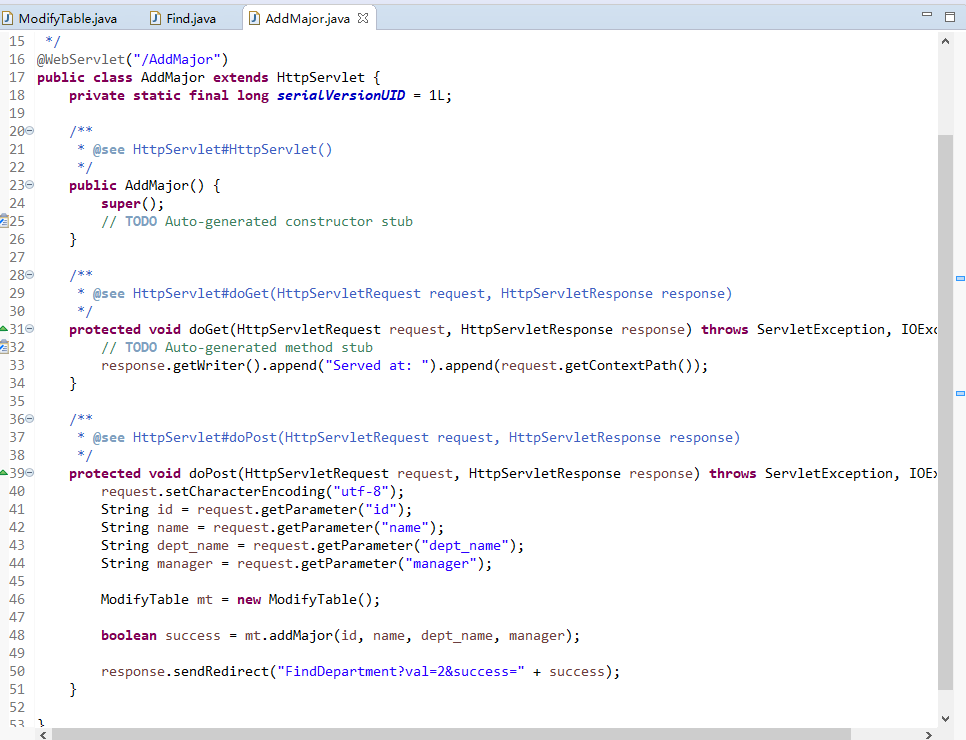


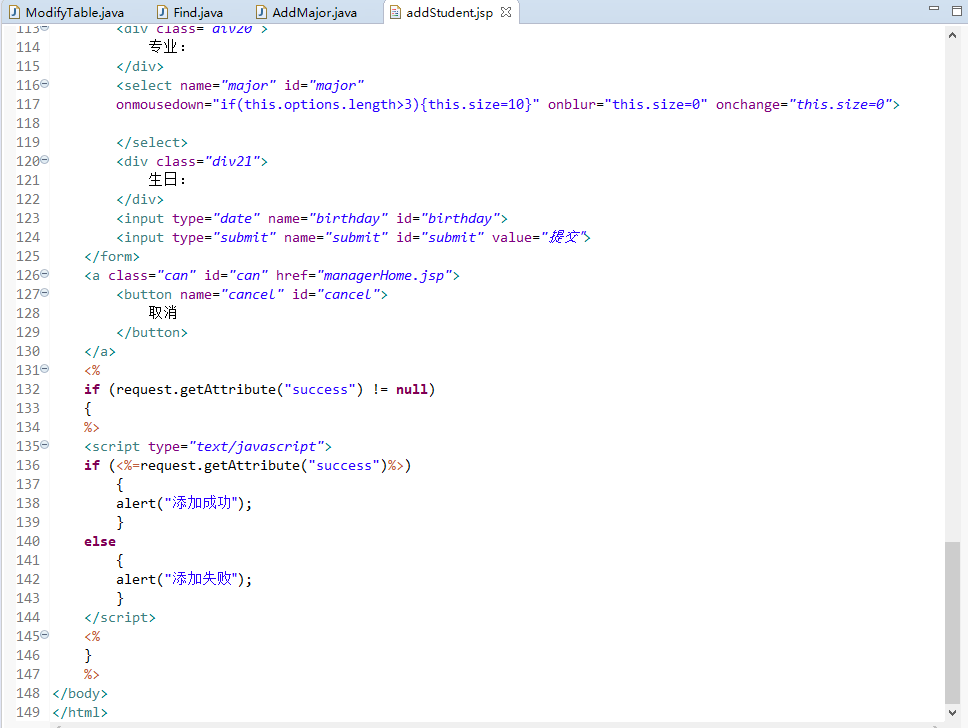














1. 测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试用例描述 | 一个输入 | 预期输出 | 实际输出 | 结果 |
| 1 | 测试用户登录是否出现异常 | 账户、密码其中之一错误 | 提示错误密码或账户 | 提示密码或账户错误 | true |
| 2 | 账户、密码正确 | 成功登入 | 进入登陆后界面 | true |
| 3 | 测试是否正确输出课程信息 | 条件查找课程 | 符合条件的所有课程 | null | false |
| 4 | 测试选课人数是否改变 | 选择有课余量课程 | 数据库中课余量减1 | 数据库中课余量减1 | true |
| 5 | 测试选课能否选课余量为0的课 | 选择课余量为0的课程 | 选课失败，数据库中课余量依旧为0 | 选课成功，数据库中课余量为负数 | false |
| 6 | 测试老师能否修改密码 | 填写账号、验证码和新密码 | 成功修改密码 | 成功修改密码 | true |
| 7 | 测试课表输出正常 | 学生查看课表 | 学生课表正常 | 显示乱七八糟 | false |
| 8 | 测试退课是否正常 | 学生删除某个课程 | 成功删除，数据库课余量加1 | 成功删除，但数据库课余量不变 | false |
| 9 | 测试教务员添加学生 | 输入学生信息 | 成功添加，数据库中有数据 | 成功添加，数据库中有数据 | true |
| 10 | 测试添加重复课程 | 输入已有课程信息 | 提示添加失败 | 程序崩溃 | false |