

# 电子科技大学

# 实验报告

## 一、实验一：网上选课系统面向对象需求分析

二、实验室名称：实验时间：2020/4/12

## 三、实验目的与任务

针对大学网上选课系统功能需求，使用系统建模工具创建该系统的需求模型，培养面向对象系统需求分析能力。

- 1) 根据大学网上选课系统的应用需求，进行系统的功能性需求分析，即用例模型分析。
- 2) 根据需求描述和用例规约，对系统进行状态（静态）分析，即初步的类模型分析。
- 3) 根据用例规约，进行系统的交互模型分析。
- 4) 进行状态机模型分析。

## 四、实验原理

应用面向对象系统需求分析方法，对网上选课系统进行需求建模分析。采用 UML 用例技术、UML 类图模型技术、UML 活动图模型技术、UML 状态图模型技术描述网上选课系统的需求模型。

用例图，是指由参与者、用例，边界以及它们之间的关系构成的用于描述系统功能的视图。用例图可以直观呈现系统与功能模块之间的互动关系，便于软件设计初期的建模。借助 UML 用例技术，开发相关人员，客户与专业人士能够以可视化的方式对问题进行探讨，减少了大量交流上的障碍，便于对问题达成共识。

类图是显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性的信息。类图是面向对象建模的主要组成部分。它既用于应用程序的系统分类的一般概念建模，也用于详细建模，将模型转换成编程代码。类图也可用于数据建模。借助 UML 类图模型技术，问题域最终被逐步转化，通过类来建模，通过编程语言构建这些类从而实现系统。

活动图是阐明了业务用例实现的工作流程。业务工作流程说明了业务为向所服务的业务主角提供其所需的价值而必须完成的工作。业务用例由一系列活动组成，它们共同为业务主

角生成某些工件。工作流程通常包括一个基本工作流程和一个或多个备选工作流程。工作流程的结构使用活动图来进行说明。借助 UML 活动图模型技术，可以简化工作流程，阐明了业务用例实现的工作流程。

状态图是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的状态对不同的事件做出反应，如状态机（它指定对象所在的状态序列）、使对象达到这些状态的事件和条件、以及达到这些状态时所发生的操作。借助 UML 状态图模型技术，可以研究系统中的类、角色。

## 五、实验内容

对大学网上选课系统进行基本调研，了解该系统的基本需求。使用系统建模工具创建该系统的需求分析模型。在实验中，使用 UML 用例图、活动图建模描述网上选课系统功能需求，使用 UML 类图、顺序图、状态图描述网上选课系统分析模型。具体内容如下：

- 1) 进行网上选课系统需求调研，总结基本用户需求描述。
- 2) 从用户需求描述，建立用例图模型，反映系统功能需求。
- 3) 采用表格描述系统各用例规约，反映用例处理要求。
- 4) 采用活动图描述系统各用例场景，与用例规约保持一致。
- 5) 从用例图模型提取系统实体类，建立系统初步类图模型。
- 6) 根据用例规格和实体类图，建立系统初步顺序图模型。
- 7) 建立系统总体状态图模型。

在完成上述需求分析建模后，撰写实验报告，描述网上选课系统的需求规格，并体现出分析过程、分析探讨、分析模型说明等报告内容。

## 六、实验设备及环境

“系统分析与设计（含 UML）”实验所涉及的硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网。

操作系统： Windows 10

系统建模工具： Power Designer 或 Rational Rose

## 七、实验步骤

对大学网上选课系统进行基本调研和分析，使用系统建模工具创建网上选课系统分析模型，其实验步骤如下：

- 1) 网上选课系统用户需求描述。
- 2) 网上选课系统用例图建模。

- 3) 网上选课系统各用例规约说明。
- 4) 网上选课系统用例场景活动图建模。
- 5) 网上选课系统实体类图建模。
- 6) 网上选课系统序列图建模。
- 7) 网上选课系统总体态图建模。

## 八、实验数据及结果分析

### （一）网上选课系统用户需求

在分析设计之前，我们通过在线问卷的形式得到了多方的反馈，根据调研结果分析，得到用户需求如下。

- 1) 学生：登录选课系统；查看课程的详细介绍；课程在线选择，包括权重值选课、抢课以及退补选；查看当前课表。
- 2) 教师：登录选课系统；注册添加课程；删除注册课程；查看开设课程的选课情况；修改个人信息。
- 3) 教务管理员：登录选课系统；添加课程到学生课表；从学生课表删除课程；设置学生选课时段，包括权重值选课时段、抢课时段和退补选时段。
- 4) 系统管理员：登录选课系统；管理系统状态，如打开或关闭网上选课系统；修改其它用户的登陆密码。

### （二）网上选课系统用例图建模

根据需求分析，可建立用例图如图 1-1 至图 1-4 所示。

图 1-1 展示的是学生用例图。设计用例主要包括登录系统、查看课程、选课和查看当前课表等。用例规约表如表 1-1 至表 1-8 所示。

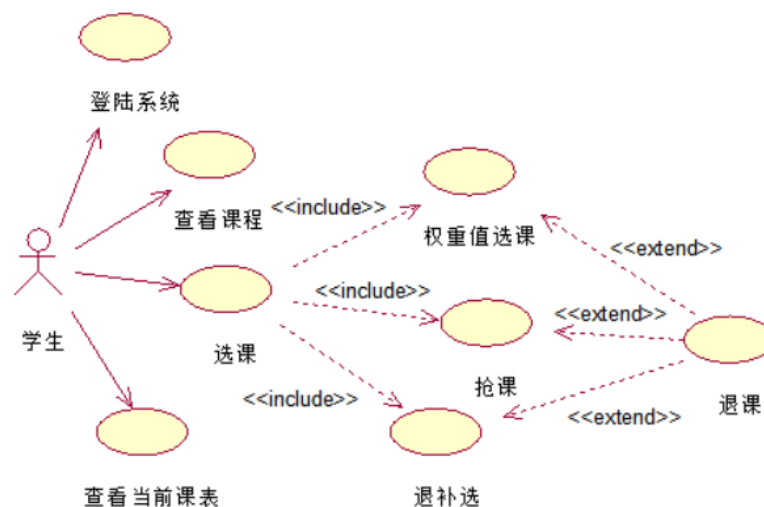


图 1-1 学生用例图

用例名称	登录系统
用例编号	STD-1
用例描述	学生登入网上选课系统
参与者	学生
前置条件	无
后置条件	学生成功登入网上选课系统
基本事件流	<ol style="list-style-type: none"> <li>1. 输入学号</li> <li>2. 输入密码</li> <li>3. 点击登录按钮</li> </ol>
备选事件流	<ol style="list-style-type: none"> <li>3a. 密码错误： <ol style="list-style-type: none"> <li>1. 显示登陆失败信息</li> <li>2. 重新输入密码</li> </ol> </li> <li>3b. 服务器正在维护： <ol style="list-style-type: none"> <li>1. 显示服务器维护信息</li> </ol> </li> </ol>
拓展点	无

表 1-1 学生登录系统用例规约

用例名称	查看课程
用例编号	STD-2
用例描述	学生对网上选课系统中已有课程进行浏览，查看课程详细信息
参与者	学生
前置条件	学生成功登入网上选课系统
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 选择课程</li> <li>2. 页面显示课程详细信息</li> </ol>
备选事件流	<ol style="list-style-type: none"> <li>1a. 点击进入课程详细信息页面前，课程被删除： <ol style="list-style-type: none"> <li>1. 显示未找到课程页面</li> <li>2. 回到上一级页面</li> </ol> </li> </ol>
拓展点	无

表 1-2 查看课程用例规约

用例名称	选课
用例编号	STD-3
用例描述	学生根据选课批次在线选课

参与者	学生
前置条件	学生成功登入网上选课系统；当前为选课时间段
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入选课页面</li> <li>2. 根据选课时段进行选课操作</li> <li>3. 选课成功</li> </ol>
备选事件流	<ol style="list-style-type: none"> <li>2a. 选课人数已满： <ol style="list-style-type: none"> <li>1. 显示选课人数已满页面</li> <li>2. 回到上一级页面</li> </ol> </li> <li>2b. 不在学生选课时间段： <ol style="list-style-type: none"> <li>1. 显示未到选课时间页面</li> <li>2. 回到上一级页面</li> </ol> </li> <li>1a. 学号不存在： <ol style="list-style-type: none"> <li>1. 显示学号不存在提示信息</li> <li>2. 清除输入学号</li> </ol> </li> <li>1b. 学号对应学生无法选择该课程： <ol style="list-style-type: none"> <li>1. 显示学生不符合选课条件信息</li> <li>2. 清除输入学号</li> </ol> </li> <li>2a. 课程与学生课表冲突： <ol style="list-style-type: none"> <li>1. 拒绝添加课程</li> </ol> </li> </ol>
拓展点	无

表 1-3 选课用例规约

用例名称	权重值选课
用例编号	STD-4
用例描述	学生通过权重值法进行选课
参与者	学生
前置条件	当前为权重值选课时段
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 选择课程</li> <li>2. 设置选择课程权重</li> <li>3. 确认选课</li> </ol>
备选事件流	<ol style="list-style-type: none"> <li>3a. 课程不需要权重值即可选课： <ol style="list-style-type: none"> <li>1. 设置权重值为 0</li> <li>2. 确认选课</li> </ol> </li> </ol>

	3b. 设置权重值高于剩余权重值： <ol style="list-style-type: none"> <li>1. 显示权重值不足提示信息</li> <li>2. 设置权重值为当前剩余权重值</li> <li>3. 确认选课</li> </ol> 3c. 设置课程权重不合法： <ol style="list-style-type: none"> <li>1. 拒绝权重设置</li> <li>2. 显示提示信息</li> <li>3. 回到“设置选择课程权重”事件流</li> </ol>
拓展点	选择课程后，学生可以选择取消选择课程，进入退课用例

表 1-4 权重值选课用例规约

用例名称	抢课
用例编号	STD-5
用例描述	学生通过抢课进行选课
参与者	学生
前置条件	当前为抢课时间段
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 选择课程</li> <li>2. 确定课程选择</li> </ol>
备选事件流	2a. 锁定前课程人数已满： <ol style="list-style-type: none"> <li>1. 提示课程人数已满</li> <li>2. 拒绝选课操作</li> </ol>
拓展点	选择课程后，学生可以选择取消选择课程，进入退课用例

表 1-5 抢课用例规约

用例名称	退补选
用例编号	STD-6
用例描述	学生在退补选时间退掉课程
参与者	学生
前置条件	当前为退补选时间段
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 选择课程</li> <li>2. 确定课程选择</li> </ol>
备选事件流	2a. 锁定前课程人数已满：

	<ol style="list-style-type: none"> <li>1. 提示课程人数已满</li> <li>2. 拒绝选课操作</li> </ol>
拓展点	选择课程后，学生可以选择取消选择课程，进入退课用例

表 1-6 退补选用例规约

用例名称	退课
用例编号	STD-7
用例描述	学生在任何选课时间段里退掉已选课程
参与者	学生
前置条件	当前为任何选课时间段
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 选择课程</li> <li>2. 确认退课</li> <li>3. 退课成功</li> </ol>
备选事件流	2a. 当前课程为不可退课的课程： <ol style="list-style-type: none"> <li>1. 提示课程不可退课</li> <li>2. 拒绝退课请求</li> </ol>
拓展点	无

表 1-7 退课用例规约

用例名称	查看当前课表
用例编号	STD-8
用例描述	学生查看当前课表排课信息
参与者	学生
前置条件	学生成功登入网上选课系统
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入课程表页面</li> <li>2. 显示当前课表信息</li> </ol>
备选事件流	2a. 课程表响应失败： <ol style="list-style-type: none"> <li>1. 提示当前无法访问课程表</li> <li>2. 回到主页面</li> </ol>
拓展点	无

表 1-8 查看当前课表用例规约

图 1-2 展示的是教师用例图。设计用例包括登录系统、注册课程、删除课程、查看选课

情况和修改个人信息。用例规约表如表 1-9 至表 1-13 所示。

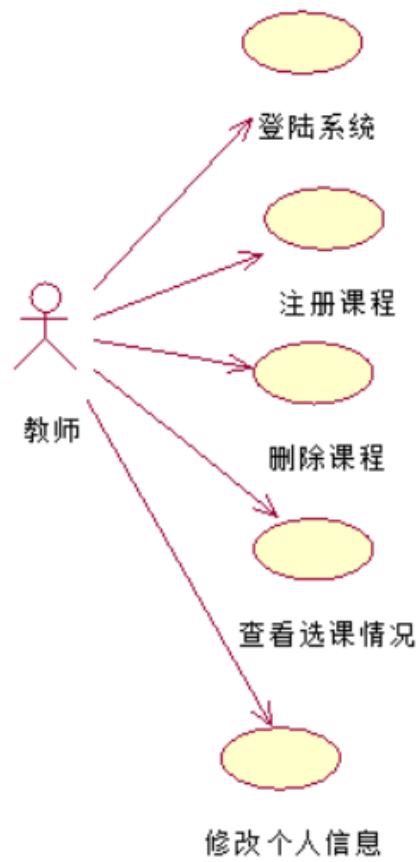


图 1-2 教师用例图

用例名称	登录系统
用例编号	TEA-1
用例描述	教师登入网上选课系统
参与者	教师
前置条件	无
后置条件	教师成功登入网上选课系统
基本事件流	1. 输入教职工号 2. 输入密码 3. 点击登录按钮
备选事件流	3a. 密码错误： 1. 显示登陆失败信息 2. 重新输入密码 3b. 服务器正在维护：



	1. 显示服务器维护信息
拓展点	无

表 1-9 教师登录系统用例规约

用例名称	注册课程
用例编号	TEA-2
用例描述	教师注册添加课程到选课系统中
参与者	教师
前置条件	教师成功登入网上选课系统
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入“我的课程”页面</li> <li>2. 选择添加课程</li> <li>3. 输入课程名称</li> <li>4. 选择开课年级</li> <li>5. 设置课程上课时间</li> <li>6. 设置课程考核方式、简介等信息</li> <li>7. 确认添加课程</li> </ol>
备选事件流	<p>7a. 未填写课程上课时间：</p> <ol style="list-style-type: none"> <li>1. 提示“课程未设置上课时间，是否继续”</li> <li>2. 课程添加成功</li> </ol>
拓展点	无

表 1-10 注册课程用例规约

用例名称	删除课程
用例编号	TEA-3
用例描述	教师删除已注册的课程
参与者	教师
前置条件	教师成功登入网上选课系统；目标课程尚未开课或已经结束
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入“我的课程”页面</li> <li>2. 选择课程</li> <li>3. 点击删除课程按钮</li> <li>4. 确认删除课程</li> </ol>
备选事件流	4a. 删除课程后想取消删除操作

	1. 点击取消删除按钮 2. 课程恢复初始状态
拓展点	无

表 1-11 删除课程用例规约

用例名称	查看选课情况
用例编号	TEA-4
用例描述	教师查看当前选择已注册课程的学生情况
参与者	教师
前置条件	教师成功登入网上选课系统
后置条件	无
基本事件流	1. 进入“我的课程”页面 2. 查看课程选择情况
备选事件流	无
拓展点	无

表 1-12 查看选课情况用例规约

用例名称	修改个人信息
用例编号	TEA-5
用例描述	教师修改个人相关资料信息
参与者	教师
前置条件	教师成功登入网上选课系统
后置条件	修改后的个人信息同步到注册课程的详细信息页面
基本事件流	1. 进入“我的信息”页面 2. 修改简介信息 3. 点击确认修改信息按钮
备选事件流	3a. 修改前后未发生改变 1. 确认修改按钮变灰（无法点击）
拓展点	无

表 1-13 修改个人信息用例规约

图 1-3 展示的是教务管理员用例图。设计用例主要包括登录系统、修改学生课程和设置

选课时段。用例规约表如表 1-14 至表 1-18 所示。

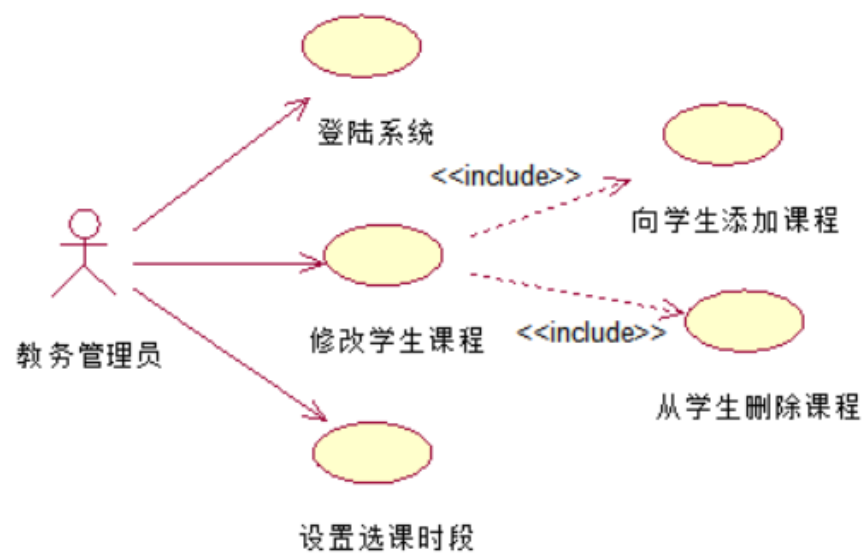


图 1-3 教务管理员用例图

用例名称	登录系统
用例编号	ADMIN-1
用例描述	教务管理员登入网上选课系统
参与者	教务管理员
前置条件	无
后置条件	教务管理员成功登入网上选课系统
基本事件流	1. 输入教职工号 2. 输入密码 3. 点击登录按钮
备选事件流	3a. 密码错误： 1. 显示登陆失败信息 2. 重新输入密码 3b. 服务器正在维护： 1. 显示服务器维护信息
拓展点	无

表 1-14 教务管理员登录系统用例规约

用例名称	修改学生课程
用例编号	ADMIN-2
用例描述	教务管理员对学生课程进行添加、删除操作

参与者	教务管理员
前置条件	教务管理员成功登入网上选课系统
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入学生课程管理页面</li> <li>2. 输入学生学号</li> <li>3. 对学生课程进行修改操作</li> <li>4. 确认课程修改</li> </ol>
备选事件流	2a. 未检索到学生学号： <ol style="list-style-type: none"> <li>1. 提示不存在该学号对应学生</li> <li>2. 清空学号输入栏</li> </ol>
拓展点	无

表 1-15 修改学生课程用例规约

用例名称	向学生添加课程
用例编号	ADMIN-3
用例描述	教务管理员对学生添加选课
参与者	教务管理员
前置条件	无
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 输入课程编号</li> <li>2. 点击添加课程按钮</li> </ol>
备选事件流	1a. 未检索到课程编号： <ol style="list-style-type: none"> <li>1. 提示不存在该编号对应课程</li> <li>2. 清空课程编号输入栏</li> </ol> 2a. 课程与学生课表其它时间段冲突： <ol style="list-style-type: none"> <li>1. 拒绝添加课程</li> </ol>
拓展点	无

表 1-16 向学生添加课程用例规约

用例名称	从学生删除课程
用例编号	ADMIN-4
用例描述	教务管理员对学生删除已选到课程
参与者	教务管理员
前置条件	无

后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 选择已选课程</li> <li>2. 点击删除课程按钮</li> <li>3. 确认删除课程</li> </ol>
备选事件流	3a. 删除课程前，课程已被学生删除： <ol style="list-style-type: none"> <li>1. 提示课程不存在</li> <li>2. 刷新页面</li> </ol>
拓展点	无

表 1-17 从学生删除课程用例规约

用例名称	设置选课时段
用例编号	ADMIN-5
用例描述	教务管理员设置学生选课不同时段对应时间
参与者	教务管理员
前置条件	教务管理员成功登入网上选课系统
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入设置选课时段页面</li> <li>2. 选择学生届数</li> <li>3. 设置权重值选课时段</li> <li>4. 设置抢课选课时段</li> <li>5. 设置退补选选课时段</li> <li>6. 确认选课时段设置</li> </ol>
备选事件流	6a. 选课时段设置不合法： <ol style="list-style-type: none"> <li>1. 不合法选课时间段标红</li> <li>2. 重新设置选课时间段</li> <li>3. 确认选课时段设置</li> </ol>
拓展点	无

表 1-18 设置选课时段用例规约

图 1-4 展示的是系统管理员用例图。设计用例包括登录系统、管理系统状态和修改登陆密码。用例规约表如表 1-19 至表 1-21 所示。

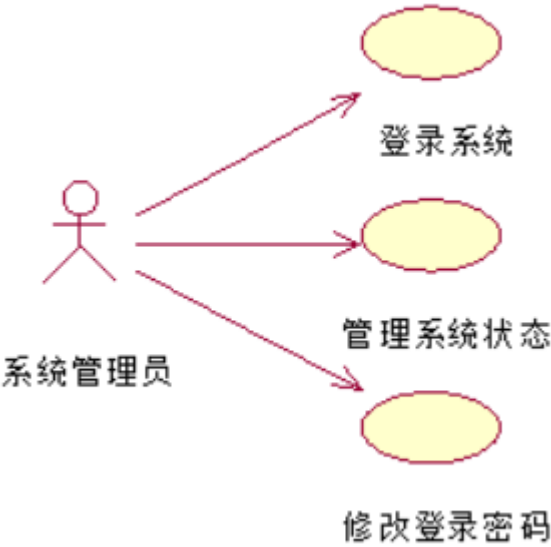


图 1-4 系统管理员用例图

用例名称	登录系统
用例编号	SYSADMIN-1
用例描述	系统管理员登入网上选课系统
参与者	系统管理员
前置条件	无
后置条件	系统管理员成功登入网上选课系统
基本事件流	1. 输入教职工号 2. 输入密码 3. 点击登录按钮
备选事件流	3a. 密码错误： 1. 显示登陆失败信息 2. 重新输入密码
拓展点	无

表 1-19 教务管理员登录系统用例规约

用例名称	管理系统状态
用例编号	SYSADMIN-2
用例描述	系统管理员设置系统开放或关闭
参与者	系统管理员
前置条件	系统管理员已登入网上选课系统
后置条件	网上选课系统开放/关闭

基本事件流	<ol style="list-style-type: none"> <li>1. 进入修改系统状态页面</li> <li>2. 设置系统开放/关闭</li> <li>3. 确认设置系统状态</li> </ol>
备选事件流	无
拓展点	无

表 1-20 管理系统状态用例规约

用例名称	修改登录密码
用例编号	SYSADMIN-3
用例描述	系统管理员重新设置某用户的网上选课系统登录密码
参与者	系统管理员
前置条件	系统管理员已登入网上选课系统
后置条件	无
基本事件流	<ol style="list-style-type: none"> <li>1. 进入修改用户密码页面</li> <li>2. 输入用户学号/教职工号</li> <li>3. 输入新密码</li> <li>4. 确认修改密码</li> </ol>
备选事件流	<p>4a. 新密码不合法:</p> <ol style="list-style-type: none"> <li>1. 显示密码规范</li> <li>2. 清空新密码输入框</li> </ol> <p>4b. 学号/教职工号对应用户不存在</p> <ol style="list-style-type: none"> <li>1. 显示用户不存在</li> <li>2. 清空学号/教职工号输入框</li> </ol>
拓展点	无

表 1-21 修改登录密码用例规约

### （三）网上选课系统用例场景活动图建模

本网上选课系统主要用例为登录系统和选课。建立活动图模型如图 1-5 至 1-7 所示。

图 1-5 是用户登录系统的活动图。系统首先显示登陆页面，由用户输入登录信息，服务器端将登录信息加密并与从数据库获取的用户信息进行比对，如果登录信息有效，则进一步从数据库获取用户身份，若无效，则显示登录失败信息。最后验证网上选课系统是否正在运行，若没有运行：用户为系统管理员，则登录进入网上选课系统。若不为系统管理员，则显示登录失败信息；若正在运行：则登录进入网上选课系统。

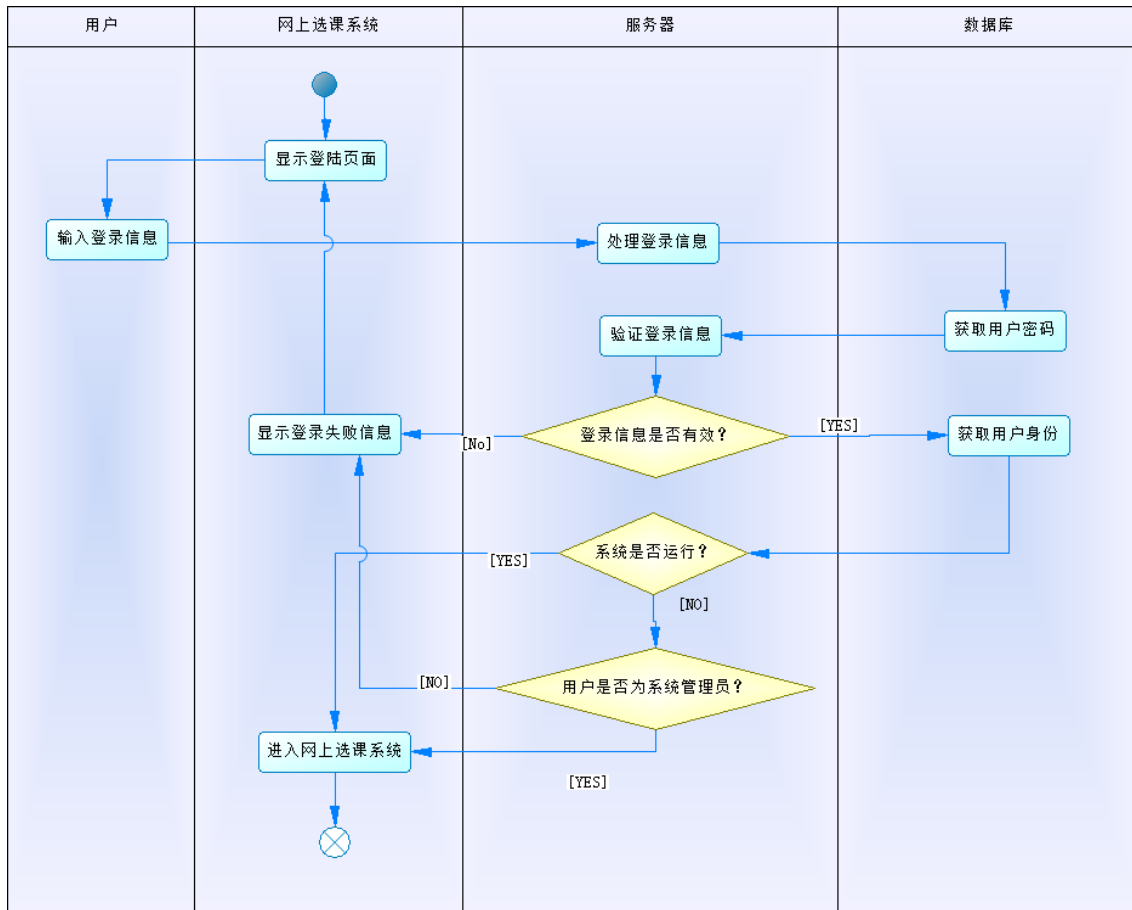


图 1-5 登录系统活动图

图 1-6 是权重值选课的活动图。首先系统显示学生可选课程，学生对课程进行选择并设置权重值，权重值选课信息发送到服务器端，验证权重值是否合法：若不合法，则在系统上显示选课失败信息，提示用户重新设置合法权重值。若合法，则将选课信息存储到数据库，系统上显示选课成功信息。

图 1-7 是抢课（或退补选）选课的活动图。首先系统显示学生可选课程，学生对课程进行选择，服务器端从数据库获取课程信息并返回到服务器端验证选课人数是否已满：若选课人数已满，则提示选课失败信息；若未滿，则将学生选课信息存储到数据库中，系统选择选课成功信息。



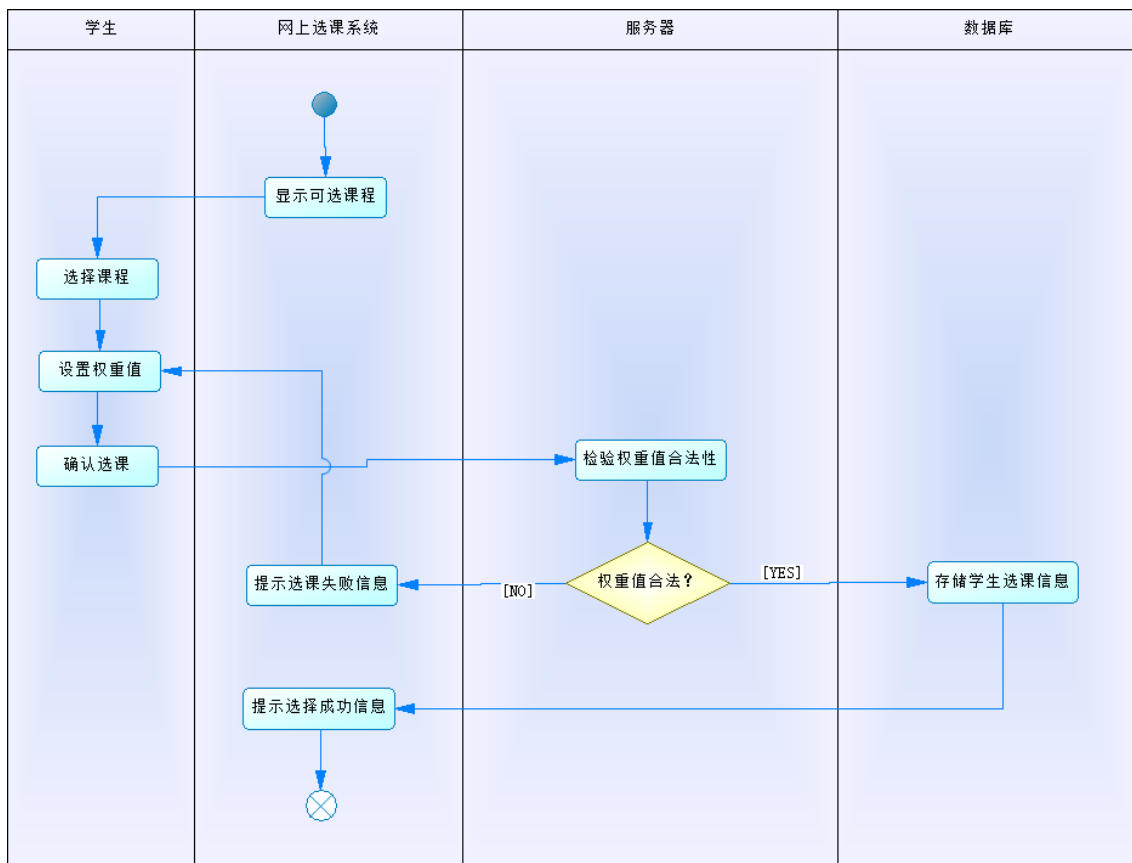


图 1-6 权重值选课活动图

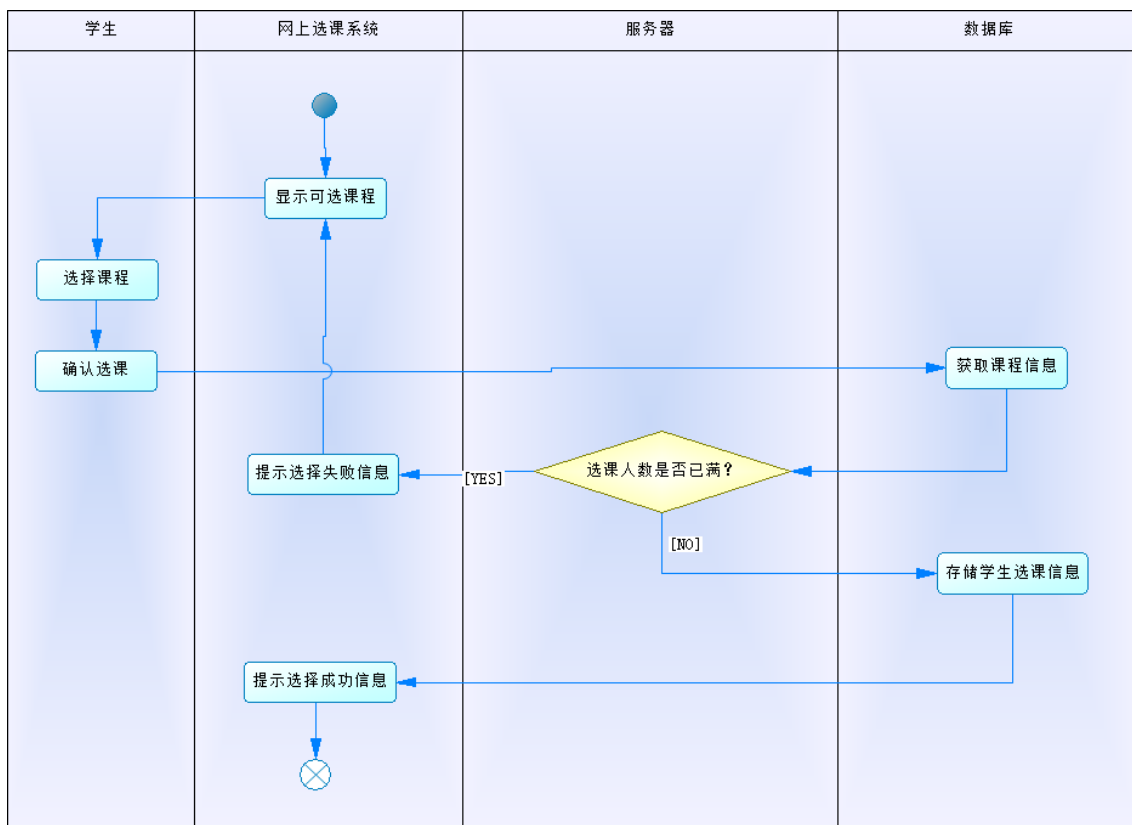


图 1-7 抢课（或退补选）选课活动图

#### （四）网上选课系统实体类图建模

分析选课系统可提取实体类包括：User 类、Student 类、Teacher 类、AcademicAdmin 类、SystemAdmin 类以及 Course 类。

如图 1-8 所示，Student、Teacher、AcademicAdmin 和 SystemAdmin 是 User 的泛化类，Student、Teacher 和 AcademicAdmin 可操作 Course 类。其中 Student 与 Course 为零或多对零或多关系，Teacher 与 Course 为一或多对零或多关系，AcademicAdmin 与 Course 为一或多对多关系。

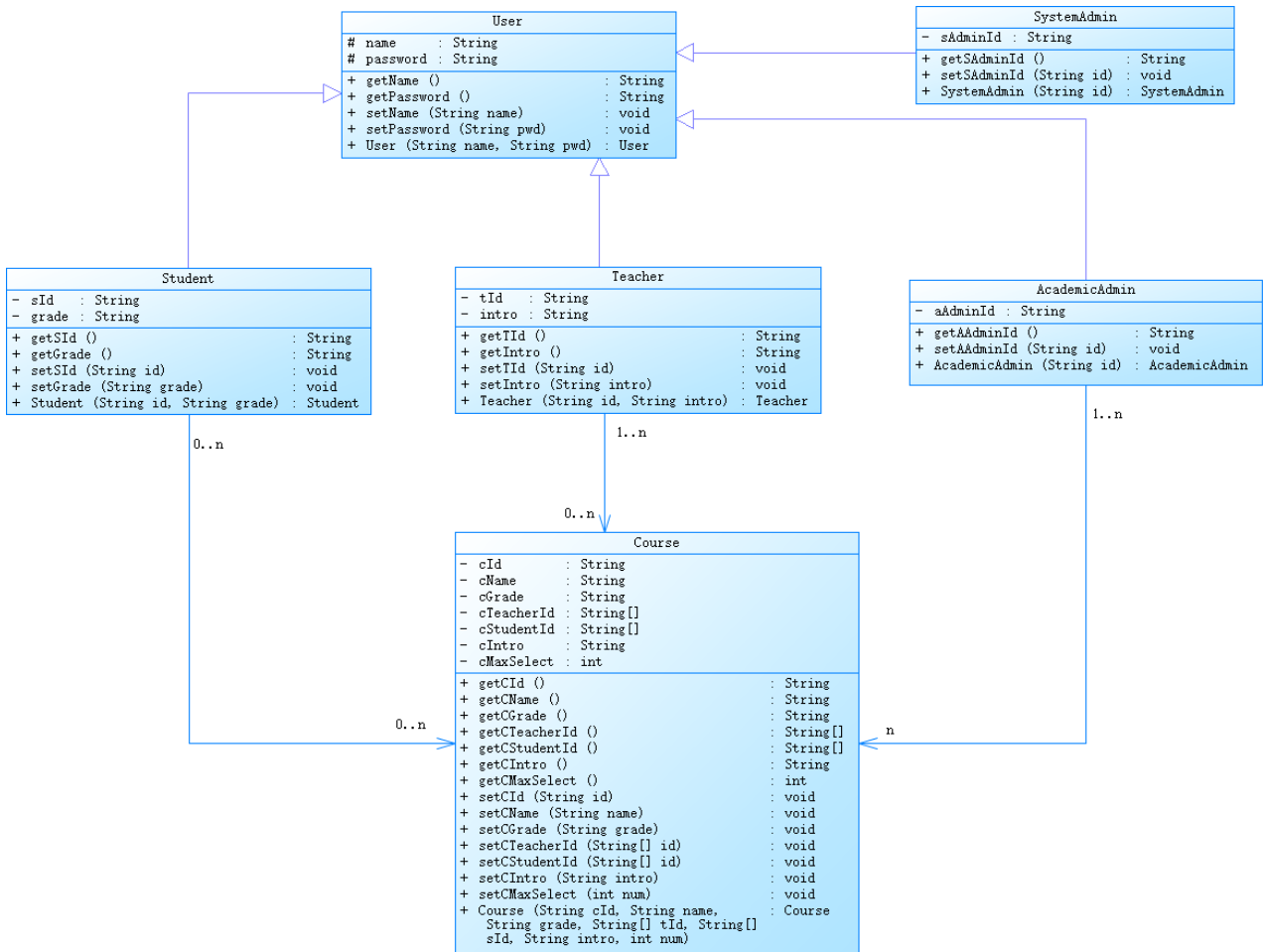


图 1-8 网上选课系统实体类图

### （五）网上选课系统序列图建模

设计登录系统和选课功能的序列图如图 1-9 至 1-11 所示。

如图 1-9 所示为登录系统的序列图。用户向 Web Interface 发送登录凭证信息，再发送至 Login Interface，由 Login Interface 和 Database Wrapper 共同验证处理登录凭证信息，最后获取服务器状态和用户身份，将处理的结果返回 Web Interface。

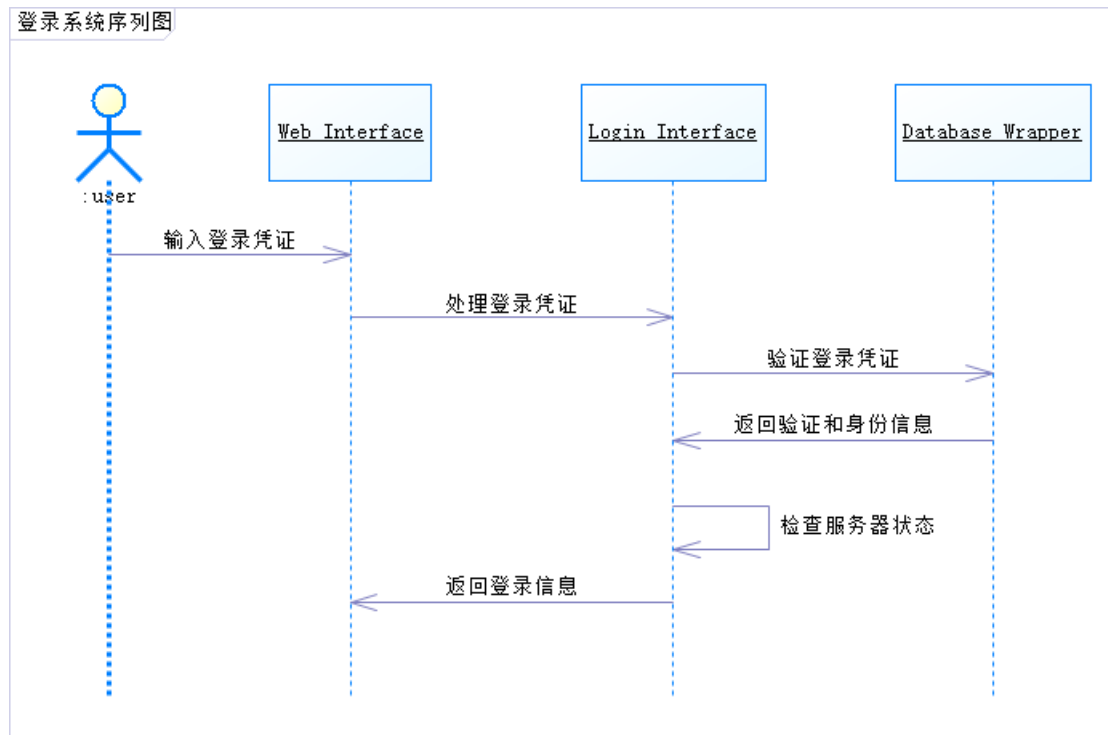


图 1-9 登录系统序列图

如图 1-10 所示为权重值选课的序列图。学生向 Web Interface 发送选课请求和权重信息，再由 Select Course Interface 检验请求合法性，并进一步存储选课信息，最后将选课结果返回到 Web Interface。

如图 1-11 所示为抢课（或退补选）选课的序列图。学生向 Select Course Interface 发送选课请求，Select Course Interface 从 Database Wrapper 获取目标课程的选课信息，检验选课人数是否已满，最后将选课结果返回 Web Interface。

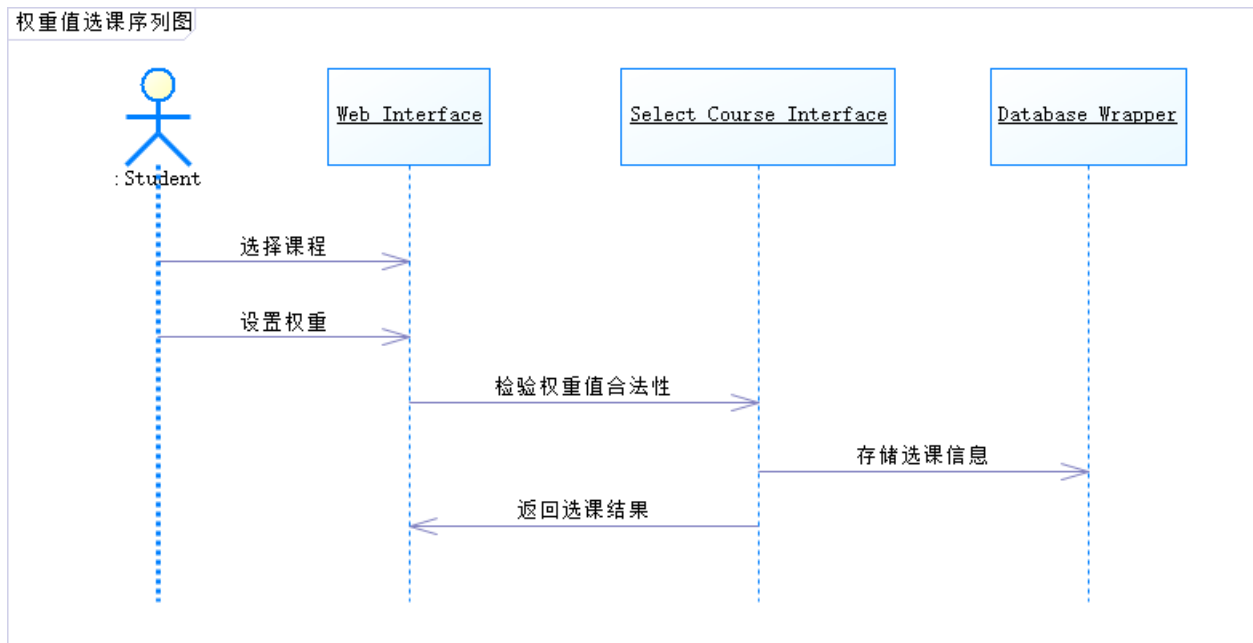


图 1-10 权重值选课序列图

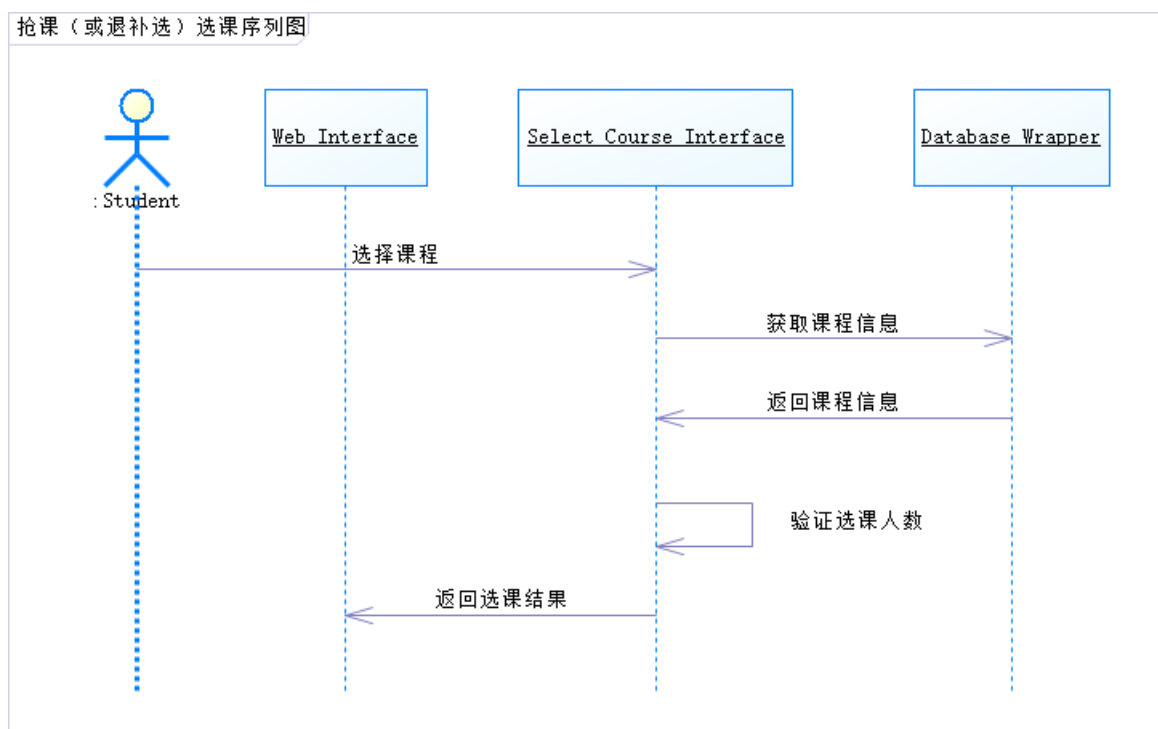


图 1-11 抢课（或退补选）选课序列图

### （六）网上选课系统课程状态图模型

课程状态图如图 1-12 所示，课程存在五种状态：创建、活动中、锁定、编辑中和已删除。

- 1) 教师注册课程时，课程类进入创建状态，由教师对课程详细信息进行编辑。
- 2) 注册完成后，课程类进入活动中状态，保存信息到数据库。若系统为打开状态且课程未进入编辑状态，则可以由学生进行选课操作。
- 3) 活动中的课程进行编辑时进入编辑中状态，学生可以进行选课操作，编辑完成后回到活动中状态。
- 4) 活动中的课程当系统关闭时进入锁定状态，学生无法进行选课操作，系统打开后回到活动中状态。
- 5) 活动中的课程被教师删除后进入已删除状态，不能再进行操作。

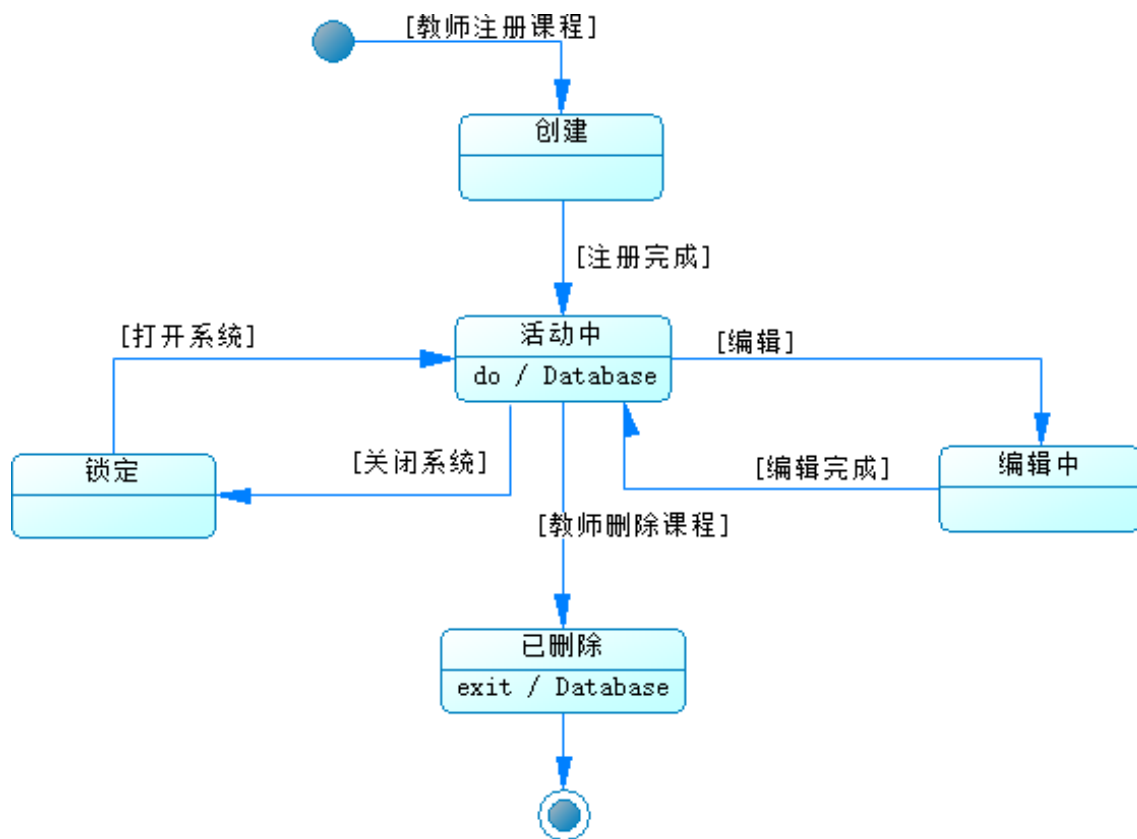


图 1-12 课程状态图

## 九、总结、改进建议及心得体会

// removed

# 电子科技大学

## 实验报告

### 一、实验二：网上选课系统面向对象设计

二、实验室名称：实验时间：2020/4/19

### 三、实验目的

针对大学网上选课系统设计，使用系统建模工具创建该系统的设计模型，培养面向对象系统设计能力。

- 1) 基于前面建立的系统分析模型，设计系统的体系结构。
- 2) 基于前面建立的系统分析模型，细化得到系统的设计模型，了解面向对象系统设计的基本方法。
- 3) 理解系统分析模型与系统设计模型的联系和区别。

### 四、实验原理

应用面向对象系统设计方法，对网上选课系统进行设计建模。采用 UML 包图模型技术、UML 类图模型技术、UML 序列图模型技术、UML 构件图技术、UML 部署图技术描述网上选课系统的设计模型。

包图是在 UML 中用类似于文件夹的符号表示的模型元素的组合。系统中的每个元素都只能为一个包所有，一个包可嵌套在另一个包中。使用包图可以将相关元素归入一个系统。一个包中可包含附属包、图表或单个元素。一个包图可以是任何一种的 UML 图组成，通常是 UML 用例图或 UML 类图。包是一个 UML 结构，借助 UML 包图模型技术，能够把诸如用例或类之类模型元件组织为组。

类图是显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图不显示暂时性的信息。类图是面向对象建模的主要组成部分。它既用于应用程序的系统分类的一般概念建模，也用于详细建模，将模型转换成编程代码。类图也可用于数据建模。借助 UML 类图模型技术，问题域最终被逐步转化，通过类来建模，通过编程语言构建这些类从而实现系统。

序列图通过描述对象之间发送消息的时间顺序显示多个对象之间的动态协作。它可以表

示用例的行为顺序，当执行一个用例行为时，时序图中的每条消息对应了一个类操作或状态机中引起转换的触发事件。利用 UML 序列图模型技术，可以直观体现程序运行时，各个对象之间的交流、时间机制。

构件图主要用于描述各种软件构件之间的依赖关系。其中，构件是系统的可替代的物理部分，它表示的是实际的事物。构件是定义了良好接口的物理实现单元。它是系统中可以替代的部分。每个构件体现了系统设计中的特定类的实现。良好定义的构件不直接依赖于其它构件而依赖于构件所支持的接口。在这种情况下，系统中的一个构件可以被支持正确的接口的其它构件所替代。接口是被软件或硬件所支持的一个操作集。通过使用命名的接口，可以避免在系统的各个构件之间直接发生依赖关系。有利于新构件的替换。借助 UML 构件图模型，可以清楚地看出系统的结构和功能。方便项目组的成员制定工作目标和了解工作情况，同时有利于软件的复用。

部署图是用来显示系统中软件和硬件的物理架构图，描述了一个运行时的硬件结点，以及在这些结点上运行的软件组件的静态视图。部署图显示了系统的硬件，安装在硬件上的软件，以及用于连接异构的机器之间的中间件。借助 UML 部署图模型技术，可以显示运行时系统的结构，同时还传达构成应用程序的硬件和软件元素的配置和部署方式。

## 五、实验内容

采用面向对象系统设计方法对大学网上选课系统进行建模设计。使用系统建模工具创建该系统的设计模型。具体实验内容如下：

- 1) 完善网上选课系统类图建模，反映系统静态结构设计。
- 2) 完善网上选课系统序列图建模，反映系统动态逻辑设计。
- 3) 网上选课系统包图与组件图建模，反映系统的软件逻辑结构关系。
- 4) 网上选课系统部署图建模，反映软件组件在系统环境结点中的物理部署关系。
- 5) 设计模型转换为代码框架，反映软件设计与软件实现的对应关系。

在完成上述模型设计后，撰写实验报告，描述网上选课系统的设计蓝图，并体现出设计思想、设计过程、设计探讨、设计说明等报告内容。

## 六、实验设备及环境

“系统分析与设计（含 UML）”实验所涉及的硬件设备为 pc 计算机、服务器以及网络环境，pc 计算机与服务器在同一局域网络。

操作系统： Windows 10

系统建模工具： Power Designer 或 Rational Rose

## 七、实验步骤



对大学网上选课系统进行设计建模，使用系统建模工具创建网上选课系统设计模型，其实验步骤如下：

- 1) 完善网上选课系统类图模型。
- 2) 完善网上选课系统序列图模型。
- 3) 创建网上选课系统包图、组件图模型。
- 4) 创建网上选课系统部署图模型。
- 5) 设计模型转换为代码框架。

## 八、实验数据及结果分析

### （一）完善网上选课系统类图模型

网上选课系统的实体类图如实验一的图 1-8 所示。在此基础上进行细化，对用户的登录系统功能以及学生的选课功能进行进一步的类图建模，如图 2-1 和 2-2 所示。完善后的实体类图如图 2-3 所示。

图 2-1 是用户的登录系统的细化类图模型。新增了 Login 类，当用户在登录界面输入表单时，将输入的账号 id 和密码 password 传入类 Login 中，调用 checkLoginStatus() 函数访问服务器验证登录凭证是否有效，实现了网上选课系统的登录功能。

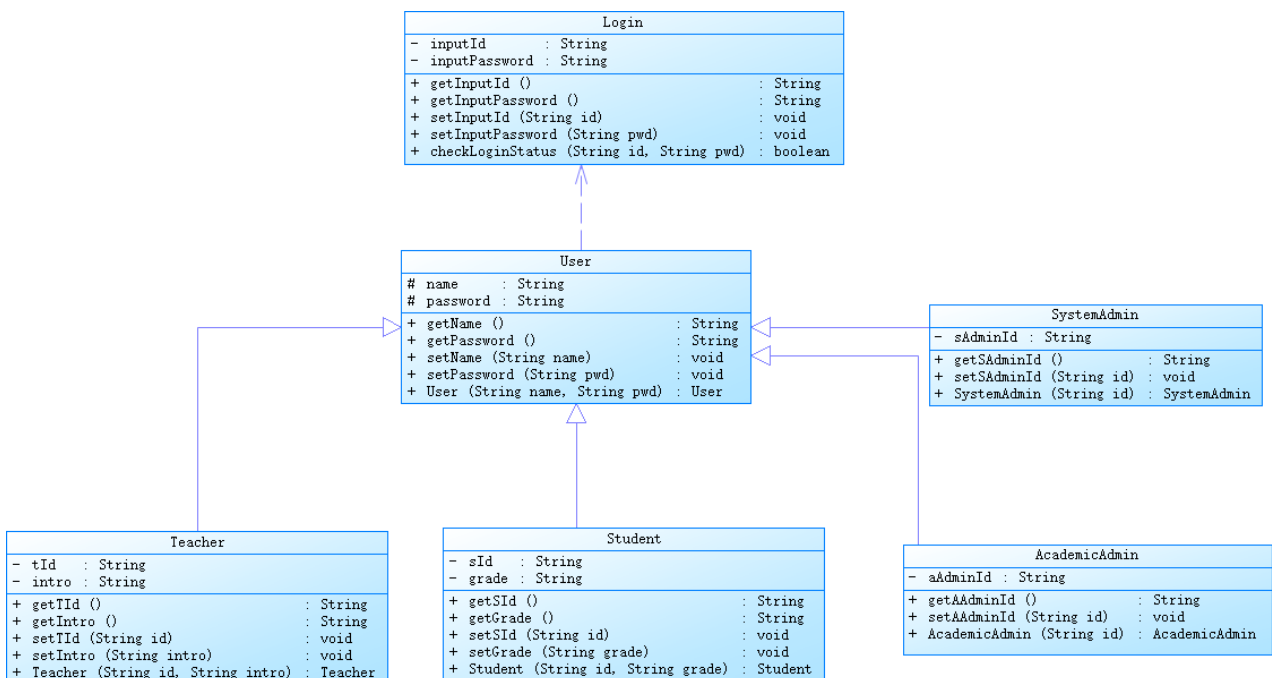


图 2-1 登录系统类图

图 2-2 是学生权重值选课的细化类图模型。新增了 SelectCourse 和 DropCourse 类，当学生选择课程时，将学号、课程号和权重信息传给 SelectCourse 类，存储到数据库中，权重选课结束以后将调用服务器相应算法处理选课信息，并最终将结果传入 Course 类；学生退课

时，将学号和课程号传给 DropCourse 类，服务器获取退课请求后从数据库删除相应选课信息，并最终将结果体现在 Course 类中。抢课和退补选的细化类图模型是本模型删除 SelectCourse 类有关权重配置的简化版，不再赘述。

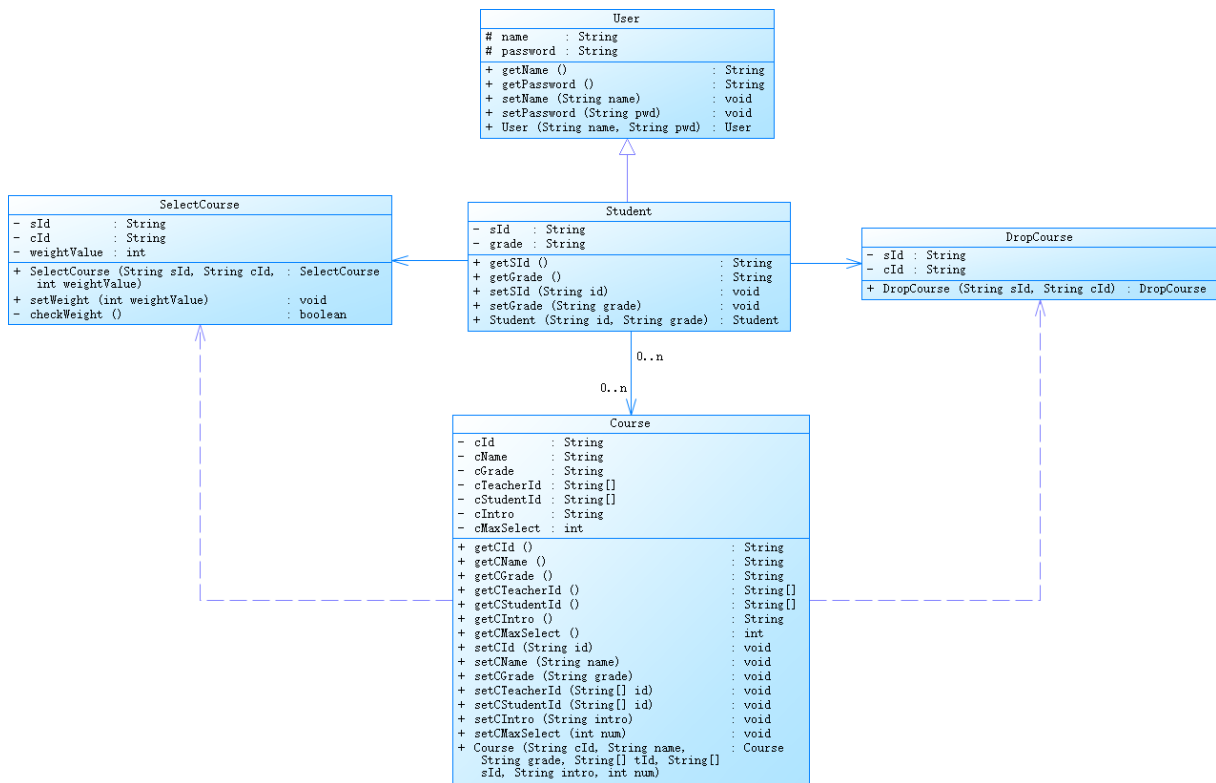


图 2-2 权重值选课类图

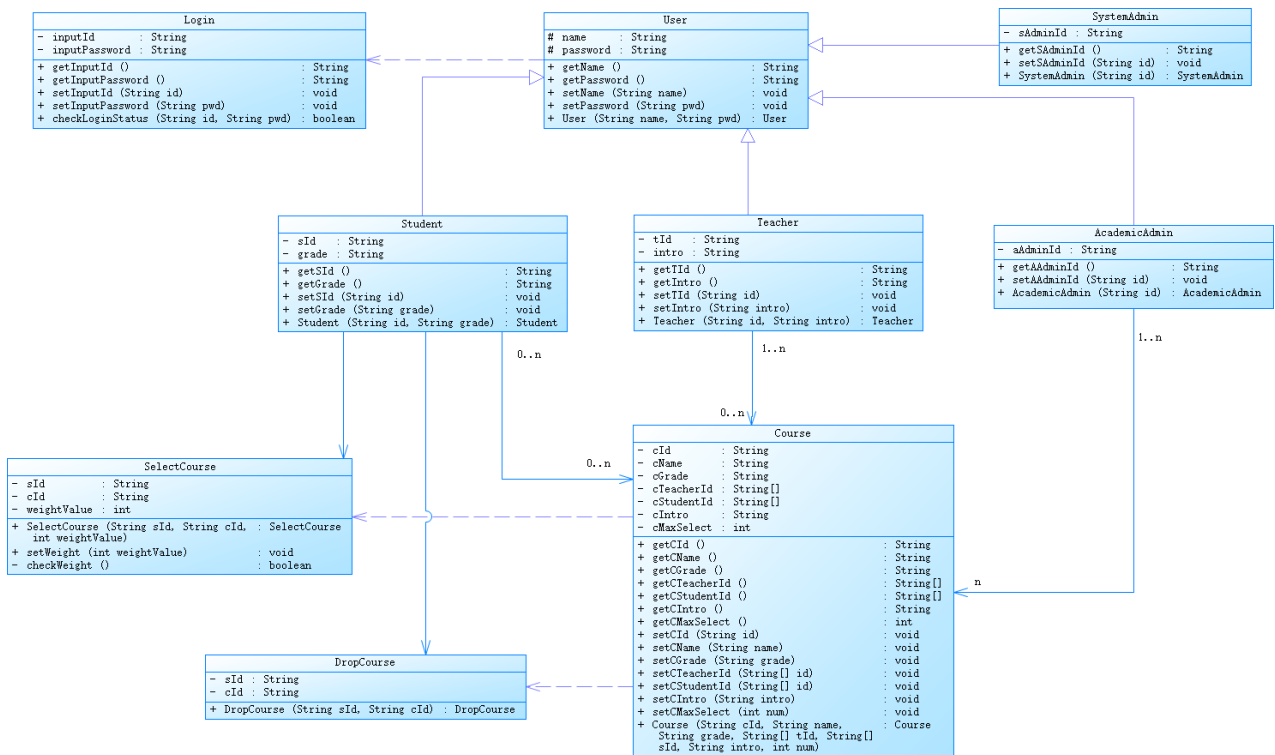


图 2-3 完善后的网上选课系统实体类图

## （二）完善网上选课系统序列图模型

对实验一中的序列图进行完善，拓展已有的操作类，细化各对象的交互逻辑，建模得到序列图如图 2-4 至 2-6 所示。

图 2-4 是用户登录系统的完善序列图。在登录页面，用户输入登录凭证，前端将登录请求发送到服务器端，服务器再与数据库存储的登录凭证相比对。如果凭证有效，则检查服务器状态。下面分为三种情况进行讨论：

- 1) 凭证有效且网上选课系统打开。用户登录成功。
- 2) 凭证有效但网上选课系统关闭。服务器从数据库中获取用户身份标识，若用户为系统管理员，则登录成功；若不是，则显示服务器维护信息，登录失败。
- 3) 凭证无效。用户登录失败。

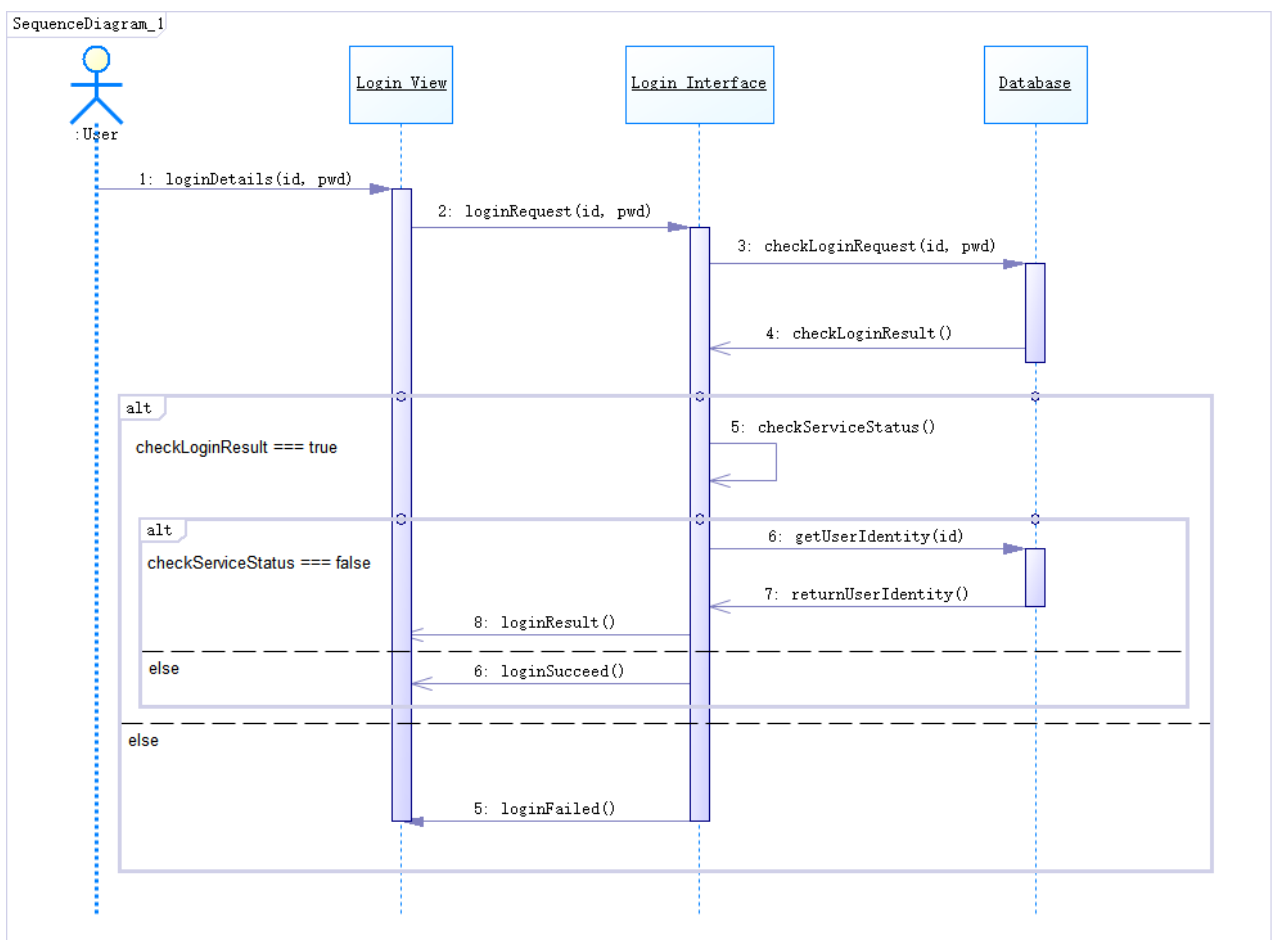


图 2-4 完善后的用户登录序列图

图 2-5 是学生权重值选课的完善序列图。在权重值选课界面，用户选择课程并赋予一定的权重，将选课请求发送到服务器。服务器首先从数据库读取用户的选课信息，计算剩余权重，通过 `checkWeightValue()` 方法验证输入权重是否合法。下面分为两种情况进行讨论：

- 1) 权重合法。添加新的选课信息到数据库中，返回选课成功信息。
- 2) 权重不合法。返回选课失败信息。

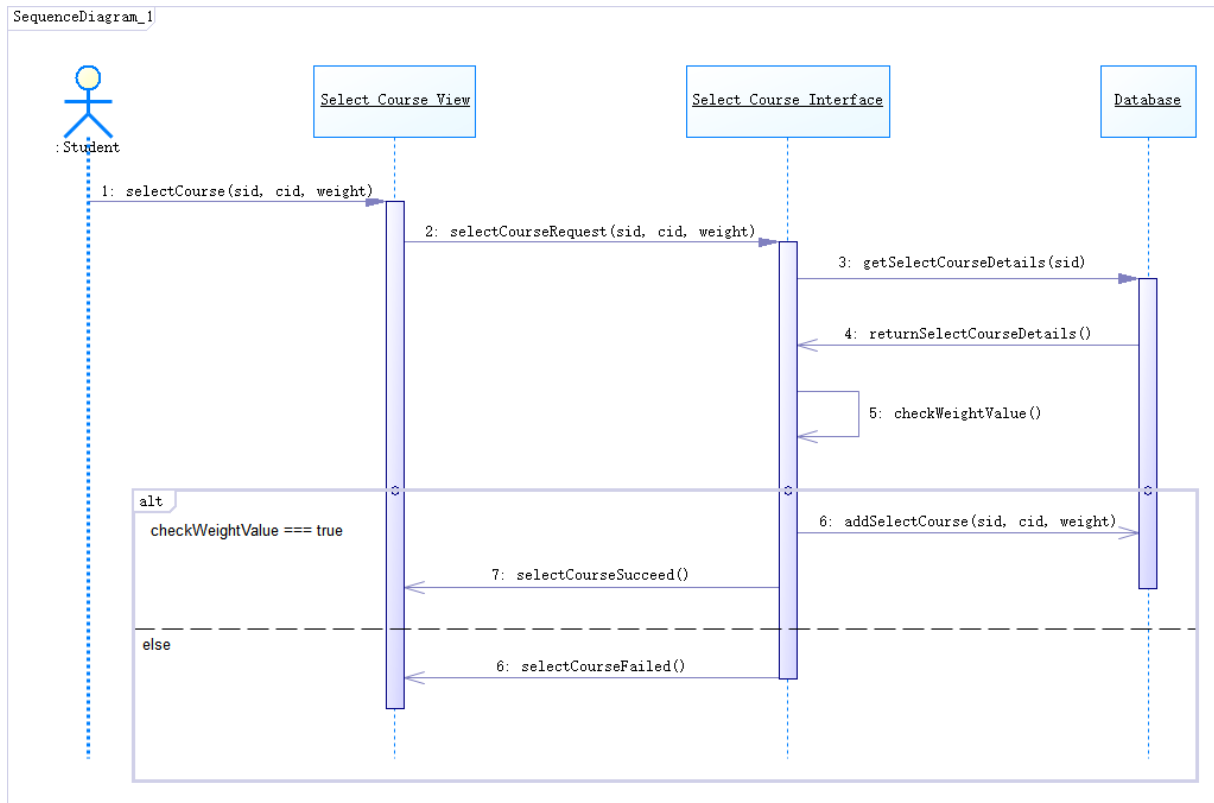


图 2-5 完善后的权重值选课序列图

图 2-6 是学生退课的完善序列图。在退课界面，学生选择课程进行退课操作，将退课请求发送到服务器。服务器从数据库得到课程选课数据，验证学生退课操作是否有效。最后从数据库中删除课程中该学生的选课信息，返回退课结果。

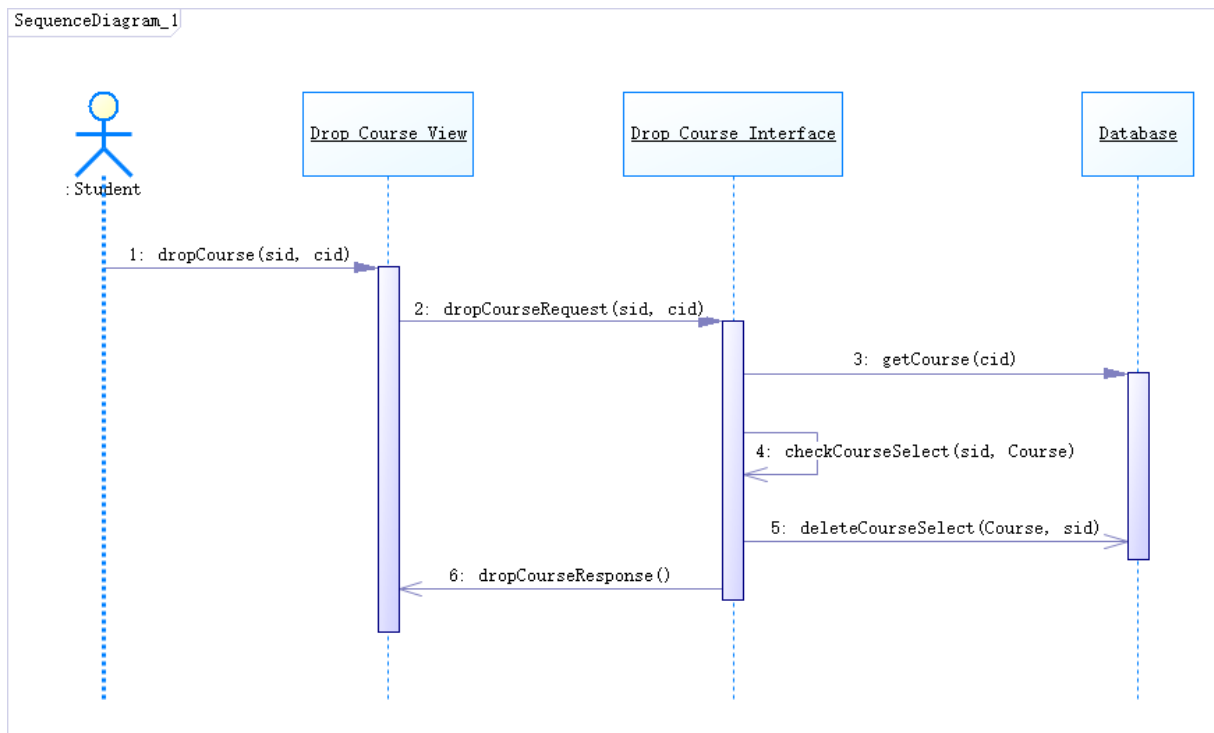


图 2-6 完善后的退课序列图

### （三）创建网上选课系统包图、组件图模型

结合用例图和类建模，可作系统包图和组件如图 2-7 和 2-8 所示。

图 2-7 为网上选课系统的包图模型。System 包引用了 UI、User、Database、Course 包，分别用于存储前端设计文件、用户类文件、数据库及后端文件和课程类及课程操作文件。其中 Course 包又依赖于 Database、SelectCourse、DropCourse、AddCourse、DeleteCourse 包，实现选课、退课、注册课程、删除课程操作与在数据库中的存储、调用。

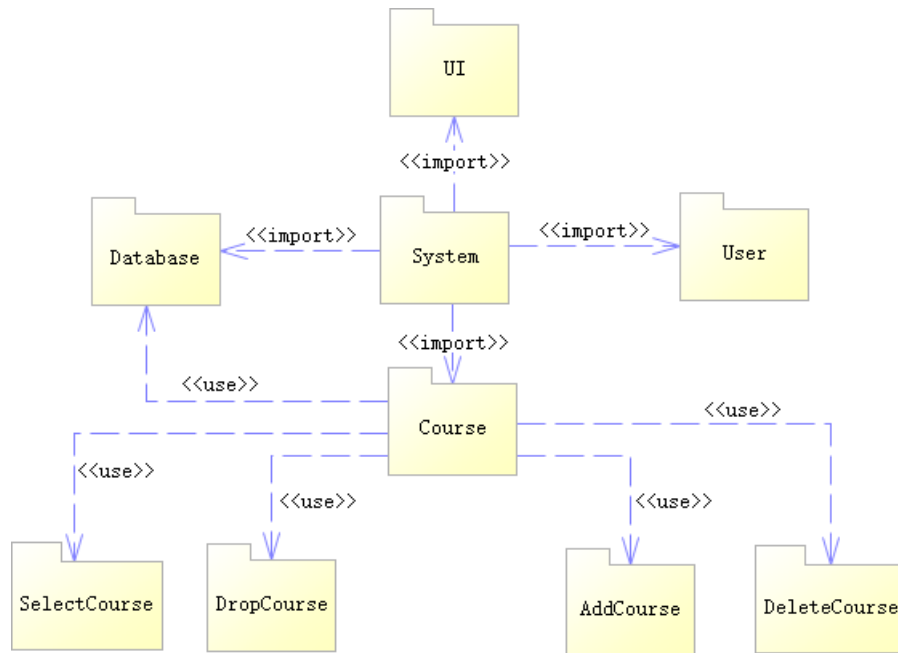


图 2-7 网上选课系统包图

图 2-8 为网上选课系统的组件图模型。网上选课系统依赖于组件 Login、Course 和 UI，其中 Course 组件依赖于 CourseOperate 和 OCSSDatabase。上述组件共同构成了网上选课系统的核心功能。

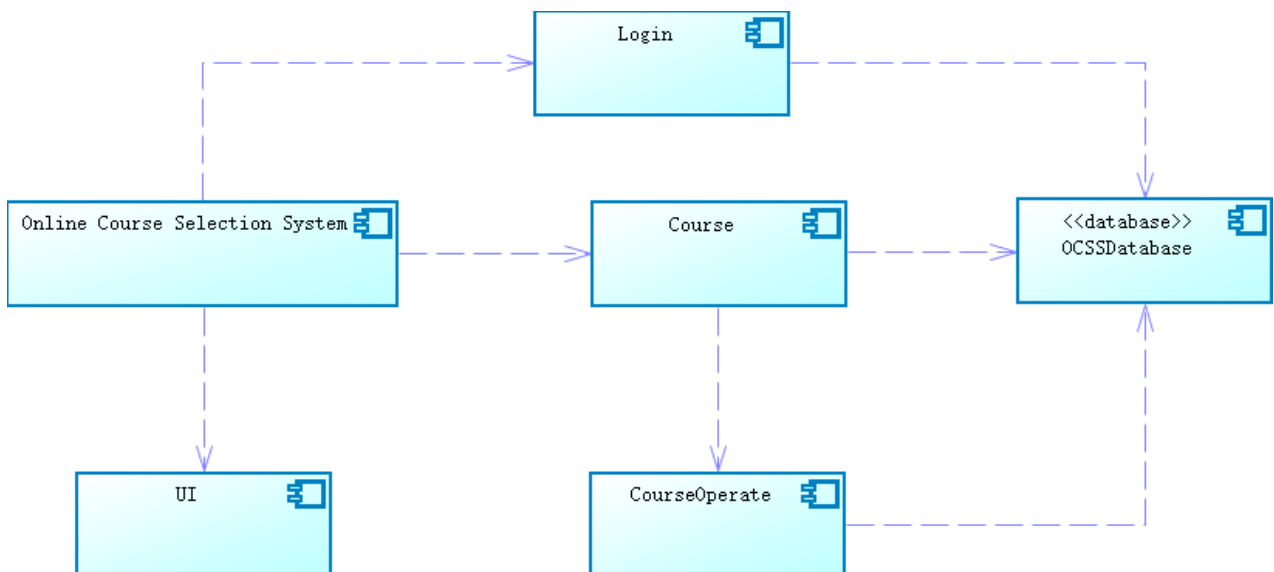


图 2-8 网上选课系统组件图

#### （四）创建网上选课系统部署图模型

根据之前建模的分析，可以建立网上选课系统部署图如图 2-9 所示。本系统包括三个结点：客户端、应用服务器和数据库服务器。其中客户端和应用服务器之间通过 TCP/IP 协议通信，应用服务器和数据库服务器之间利用 JDBC 通信。将构件图中的构件放在部署图中，其中 UI 构件在客户端结点中，Login、Course 和 CourseOperate 构件在应用服务器结点中，OCSSDatabase 构件在数据库服务器结点中，并新增细化的数据库构件 UserDatabaseOperate 和 CourseDatabaseOperate。

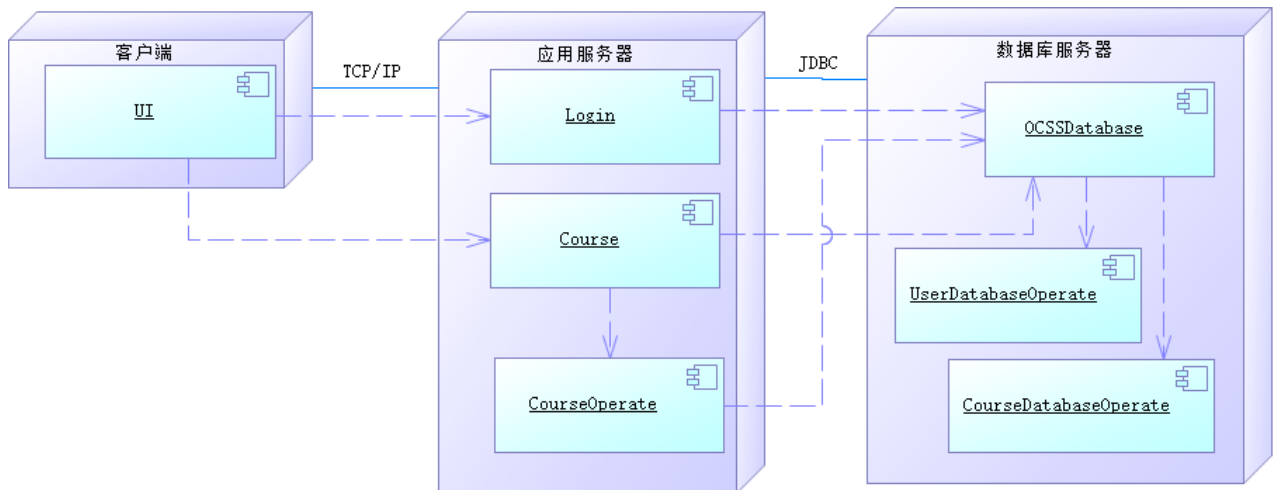


图 2-9 网上选课系统部署图

## (五) 网上选课系统设计模型框架代码转换

将完善后的实体类图转化为 Java 程序如代码 2-1 至 2-9 所示。AcademicAdmin、Student、SystemAdmin 和 Teacher 是 User 的泛化类，用于存储不同用户的信息，实现不同用户有不同操作权限的功能。用户可以通过 Login 类登入网上选课系统。Course 类存储课程信息和选课信息，Teacher 可以新建课程类，而 Student 可以通过 SelectCourse 类来选择 Course 类，或是通过 DropCourse 类来删除已选择的 Course 类。

代码 2-1 AcademicAdmin.java

```

/*****
* Module: AcademicAdmin.java
* Author: 40855
* Purpose: Defines the Class AcademicAdmin
*****/

import java.util.*;

/** @pdOid d1baa010-f425-4330-af30-63bdf1a81727 */
public class AcademicAdmin extends User {
    /** @pdOid fc36e8c3-e550-4f4e-bbf7-21480f144d72 */
    private String aAdminId;

    /** @pdRoleInfo migr=no name=Course assc=Association_3 coll=java.util.Collection
impl=java.util.HashSet mult=n */
    public java.util.Collection<Course> course;

    /** @pdOid 6c33d7f2-43a2-4e72-aa1e-3f8edac75493 */
    public String getAAdminId() {
        // TODO: implement
        return null;
    }

    /** @param id
     * @pdOid a08dd0e0-1f21-42d6-b470-326c972787f0 */
    public void setAAdminId(String id) {
        // TODO: implement
    }
}

```

```

/** @param id
 * @pdOid 4d8c9dbc-764d-4c18-ba9a-74fb122095cb */
public AcademicAdmin AcademicAdmin(String id) {
    // TODO: implement
}

/** @pdGenerated default getter */
public java.util.Collection<Course> getCourse() {
    if (course == null)
        course = new java.util.HashSet<Course>();
    return course;
}

/** @pdGenerated default iterator getter */
public java.util.Iterator getIteratorCourse() {
    if (course == null)
        course = new java.util.HashSet<Course>();
    return course.iterator();
}

/** @pdGenerated default setter
 * @param newCourse */
public void setCourse(java.util.Collection<Course> newCourse) {
    removeAllCourse();
    for (java.util.Iterator iter = newCourse.iterator(); iter.hasNext();)
        addCourse((Course)iter.next());
}

/** @pdGenerated default add
 * @param newCourse */
public void addCourse(Course newCourse) {
    if (newCourse == null)
        return;

```



```

        if (this.course == null)
            this.course = new java.util.HashSet<Course>();
        if (!this.course.contains(newCourse))
            this.course.add(newCourse);
    }

    /** @pdGenerated default remove
     * @param oldCourse */
    public void removeCourse(Course oldCourse) {
        if (oldCourse == null)
            return;
        if (this.course != null)
            if (this.course.contains(oldCourse))
                this.course.remove(oldCourse);
    }

    /** @pdGenerated default removeAll */
    public void removeAllCourse() {
        if (course != null)
            course.clear();
    }
}

```

代码 2-2 Course.java

```

/*****
 * Module: Course.java
 * Author: 40855
 * Purpose: Defines the Class Course
 *****/

import java.util.*;

/** @pdOid 87f0ce44-c399-4d1a-acdb-26beffd41f4d */
public class Course {
    /** @pdOid 78ff61e7-559c-47f9-a37c-e92fc4f0855e */

```

```

private String cId;
/** @pdOid b959edaf-644c-4f8a-9558-9d4505813ba9 */
private String cName;
/** @pdOid b664106e-e8c0-4f39-82be-0bb8644e813c */
private String cGrade;
/** @pdOid cd7b8fbc-a816-478b-a1df-90facb4294be */
private String[] cTeacherId;
/** @pdOid 124b6c2d-45d1-4454-b52e-846a22a87f91 */
private String[] cStudentId;
/** @pdOid 86aef5ae-9fca-4f33-baab-0beefabaecc3 */
private String cIntro;
/** @pdOid e1bc013b-d2a7-4157-860a-5a50354e80d0 */
private int cMaxSelect;

/** @pdOid 41f92502-5870-49d6-bfef-0b49f1e5f53c */
public String getCId() {
    // TODO: implement
    return null;
}

/** @pdOid c4988139-cdfe-4173-be8e-8e9c2bdf9f67 */
public String getCName() {
    // TODO: implement
    return null;
}

/** @pdOid 776340b4-fec6-4c3e-a237-3d29fb48ba47 */
public String getCGrade() {
    // TODO: implement
    return null;
}

/** @pdOid 7a5eb374-2eb8-4eba-91df-1a62541365ca */
public String[] getCTeacherId() {
    // TODO: implement

```

```
        return null;
    }

    /** @pdOid 740fec6f-e1a0-4cbd-8290-3fc9a49e0d53 */
    public String[] getCStudentId() {
        // TODO: implement
        return null;
    }

    /** @pdOid 8c3602f5-a7b2-41f5-bc73-fd9046f78877 */
    public String getCIntro() {
        // TODO: implement
        return null;
    }

    /** @pdOid 8b6d27ea-df76-4df4-8c92-c948be6ec788 */
    public int getCMaxSelect() {
        // TODO: implement
        return 0;
    }

    /** @param id
     *  * @pdOid 7a18f637-6aff-493e-a67b-1554926c4238 */
    public void setCId(String id) {
        // TODO: implement
    }

    /** @param name
     *  * @pdOid c0b04975-f5f5-4220-8873-79f21c48c488 */
    public void setCName(String name) {
        // TODO: implement
    }

    /** @param grade
     *  * @pdOid 71ec58c2-aae7-4da4-b979-146f12304b6b */
```

```

public void setCGrade(String grade) {
    // TODO: implement
}

/** @param id
 * @pdOid 94e5a035-50c7-42cb-aa27-9ac3abfe07d7 */
public void setCTeacherId(String[] id) {
    // TODO: implement
}

/** @param id
 * @pdOid 0abf179d-00f4-4e67-ae11-e62abb2a7df6 */
public void setCStudentId(String[] id) {
    // TODO: implement
}

/** @param intro
 * @pdOid d6f4bbdf-04c2-42eb-a071-6e8e7847bde0 */
public void setCIntro(String intro) {
    // TODO: implement
}

/** @param num
 * @pdOid b0396321-efb0-4c6d-992f-b81670b91b27 */
public void setCMaxSelect(int num) {
    // TODO: implement
}

/** @param cId
 * @param name
 * @param grade
 * @param tId
 * @param sId
 * @param intro
 * @param num

```

```

    * @pdOid 2320c0c3-15d9-4cb7-b82c-3d0f4b4205cf */
    public Course Course(String cId, String name, String grade, String[] tId, String[] sId, String intro, int num) {
        // TODO: implement
    }
}

```

## 代码 2-3 DropCourse.java

```

/*****
 * Module: DropCourse.java
 * Author: 40855
 * Purpose: Defines the Class DropCourse
 *****/

import java.util.*;

/** @pdOid ebc54f26-4b6b-4667-9faf-546cbb4f5962 */
public class DropCourse {
    /** @pdOid cb2dce36-fd19-470b-9379-9ce8d0e0b26b */
    private String sId;
    /** @pdOid 2e9881ed-293f-4c7e-9c2d-eeb9c18ae8d0 */
    private String cId;

    /** @param sId
     * @param cId
     * @pdOid 0498366f-4cea-488e-be79-ee9e9fd6577b */
    public DropCourse DropCourse(String sId, String cId) {
        // TODO: implement
    }
}

```

## 代码 2-4 Login.java

```

/*****
 * Module: Login.java
 * Author: 40855
 * Purpose: Defines the Class Login

```

```
*****/
```

```
import java.util.*;
```

```
/** @pdOid 43f29e59-6106-4fa7-9529-36a27697df06 */
```

```
public class Login {
```

```
    /** @pdOid 2ec39192-1067-44b2-b2ed-2eaffcc357ad */
```

```
    private String inputId;
```

```
    /** @pdOid 8e7c374f-f9e9-4e27-9157-a75d057d6d13 */
```

```
    private String inputPassword;
```

```
    /** @pdOid 7c925699-fcaa-47a1-ba4e-1018a16b9a58 */
```

```
    public String getInputId() {
```

```
        // TODO: implement
```

```
        return null;
```

```
    }
```

```
    /** @pdOid 5d6cb74c-24c0-47ce-b764-a293644e0d24 */
```

```
    public String getInputPassword() {
```

```
        // TODO: implement
```

```
        return null;
```

```
    }
```

```
    /** @param id
```

```
        * @pdOid dfa1dc5e-6d00-40e3-a818-98924e4f0677 */
```

```
    public void setInputId(String id) {
```

```
        // TODO: implement
```

```
    }
```

```
    /** @param pwd
```

```
        * @pdOid 59c39b42-7b3e-4f8b-a47d-8dc26552807d */
```

```
    public void setInputPassword(String pwd) {
```

```
        // TODO: implement
```

```
    }
```

```

/** @param id
 * @param pwd
 * @pdOid 079b61c0-b6c4-4005-a189-ff7f9f7699f9 */
public boolean checkLoginStatus(String id, String pwd) {
    // TODO: implement
    return false;
}
}

```

## 代码 2-5 SelectCourse.java

```

/*****
 * Module: SelectCourse.java
 * Author: 40855
 * Purpose: Defines the Class SelectCourse
 *****/

import java.util.*;

/** @pdOid 88179403-513d-4eaf-9891-088bcf93b088 */
public class SelectCourse {
    /** @pdOid fd87b09f-d3b7-4af8-b830-a9eb07bd6760 */
    private String sId;
    /** @pdOid ef8edbe7-bf3a-4edf-9762-c446c932a4bf */
    private String cId;
    /** @pdOid 06a40534-0039-440c-a4d6-384fb91d4c1c */
    private int weightValue;

    /** @pdOid 56fb2ac0-dad3-422e-a93d-ca0b6fbb9cd9 */
    private boolean checkWeight() {
        // TODO: implement
        return false;
    }

    /** @param sId
     * @param cId

```

```

    * @param weightValue
    * @pdOid 49c11be4-2364-4255-bc33-8c3d36749fe0 */
    public SelectCourse SelectCourse(String sId, String cId, int weightValue) {
        // TODO: implement
    }

    /** @param weightValue
    * @pdOid cf3e5082-f387-4026-b7a0-4d2805aac46a */
    public void setWeight(int weightValue) {
        // TODO: implement
    }
}

```

## 代码 2-6 Student.java

```

/*****
 * Module: Student.java
 * Author: 40855
 * Purpose: Defines the Class Student
 *****/

import java.util.*;

/** @pdOid d141d7a2-d541-4154-a712-f4b7877ea5ad */
public class Student extends User {
    /** @pdOid f0aafd35-f3f3-408f-9be9-3cea791d8e12 */
    private String sId;
    /** @pdOid 4a255b60-bec9-4d29-8435-5bdf2c1311bb */
    private String grade;

    /** @pdRoleInfo migr=no name=Course assc=Association_1 coll=java.util.Collection
    impl=java.util.HashSet mult=0..n */
    public java.util.Collection<Course> course;
    /** @pdRoleInfo migr=no name=SelectCourse assc=Association_6 coll=java.util.Collection
    impl=java.util.HashSet */
    public SelectCourse selectCourse;

```



```

/** @pdRoleInfo migr=no name=DropCourse assc=Association_5 coll=java.util.Collection
impl=java.util.HashSet */
public DropCourse dropCourse;

/** @pdOid ed3c8dda-3ed8-4c99-9b5d-40fed5b95fe4 */
public String getId() {
    // TODO: implement
    return null;
}

/** @pdOid ec442614-7e69-4d82-8a2b-bddac96f21c7 */
public String getGrade() {
    // TODO: implement
    return null;
}

/** @param id
 * @pdOid 35113315-a313-4281-adbb-3625cd7c22fa */
public void setId(String id) {
    // TODO: implement
}

/** @param grade
 * @pdOid 9bad51b1-4f09-4ac7-a3c0-3a96da9f4fbf */
public void setGrade(String grade) {
    // TODO: implement
}

/** @param id
 * @param grade
 * @pdOid fe968f56-c498-494b-9d2c-6b6e8adffce6 */
public Student Student(String id, String grade) {
    // TODO: implement
}

```

```
/** @pdGenerated default getter */
public java.util.Collection<Course> getCourse() {
    if (course == null)
        course = new java.util.HashSet<Course>();
    return course;
}

/** @pdGenerated default iterator getter */
public java.util.Iterator getIteratorCourse() {
    if (course == null)
        course = new java.util.HashSet<Course>();
    return course.iterator();
}

/** @pdGenerated default setter
    * @param newCourse */
public void setCourse(java.util.Collection<Course> newCourse) {
    removeAllCourse();
    for (java.util.Iterator iter = newCourse.iterator(); iter.hasNext();)
        addCourse((Course)iter.next());
}

/** @pdGenerated default add
    * @param newCourse */
public void addCourse(Course newCourse) {
    if (newCourse == null)
        return;
    if (this.course == null)
        this.course = new java.util.HashSet<Course>();
    if (!this.course.contains(newCourse))
        this.course.add(newCourse);
}

/** @pdGenerated default remove
```

```

    * @param oldCourse */
public void removeCourse(Course oldCourse) {
    if (oldCourse == null)
        return;
    if (this.course != null)
        if (this.course.contains(oldCourse))
            this.course.remove(oldCourse);
}

/** @pdGenerated default removeAll */
public void removeAllCourse() {
    if (course != null)
        course.clear();
}
}

```

代码 2-7 SystemAdmin.java

```

/*****
 * Module:  SystemAdmin.java
 * Author:  40855
 * Purpose: Defines the Class SystemAdmin
 *****/

import java.util.*;

/** @pdOid f6510b61-e20e-4b5d-92b0-204878beb0d1 */
public class SystemAdmin extends User {
    /** @pdOid 274beec3-efd7-431c-8b12-11fd044e167a */
    private String sAdminId;

    /** @pdOid 326847dc-8c8a-4aa2-8f2f-32093977e2bf */
    public String getSAdminId() {
        // TODO: implement
        return null;
    }
}

```

```

/** @param id
 * @pdOid d9df43cd-213e-4ff1-a971-9c9e60d72f93 */
public void setSAdminId(String id) {
    // TODO: implement
}

/** @param id
 * @pdOid 2ade2a74-2d0d-4549-b878-2402ebd1d239 */
public SystemAdmin SystemAdmin(String id) {
    // TODO: implement
}
}

```

## 代码 2-8 Teacher.java

```

/*****
 * Module: Teacher.java
 * Author: 40855
 * Purpose: Defines the Class Teacher
 *****/

import java.util.*;

/** @pdOid 835a09ca-f71b-4b5d-90b1-062cb5f9516f */
public class Teacher extends User {
    /** @pdOid ea7765db-c77e-4760-8ea9-31f13938215b */
    private String tId;
    /** @pdOid c6ae4cf2-9603-4b58-a58c-19522a4722ca */
    private String intro;

    /** @pdRoleInfo migr=no name=Course assc=Association_2 coll=java.util.Collection
impl=java.util.HashSet mult=0..n */
    public java.util.Collection<Course> course;

    /** @pdOid 440c70e4-cf10-42ea-a410-f3e3112b2478 */

```

```
public String getTid() {
    // TODO: implement
    return null;
}

/** @pdOid fc4ab699-323f-4c34-bc8f-6770bbf2d5e0 */
public String getIntro() {
    // TODO: implement
    return null;
}

/** @param id
 * @pdOid 6c0c6c55-cc1e-4014-804e-108eab78ac97 */
public void setTid(String id) {
    // TODO: implement
}

/** @param intro
 * @pdOid 5bb9460d-2c4d-4be9-ad08-409757eb196a */
public void setIntro(String intro) {
    // TODO: implement
}

/** @param id
 * @param intro
 * @pdOid fb5c2625-a622-4999-bf45-81a957a5d56e */
public Teacher Teacher(String id, String intro) {
    // TODO: implement
}

/** @pdGenerated default getter */
public java.util.Collection<Course> getCourse() {
    if (course == null)
        course = new java.util.HashSet<Course>();
}
```

```
        return course;
    }

    /** @pdGenerated default iterator getter */
    public java.util.Iterator getIteratorCourse() {
        if (course == null)
            course = new java.util.HashSet<Course>();
        return course.iterator();
    }

    /** @pdGenerated default setter
     * @param newCourse */
    public void setCourse(java.util.Collection<Course> newCourse) {
        removeAllCourse();
        for (java.util.Iterator iter = newCourse.iterator(); iter.hasNext();)
            addCourse((Course)iter.next());
    }

    /** @pdGenerated default add
     * @param newCourse */
    public void addCourse(Course newCourse) {
        if (newCourse == null)
            return;
        if (this.course == null)
            this.course = new java.util.HashSet<Course>();
        if (!this.course.contains(newCourse))
            this.course.add(newCourse);
    }

    /** @pdGenerated default remove
     * @param oldCourse */
    public void removeCourse(Course oldCourse) {
        if (oldCourse == null)
            return;
        if (this.course != null)
```

```

        if (this.course.contains(oldCourse))
            this.course.remove(oldCourse);
    }

    /** @pdGenerated default removeAll */
    public void removeAllCourse() {
        if (course != null)
            course.clear();
    }
}

```

## 代码 2-9 User.java

```

/*****
 * Module:   User.java
 * Author:   40855
 * Purpose:  Defines the Class User
 *****/

import java.util.*;

/** @pdOid 7ec39155-dbd4-4b95-b9b9-3b3c988e8cbf */
public class User {
    /** @pdOid f83208d4-e952-4773-9000-939c7a3e6e3e */
    protected String name;
    /** @pdOid 57ad67ef-a717-4030-9350-e8e1bb7092a6 */
    protected String password;

    /** @pdOid 536ec7f1-0f87-43a8-a94c-f6dc441f0bcf */
    public String getName() {
        // TODO: implement
        return null;
    }

    /** @pdOid 4885ee40-283f-4750-a618-14db17024ccc */
    public String getPassword() {

```

```
// TODO: implement
return null;
}

/** @param name
 * @pdOid 956032af-dbc9-4ee6-af38-7d9c6787b143 */
public void setName(String name) {
    // TODO: implement
}

/** @param pwd
 * @pdOid 2b2cd7bd-436a-4587-8f5e-f6e6a6879959 */
public void setPassword(String pwd) {
    // TODO: implement
}

/** @param name
 * @param pwd
 * @pdOid 2a38ada4-6a83-4dd8-b824-287420be218e */
public User User(String name, String pwd) {
    // TODO: implement
}
}
```



## 九、总结、改进建议及心得体会

// removed