学生学籍管理系统报告

题目：学生学籍管理系统

姓名：刘忠坤

学号：201712103

班级：软件工程1704班

完成日期：2019.12.28

指导老师：李天俐

目录

**[一、 系统概述 3](#_Toc27166)**

**[二、 需求分析 3](#_Toc14796)**

[2.1 用户需求 3](#_Toc2357)

[2.2系统功能模块结构图 4](#_Toc18892)

[2.3系统运行效果 4](#_Toc32003)

**[三、 用例建模 7](#_Toc16413)**

**[3.1确定系统边界 7](#_Toc28035)**

[3.2识别参与者 7](#_Toc6755)

[3.3识别用例 7](#_Toc26875)

[3.4绘制用例图 8](#_Toc24632)

[3.5 用例描述 9](#_Toc26978)

**[四、 静态建模 14](#_Toc31091)**

[4.1 识别对象类 14](#_Toc7682)

[4.2识别属性 14](#_Toc27580)

[4.3绘制类图 14](#_Toc25415)

[4.4设计数据表 14](#_Toc25007)

**[五、 动态建模 15](#_Toc25039)**

[5.1绘制顺序图 15](#_Toc12311)

[5.2绘制协作图 17](#_Toc22331)

[5.3绘制活动图 18](#_Toc29195)

[5.4绘制状态图 21](#_Toc24150)

**[六、 架构建模 22](#_Toc23286)**

[6.1绘制包图 22](#_Toc31105)

[6.2绘制组建图 23](#_Toc5581)

[6.3绘制部署图 24](#_Toc31120)

**[七、 总结 24](#_Toc767)**

1. **系统概述**

学生的学籍是每个学生拥有的重要信息，为实现更加方便地管理学生学籍

信息，我们开发了学生学籍管理系统。系统可以像纸质文件一样便于观看，更重要的是该系统能够更快速的管理学生的学籍信息，更准确的查找相关信息。

Struts2框架提供了基于Java技术开发Web应用系统的一种方案，它属于基于系统体系结构的开发方法。Struts2框架要求具备以下开发环境。

1. Java语言基础，如jdk1.6。
2. Java技术开发环境，如MyEclipse10。
3. Web服务器，如tomcat6.0。
4. 数据库管理系统，如MySql5.0。
5. Struts2技术框架，如struts-2.3.16。
6. B/S系统的开发技术，如JSP、Servlet、JavaBean、XML、HTML.

本案例是取自一个可运行的基于Struts2技术开发的学生学籍管理系统。

1. **需求分析**

本系统基于B/S设计，应用系统可以发布到网上，用户通过浏览器随时操作学生学籍管理系统。

## 2.1 用户需求

通过学生学籍管理系统，用户可以在系统中任意添加、修改、删除学生学籍数据，包括基本信息管理、成绩信息管理、查询信息管理。要实现的功能包括以下4个方面。

1. **登录与注册：**

没有账号的用户，必须通过系统注册一个账号，所有希望通过系统管理学籍信息的用户，必须使用账号登录系统。

1. **学生学籍管理：**

将学生信息存储于系统中，管理员可以管理学生基本信息，如查询学生基本信息、添加新学生基本信息、修改和删除学生基本信息等；

1. **成绩管理：**

将成绩信息存储于系统中，管理员可以添加成绩、修改成绩和删除成绩，并可以查阅总体成绩的情况；

1. **信息查询：**

管理员可以查询特定的学生信息和成绩的大体情况。

**2.2系统功能模块结构图**

根据用户需求，分析与设计了5个功能模块，它们之间的关系如图2.1所示。

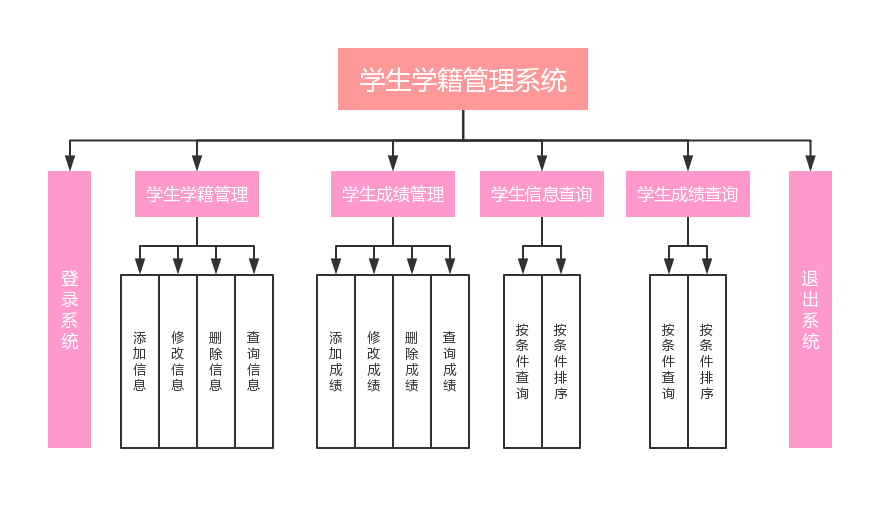


图2.1 系统功能模块结构图

**2.3系统运行效果**

开发的原型系统，最终的运行效果如图2.2~2.7所示



图2.2 系统登录界面



图2.3 系统登录后的主界面



图2.4 查看学生学籍信息界面



图2.5 增加学生学籍信息界面



图2.6 修改学生学籍信息界面



图2.7 增加学生成绩信息

1. **用例建模**

用例建模主要完成用例模型的分析与设计。用例模型主要由用例、用例描述、用例组成。用例模型用来描述从系统的外部用户（即参与者）观看系统应该具备哪些功能。

# **3.1确定系统边界**

在进行用例分析和用例建模之前，必须要先确定系统边界。系统边界就是讲系统的功能特性与系统的外部环境分离开来的逻辑边界线。本系统的全部功能是管理学生学籍信息、管理学生成绩信息。

**3.2识别参与者**

识别出系统的角色。角色可以是用户、外部系统，甚至是外部处理，通过某种途径与系统交互。重要的是着重从系统外部执行者的角度来描述系统需要提供哪些功能，并指明这些功能的执行者（角色）是谁。尽可能地确保所有角色都被完全识别出来。

**3.3识别用例**

确定系统边界和参与者后，要找出本系统的参与者可能会执行哪些任务就非常容易了。显然，系统的用户需要注册账号、登录系统、管理学生学籍信息、管理学生成绩信息，查询信息等。这些操作都是由用户发起的用例。经过分析，可以得到以下用例。

1. 登录
2. 注册
3. 学生学籍信息管理
   1. 增加学生学籍信息
   2. 修改学生学籍信息
   3. 查看学生学籍信息
   4. 删除学生学籍信息
4. 学生成绩信息管理
   1. 增加学生成绩信息
   2. 修改学生成绩信息
   3. 查看学生成绩信息
   4. 删除学生成绩信息

**3.4绘制用例图**

只有正确地分析了用例之间的关系，才能正确建立用例模型。

**1.“登录”用例与“注册”用例之间的关系**

有账号的用户可以直接登录系统，也可以作为新用户再次申请一个新账号。因此，用户与“登录”用例和“注册”用例各自存在直接的通信关系。

无账号的用户可以直接申请一个新账号,但是,如果无账号的用户直接登录系统，则拒绝他登录，转而导向注册操作界面，督促他注册一个新号,因此,“登录”用例和“注册”用例之间存在扩展关系,“登录”用例是基本用例,“注册”用例是扩展用例。

**2.“学生学籍信息管理”用例与它的3 个子用例“增加学生学籍信息”、“修改学生学籍信息”、“查看学生学籍信息”之间的关系**

用户在办公室信息管理操作界面可以修改和查看办公室信息。具体操作哪之个间功的能关由系用户根据需要决定,因此,“学生学籍信息管理”用例与“修改学生学籍信息”、“查看学生学籍信息”、“增加学生学籍信息”之间分别存在扩展关系。为了更准确地表达出用户操作子功能的逻辑，可以为“学生学籍信息管理”用例与它的3个子用例之间设计版型《use》关系，表示“学生学籍信息管理”用例使用某个子用例完成用户选择的子功能操作。

**3.“学生成绩信息管理”用例与它的3 个子用例“增加学生成绩信息”、“修改学生成绩信息”、“查看学生成绩信息”之间的关系。**

参照“学生学籍信息管理”用例与它的3 个子用例之间的关系的确定方法，也可以设计成《use》关系。

根据上述分析,设计的该系统的用例图如图 2.8 所示。



图2.8 系统的用例图

**3.5 用例描述**

用例图只能反映出用例之间的关系。但是,每个用例具体完成什么任务，从用例图上是看不出来的。必须依靠每个用例的用例描述进行说明。下面给出每个用例的用例描述。

1．登录用例

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 |  |
| 主成功场景 | 1. 用户访问学生学籍管理系统 2. 系统显示登录页面 3. 用户录入登录信息（账号，密码），点击登录按钮 4. 系统验证账号密码信息是否正确 5. 系统显示系统主页 |
| 扩展1 | 4a、录入信息格式错误  4a1、系统重新显示登录页面，并提示错误  错误提示：“账号格式不正确”，“密码格式不正确”  4a2、回到步骤3 |
| 扩展2 | 4b、登录验证失败  4b1、系统重新显示登录页面，并提示错误  错误提示：“账号不存在”，“密码错误”  4b2、回到步骤3 |
| 备注 |  |

1. 修改密码用例

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 登录完成 |
| 主成功场景 | 1. 用户访问修改密码页面 2. 系统显示修改密码页面 3. 用户录入信息（新密码，确认新密码），点击修改密码按钮 4. 系统验证密码信息是否符合输入条件 5. 修改成功系统提示：“密码修改成功” |
| 扩展1 | 4a、密码不能为空  4a1、系统重新显示修改密码页面，并提示错误  错误提示：“密码不能为空”  4a2、回到步骤3 |
| 扩展2 | 4b、密码和确认密码不一致  4b1、系统重新显示修改密码页面，并提示错误  错误提示：“新密码和确认密码不一致”  4b2、回到步骤3 |
| 备注 |  |

1. 增加学生基本信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 登录完成 |
| 主成功场景 | 1. 管理员访问增加基本信息页面 2. 系统显示增加基本信息页面 3. 管理员录入学生基本信息（学生的学号,姓名,性别,班级,专业,民族,出生日期,院系名称,专业名称,班级名称），点击增加按钮 4. 系统验证新增学生基本信息是否符合输入条件 5. 系统保存新学生基本信息，并显示增加成功页面 |
| 扩展1 | 4a、填写的学生基本信息学生的学号,姓名,性别,班级,专业,民族,出生日期,院系名称,专业名称,班级名称)不符合要求  4a1、系统重新显示增加学生基本信息页面，并提示相应输入域的格式要求  4a2、回到步骤3 |
| 扩展2 | 4b、新增加的学号已经存在  4b1、系统重新显示增加基本信息页面并提示错误:  “此学号XX已经存在”  4b2、回到步骤5 |
| 备注 |  |

1. 查询学生基本信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 登录完成 |
| 主成功场景 | 1. 用户访问查询学生基本信息页面 2. 系统显示查询学生基本信息页面 3. 用户录入需要查询的信息（学生的学号,姓名,性别,班级,专业,民族,出生日期,院系名称,专业名称,班级名称），点击查询按钮 4. 系统验证查询的学生基本信息是否符合输入条件 5. 系统显示查询到的学生基本信息 |
| 扩展1 | 4a、没有查询到匹配的学生基本信息  4a1、系统显示“没有匹配的学生基本信息”  4a2、回到步骤2 |
| 备注 |  |

1. 修改学生基本信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 用户成功查询到学生基本信息,并选中需要修改的信息 |
| 主成功场景 | 1. 用户访问修改基本信息页面 2. 系统显示修改基本信息页面 3. 用户将需要修改的内容输入对应的文本框中，并点击修改按钮 4. 系统验证修改的学生基本信息是否符合输入条件 5. 系统显示用户信息修改成功页面 |
| 扩展1 | 4a、输入的信息为空  4a1、系统重新显示修改基本信息页面并提示错误：  “XXX不能为空”  4a2、回到步骤2 |
| 扩展2 | 4b、输入的信息不符合要求  4b1、系统重新显示修改基本信息页面并提示错误：  “XXX不符合要求”  4b2、回到步骤2 |
| 备注 |  |

1. 删除学生基本信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 用户成功查询到学生基本信息，并选中需要删除的信息 |
| 主成功场景 | 1. 用户访问删除基本信息页面 2. 系统显示删除基本信息页面 3. 用户确认需要删除信息的内容，并点击删除按钮 4. 系统验证要删除的学生基本信息是否符合输入条件 5. 系统显示删除成功页面 |
| 扩展1 | 4a、输入的信息不存在  4a1、系统重新显示删除基本信息页面并提示错误：  “XXX不存在”  4a2、回到步骤2 |
| 扩展2 | 4b、输入的信息不符合要求  4b1、系统重新显示删除基本信息页面并提示错误：  “XXX不符合要求”  4b2、回到步骤2 |
| 备注 |  |

1. 增加学生成绩信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 登录完成 |
| 主成功场景 | 1. 管理员访问增加成绩信息页面 2. 系统显示增加成绩信息页面 3. 管理员录入学生成绩信息（学号,课程名称,学期,成绩,学分），点击增加按钮 4. 系统验证新增学生成绩信息是否符合输入条件 5. 系统计算总成绩并保存学生成绩信息 6. 系统显示学生成绩增加成功页面 |
| 扩展1 | 4a、填写的学号、成绩等信息不符合要求  4a1、系统重新显示增加成绩信息页面，并提示相应输入域的格式要求  4a2、回到步骤3 |
| 扩展2 | 4b、新增加的学号已经存在  4b1、系统重新显示增加基本信息页面并提示错误:  “此学号XX已经存在”  4b2、回到步骤5 |
| 备注 | 学生成绩信息的总成绩由服务器计算填入 |

1. 查询学生成绩信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 登录完成 |
| 主成功场景 | 1. 用户访问查询成绩信息页面 2. 系统显示查询成绩信息页面 3. 用户录入信息（学生的学号，或者某个学科的排名,某个学期），并点击查询按钮 4. 系统验证查询的信息是否符合输入条件 5. 系统显示查询到的学生成绩信息 |
| 扩展1 | 4a、没有查询到匹配的成绩信息  4a1、系统显示“没有匹配的成绩信息”  4a2、回到步骤2 |
| 备注 |  |

1. 修改学生成绩信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 用户成功查询到学生成绩信息,并选中需要修改的信息 |
| 主成功场景 | 1. 用户访问修改成绩信息页面 2. 系统显示修改成绩信息页面 3. 用户将需要修改的内容输入对应的文本框中，并点击修改按钮 4. 系统验证修改的学生成绩信息是否符合输入条件 5. 系统显示用户信息修改成功页面 |
| 扩展1 | 4a、输入的信息为空  4a1、系统重新显示修改成绩信息页面并提示错误：  “XXX不能为空”  4a2、回到步骤2 |
| 扩展2 | 4b、输入的信息不符合要求  4b1、系统重新显示修改成绩信息页面并提示错误：  “XXX不符合要求”  4b2、回到步骤2 |
| 备注 |  |

1. 删除学生成绩信息

|  |  |
| --- | --- |
| 执行者 | 管理员 |
| 前置条件 | 管理员成功查询到学生成绩信息,并选中需要删除的信息 |
| 主成功场景 | 1. 管理员访问删除成绩信息页面 2. 系统显示删除成绩信息页面 3. 管理员确认需要删除信息的内容，并点击删除按钮 4. 系统验证要删除的学生成绩信息是否符合输入条件 5. 系统显示删除成功页面 |
| 扩展1 | 4a、输入的信息不存在  4a1、系统重新显示删除成绩信息页面并提示错误：  “XXX不存在”  4a2、回到步骤2 |
| 扩展2 | 4b、输入的信息不符合要求  4b1、系统重新显示删除成绩信息页面并提示错误：  “XXX不符合要求”  4b2、回到步骤2 |
| 备注 |  |

1. **静态建模**

**4.1 识别对象类**

根据“管理员管理”用例处理的信息来看，都是与管理员基本情况有关的信息。因此可以考虑建立一个“管理员”类，用于处理管理员基本情况。

根据“学生信息管理”用例处理的信息来看，都是与学生基本情况有关的信息。因此可以考虑建立一个“学生”类，用于处理学生基本情况。

**4.2识别属性**

1. 管理员类（admin）:管理员名(accounts)、管理员密码(password)。
2. 学生信息类（students）:学生学号(number)、姓名(Name)、性别(gender)、民族(nation)、 血型(bloodtype)、出生日期(birthday)、院系(yxname)、专业(zyname)、班级(bjname)
3. 学生成绩类(stuscore):学生学号(number)、课程名称(kcname)、学期(term)、成绩(score)、学分(credit)

**4.3绘制类图**

确定的实体类之间的关系如图18所示

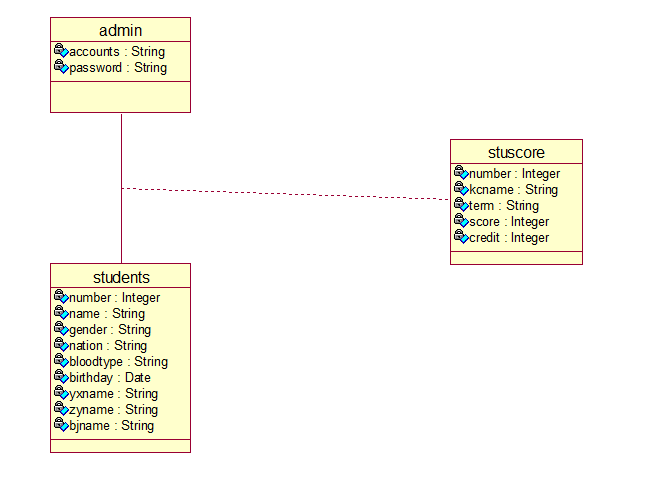


图18

**4.4设计数据表**

本系统使用MySql5.0。图20-图22为设计的数据表的结构。

表admin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 长度 | 备注 |
| accounts | 字符串 | 否 | 20 | 主键，账号 |
| password | 字符串 | 否 | 20 | 密码 |

表students

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 长度 | 备注 |
| number | 整型 | 否 | 11 | 主键，学号 |
| name | 字符串 | 是 | 20 | 姓名 |
| gender | 字符串 | 是 | 20 | 性别 |
| nation | 字符串 | 是 | 20 | 民族 |
| bloodtype | 字符串 | 是 | 20 | 血型 |
| birthday | 日期 | 是 |  | 出生日期 |
| yxname | 字符串 | 是 | 20 | 院系名称 |
| zyname | 字符串 | 是 | 20 | 专业名称 |
| bjname | 字符串 | 是 | 20 | 班级名称 |

表stuscore

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否为空 | 长度 | 备注 |
| number | 整型 | 否 | 11 | 学号 |
| kcname | 字符串 | 是 | 20 | 课程名称 |
| term | 字符串 | 是 | 20 | 学期 |
| score | 整型 | 是 | 11 | 成绩 |
| credit | 整型 | 是 | 11 | 学分 |

1. **动态建模**

**5.1绘制顺序图**

描述相互合作的对象之间的交互关系和连接关系。

强调时间和顺序。

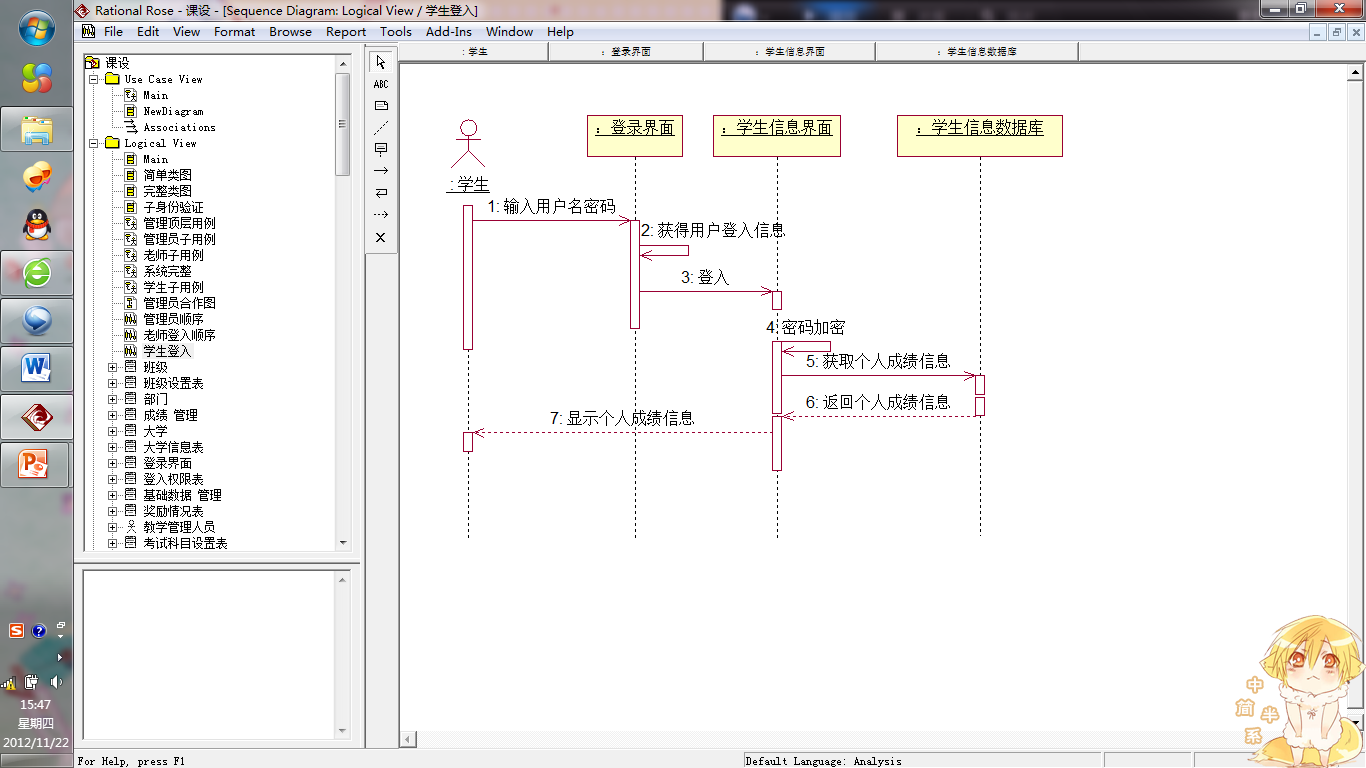


图3-8系统中用例学生查询成绩等信息顺序图

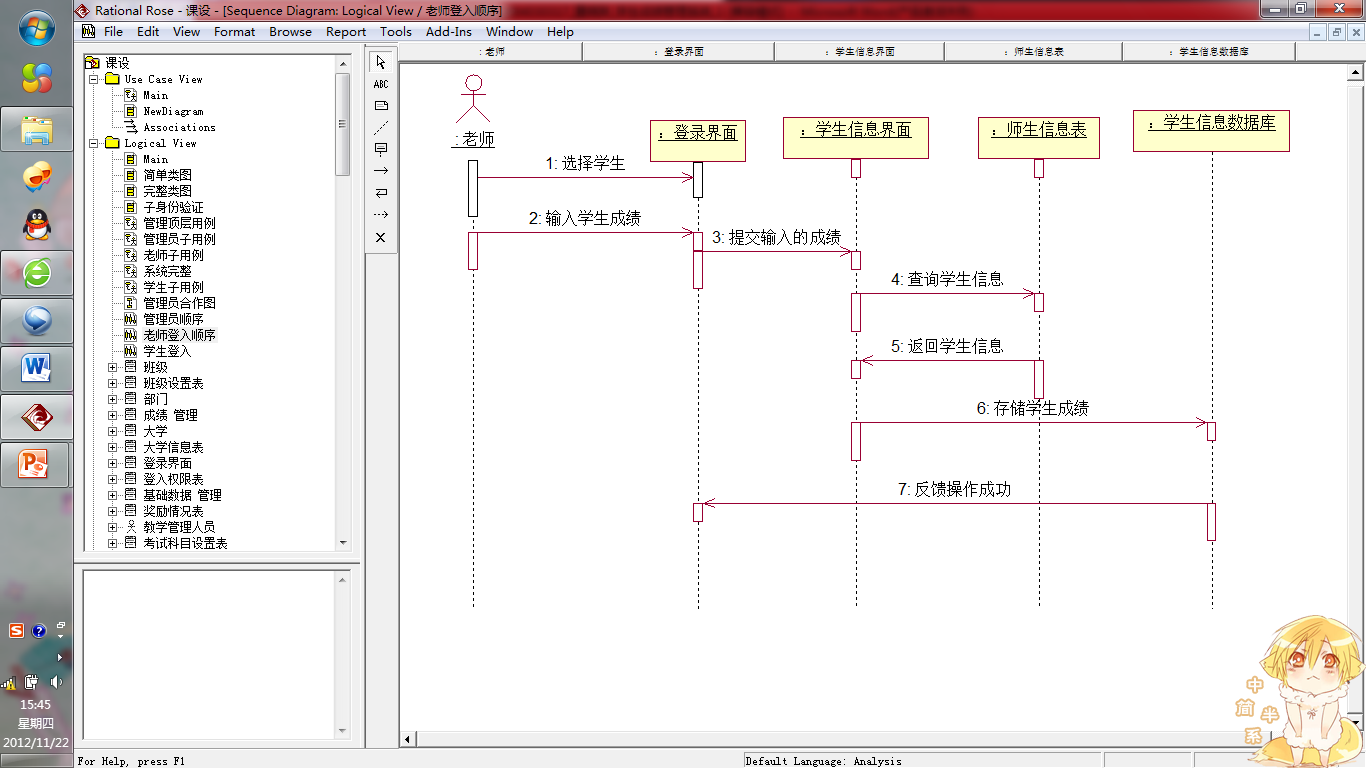


图3-9 系统中用例老师录入成绩等信息顺序图

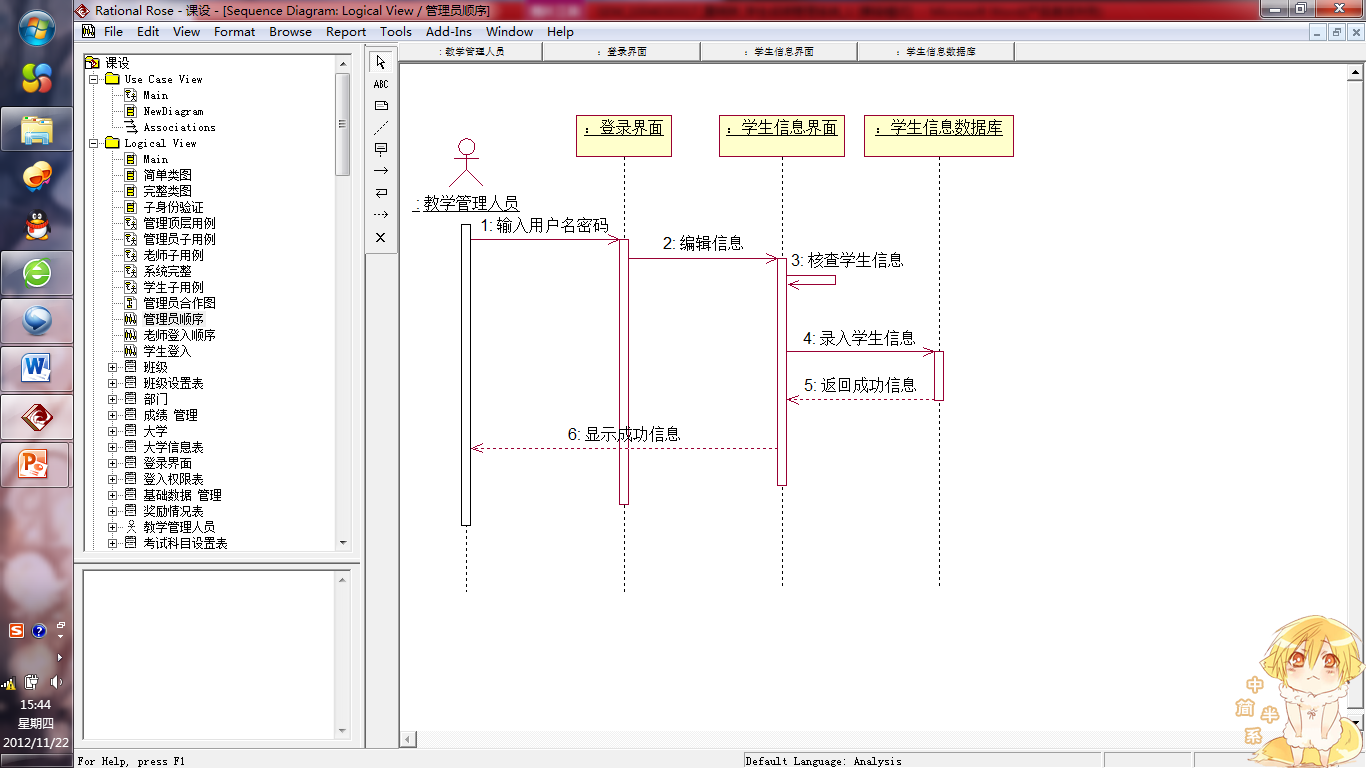


图3-10系统中用例教学管理人员修改信息顺序图

**5.2绘制协作图**

描述相互合作的对象之间的交互关系和连接关系。

强调对象间的相互关系。

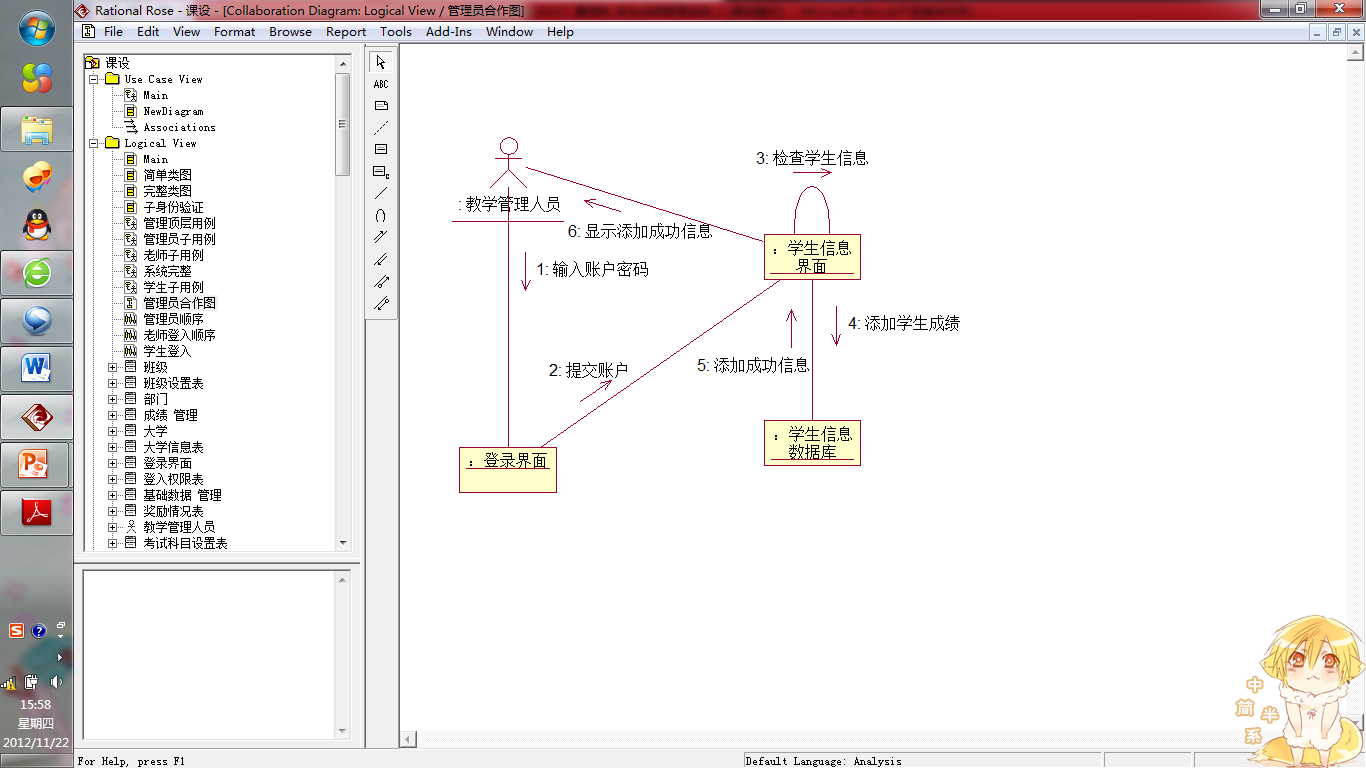


图3-11教学管理员添加学生信息合作图

**5.3绘制活动图**

描述系统中各种活动的执行顺序，刻画一个方法中所要进行的各项活动的流程。

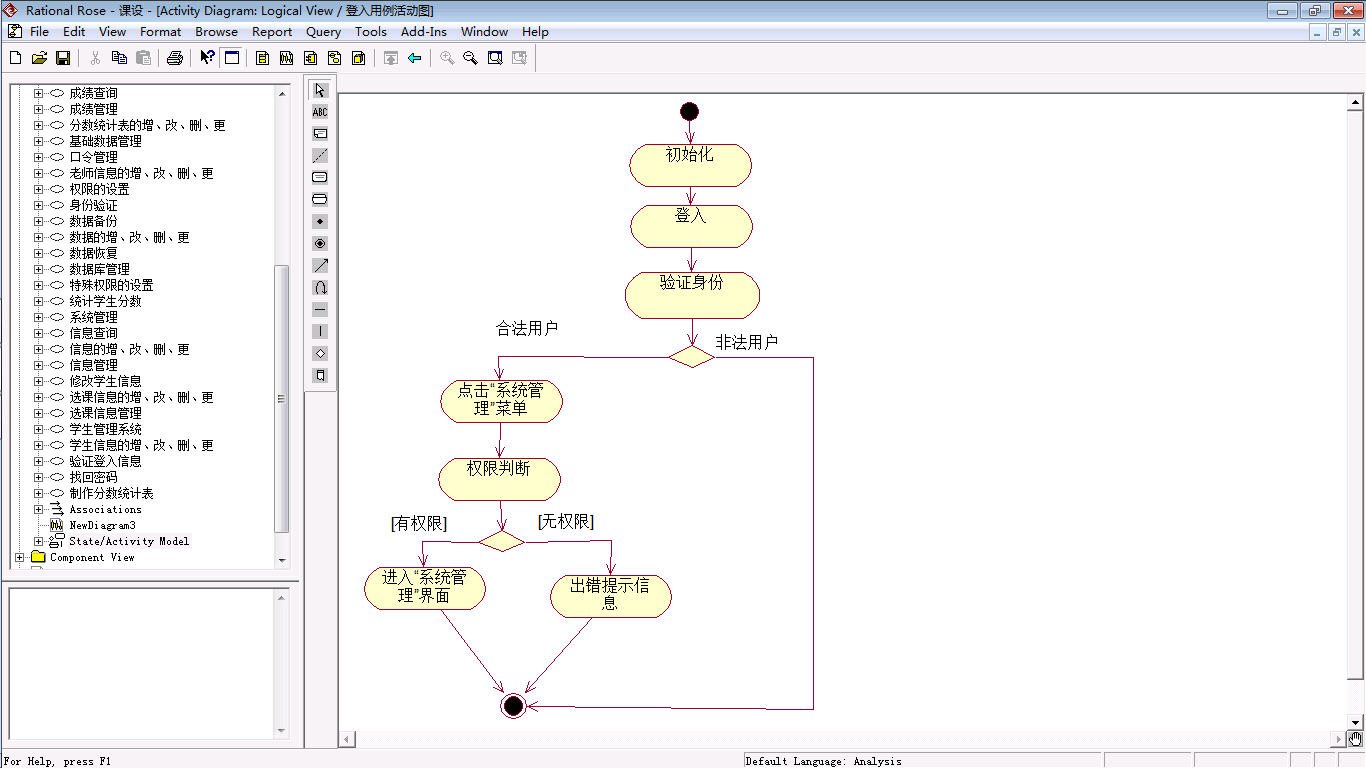


图3-13学生成绩管理系统中登入活动图

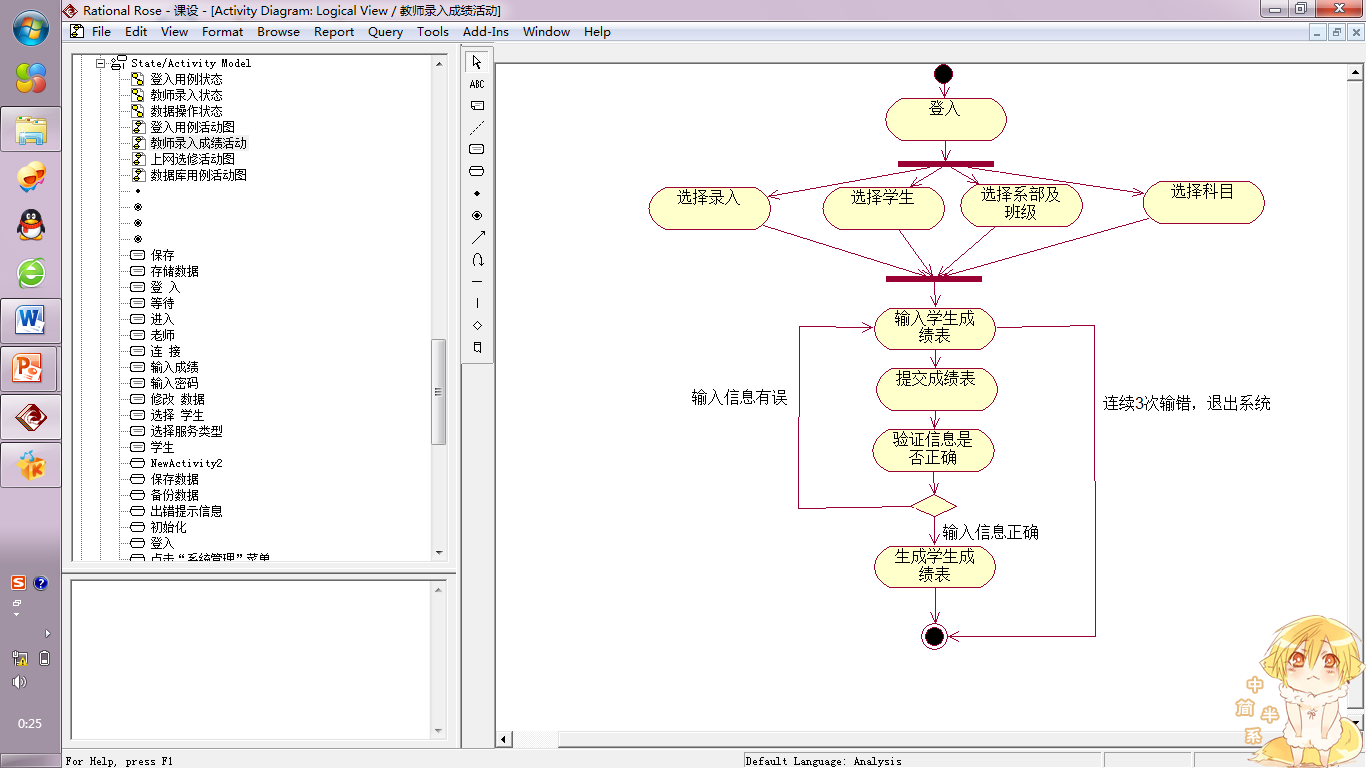


图3-14学生成绩管理系统中教师录入成绩活动图

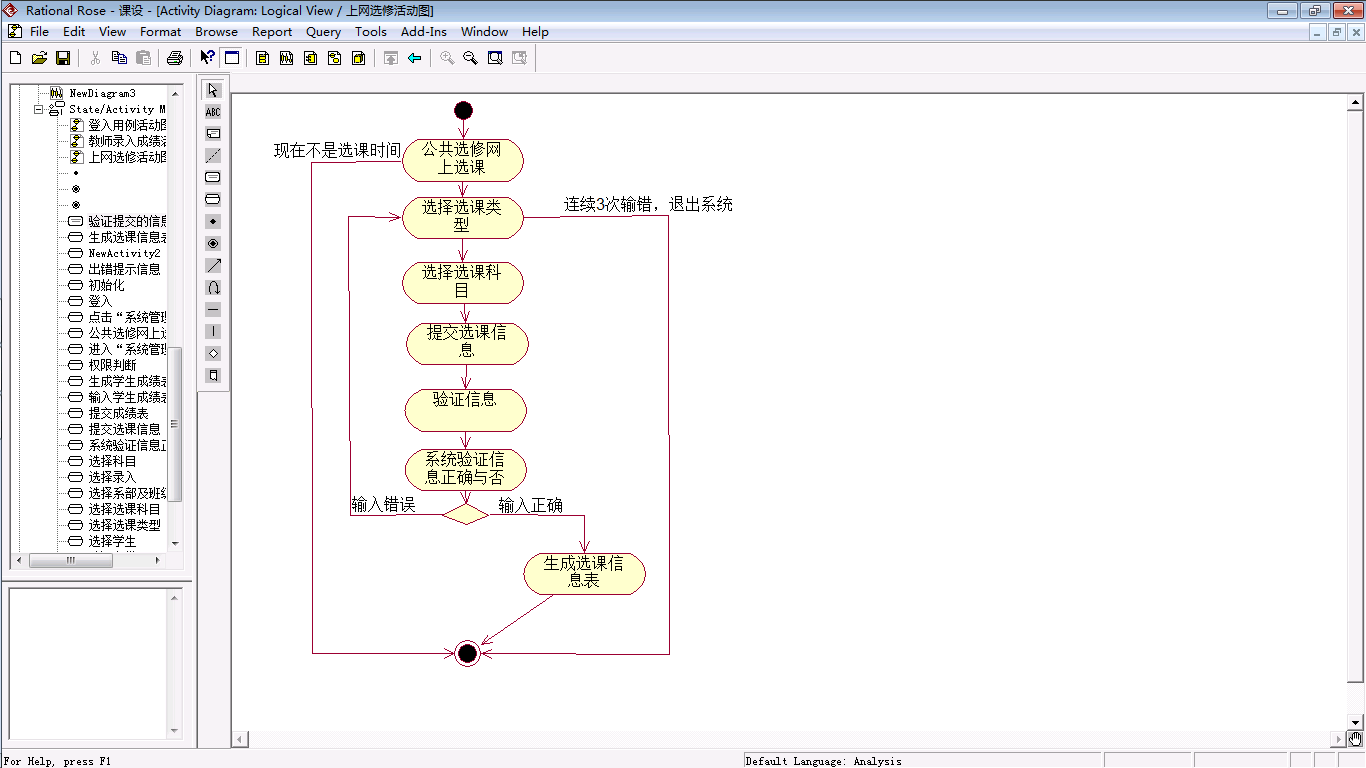


图3-15学生成绩管理系统中网上选修科目活动图

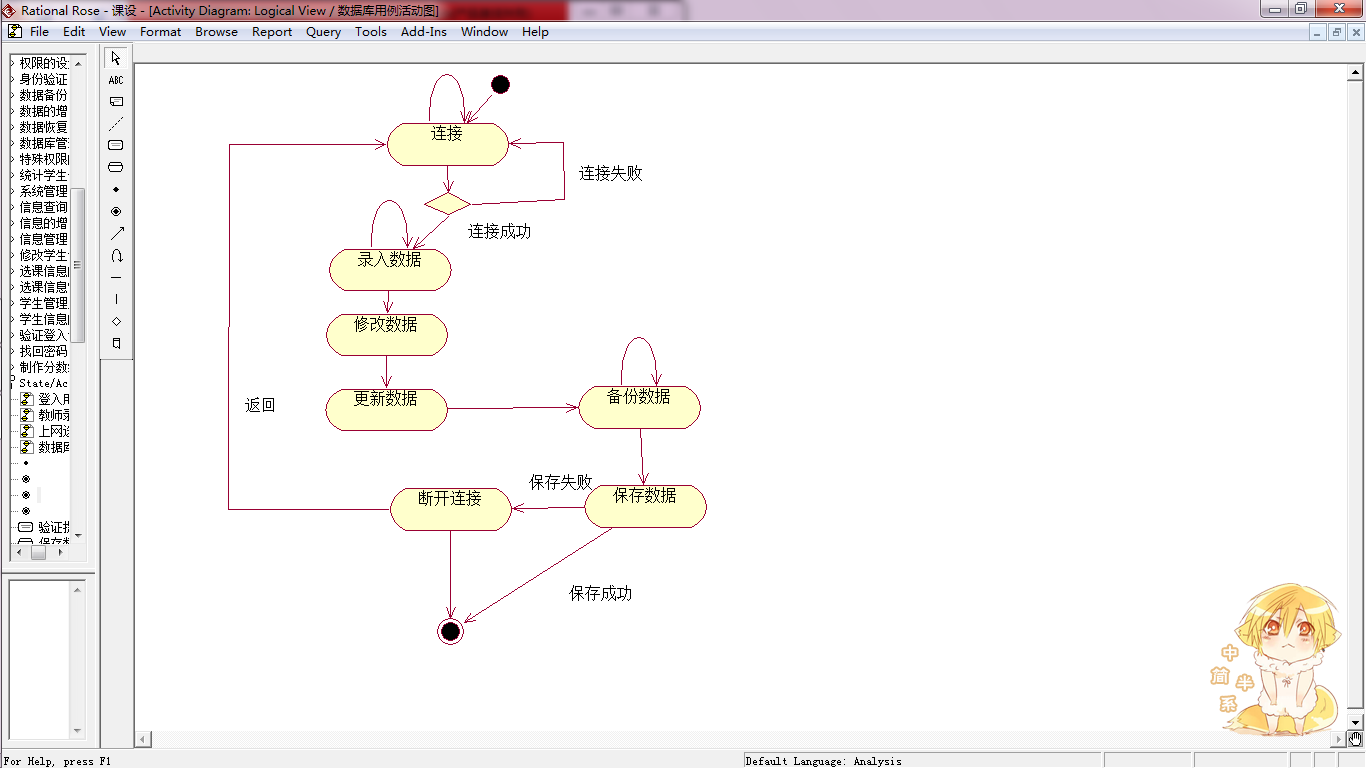


图3-16 系统中数据库操作用例活动图

**5.4绘制状态图**

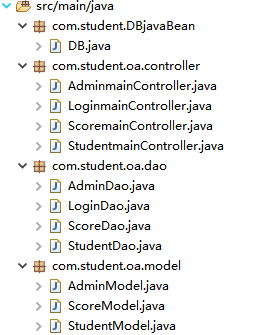
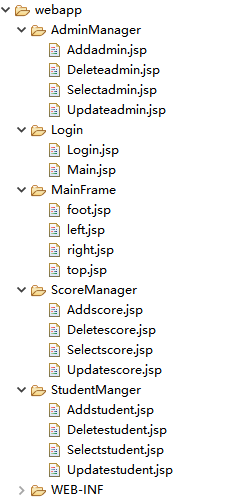
描述特定对象的所有可能的状态及其引起的状态迁移



图3-12学生成绩管理系统中数据操作状态图

1. **架构建模**

**6.1绘制包图**



1. JSP文件的目录结构图 (b)Java文件的目录结构图

在图(a)中，描述了JSP文件存放的目录结构形式。Webapp是一个根包，下面有6个一级子包，分别是AdminManager、ScoreManager、StudentManager、Login、MainFrame、WEB-INF。

在图(b)中，描述了Java文件存放的目录结构形式。src/main/java是一个根包，下面有4个一级子包，分别是com.student.oa.controller、com.student.oa.dao、com.student.oa.model、com.student.DBJavaBean。

此外，整个系统需要一个包StudentSystem,Webapp包和src/main/java包是StudentSystem的一级子包。根据上面的分析可以获得包的嵌套关系。如下图所示：

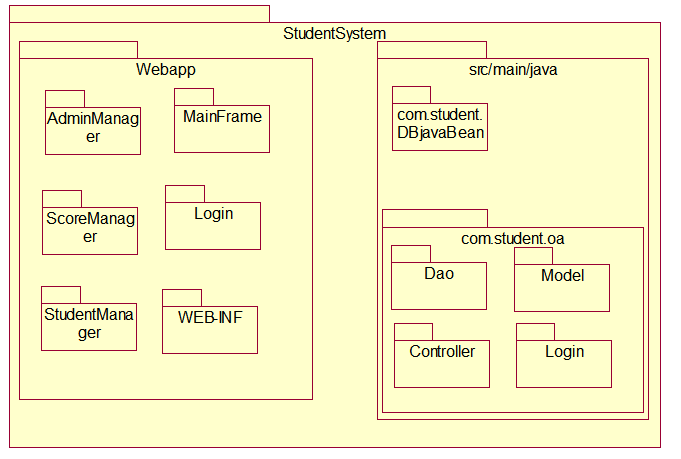


图41

包之间的关系如图所示：

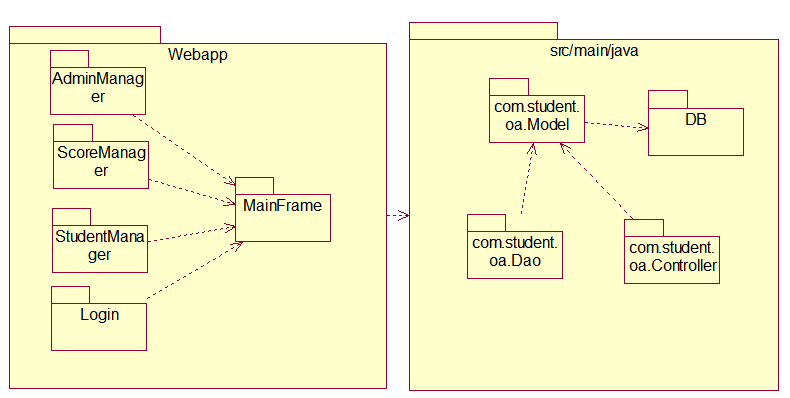


图42

**6.2绘制组建图**

各组件之间是否含有依赖关系，取决于各组件所含的JSP文件或者Java文件之间是否含有依赖关系。组建图如图所示：



图43

**6.3绘制部署图**

部署图描述了位于结点实力上的运行组件实例的安排，描述系统的实际物理结构。一个软件系统只有一个部署图，部署图常用于帮助理解分布式系统。如图44为本系统的部署图。

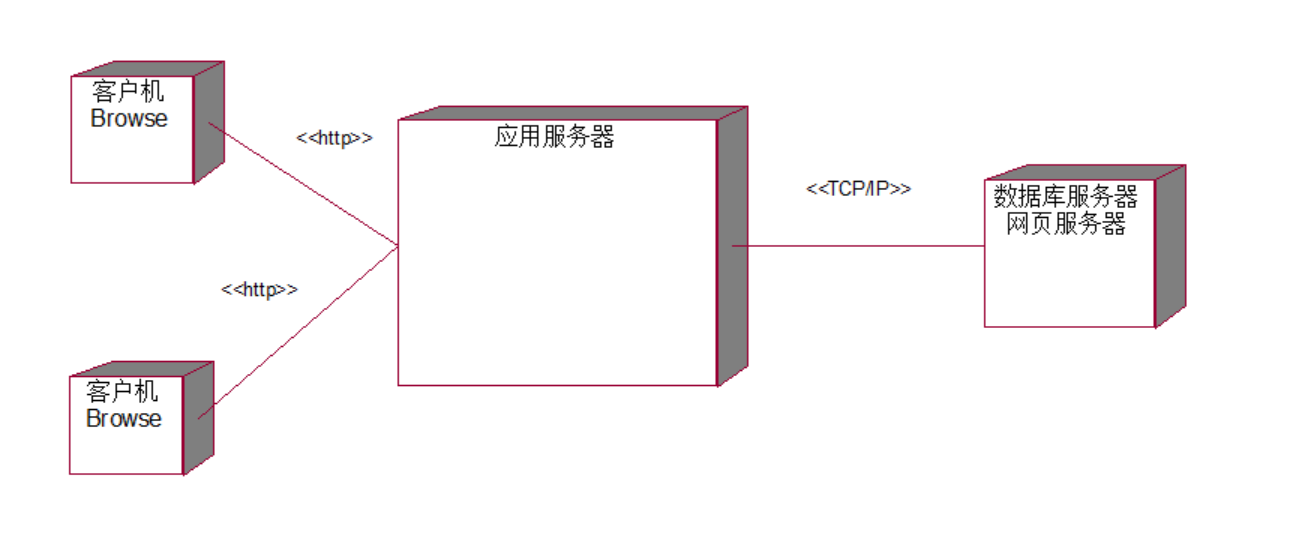


图44

1. **总结**

通过这一学期的学习，我了解到UML是一种功能强大的,面向对象的可视化系统分析的建模语言,它的各个模型可以帮助开发人员更好地理解业务流程,建立更可靠,更完善的系统模型.从而使用户和开发人员对问题的描述达到相同的理解,以减少语义差异,保障分析的正确性。理解了用例图、类图、顺序图、协作图、活动图、状态图、包图、组建图、部署图等图的概念与意义，帮助我更好的理解了软件开发流程。