

《J2EE框架与项目开发》课程设计报告

2016 - 2017 学年 第 2 学期

|  |  |
| --- | --- |
| 项目名称 | 在线考试系统 |
| 姓 名 | 张立飞、胡晓天 |
| 学 号 | 092214109、092214119 |
| 班 级 | 软件工程（单招）141 |
| 实验地点 | 宿舍 |
| 指导教师 | 周荣 |

计算机科学与工程学院

2017

**目录**

1. 设计目的和任务

1.1项目的目标

1.2项目的任务

2. 开发环境

2.1 硬件环境

2.2 软件环境

3.设计题目

3.1 题目详细描述

3.2 功能要求

4. 设计与实现

4.1设计部分

4.2实现部分

5. 总结

6. 参考资料

7. 附录

1. 设计目的和任务

1.1项目的目标：

（1）掌握Struts1.x的基本框架

（2）掌握工厂设计模式的应用

（3）掌握MVC架构

（4）提高解决问题的能力

1.2项目的任务：

完成在线考试系统，前台用bootstrap实现。 后台用Struts1.x实现。设计并实现一款在线考试系统。

**具有以下功能：**

1、支持多提醒：单选、多选、判断

2、可以设置考试开始时间，即未到时间不开始考试

3、可以设置考试时长，即考试时可以计时

4、考试成员可以通过数据库管理软件导入

5、考试题目再考试前按规定随机抽取

6、考试系统具有防止刷新功能和倒计时功能

7、考试时间结束后自动交卷

8、系统能自动统计每个学生的得分

9、可以设置每场考试的考试信息

1. 开发环境

2.1硬件环境

Cpu : Intel Core i7 4700HQ @2.40GHZ

Mainboard : ASUSTek COMPUTER INC X450JN

Memory : DDR3 8GBytes

Graphics : Intel HD Graphics 4600

Graphics : NVIDIA GeForce 840M

2.2软件环境

Coding IDE : IntelliJ IDEA 2017.3

Java SDK : 8u111

Apache : apache-tomcat 9.0.0.M13

Database : mysql – 5.6.24

Design : starUml v2.80

1. 设计题目

3.1题目详细描述

1. 项目名称为网络考试系统
2. 软件开发形式使用Struts或者Spring进行。软件必须考虑外观美观，并考虑设备适应性，建议使用bootstrap框架.
3. 每个项目可根据自己的能力，在完成基本功能的基础上，进行适当的功能扩展（在项目报告中要提现出来）。

3.2功能要求

官方:

设计一个在线考试系统（支持单选，多选，判断和填空题型，假设每个填空题只有一空且只有一个正确答案），要求：在系统中可设置一次考试的考试开始时间和考试时长，该场考试的考生名单可以通过数据库管理软件导入，每个学生的题目在考试前按照规定进行随机抽取（同一个学生的题目不可重复，题目顺序以及选择题的选项顺序可打乱），考试系统应有防刷新功能和倒计时功能，考试时间结束时自动收卷。系统能自动统计每个学生的得分。

个人设计:

依据用户需求，将用户分为三个角色：教师、学生、管理员。

教师需求：

登陆系统后，点击添加一门课程，在添加课程界面中输入课程名，添加课程介绍信息，点击添加后添加了一门课程。点击可以查看课程信息或修改课程信息，删除课程信息。

一门课程可以添加多个题库，点击添加题库，添加界面中输入题库名，使用导入按钮导入题目，并在下方显示出题目列表，点击确定，每个题目都是可编程的，可以修改选项内容和答案。保存题库信息。可以随时修改或删除题目，可以单独添加题目。

一门课程可以添加多次考试，在考试中输入考试名，考试时间，各个题目数量与分值，考试时限，限制交卷时间。选择考试班级与考试所用题库。

一门课程可以增加多个班级，点击增加班级，在增加界面中输入班级名，使用导入按钮导入学生，并在下方显示出学生的。每个学生都是可编程的，可以修改人名密码。

每场考试结束后，教师可以打开课程下的考试，查看各个学生的成绩，并导出成Excel表。

学生需求：

学生登陆系统后能够 看到自己所需要参与的考试，点击选择考试并输入考试。

学生按要求管理，到时交卷。交卷后能看到自己的成绩。

管理员需求：

能够查看所有的老师，所有的课程，所有的班级，所有的学生，所有的考试。

1. 设计与实现

4.1设计部分：见附录1

4.2实现部分：

1. 利用JavaMail开源库实现重置密码服务

利用Oracle提供的JavaMail Api可以实现向目标邮件服务器发送邮件。（见参考资料1）

模仿当下的认证模式，即假如用户忘记了密码，则可以输入邮箱地址，由我方发送一个认证邮件给用户，当用户成功登陆到邮箱，则可以点击认证邮件中的重置密码链接，从而依据用户信息重置用户密码。（见参考资料 2）

Mail mail = new Mail();

String eMailBody = "http://localhost:8080/reset.do?username=" + mailAddress.substring(0, mailAddress.indexOf('@'));

mail.setTarget(mailAddress);

mail.setMail("请点击下方链接重置您的密码（新密码为123456）：\n" + eMailBody + "\n注：完成后自动为您跳转到登陆界面");

mail.sendMail();

代码段 1-发送一个邮件

1. 利用POI开源库实现读取Excel文件

在项目过程中，需要从文件中导入用户信息。思路有两种，一种利用数据库自带的命令方式将Excel文件导入进入数据库。一种是读取Excel文件中的信息，然后将信息利用SQL语句写入数据库。

Apache 提供了一种POI开源库可以读取Excel文件。（见参考资料3）。先利用开源库读取Excel文件内容，然后封装创建题目与创建学生类。利用封装的接口类来将文件内容转换为题目与学生列表。

public class CreateStudents {

private String filePath = null;

private ImportExcel poi = new ImportExcel();

public CreateStudents(String filePath) {

this.filePath = filePath;

boolean isSuccess = poi.read(this.filePath) != null;

if(!isSuccess) {

this.poi = null;

}

}

public ArrayList<Student> createStudents() {

if(this.poi == null) {

return null;

}

ArrayList<Student> students = new ArrayList<>();

int sheets = this.poi.getTotalSheets();

for(int i = 0; i < sheets; ++i) {

for(int j = 1; j < this.poi.getTotalRows(i); ++j) {

if(this.poi.getTotalCells(i) == 3) {

String username = this.poi.getSheetsData(i, j, 1);

String password = this.poi.getSheetsData(i, j, 2);

Student student = new Student(username, password, "Student");

students.add(student);

}

}

}

return students;

}

}

代码段 2-封装一个Excel信息转换类

1. 利用Java倒计时+JS实现倒计时功能

在项目实现过程中，需要实现考试时倒计时功能。在实际过程中，如果刷新网页，利用token虽然可以防止题目刷新，但是无法防止倒计时上的刷新。所以思路如下：

在用户点击考试时记录考试时间，考试时间有限制，所以使用考试时间加上考试持续的时间，就等于最终要交卷的时间。将最终交卷的时间保存在session中，在填写题目的页面中获取交卷时间，计算交卷时间与当前时间的时间差来倒计时。

<%

Date finishDate = (Date) request.getSession().getAttribute("finishDate");

String finishTime = new SimpleDateFormat("HH:mm:ss").format(finishDate);

%>

<script>

var hourtime = "<%=finishTime%>";

var endTime = new Date(new Date().toLocaleDateString()+" "+ hourtime);

function displayTime() {

var elt = document.getElementById("clock");

var now = new Date();

var leftTime = endTime.getTime() - now.getTime();

if(leftTime < 0) {

elt.innerHTML = "Over";

submitPapar();

} else {

leftTime = parseInt(leftTime / 1000);

var o = Math.floor(leftTime / 3600);

var m = Math.floor(leftTime / 60 % 60);

var s = leftTime % 60;

elt.innerHTML = o + "小时:" + m + "分:" + s + "秒";

setTimeout(displayTime, 100);

}

}

displayTime();

</script>

代码段 3-考试倒计时处理

代码段4-计算考试结束时间

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date date = new Date();

String finishTime = simpleDateFormat.format(new Date(date.getTime() + exam.examMinutes \* 60 \* 1000));

Date finishDate = simpleDateFormat.parse(finishTime);

request.getSession().setAttribute("finishDate", finishDate);

1. 利用JS实现网页静止刷新功能

<script language="javascript">

//防止页面后退

history.pushState(null, null, document.URL);

window.addEventListener('popstate', function () {

history.pushState(null, null, document.URL);

});

document.onkeydown = function() {

if(event.keyCode==116) {

event.keyCode=0;

event.returnValue = false;

}

}

document.oncontextmenu = function() {

event.returnValue = false;

}

function submitPapar(){

document.examPapar.submit();

}

</script>

1. 总结

这个项目虽然不是大学期间的最后一个项目但也基本上算是大学期间最后做的一个有实验报告的实验了。所以，做如下总结：

这个项目可以说时开创了我们大学期间的小组合作史，以前没有任何一个软件实验是小组合作的。这就导致同学们的合作能力有很大的缺失，包括我自己。而且，有很多“抱大腿”的现象，这样的合作就没有什么意义了，但是我所接触到的大学生活，这种现象确实也不会让人惊讶。所以包括我自己在内，都是一种渴望划水的状态。

这个项目拿到手上看的时候，我就觉得这是一个复杂的项目。有的人会觉得这个项目就导入题目，创建考试，学生考试这几个重要的功能，不复杂啊。但我觉得一个项目的考虑应该从真正去实施上去考虑，这个项目真实发布情况下，用户的需求是什么样子的。假如您在看这份总结的话，老师的需求都偏于实验化，学生也懒得去思考用户需要的是什么，所以往往同学们做出来的东西都没有新意。所以我对于这个项目的考虑是，教师角色应该拥有什么样的元素，怎样将真实世界的元素抽象成为面向对象的设计结构。所以就有了附录1内的设计方案。

项目的需求分析我做了不多不少整整一个星期，先分析官方给出的需求，然后加上自己的理解，做出自己理想中的需求，在做需求的时候，我会想到一些复杂而又实用的系统，例如日志系统、恢复系统，虽然这些功能没有全部实现，但我觉得这是软件的创新点，是未来假如软件能够售出的加分点。完成需求之后就开始设计用例，设计用例完了设计数据库并生成E-R图，在后期的过程中不停的完善。然后设计类图，类图设计完了设计动态模型。也许是自己之前的学科学得不到位，所以对于系统的细节结构，系统细节的动态模型总是不知道如何去建立。所以动态模型这部分做得很不理想。

不过，做了这么多的项目，我心里是清楚的，从以前开始，我对于系统的设计都很片面，后来做多了反而越做越顺手，越做越细致，所以我觉得软件项目管理，软件架构师确实不是一时半会儿能够做到的，也许学四个月的java就能成为一名软件工程师，但没有两三年的设计功夫确实无法达到项目经理的高度。希望在以后的生活中我能学到更多。

在本次项目中我开始全面使用git用作项目的代码管理。虽然前面也使用过Coding公司的代码库，但总觉得使用的并不是很规范。所以在这次项目过程中我全面使用git来管理，每天都推送更新版本。不过我发现，在这样小的项目里面，git用不到分支，用不到回退这样的操作，我的操作就像是一条脊椎骨似的直线。而不是树形的结构。我的思考是小的非盈利的项目，不存在版本的发布，所以就不存在分支，这就导致我都项目不是树形结构，以后到公司就应该正常了。

完成项目的代码编写也总共消耗了一周的时间，在刚开始编写的时候思路清晰，积极编写注释和署名批注，越到后面，时间的压力，越来越感到这个项目后台对于一个人的庞大，在两周的情况下（有其他的作业），编写这样的后台至少要写3000行代码，对于一般队伍来说是比较困难的，所以这次虽然组队了，但抄袭、购买的现象竟然还是不少。而我也感到了前所未有的吃力。我需要在两周内完成j2ee的作业的编写。一周内写完所有的代码确实工作量大。越往后，代码质量越低，这是这个项目最大的遗憾。假如时间能够充裕一点，应该会更好吧。

在最后，还是感谢老师一年来的照顾。

1. 参考资料
2. JavaMail邮件Api下载地址

<http://www.oracle.com/technetwork/java/javamail/index.html>

1. JavaMail使用

<http://www.cnblogs.com/hehaiyang/p/4278857.html>

1. POI Excel文件读取Api下载地址

<http://poi.apache.org>

1. Struts1.x上传文件参考

<http://blog.csdn.net/captaingan/article/details/7316086>

1. 附录
2. **软件设计文档**
3. **项目文件意义介绍**

# 第1章 需求分析

在线考试系统是利用现代信息技术，通过将题库上传到网络数据库中，当学期结束，教师需要对学生进行考核评分时所产生试卷进行评分的一种在线学习产物。在线考试系统可以将用户划分为三类人：学生、教师、系统管理员。

一个基础的在线考试系统，大致可以划分为一下的流程。系统管理员通过添加人员，可以添加教师，并管理系统缺陷等。教师可以通过添加课程、班级、题库等元素单元来创建考试。学生登陆进入系统，可以查询到已经可以开考的考试和已考试的成绩。

如图所示，为系统整体功能模块图，该系统分为3大子系统，分别为学生考试子系统、教师考试管理子系统，系统管理子系统。

学生考试子系统

教师考试管理子系统

系统管理子系统

在线考试系统

图1-1系统整体功能模块图

## 1.1 学生考试子系统的需求分析

在学生考试子系统中，其子系统的一些主要功能如下：

1) 学生可以查询到自己还没有完成的考试

2) 学生可以看到自己已经考的试的成绩情况

3) 学生可以参与在线考试

4) 考试的题目选项随机出现

5) 考试系统具有放刷新功能和倒计时功能

6) 考试结束后自动交卷

其功能细化图如下：

学生考试子系统功能模块

考试查询

考试成绩查询

在线考试

生成试卷

试卷评分

图1-2学生考试子系统功能模块图

## 1.2 系统管理员管理子系统的需求分析

在系统管理员管理子系统中，其功能概述如下：

1) 添加/删除教师

2) 通过教师ID管理教师的课程班级等内容

其功能细化图如下：

教师成员管理

教师管理内容管理

系统管理员管理子系统

图1-3系统管理员管理子系统功能模块图

## 1.3 教师考试管理子系统的需求分析

在教师考试管理子系统中，其功能概述如下：

1) 教师可以管理课程，如增加、删除、查看、修改

2) 教师可以管理题库，如增加、删除、查看、修改

3) 教师可以管理题库中的题目，如上传、删除、修改、查看

4) 教师可以管理班级，如增加、删除、查看、修改

5) 教师可以管理班级中的学生，如上传、查看、修改

6) 教师可以管理考试，如增加、删除、查看

7) 教师可以向系统管理员发送缺陷反馈

其功能细化图如下：

教师考试管理系统

题库管理

考试管理

缺陷信息管理

课程管理

班级管理

图1-4教考试管理整体功能模块

### 1.3.1课程管理子系统

在课程管理子系统中，其功能概述如下：

1) 添加课程信息

2) 删除课程信息

3) 更新课程信息

4) 查看课程信息

其功能细化图如下：

课程信息管理

查看课程信息

添加课程信息

删除课程信息

更新课程信息

图1-5课程信息管理系统功能模块图

### 1.3.2题库管理子系统

在题库信息管理子系统中，其功能概述如下：

1) 添加题库信息

2) 删除题库信息

3) 更新题库信息

4) 查看题库信息

其功能细化图如下：

题库信息管理

查看题库信息

添加题库信息

删除题库信息

更新题库信息

图1-6题库信息管理系统功能模块图

### 1.3.3班级管理子系统

在班级信息管理子系统中，其功能概述如下：

1) 添加班级信息

2) 删除班级信息

3) 更新班级信息

4) 查看班级信息

其功能细化图如下：

班级信息管理

查看班级信息

添加班级信息

删除班级信息

更新班级信息

图1-7班级信息管理系统功能模块图

### 1.3.4考试管理子系统

在考试管理子系统中，其功能概述如下：

1) 增加考试信息

2) 删除考试信息

3) 查看考试信息

其功能细化图如下：

考试信息管理

查看考试信息

删除考试信息

添加考试信息

图1-8考试信息管理系统功能模块图

# 第2章 系统用例模型

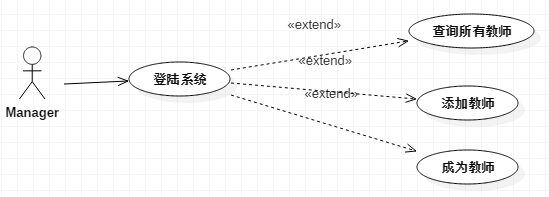
## 2.1 参与者的识别

三个参与者：教师、管理员、学生

## 2.1 管理员模型

通过分析，在管理员管理子系统可以完成以下功能：管理员通过登陆系统后可以查询出所有的教师，可以添加教师成员或成为教师成员来管理教师资源。

### 2.1.1 管理员用例图



### 2.1.2 管理员用例图相关说明

用 例：登录系统

简 述：此用例是三个角色的公用用例，用于进行身份认证进入系统

参 与 者：教师或管理员或学生

前置条件：无

后置条件：系统对应进入各角色主页

基本路径：

1. 用户打开网站主页，系统调用出登陆首页
2. 用户输入用户名密码点击登陆转到6
3. 用户忘记密码输入邮箱地址转到4
4. 系统获取邮箱地址向邮箱发送重置密码链接5
5. 用户点击重置连接，系统完成重置密码操作，转到1
6. 系统向数据库查询用户信息，如果没有用户信息转到1，如果存在用户信息则进入用户首页。

用 例：查询所有教师

简 述：查询出系统中国所有的教师

参 与 者：管理员

前置条件：session中存在Manage对象（即管理员已登陆）

后置条件：

基本路径：

1. 管理员登陆后自动调用此方法或添加教师取消后调用此方法
2. 系统调用方法获取所有的教师信息
3. 系统将信息在新界面上显示出信息

用 例：添加教师

简 述：在系统中添加一个教师

参 与 者：管理员

前置条件：session中存在Manage对象（即管理员已登陆）

后置条件：

基本路径：

1. 管理员向系统提交添加请求，系统跳转到相应页面
2. 用户输入新教师的姓名与登陆密码，并提交
3. 系统先检测是否存在此用户，如果存在就将其重命名，如果不存在则直接保存进入数据库
4. 调用查询所有教师用例进入界面

用 例：成为教师

简 述：管理员成为教师管理教师的管理对象

参 与 者：管理员

前置条件：session中存在Manage对象（即管理员已登陆）并选择好教师ID

后置条件：

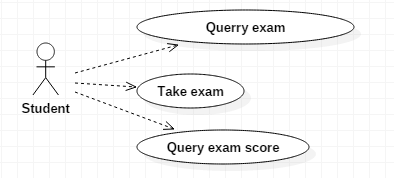
基本路径：

1. 管理员向系统提交添加请求
2. 系统获取教师对象的基本信息保存
3. 跳转页面

## 2.2 学生用例模型

通过分析，在学生考试子系统中，该系统可以完成以下功能：查询所有的未考科目，参与考试，成绩查看。

### 2.2.1 学生用例图



### 2.2.2 学生用例图相关说明

用 例：查询所有考试

简 述：查询学生所有未考过的试

参 与 者：学生

前置条件：session中存在Student对象（即学生已登陆）

后置条件：

基本路径：

1. 用户提出登陆请求后，系统自动调用此用例进入学生主界面
2. 系统获取学生所有的考试信息，去除有成绩的考试，将剩下的信息保存并跳转到学生主界面
3. 显示学生考试界面

用 例：查询所有考试和分数

简 述：查询学生所有的考试记录和考试的分数

参 与 者：学生

前置条件：session中存在Student对象（即学生已登陆）

后置条件：

基本路径：

1. 用户提交查询请求
2. 系统从数据库中查询此学生所有的考试和分数
3. 系统将数据保存，并跳转页面
4. 系统将信息在新页面上显示

用 例：参与考试

简 述：学生参加一场考试

参 与 者：学生

前置条件：session中存在Student对象（即学生已登陆）且获取考试ID

后置条件：

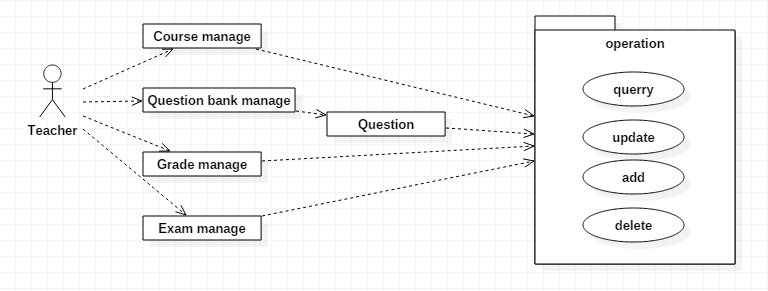
基本路径：

1. 学生选择一场考试后点击参与进入此用例
2. 系统获取考试ID，从数据库中获取此考试的相关信息
3. 系统依据考试的相关要求信息从数据库中获取对应数量的题目
4. 系统将题目信息保存并显示在新页面上
5. 用户此时开始答题（答题过程中系统限制其无法取消与刷新后退）
6. 答题完成后用户点击提交
7. 系统获取学生所有答题结果，与系统中保存的题目相比较，开始评分
8. 系统自动调用查询所有考试和分数用例，显示分数

## 2.3 教师用例模型

通过分析，在教师考试管理子系统中，该系统可以完成以下功能：课程、班级、题库的管理，考试的管理以及反馈信息的添加。

### 2.3.1 教师用例图



### 2.2.3.2 教师考试管理用例图相关说明

用 例：课程管理

简 述：管理课程例如增删改查等

参 与 者：教师

前置条件：

后置条件：

基本路径：

1. 用户登陆后系统自动调用查询所有课程功能跳转到教师管理首页
2. 系统调用数据访问层方法获取所有的课程并跳转界面，显示在界面上
3. 用户请求添加课程
4. 系统跳转到添加课程页面
5. 用户在添加课程页面输入新课程信息，点击提交
6. 系统将新课程信息写入数据库，转到2
7. 用户请求更新课程
8. 系统从数据库中查询到当前课程的信息，跳转页面，显示在更新界面上
9. 用户对课程信息做出更改操作，并保存
10. 系统将新信息保存进入数据库，转到2
11. 用户请求删除课程
12. 系统获取课程ID，并从数据库中删除此课程信息，转到2

用 例：题库管理

简 述：管理题库例如增删改查等

参 与 者：教师

前置条件：

后置条件：

基本路径：

1. 当用户点击某个课程后系统调用此方法
2. 系统调用数据访问层方法获取此用户所有的题库并跳转界面，显示在界面上
3. 用户请求添加题库
4. 系统跳转到添加题库页面
5. 用户在添加题库页面输入新课程信息，并选择需要上传的文件点击提交
6. 系统获取上传的文件并读取文件内的题目信息，然后将信息写入数据库，转到2
7. 用户请求更新题库
8. 系统从数据库中查询到当前课程的信息，跳转页面，显示在更新界面上
9. 用户对题库信息做出更改操作（例如修改题库名，删除题目，修改题目），并保存
10. 系统将新信息保存进入数据库，转到2
11. 用户请求删除题库
12. 系统获取题库ID，并从数据库中删除此题库内所有的题目，然后删除题库信息，转到2

用 例：班级管理

简 述：管理班级例如增加修改删除等

参 与 者：教师

前置条件：

后置条件：

基本路径：

1. 用户请求查看所有的班级
2. 系统调用数据访问层方法获取此用户所有的班级并跳转界面，显示在界面上
3. 用户请求添加班级
4. 系统跳转到添加班级页面
5. 用户在添加班级页面输入班级信息，并选择需要上传的文件（学生信息）点击提交
6. 系统获取上传的文件并读取文件内的题目信息，然后将信息写入数据库，转到2
7. 用户请求更新班级
8. 系统从数据库中查询到当前班级的信息，跳转页面，显示在更新界面上
9. 用户对班级信息做出更改操作（例如修改班级名，删除学生，修改学生），并保存
10. 系统将新信息保存进入数据库，转到2
11. 用户请求删除班级
12. 系统获取班级ID，并从数据库中删除此班级内所有的学生，然后删除班级信息，转到2

用 例：考试管理

简 述：管理考试信息的增加与删除

参 与 者：教师

前置条件：

后置条件：

基本路径：

1. 用户请求查看所有的考试
2. 系统调用数据访问层方法获取此用户所创建的所有考试信息并跳转界面，显示在界面上
3. 用户请求添加考试
4. 系统跳转到添加考试页面
5. 用户在添加班级页面输入考试信息，点击提交
6. 系统将信息写入数据库，转到2
7. 用户请求删除考试
8. 系统获取考试ID，并从数据库中删除此考试所有的成绩，然后删除考试信息

用 例：反馈管理

简 述：向服务器管理员提交反馈信息

参 与 者：教师

前置条件：

后置条件：

基本路径：

1. 用户请求向系统提交反馈信息
2. 系统跳转到信息页面
3. 用户输入信息
4. 系统将信息保存进入数据库
5. 系统查询所有的反馈信息
6. 跳转页面显示在界面上

# 第3章 系统静态模型

## 3.1 系统中的类

系统中的类有三个参与者相关的类和其他的几个相关类。

### 3.1.1 参与者相关的类

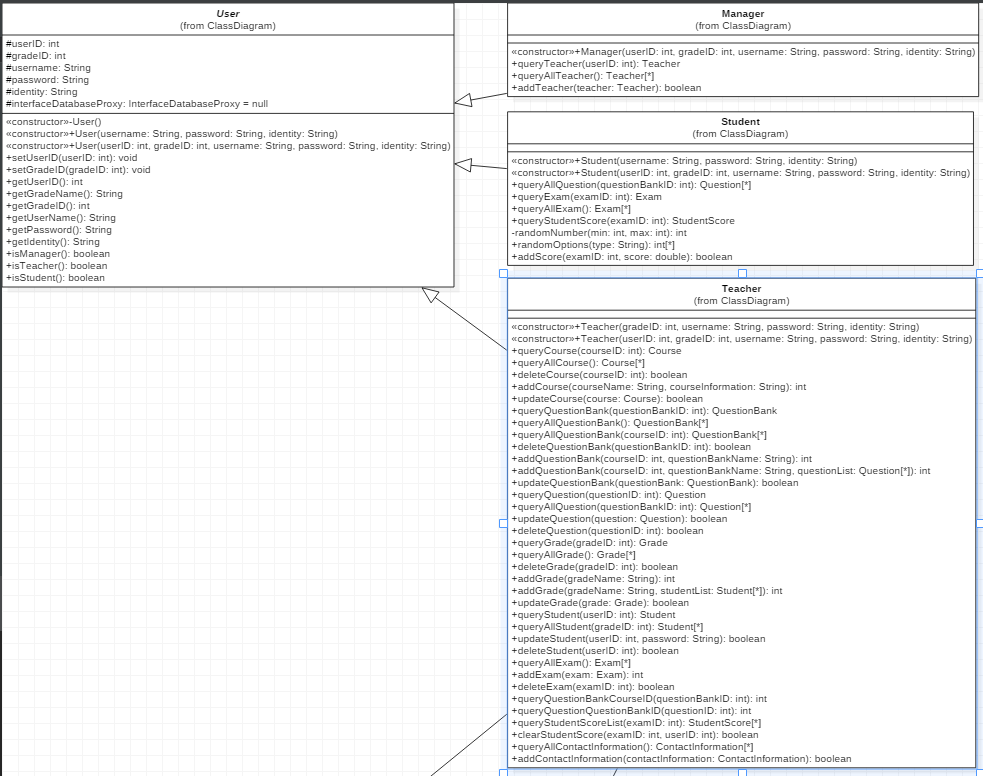
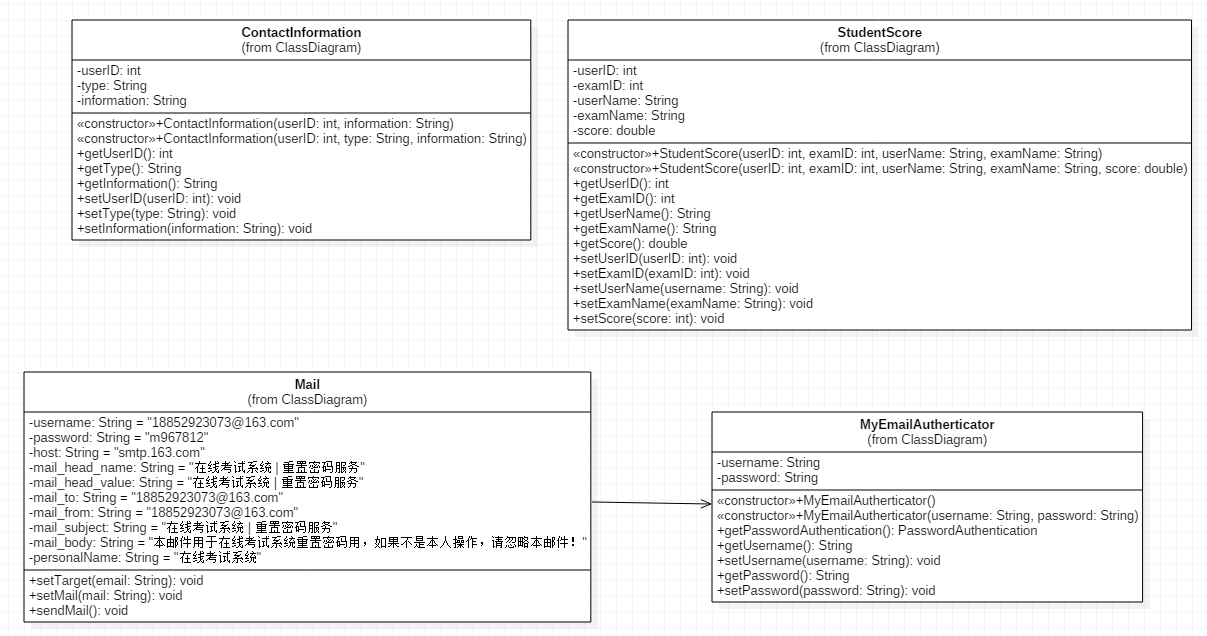


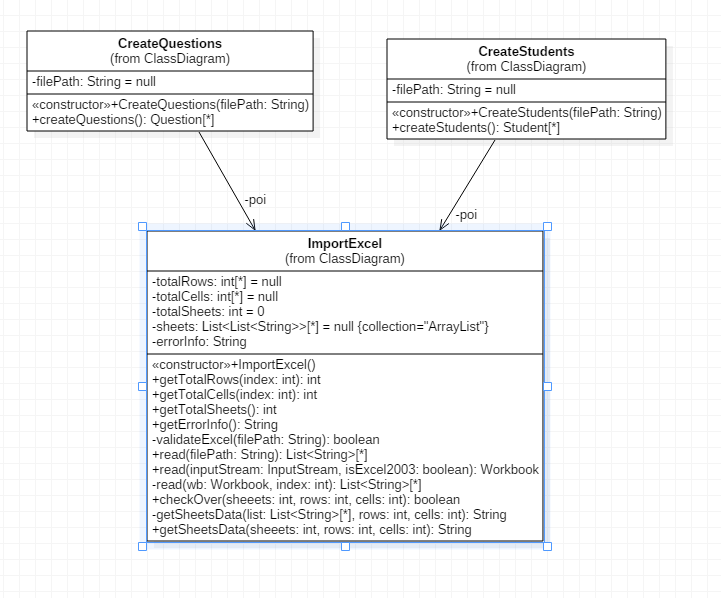
图3-1参与者相关的类

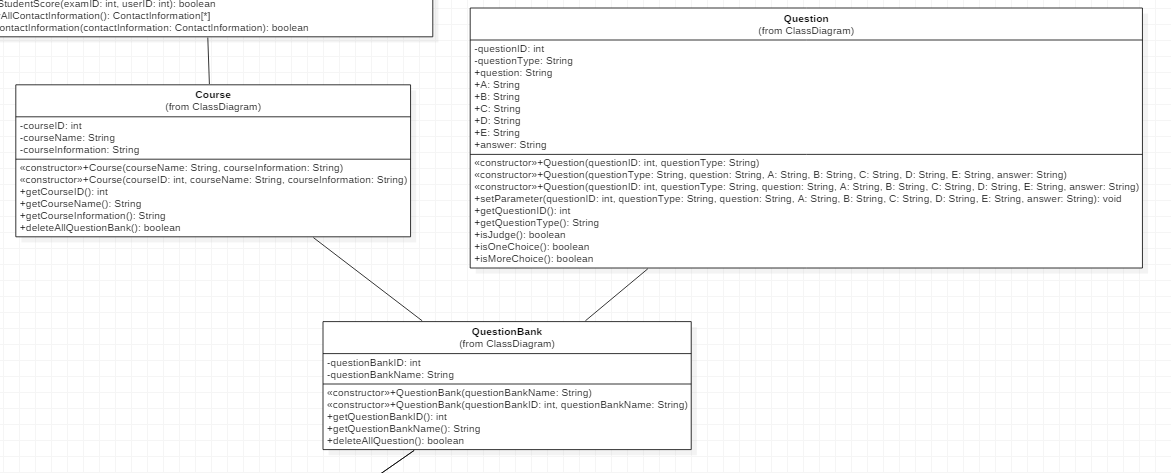
类图说明：

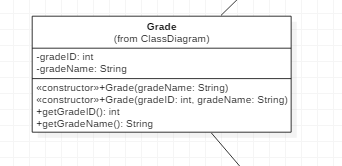
1. User类：此类是一个抽象类，用户提供用户最基本的属性与方法，例如用户ID，班级ID，用户名，密码，身份
2. Manager类，此类是user类的子类，用于描述管理员对象
3. Student类，此类是user类的子类，用于描述学生对象
4. Teacher类，此类是user类的子类，用于描述教师对象

### 3.1.2 系统中其他的相关类









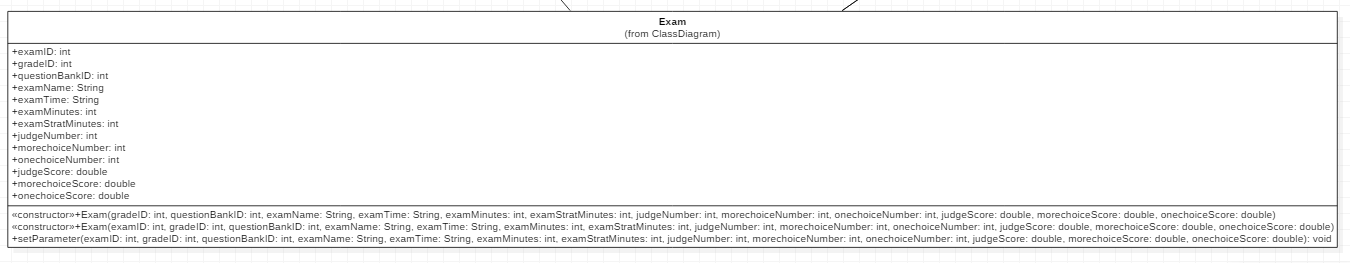


图3-2系统中用到的其他类

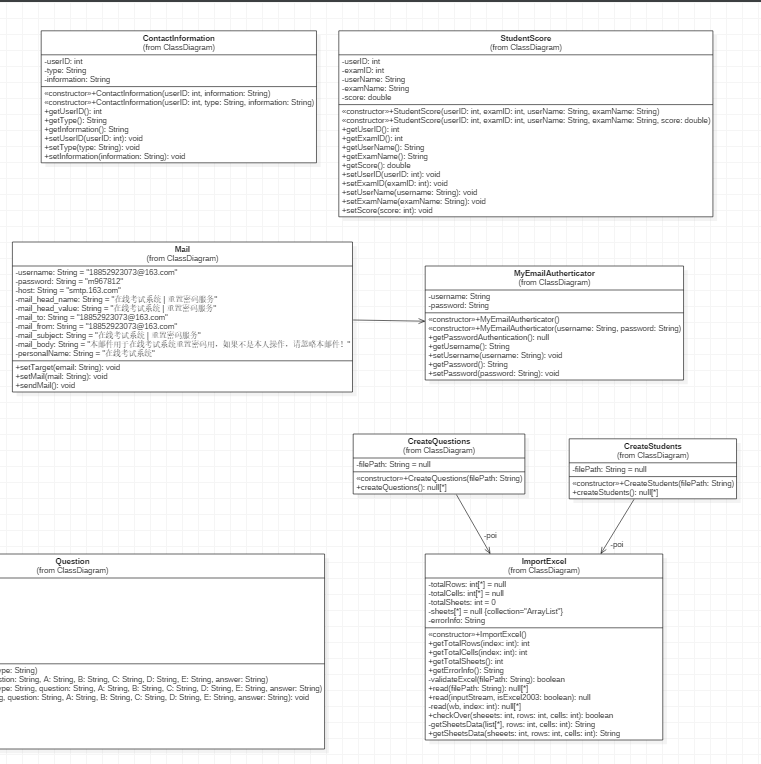
类图说明：

|  |  |  |
| --- | --- | --- |
| 序号 | 类名 | 类说明 |
| 1 | ContactInformation | 此类用于抽象反馈信息 |
| 2 | StudentScore | 此类用于保存学生的分数等相关信息 |
| 3 | Mail | 此类用于发送重置邮件 |
| 4 | CreateQuestions | 此类用于将读取的Excel数据转化为问题对象 |
| 5 | CreateStudents | 此类用于将读取的Excel数据转化为学生对象 |
| 6 | ImportExcel | 此类用于从Excel文件中读取数据 |
| 7 | Course | 此类用于描述课程信息对象 |
| 8 | Question | 此类用于描述问题信息对象 |
| 9 | QuestionBank | 此类用于描述题库对象 |
| 10 | Grade | 此类用于描述班级对象 |
| 11 | Exam | 此类用于描述考试对象 |

## 3.2 系统中类与类的关系

### 3.2.1 各个类之间的关系图

教师管理对象之间的关系



其他类之间的关系

图3-3各个类之间的关系

### 3.2.2 类之间关系说明

在关系表示中要标明类与类之间一对多或者多对多等数量关系，如一个书目的图书可以是很多本。一个借阅者可以有0或多个阅览或预定。每一个借阅和预定都和多本书籍相联系。根据这些信息，连接起来的类图如上3-3所示。在上述创建的类图中的类仅仅包含了类的属性，没有包含类的操作，可以通过系统的动态模型来确定类的操作。

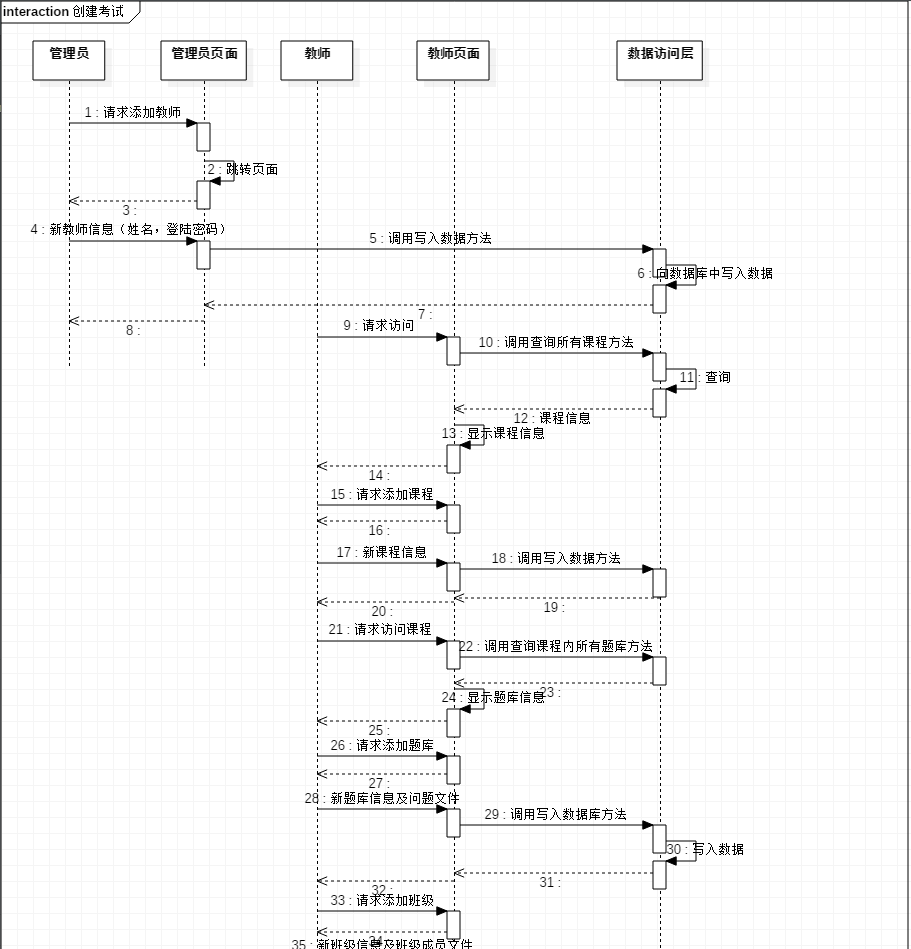
## 3.3 系统中的静态数据对象E-R图

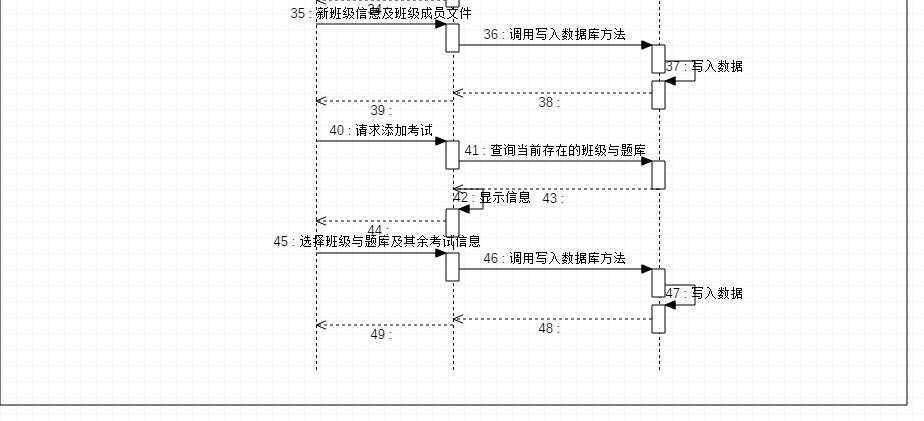
# 第4章 系统动态模型

根据系统的用例模型还可以通过对象之间的相互作用来考虑系统对象的行为。这种交互作用通过两种方式进行考察，一种是通过相互作用的一组对象为中心考察，计通过交互图惊醒，包括序列图和协作图；另一种是通过一独立的对象为中心进行考察，包括活动图和状态图。对象之间的相互作用构成了系统的动态模型。

## 4.1创建考试

### 4.1.1 创建考试序列图

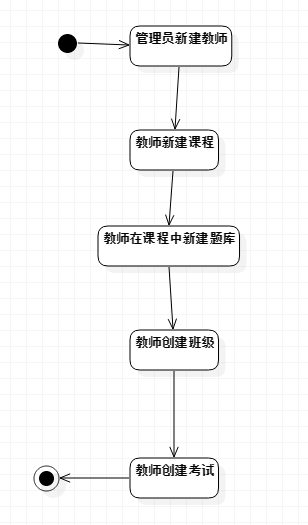




创建考试序列图描述：

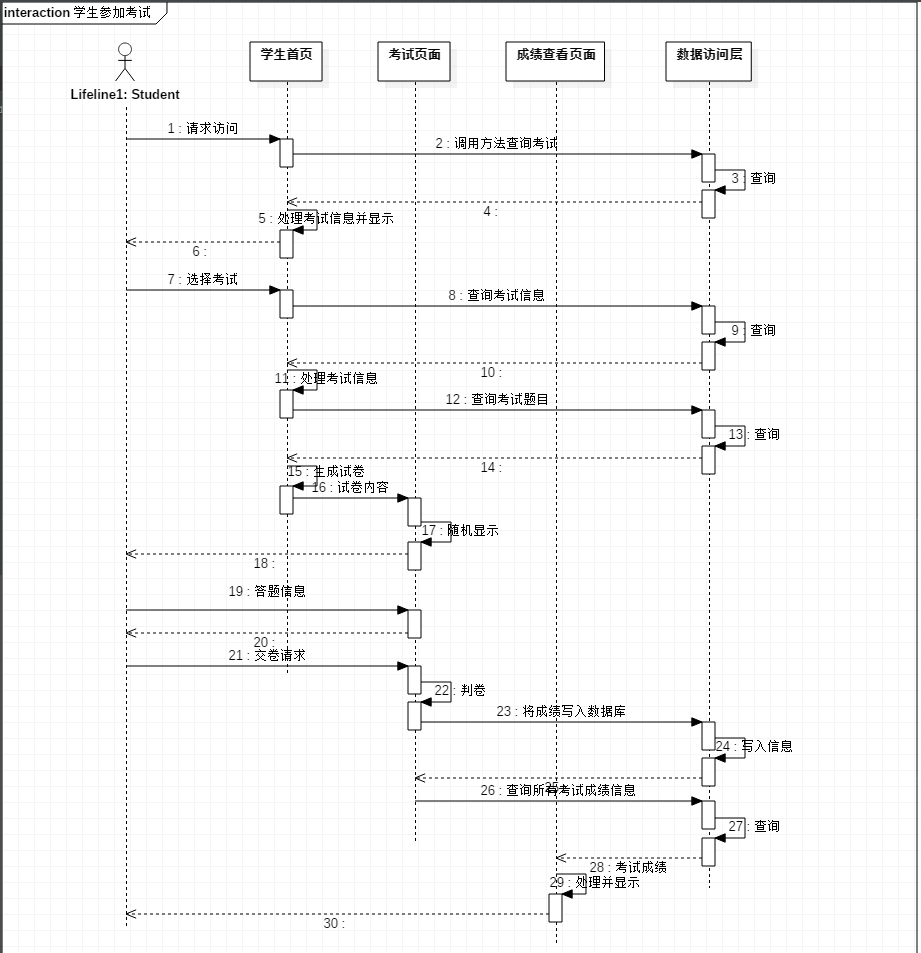
1. 管理员登陆系统，向系统发送请求添加教师
2. 管理员在添加教师页面上填写信息并提交
3. 系统获取消息，调用系统消息将信息写入数据库
4. 教师登陆系统后，请求添加课程
5. 教师在添加课程界面上填写信息并提交
6. 系统获取课程信息写入数据库
7. 系统进入课程信息页面，显示目标课程内所有的题库信息
8. 教师请求添加题库
9. 教师在添加题库页面填写题库信息，并选择文件上传，提交
10. 系统先获取文件，然后调用iop库读取Exce文件信息
11. 系统将Excel文件信息转化为多个问题对象
12. 系统将多个问题对象通过数据访问层方法写入数据库
13. 教师请求添加考试
14. 教师在添加考试界面上填写考试信息，选择考试班级和考试题库
15. 系统将获取到的信息写入数据库

### 4.1.2 创建考试活动图

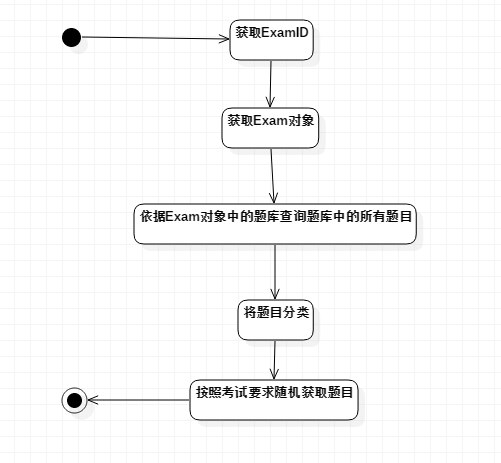


## 4.2学生参与考试

### 4.2.1 学生参与考试序列图



### 4.2.2 学生参与考试活动图

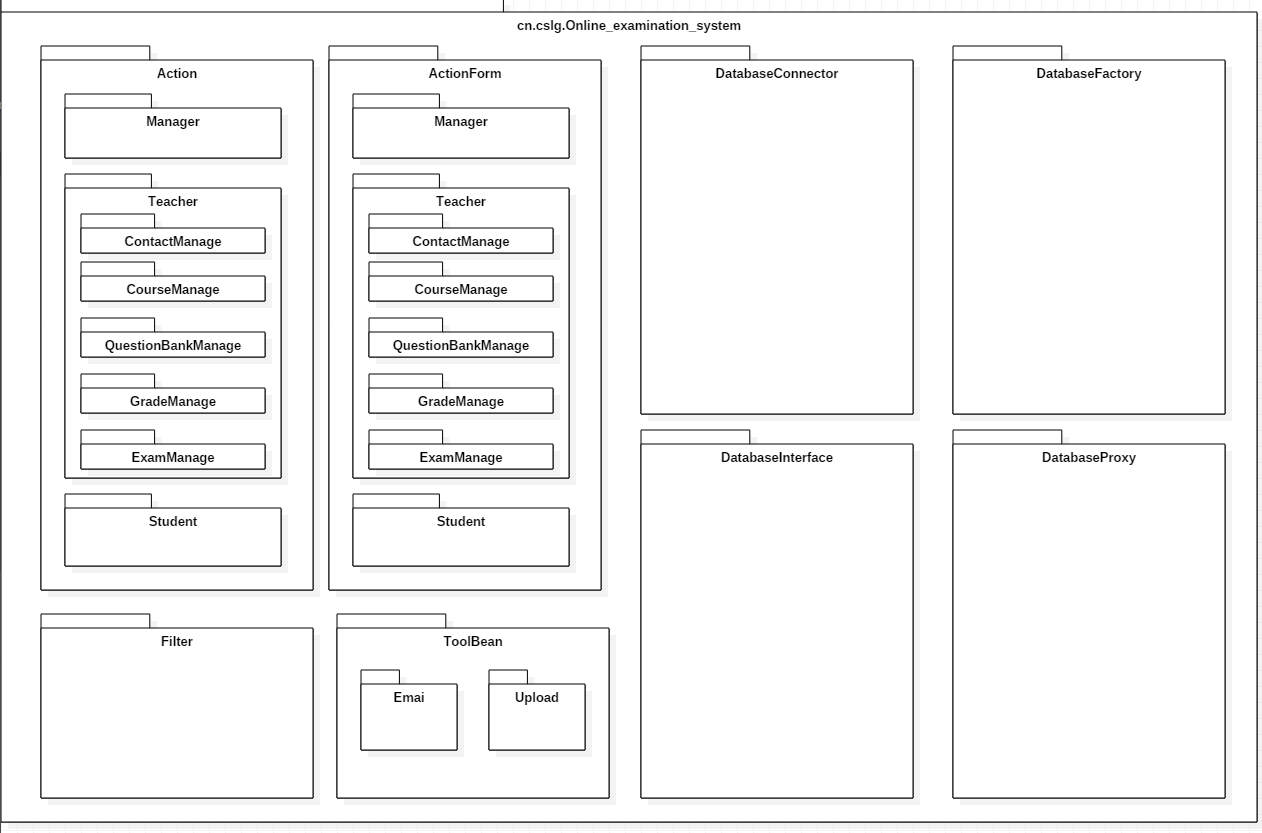


# 第5章 系统部署模型

前面的静态模型和动态模型都是按照路基的关电对烯烃进行的概念建模，还需要对系统的实现结构进行建模。对系统的实现结构进行建模的方式包括两种，即构件图和部署图。

## 5.1 在线考试系统的包图

在在线考试系统中，可以对系统的主要参与者和主要业务实体类的包分别创建对应的包来进行管理，这些包结构的关系如下。



本系统采用struts1.x框架+工厂类设计模式

|  |  |  |
| --- | --- | --- |
| 序号 | 包名 | 包意义 |
| 1 | Action | Struts操作 |
| 2 | ActionForm | Struts表单 |
| 3 | Filter | 过滤器 |
| 4 | Toolbean | 工具类 |
| 5 | DatabaseConnector | 数据库连接包 |
| 6 | DatabaseProxy | 数据库代理包 |
| 7 | DatabaseInterface | 数据库接口包 |
| 8 | DatabaseFactory | 数据库工厂包 |
| 9 | Manage | 管理员包 |
| 10 | Student | 学生包 |
| 11 | Teacher | 教师包 |
| 12 | ContactManage | 反馈信息功能包 |
| 13 | CourseManage | 课程管理包 |
| 14 | QuestionBankManage | 题库管理包 |
| 15 | GradeManage | 班级管理包 |
| 16 | ExamManage | 考试管理包 |

## 5.2 在线考试系统部署图

### 5.2.1部署图

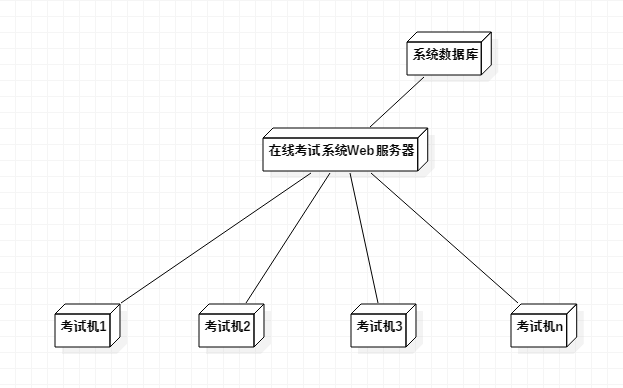


图5-5图书管理系统部署图

### 5.2.2部署图说明

本系统作为一个Web系统，其各个子系统，如教师管理子系统，管理员管理子系统统一在web服务器上发布。用户直接通过用户名密码登陆，系统可以自动判断用户身份。

**附录2**

