

软件工程课程 B6 小组

# 在线测试子系统 总体设计报告

邹天旻 3090102959

赵欣欣 3090104129

颜梦佳 3090102574

吕萌 3090103393

# 目录

一、引言.....	6
1.1 标识.....	6
1.2 编写目的.....	6
1.2.1 目的.....	6
1.2.2 预期读者.....	6
1.3 背景.....	7
1.4 系统概述.....	8
1.4.1 目标.....	8
1.4.2 用户的特点.....	8
1.4.3 假定和约束.....	8
1.5 定义.....	9
1.6 基线.....	9
1.7 参考资料.....	9
二、总体设计.....	10
2.1 需求规定.....	10
2.1.1 系统功能.....	10
2.1.2 系统性能.....	11
2.1.3 安全性需求.....	12
2.1.4 可维护性需求.....	12
2.2 运行环境.....	12
2.2.1 设备.....	12
2.2.2 支持软件.....	13
三、系统结构.....	13
3.1 系统功能结构.....	13
3.2 技术简介.....	16
3.3 部署图.....	17
3.4 类图.....	18
3.5 内部接口.....	18
3.6 顺序图.....	19
3.6.1 用户登录.....	20
3.6.2 试题管理.....	21
3.6.3 试卷管理.....	23
3.6.4 在线答题.....	23
3.6.5 成绩查询.....	24
四、执行概念.....	24
4.1 学生登录.....	24
4.1.1 模块概述.....	24
4.1.2 IPO 图.....	25
4.1.3 功能.....	25
4.1.4 输入项.....	25
4.1.5 输出项.....	25
4.1.6 设计方法（算法）.....	25

4.1.7	流程图.....	26
4.2	教师登录.....	26
4.2.1	模块概述.....	26
4.2.2	IPO 图.....	27
4.2.3	功能.....	27
4.2.4	输入项.....	27
4.2.5	输出项.....	27
4.2.6	设计方法（算法）.....	27
4.2.7	流程图.....	28
4.3	添加试题.....	28
4.3.1	模块概述.....	28
4.3.2	IPO 图.....	29
4.3.3	功能.....	29
4.3.4	输入项.....	29
4.3.5	输出项.....	29
4.3.6	设计方法（算法）.....	30
4.3.7	流程图.....	30
4.4	修改试题.....	31
4.4.1	模块概述.....	31
4.4.2	IPO 图.....	31
4.4.3	功能.....	31
4.4.4	输入项.....	31
4.4.5	输出项.....	32
4.4.6	设计方法（算法）.....	32
4.4.7	流程图.....	33
4.5	删除试题.....	33
4.5.1	模块概述.....	33
4.5.2	IPO 图.....	34
4.5.3	功能.....	34
4.5.4	输入项.....	34
4.5.5	输出项.....	34
4.5.6	设计方法（算法）.....	34
4.5.7	流程图.....	35
4.6	查询试题.....	35
4.6.1	模块概述.....	35
4.6.2	IPO 图.....	36
4.6.3	功能.....	36
4.6.4	输入项.....	36
4.6.5	输出项.....	36
4.6.6	设计方法（算法）.....	37
4.6.7	流程图.....	37
4.7	手动试卷生成.....	37
4.7.1	模块概述.....	37
4.7.2	IPO 图.....	38

4.7.3 功能.....	38
4.7.4 输入项.....	38
4.7.5 输出项.....	38
4.7.6 设计方法（算法） .....	38
4.7.7 流程图.....	39
4.8 自动试卷生成.....	39
4.8.1 模块概述.....	39
4.8.2 IPO 图.....	40
4.8.3 功能.....	40
4.8.4 输入项.....	40
4.8.5 输出项.....	40
4.8.6 设计方法（算法） .....	40
4.8.7 流程图.....	41
4.9 成绩统计.....	42
4.9.1 模块概述.....	42
4.9.2 IPO 图.....	42
4.9.3 功能.....	42
4.9.4 输入项.....	42
4.9.5 输出项.....	42
4.9.6 设计方法（算法） .....	43
4.9.7 流程图.....	43
4.10 学生在线答题.....	44
4.10.1 模块概述.....	44
4.10.2 IPO 图.....	44
4.10.3 功能.....	44
4.10.4 输入项.....	44
4.10.5 输出项.....	44
4.10.6 设计方法（算法） .....	45
4.10.7 流程图.....	46
4.11 历史查询.....	46
4.11.1 模块概述.....	46
4.11.2 IPO 图.....	47
4.11.3 功能.....	47
4.11.4 输入项.....	47
4.11.5 输出项.....	47
4.11.6 设计方法（算法） .....	47
4.11.7 流程图.....	48
五、用户界面设计.....	49
5.1 用户登录.....	49
5.2 题库管理界面.....	50
5.3 试卷生成界面.....	50
5.4 学生答题界面.....	51
5.5 成绩统计界面.....	52
5.6 学生历史数据查询.....	52

六、数据库设计.....	53
6.1 E-R 图 .....	53
6.2 题目实体.....	53
6.3 试卷实体.....	54
6.4 考试成绩记录实体.....	55
七、接口设计.....	55
7.1 基础信息管理子系统.....	55
7.2 自动排课子系统.....	56
7.3 选课子系统.....	56
7.4 资源共享子系统.....	56
7.5 论坛交流子系统.....	56
7.6 成绩管理子系统.....	56
八、系统开发难点分析.....	56
九、故障处理.....	57
9.1 出错信息表.....	57
9.2 补救措施.....	57
9.3 系统维护设计.....	57

# 一、引言

## 1.1 标识

文件状态:	文件标识:	总体设计报告
[ ] 草稿	当前版本:	1.0
[√] 正式发布	作 者:	软件工程 B6 组
[ ] 正在修改	完成日期:	2012 年 5 月 6 日

## 1.2 编写目的

从本阶段开始,项目进入正式开发阶段。这份总体设计报告的编写目的,是以本项目的  
需求分析说明书为依据,从总体设计的角度,明确系统管理子系统的总体架构、流程、数据  
结构、数据库设计。

### 1.2.1 目的

为编码人员提供依据

为修改、维护提供条件

明确各模块外部接口、内部接口、用户接口

项目负责人将按计划说明书的要求布置和控制开发工作全过程

### 1.2.2 预期读者

这份设计报告的预期读者包括:

软件用户

项目经理

项目开发人员

软件质量分析员

软件维护人员

## 1.3 背景

### 软件系统名称

教学服务系统——在线测试子系统

### 任务提出者

浙江大学软件工程基础课程任课老师——王章野

### 开发者

浙江大学 2011-2012 学年秋学期软件工程基础课程第十三小组，小组成员为：

邹天旻 3090102959

赵欣欣 3090104129

颜梦佳 3090102574

吕 萌 3090103393

### 用户

教学服务系统管理员，包括学生以及教师

### 实现该软件的计算机网络

由若干台 PC 机组成的局域网

### 该软件系统同其他系统或其他机构的基本的相互来往关系

小组可以使用一个网上现有的教学服务系统（如浙江大学教务管理系统）来加深对教学服务系统的了解，尤其是对管理员部分的理解，同时通过学习一些基本的教学服务知识以及程序设计技术来为开发工作打下基础。

## 相关背景介绍

为全面提高学生创新和实践能力,浙江大学软件工程基础课程分为课堂教学和综合性实验两部分。综合性实验采取分组形式完成,全面锻炼学生的系统设计与实现能力、测试能力、组织文档能力以及团队合作能力。综合性实验从实际问题中抽取,贯穿全部重要知识点,通过阶段性验收,锻炼学生综合运用每个环节所学知识解决实际问题的能力。本次课程,教师选取教务管理系统作为综合性实验题目。该说明书对教务管理系统的在线测试子系统进行详细描述。

## 1.4 系统概述

### 1.4.1 目标

我们旨在完成在教学服务系统的在线测试子系统。在线测试子系统为教师提供题库管理、试卷生成、获取成绩统计信息等服务,为学生提供在线答题、成绩查询等服务。

教师可以通过教师用户界面进行如下操作:题库管理(包括添加选择题、是非题,包括题目和答案,形成题库,并进行查询、添加、删除、修改等操作),试卷生成(包括手工从题库选取题目,组成试卷;或设定测试范围,由系统自动生成试卷),成绩统计(按学生、试卷、题型、考察点,显示多种统计数据,提供直观图示,供进一步分析)。

学生可以通过学生用户界面进行如下操作:在线答题(浏览并选择试卷,在线答题;测试开始后开始计时,中途退出无效,提交试卷后自动评分),历史成绩查询。

### 1.4.2 用户的特点

教学服务系统(包括在线测试子系统)的用户为教师和学生。

教师对于题库中的内容有相当程度的了解,他试图对学生进行针对性测试,并要求获得全面综合的反馈信息,这过程中涉及大量的题库数据处理、成绩数据统计;而学生需要流畅的答题环境,以求在答题结束后对自身水平有更深入的了解。

因此,本软件的客户端界面一方面应尽量方便教师与学生的各种操作,另一方面,后台数据处理性能应尽量实时、准确。

### 1.4.3 假定和约束

此次软件开发出于实验性目的,暂无经费划拨,希望项目开发团队能够充分利用现有资源高效地开发出符合需求的系统。

教学服务系统由7个子系统构成,每个子系统完成其中一项功能,最后进行系统集成。此项目开发期限较短,因此各个子系统务必要安排好工作进度,保证有一定的时间进行系统集成。系统管理子系统与其他几个子系统均有关联,因此,务必做好与其他几个子系统开发



团队的沟通协商工作。

此次软件开发本着实验教学的目的，同时访问本系统的用户较少，故对数据处理效率的能力等方面要求不高。

## 1.5 定义

### 权限

指用户职能的范围，即各种用户所登录界面、所接触数据、所进行操作等的范围。

### PHP

PHP，是英文超级文本预处理语言 Hypertext Preprocessor 的缩写。PHP 是一种 HTML 内嵌式的语言，是一种在服务器端执行的嵌入 HTML 文档的脚本语言，语言的风格有类似于 C 语言，被广泛的运用。

### MySQL

MySQL 是一个小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司。在 2008 年 1 月 16 号被 Sun 公司收购。而 2009 年，SUN 又被 Oracle 收购。MySQL 是一种关联数据库管理系统，关联数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内。这样就增加了速度并提高了灵活性。MySQL 的 SQL“结构化查询语言”。SQL 是用于访问数据库的最常用标准化语言。MySQL 软件采用了 GPL（GNU 通用公共许可证）。由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，许多中小型网站为了降低网站总体拥有成本而选择了 MySQL 作为网站数据库。

## 1.6 基线

[1] “在线测试子系统”需求分析报告 1.0

## 1.7 参考资料

《在线测试子系统 总体设计报告》（课程资料）

提供者 软件工程基础课程任课教师—王章野老师

《课程资源共享与发布子系统 软件总体设计说明书》（课程资料）

提供者 软件工程基础课程任课教师—王章野老师

MVC - 维基百科，自由的百科全书

<http://zh.wikipedia.org/wiki/MVC>

Yii Framework: Best for Web 2.0 Development

## 二、总体设计

### 2.1 需求规定

#### 2.1.1 系统功能

##### 1. 为教师、学生提供登录界面

在线测试子系统提供登录界面让用户（包括教师与学生）输入用户账号、密码、类型，并对账号和密码进行验证，一旦账号和密码通过认证，用户便可以进入在线测试子系统的后台的相应页面进行相关操作。

##### 2. 为教师提供添加、删除、修改、查询题目的功能

在线测试子系统中，教师登录成功后，可以浏览子系统中已经存在的题目、修改已经存在的题目、删除已经存在的题目以及添加新题目。  
在线测试子系统为此功能提供后台操作页面和数据库更新方法。教师添加新的新题时，需要输入题目属性。

##### 3. 为教师提供手动生成试卷的功能

教师登录成功后，可以选择手动生成试卷功能，进入新界面，浏览在线测试子系统中即时的存在所有题目，并根据自己的需要选择要进行考察的题目并组成新的试卷。  
在线测试子系统为此功能提供后台操作页面和数据库更新方法。教师最终生成的试卷经过系统检查，确认不存在异常情况（空卷）并存入数据库。

##### 4. 为教师提供自动生成试卷的功能

教师登录成功后，可以选择自动生成试卷功能，进入新界面，教师选择考察范围、综合难度等试卷参数并确认后，系统调用自动生成试卷算法生成合乎要求的试卷。  
在线测试子系统为此功能提供后台操作页面、自动生成试卷算法和数据库更新方法。教师浏览最终生成的试卷，确认后经过系统检查，确认不存在异常情况（空卷）并存入数据库。

## 5. 为学生提供在线测试的功能

学生登录成功后并当前存在有待完成的考试，可以选择开始参加考试，系统及时并即时记录学生的答案。答题时间到后，保存对所有答案，并与正确对答案进行对比，显示学生的成绩。

在线测试子系统为此功能提供后台操作页面、批卷算法和数据库更新方法。

## 6. 为学生提供查看历史成绩的功能

学生登陆成功后并存在已完成试卷，可以选择查看历史成绩，显示其所有参加过的考试的成绩，并统计其平均分。

在线测试子系统为此功能提供后台操作页面、批卷算法和数据库更新方法。

## 7. 为教师提供成绩统计的功能

教师以教师账号（一般为学工号）登录成功后，可以选择根据不同的统计方式（按学生、试卷、题型、考察点），查看学生成绩统计信息，包括直观图示供教师分析。

在线测试子系统为此功能提供后台操作页面、按功能统计算法、直观图显示算法。

### 2.1.2 系统性能

对于在线测试系统，良好的交互环境以及与系统数据库的完善连接是非常重要的，需要根据教师、学生的各种操作来进行设计，优化交互界面，使用户的工作更加高效、方便。为此，我们提出以下几个方面的要求：

- 1、界面设计简洁直观，布局合理，主题统一，信息清晰，突出重点。
- 2、操作界面友好，操作方便，容易上手。
- 3、系统具有良好的反应速度，给用户良好的使用体验。我们要求在良好的网络情况下，系统应

具有以下时间特性要求：

单个用户在线时，网页响应时间少于 1s，信息搜索响应时间少于 2s。

100 个用户在线时，网页响应时间少于 2s，信息搜索响应时间少于 5s。

当然，具体的系统性能可能还会与具体的服务器端数据库、网络传输延时以及并发访问该用户量有关。

### 2.1.3 安全性需求

#### 1. 保密性

对于个人信息验证功能，应避免未被授权的用户浏览其他用户的账户密码信息。完善的访问控制要避免系统中已注册的合法用户非法访问和使用系统资源。用户密码等敏感信息需要加密储存与传输。在用户登录过程中，应避免数据库嵌入、密码强制破解、伪造会话侵入等情况。

#### 2. 完整性

系统中的用户信息需要保持完整性，要避免各种形式的数据丢失。同时，也要避免未被授权用户偶然或恶意地插入、删除、修改系统中的各种用户信息。

### 2.1.4 可维护性需求

1、系统设计时应备齐各类文档，操作人员只需阅读文档就可知道所有的操作，维护人员也可以通过文档了解系统结构、设计思路，以便快速掌握整个系统。

2、程序设计时应保持良好的编程风格，结构清晰、注释明确，使调试、测试人员能快速定位各种错误。

3、交付时应文档齐全，说明详尽，文档符合相关标准。

## 2.2 运行环境

### 2.2.1 设备

由于条件有限，我们不能提供专门的服务器运行系统，请开发人员利用配置较高的 PC 作为服务器。

服务器端的设备要求如下：

#### 计算机

CPU:  $\geq 2.0\text{GHz}$

内存:  $\geq 1.0\text{GB}$

#### 外围设备

键盘: 能用即可

鼠标：能用即可  
显示器：能用即可  
硬盘： $\geq 100\text{GB}$   
硬盘转速： $\geq 7200\text{rpm}$

#### 通讯设备

网线：具有良好的数据传输能力  
网卡：100M

### 2.2.2 支持软件

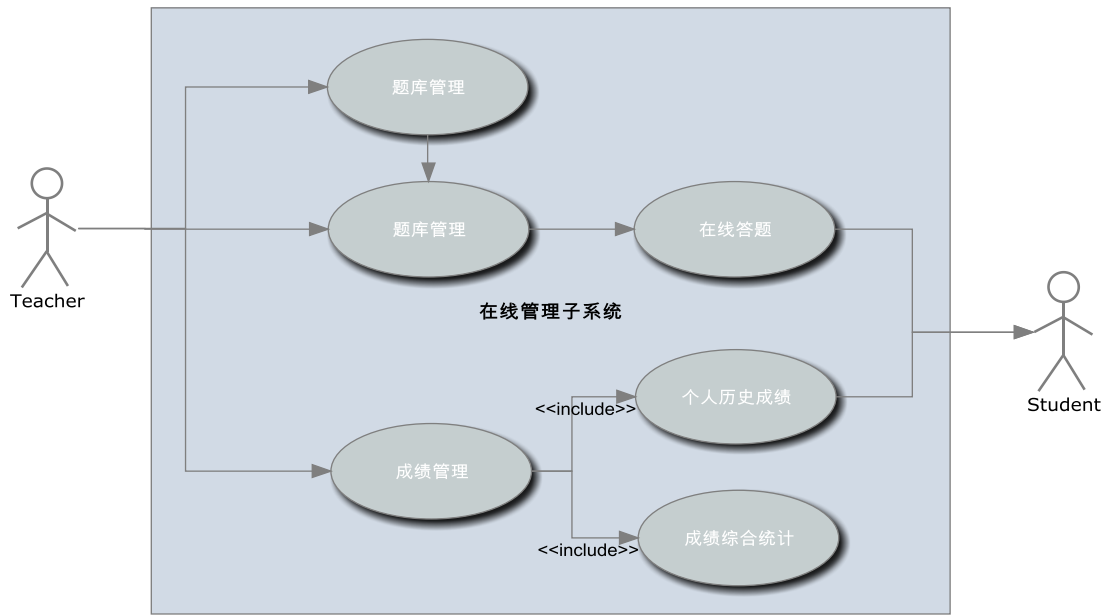
操作系统：Windows XP 或 Windows 7  
数据库平台：MySQL  
Web 服务器：Apache Tomcat  
开发工具：能支持网页开发的工具均可，如 EditPlus。  
测试工具：请测试人员自行选择。  
建模工具：根据项目状况，请自行选择合适方便的建模工具。  
办公软件：Microsoft Office2007/2010 系列产品  
浏览器：IE 浏览器、谷歌浏览器等

## 三、系统结构

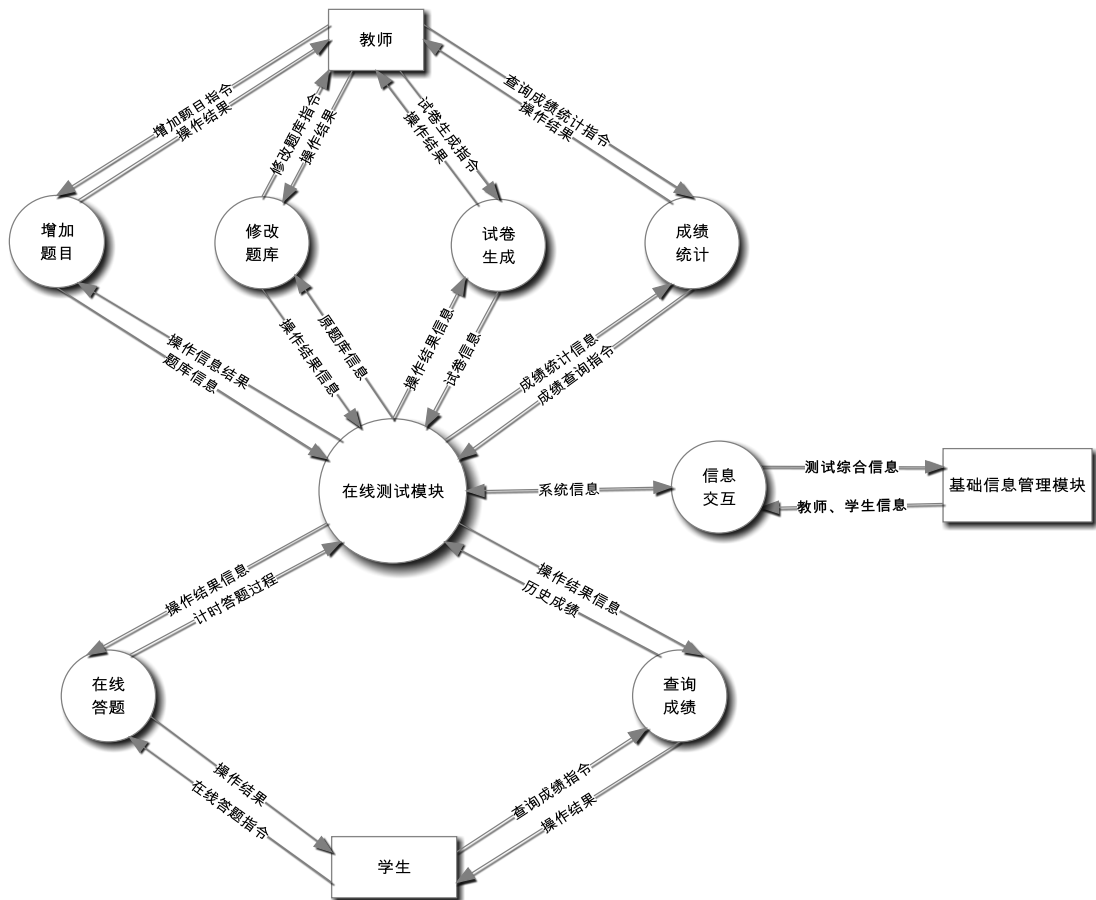
### 3.1 系统功能结构

在之前的需求分析中，我们组对于我们的系统进行了相应的用例分析，具体如下所示。

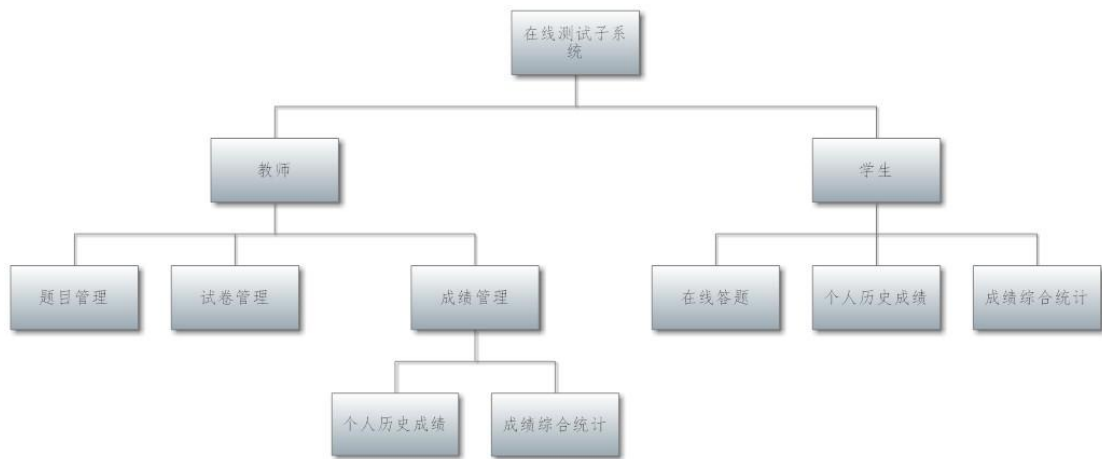
用例图：在线测试子系统



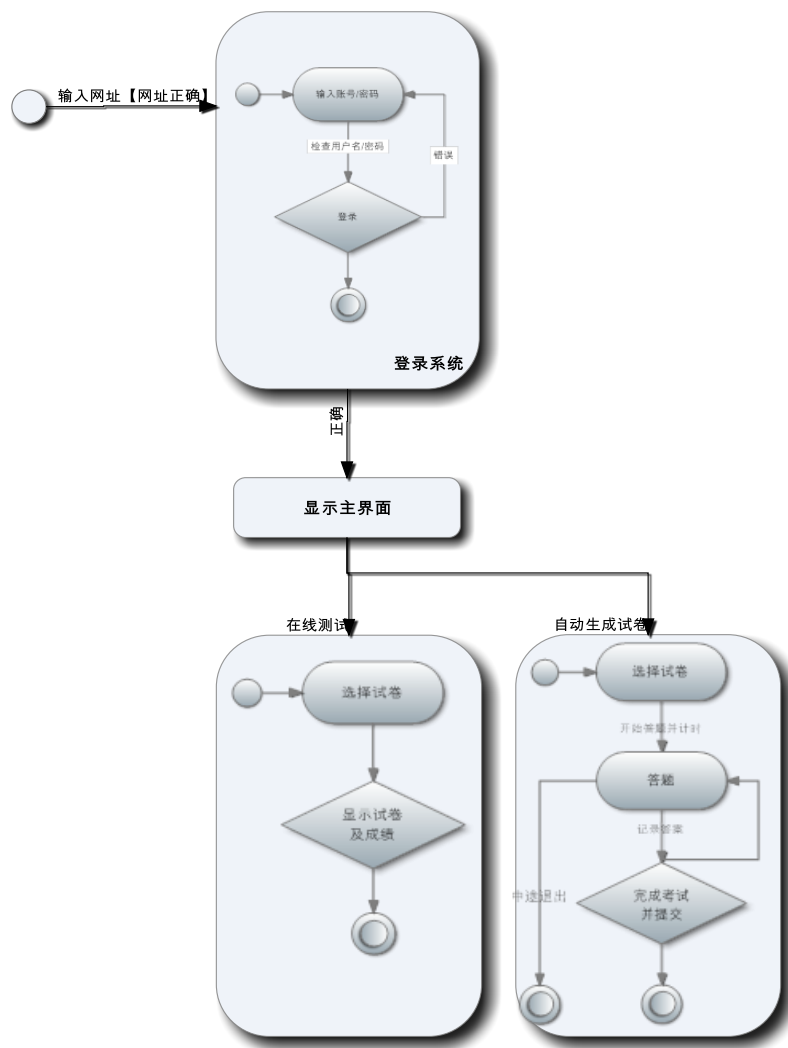
此外，我们也在过去的需求报告中，基于数据流进行了相应的分析如下：

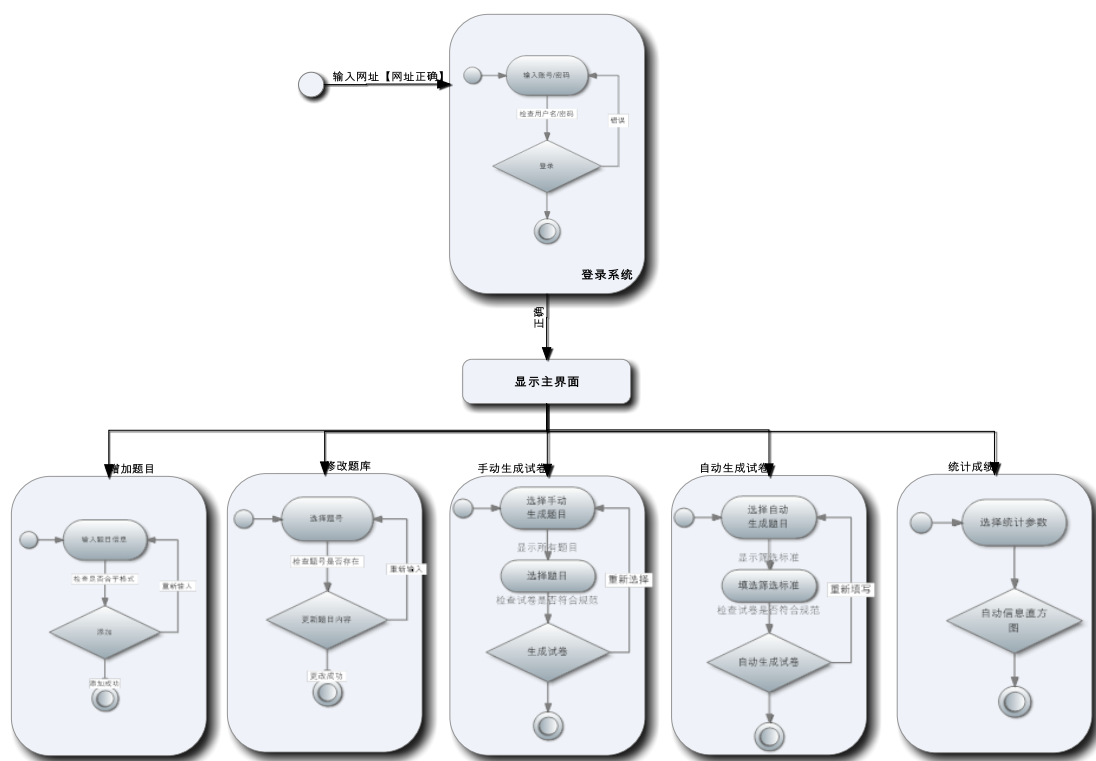


基于上面我们对于系统需求的分析，我们的系统地总体结构设计也比较显然了。这里我们的在线测试子系统的系统层次图如下所示。



而系统的运行的整体流程则如下所示，这两幅状态图在我们需求分析中也已经有了，这里引用过来，并且作为程序流程设计的参考。





## 3.2 技术简介

首先，在我们系统中，我们希望采用 MVC 的模式进行设计和实现。“MVC 模式（三层架构模式）（Model-View-Controller）是软件工程中的一种软件架构模式，把软件系统分为三个基本部分：模型（Model）、视图（View）和控制器（Controller）。”在这种模式下，程序的开发人员能够将模型、视图、控制器这三个模块独立开来，从而能够对各个模块进行更为方便的修改和完善。

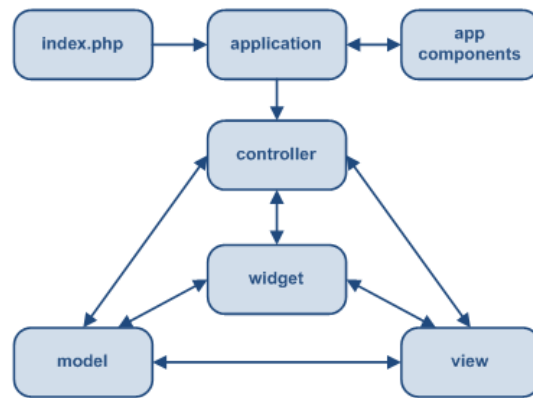
在这个模式中，“模型”不依赖“视图”和“控制器”，它用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法，能够对数据库中的数据进行直接访问。模型中数据的变化一般会通过一种刷新机制被公布，而为了实现该机制，相应的视图必须事先注册，以了解相应的数据改变。“视图”的任务则是实现数据有目的的显示。它需要访问相应的数据模型从而实现上面我们提到过的刷新功能。“控制器”则负责处理事件并作出响应。

以上即为 MVC 模式的说明，而在我们具体实现这个程序的过程中，我们在 WAMP 的基础之上，又采用了 YII Framework 作为网站建设的基础。

YII Framework 是一种免费、开源的、基于组件的网络应用开发框架，由 PHP5 编写。YII 采用了严格的 OOP 思想，在效率、扩展性和可维护性方面都有着不俗的表现。

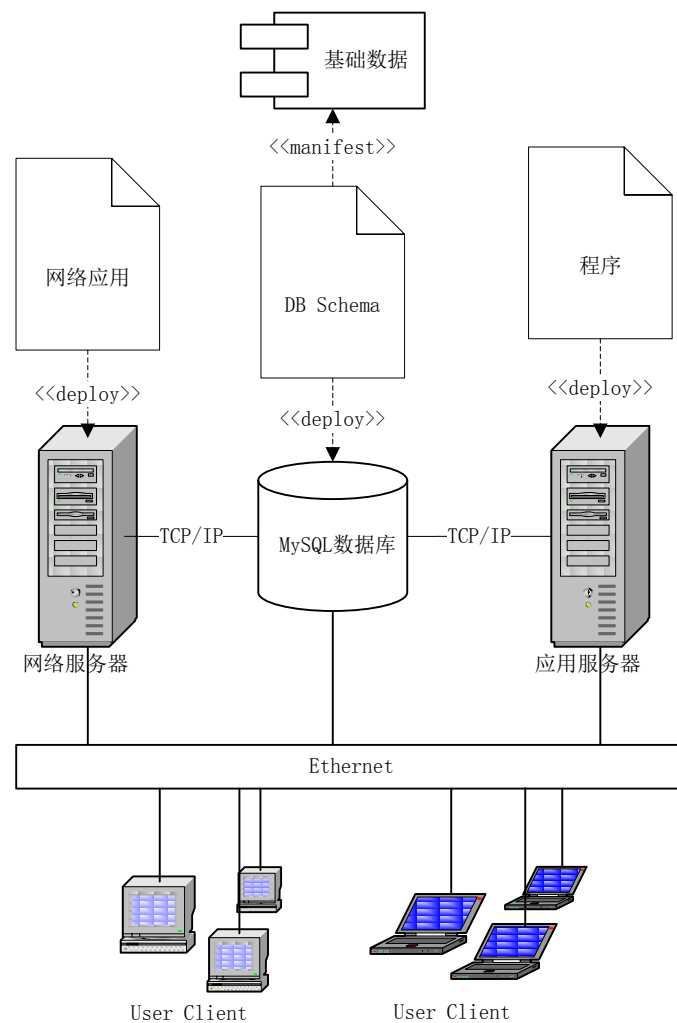
除了使用了 MVC 的模式之外，YII 还用了一些相关结构对前台进行了一些封装。在这个框架中，我们可以看到，基于 YII 创建的网站一般具有如下的结构，这也是我们的网站的结构：



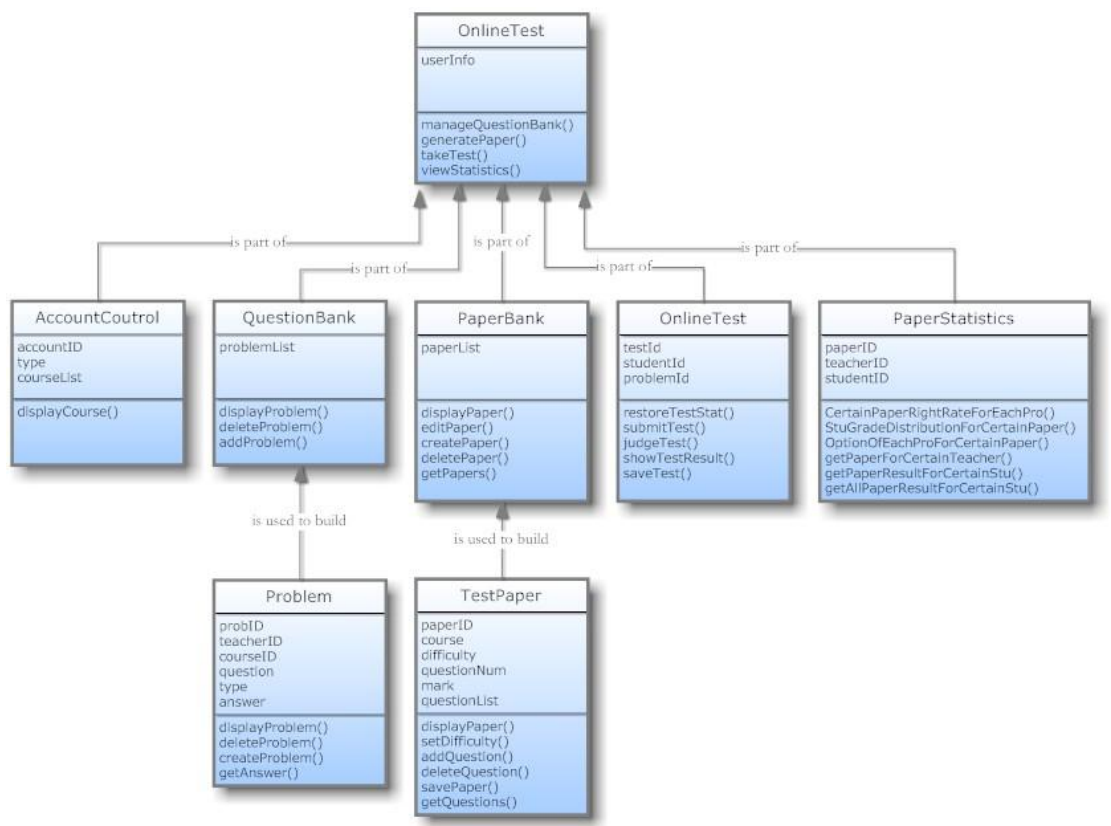


### 3.3 部署图

现在，我们将就系统运行时的结构以及相应的硬件配置及其软件部署进行一个设计。这里我们将采用 UML 中的部署图对整个系统作一个规划。具体设计如下图所示：

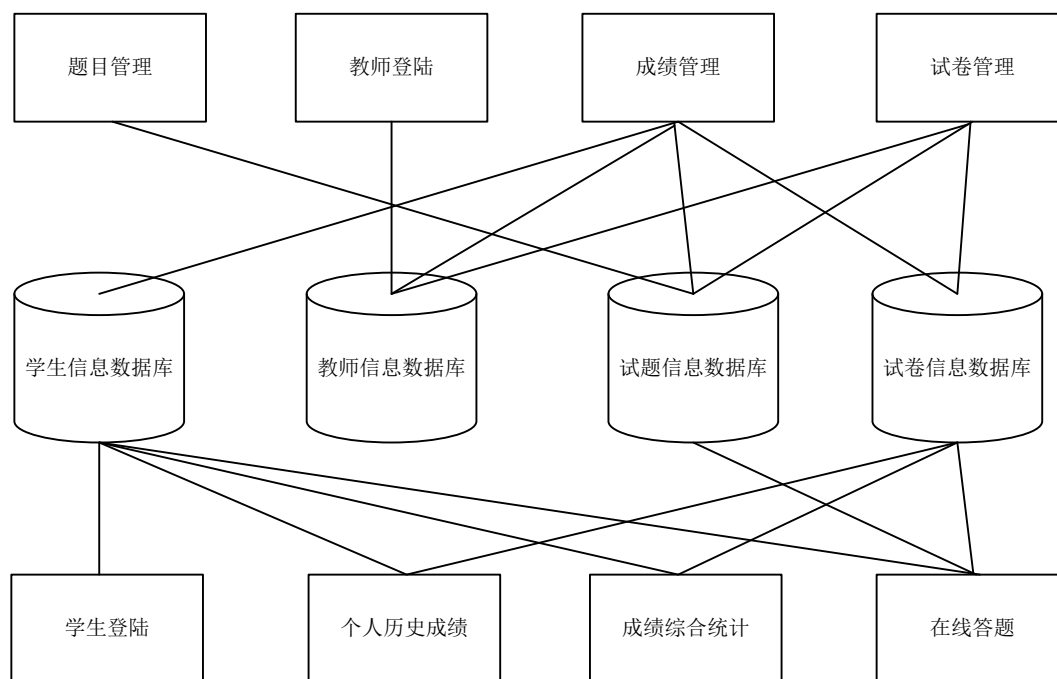


### 3.4 类图



### 3.5 内部接口

对于在线测试子系统,该子系统内部的各个模块之间也有着一定的合作与协同。特别的,在数据库上,该子系统的各个模块在数据库的共享上需要进行相应的设计。下面就是在线测试子系统内各个模块对数据库的共享情况:



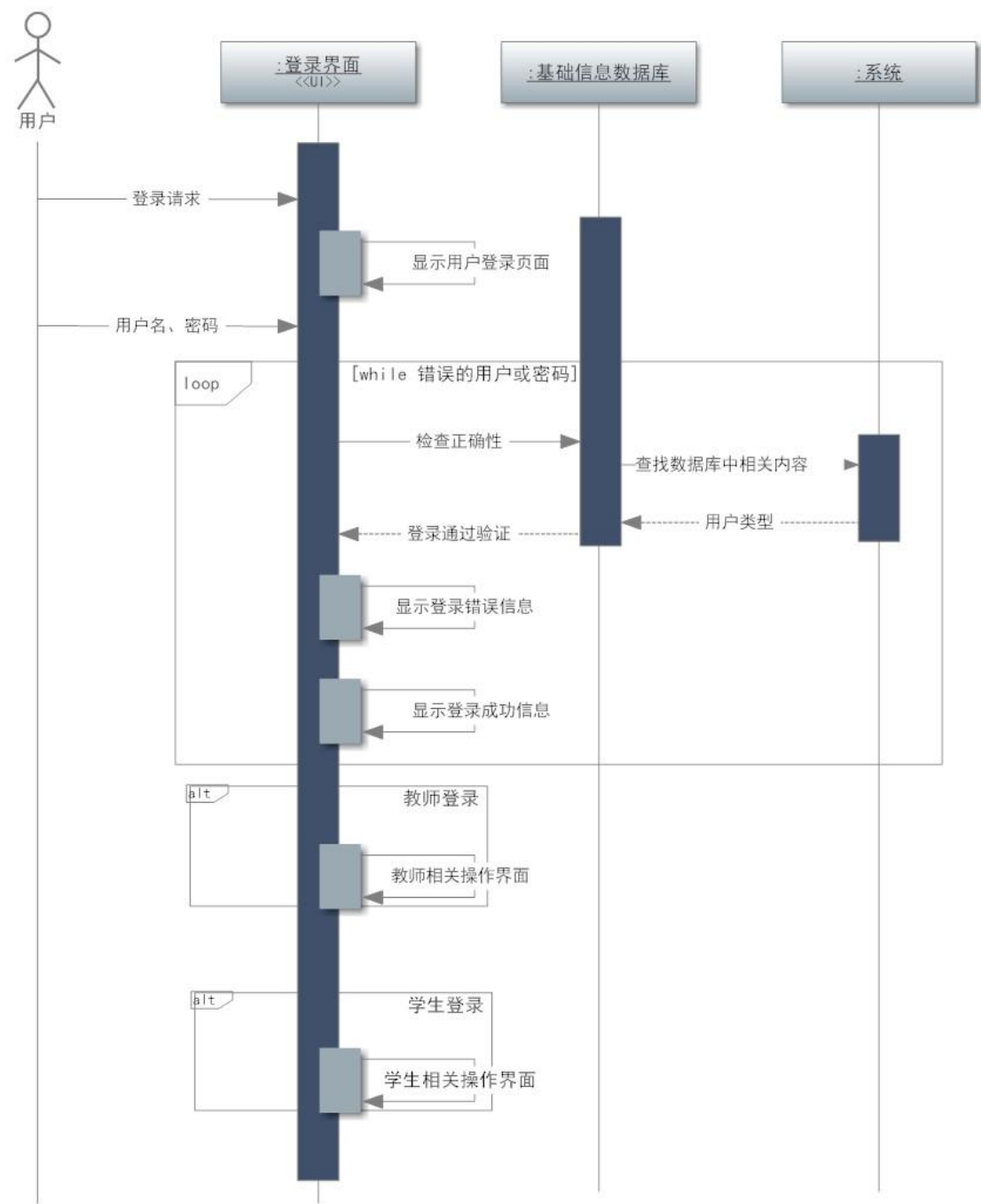
### 3.6 顺序图

为了更好地对整个在线测试子系统的工作流程进行描述,这里给出在线测试子系统的顺序图。下面将根据系统的功能,对各个模块分别给出对应的顺序图。当然,在这里为了让我们的设计思路更加明显,这里我们以功能为依据,对各个模块进行分类。

此外,这里我们的描述都通过文字描述进行,而不采用相关函数名等。相比于采用伪代码进行描述,这样的方式能够让我们的叙述更加直白易懂。

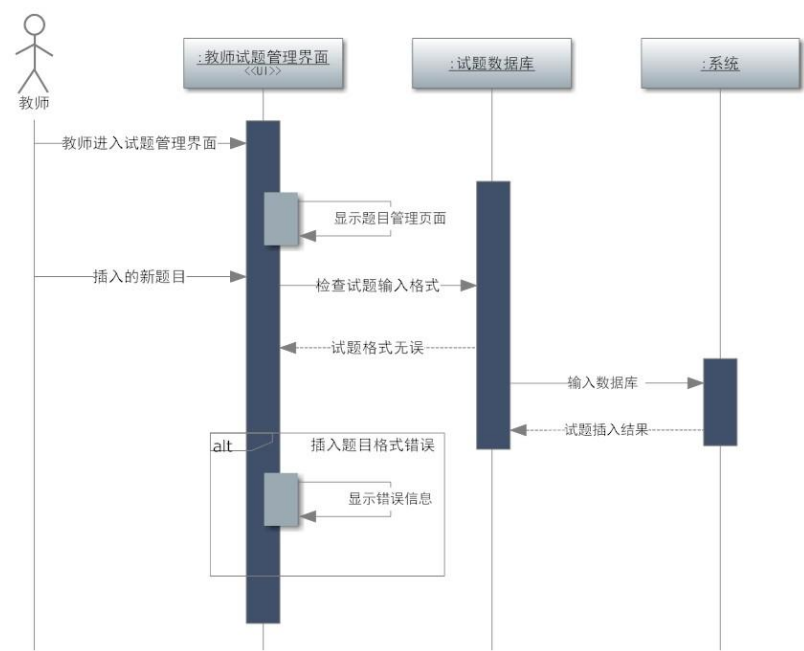
3.6.1 用户登录

用户登录

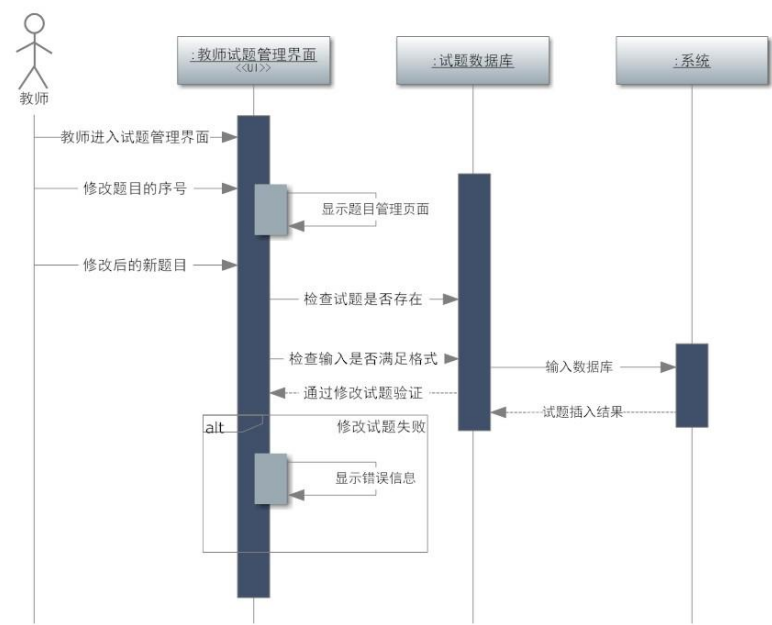


3.6.2 试题管理

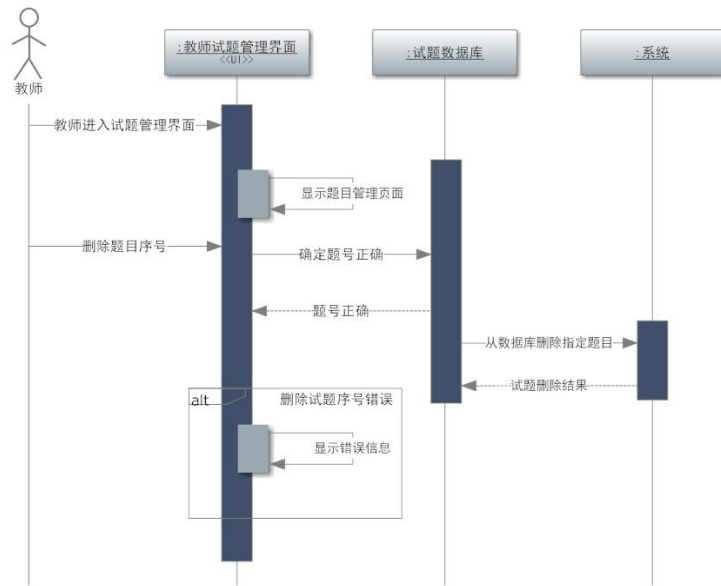
教师试题管理模块——增加试题



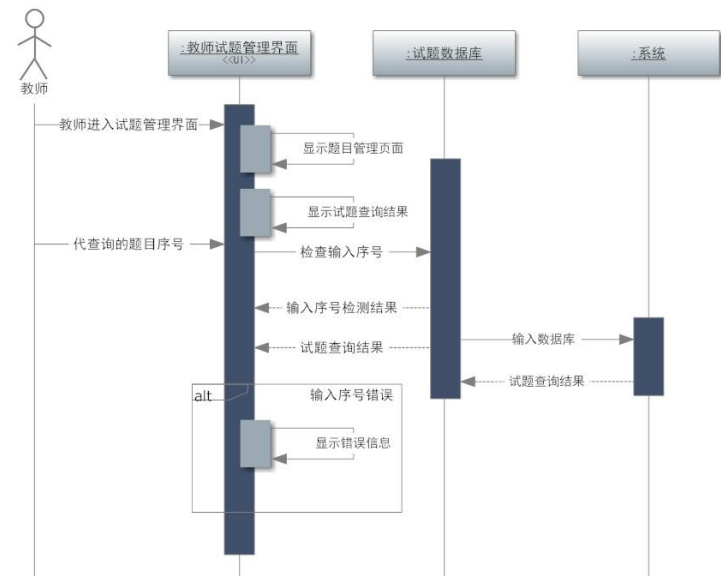
教师试题管理模块——修改试题



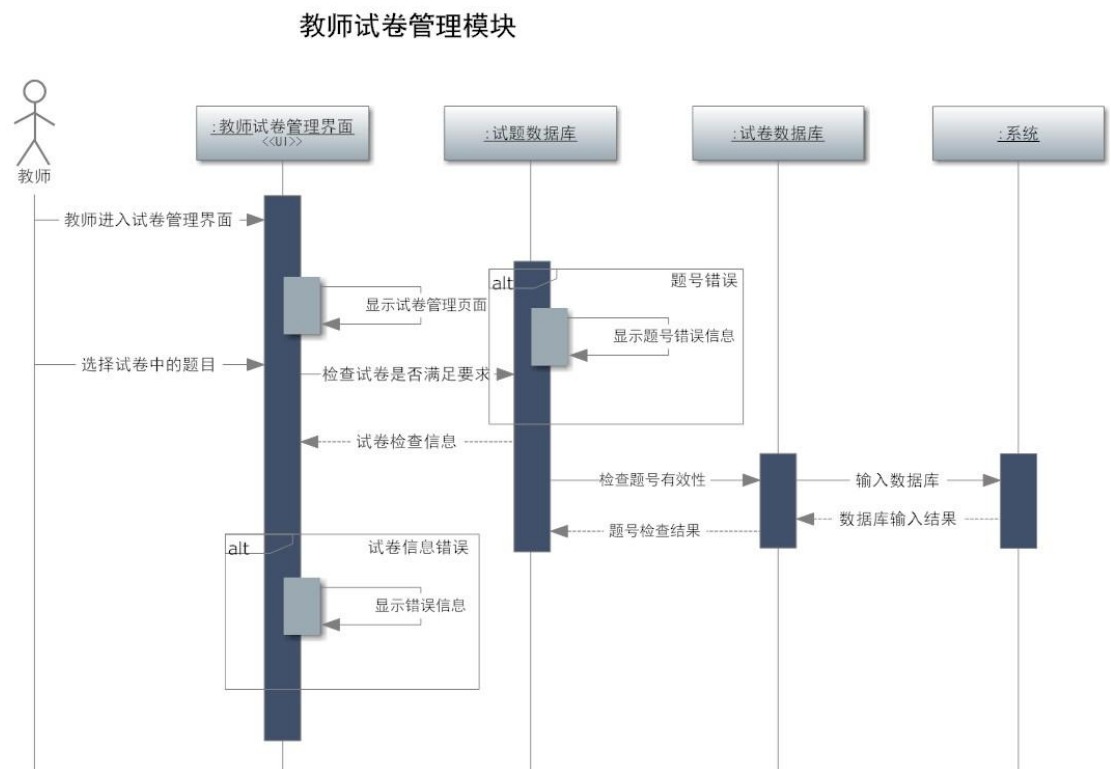
## 教师试题管理模块——删除试题



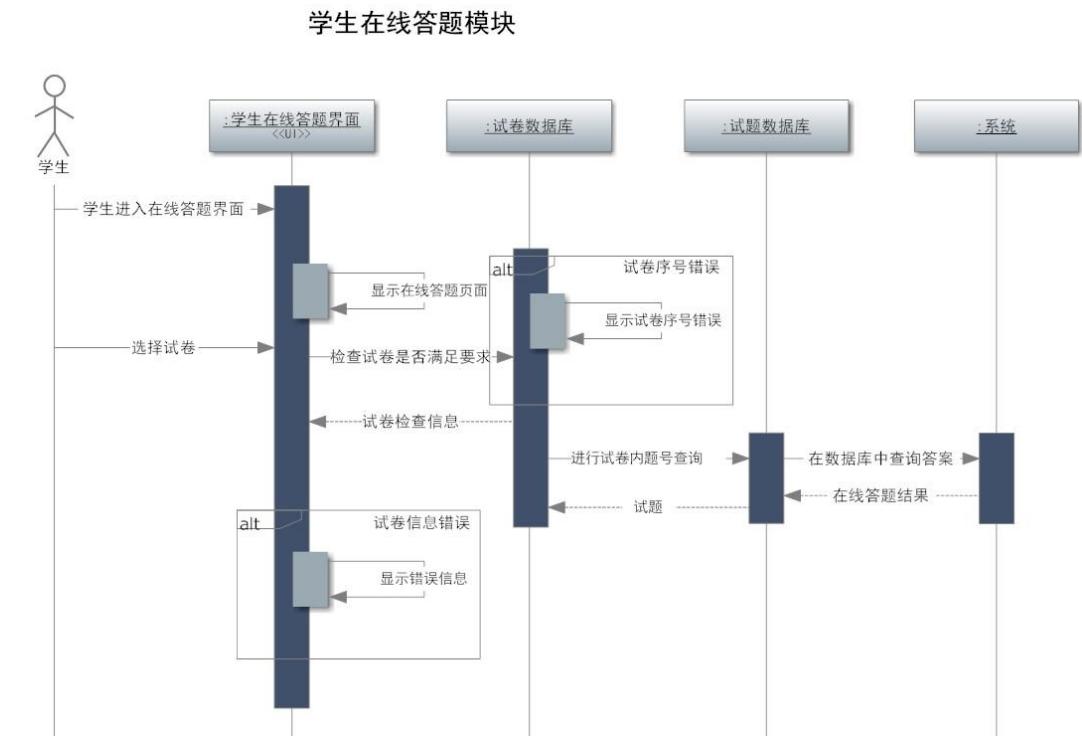
## 教师试题管理模块——查询试题



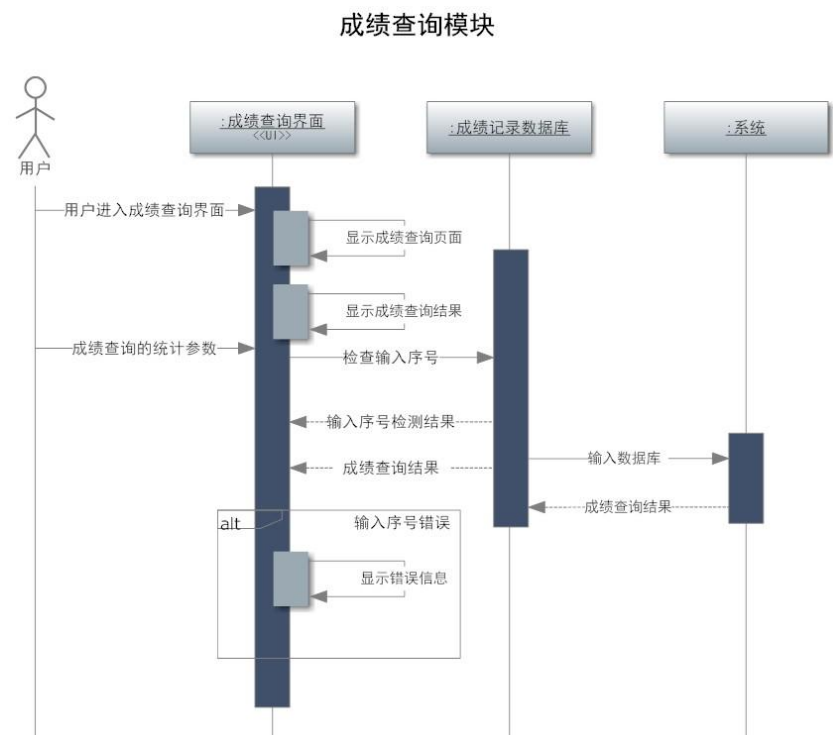
3.6.3 试卷管理



3.6.4 在线答题



### 3.6.5 成绩查询



## 四、执行概念

### 4.1 学生登录

#### 4.1.1 模块概述

本模块在用户使用我们的在线测试子系统时的第一个界面,在这个界面进行身份认证成功后方可使用在线测试子系统的功能。存在的目的是对教师或者学生的用户身份进行认证,从而控制用户对于在线测试子系统的访问。



4.1.2 IPO 图



4.1.3 功能

输入账号密码后登陆，继而进行身份认证。

4.1.4 输入项

名称	标识	类型和格式	输入方式
账户	s_id	char(11)	在账号框手工输入
密码	s_password	tinytext	在密码框手工输入
登陆	login	button	单击按钮

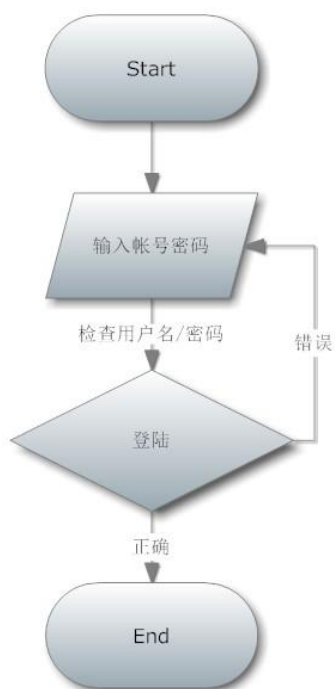
4.1.5 输出项

名称	标识	类型和格式	输出方式
登陆结果	loginresult	bool	由脚本输出

4.1.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
学生登录	登陆界面	输入登录信息，获得输入信息并送至Model进行判断，得到判	存储学生信息，并执行信息的判断方法	显示登录结果	登录有两种结果：成功；失败

### 4.1.7 流程图



## 4.2 教师登录

### 4.2.1 模块概述

本模块在用户使用我们的在线测试子系统时的第一个界面,在这个界面进行身份认证成功后方可使用在线测试子系统的功能。存在的目的是对教师或者学生的用户身份进行认证,从而控制用户对于在线测试子系统的访问。和上面学生登陆不同的是,这个登陆功能主要针对教师。

4.2.2 IPO 图



4.2.3 功能

输入账号密码后登陆，继而进行身份认证。

4.2.4 输入项

名称	标识	类型和格式	输入方式
账户	t_id	char(11)	在账号框手工输入
密码	t_password	tinytext	在密码框手工输入
登陆	login	button	单击按钮

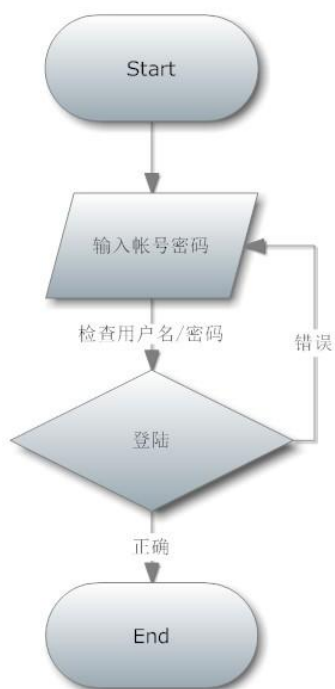
4.2.5 输出项

名称	标识	类型和格式	输出方式
登陆结果	loginresult	bool	由脚本输出

4.2.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
学生登录	登陆界面	输入登录信息，获得输入信息并送至Model进行判断，得到判	存储学生信息，并执行信息的判断方法	显示登录结果	登录有两种结果：成功；失败

## 4.2.7 流程图

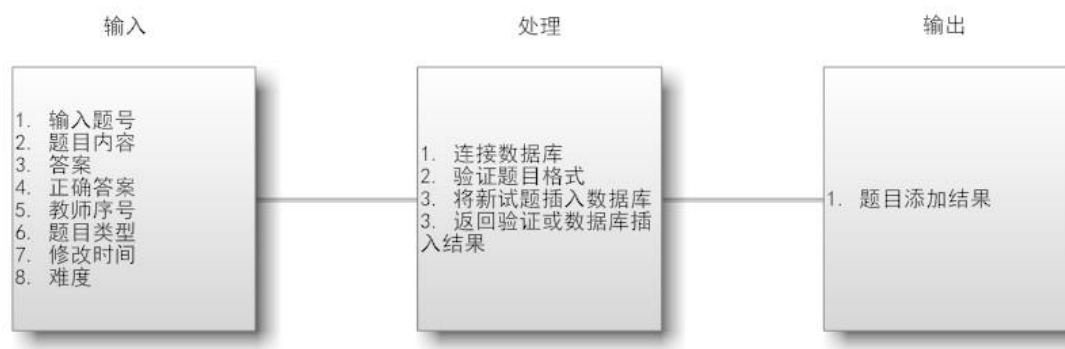


## 4.3 添加试题

### 4.3.1 模块概述

在这个模块中，教师能够对我们的在线测试系统添加试题。此时，教师需要根据我们给出的试题的规范，便能够向数据库中添加题目。

4.3.2 IPO 图



4.3.3 功能

根据相应的输入输出规范向题库中增加试题。

4.3.4 输入项

名称	标识	类型和格式	输入方式
题号	QuestionNo	int(10)	手工输入
题目内容	QContent	varchar(100)	手工输入
答案	QAnswer	varchar(120)	手工输入
正确答案	QAnswer	char(1)	手工输入
教师序号	TeacherNo	int(11)	手工输入
题目类型	QType	enum('Selection', 'Blank', 'TorF')	手工输入
修改时间	LModTime	timestamp	自动获取
难度	HardnessHardness	int(1)	手工输入

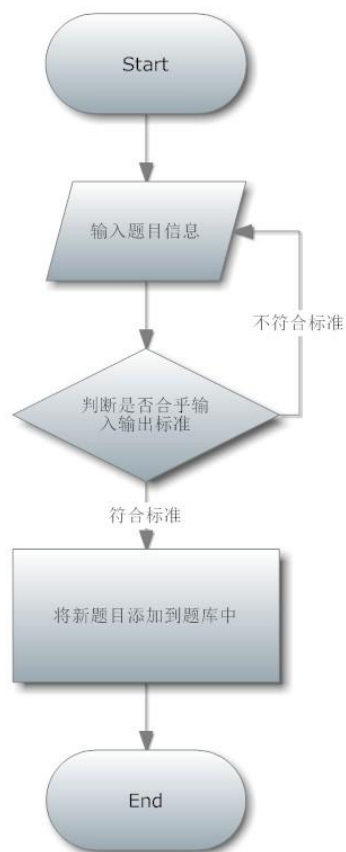
4.3.5 输出项

名称	标识	类型和格式	输出方式
题目添加结果	QuestionAddResult	bool	由脚本输出

### 4.3.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
添加试题	教师添加试题界面	输入新加入的试题信息，获得输入信息并送至Model进行题目格式判断，得到判断结果送至View	存储试题信息，并执行信息的判断方法	显示添加题目的结果	教师添加题目有两种结果：成功；失败

### 4.3.7 流程图

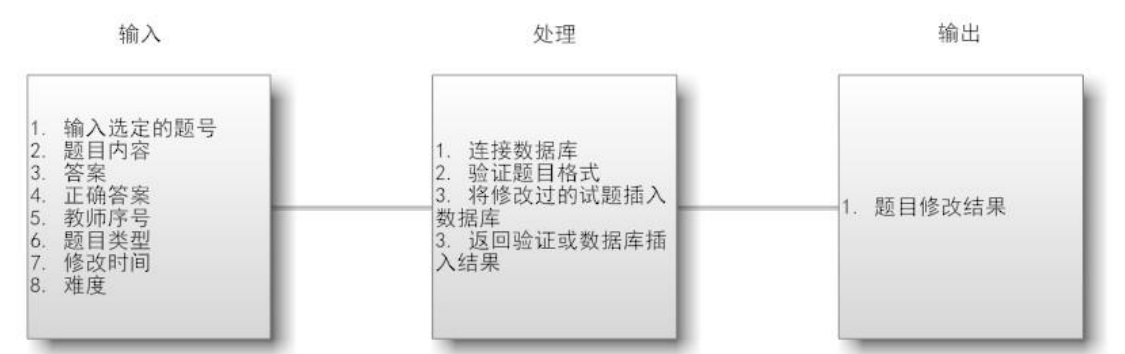


## 4.4 修改试题

### 4.4.1 模块概述

在这个模块中，教师能够对我们的在线测试系统数据库中的试题进行修改。此时，教师需要根据我们给出的试题的规范，便能够对数据库中的某一个选定题号的题目进行相应的修改。

### 4.4.2 IPO 图



### 4.4.3 功能

根据相应的输入输出规范对题库中的某一道选定的试题做出修改操作。

### 4.4.4 输入项

名称	标识	类型和格式	输入方式
题号	QuestionNo	int(10)	手工输入
题目内容	QContent	varchar(100)	手工输入
答案	QPAnswer	varchar(120)	手工输入
正确答案	QRAnswer	char(1)	手工输入
教师序号	TeacherNo	int(11)	手工输入
题目类型	QType	enum('Selection', 'Blank', 'TorF')	手工输入
修改时间	LModTime	timestamp	自动获取
难度	HardnessHardness	int(1)	手工输入

### 4.4.5 输出项

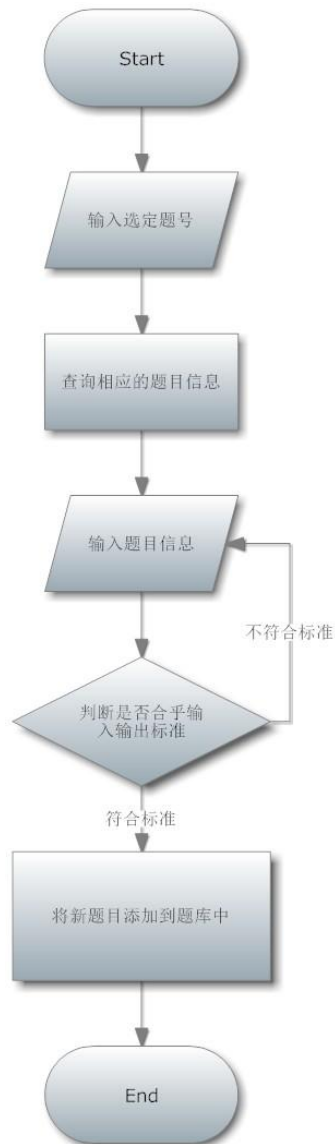
名称	标识	类型和格式	输出方式
题目修改结果	QuestionChgResult	bool	由脚本输出

### 4.4.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
修改试题	教师修改试题界面	输入题号信息与修改后的试题信息，获得输入信息并送至 Model 进行题目格式判断，得到判断与修改结果送至 View	存储试题信息，并执行信息的判断方法	显示修改题目的结果	教师修改题目有两种结果：成功；失败



#### 4.4.7 流程图

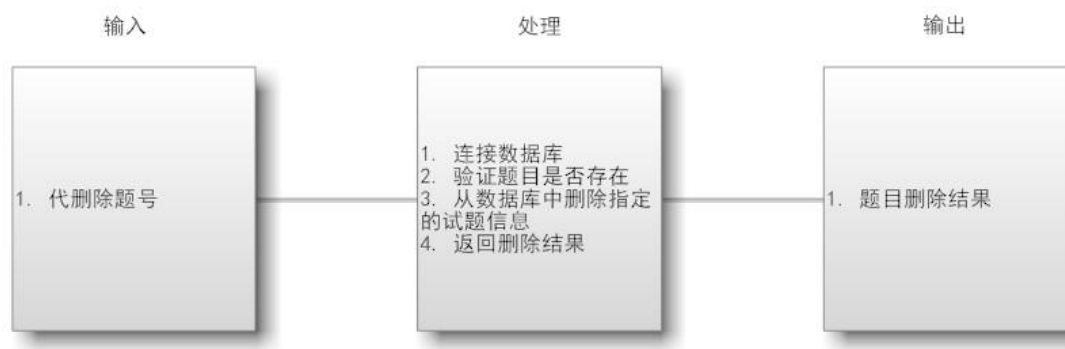


### 4.5 删除试题

#### 4.5.1 模块概述

在这个模块中，教师能够对我们的在线测试中已有的试题进行相关的删除工作。此时，教师需要输入相应题号对某一个选答题号的题目进行删除。

4.5.2 IPO 图



4.5.3 功能

根据相应的输入题号，对题库中某一道已有的试题进行删除操作。

4.5.4 输入项

名称	标识	类型和格式	输入方式
题号	QuestionNo	int(10)	手工输入

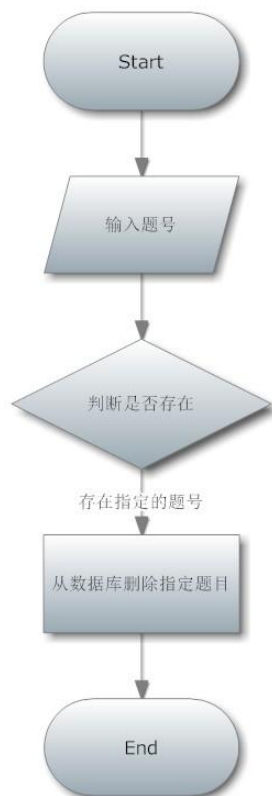
4.5.5 输出项

名称	标识	类型和格式	输出方式
题目删除结果	QuestionDelResult	bool	由脚本输出

4.5.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
删除试题	教师删除试题界面	输入题号信息，获得输入信息并送至 Model 进行查找与删除，得到删除结果送至 View	存储试题信息，并执行信息的判断方法	显示删除题目的结果	教师删除题目有两种结果：成功；失败

### 4.5.7 流程图

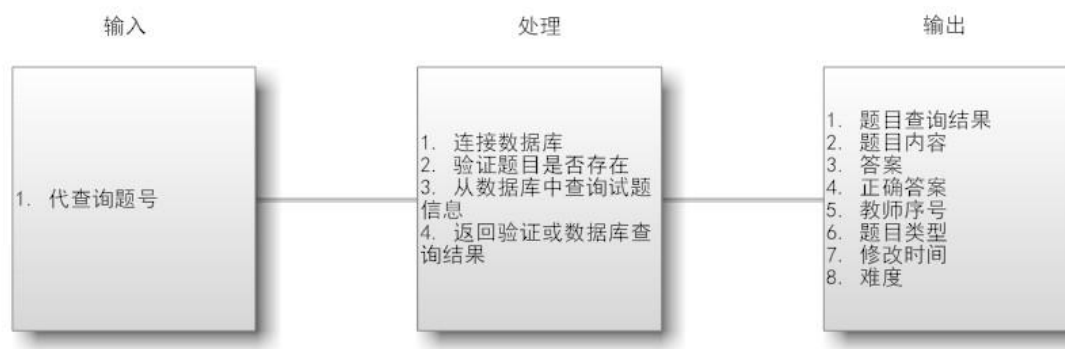


## 4.6 查询试题

### 4.6.1 模块概述

在这个模块中，教师能够对我们的在线测试中已有的试题进行相关的查询工作。此时，教师需要输入相应题号对某一个选定题号的题目进行查询，并且返回相应的查询结果。

4.6.2 IPO 图



4.6.3 功能

根据相应的输入题号，对题库中某一道已有的试题进行查询，并且返回相应的试题信息作为查询的结果。

4.6.4 输入项

名称	标识	类型和格式	输入方式
题号	QuestionNo	int(10)	手工输入

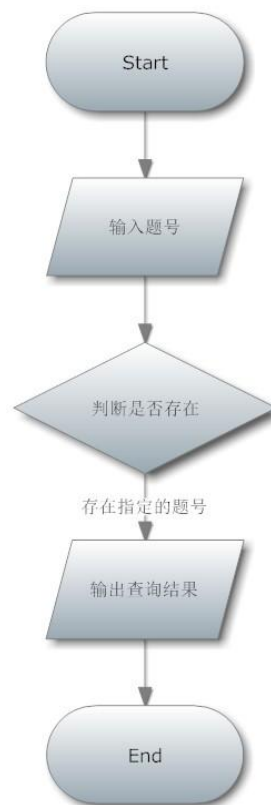
4.6.5 输出项

名称	标识	类型和格式	输入方式
题目查询结果	QuestionQueResult	bool	由脚本输出
题号	QuestionNo	int(10)	由脚本输出
题目内容	QContent	varchar(100)	由脚本输出
答案	QAnswer	varchar(120)	由脚本输出
正确答案	QAnswer	char(1)	由脚本输出
教师序号	TeacherNo	int(11)	由脚本输出
题目类型	QType	enum('Selection', 'Blank', 'TorF')	由脚本输出
修改时间	LModTime	timestamp	由脚本输出
难度	HardnessHardness	int(1)	由脚本输出

### 4.6.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
查询试题	教师删除试题界面	输入题号信息, 获得输入信息并送至 Model 进行查找, 得到结果送至 View	存储试题信息, 并执行信息的判断方法	显示查询题目的结果	教师查询题目有两种结果: 成功; 失败

### 4.6.7 流程图



## 4.7 手动试卷生成

### 4.7.1 模块概述

教师通过手动指定题号进行试卷的生成, 并且将相应的试卷的信息存入我们的在线测试系统的数据库中。

4.7.2 IPO 图



4.7.3 功能

教师通过指定试卷的题号等信息，手动在数据库中创建一张新的试卷。

4.7.4 输入项

名称	标识	类型和格式	输入方式
试卷号	PaperNo	int(10)	手工输入
试题题量	NumQues	int(2)	手工输入
创建时间	CreTime	timestamp	自动获取
第#题题号	Q#No	int(10)	手工输入

4.7.5 输出项

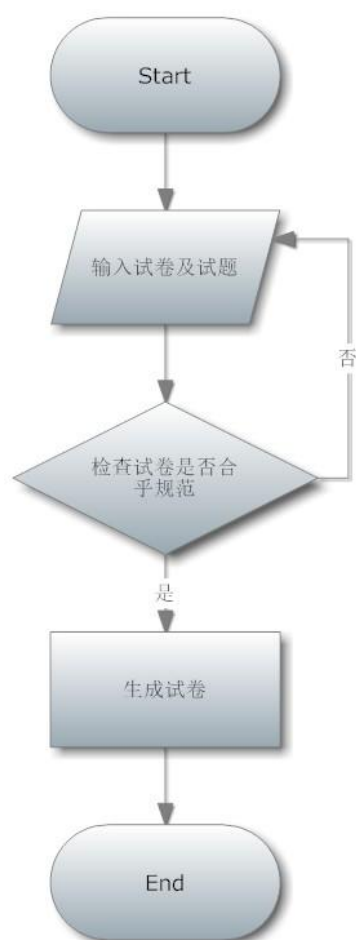
名称	标识	类型和格式	输出方式
创建试卷结果	PaperMannCrtResult	bool	由脚本输出

4.7.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
手动创建试卷	教师手动创建试卷界面	输入新创建的试卷的题	存储试题信息, 并执行信	显示手动创建试卷的结	教师手动创建试卷有两

号信息,获得 输入信息并 送至 Model 进行查找,得 到结果送至 View	息的判断方 果 法	种结果: 成 功; 失败
--	--------------	-----------------

### 4.7.7 流程图

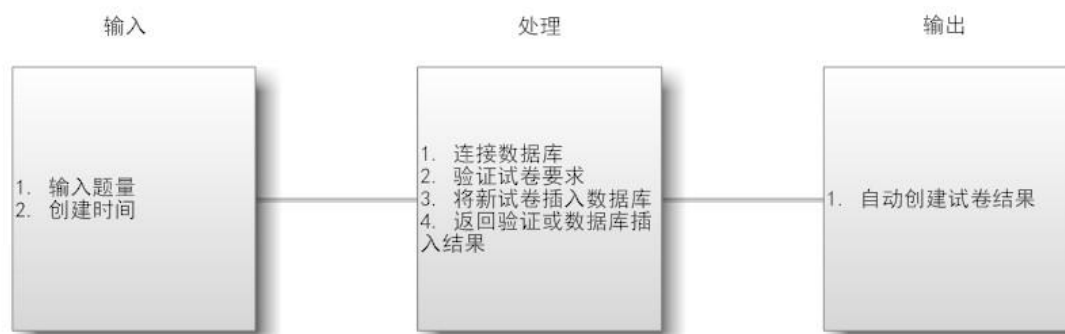


## 4.8 自动试卷生成

### 4.8.1 模块概述

在这个模块中,教师发出自动生成试卷的要求后,在线测试子系统进行自动的试卷生成,并且将相应的试卷的信息存入我们的在线测试系统的数据库中。

4.8.2 IPO 图



4.8.3 功能

自动生成试卷并且相应的存入数据库。

4.8.4 输入项

名称	标识	类型和格式	输入方式
试卷号	PaperNo	int(10)	手工输入
试题题量	NumQues	int(2)	手工输入
创建时间	CreTime	timestamp	自动获取

4.8.5 输出项

名称	标识	类型和格式	输出方式
创建试卷结果	PaperAutoCrtResult	bool	由脚本输出
试卷号	PaperNo	int(10)	由脚本输出
试题题量	NumQues	int(2)	由脚本输出
创建时间	CreTime	timestamp	由脚本输出
第#题题号	Q#No	int(10)	由脚本输出

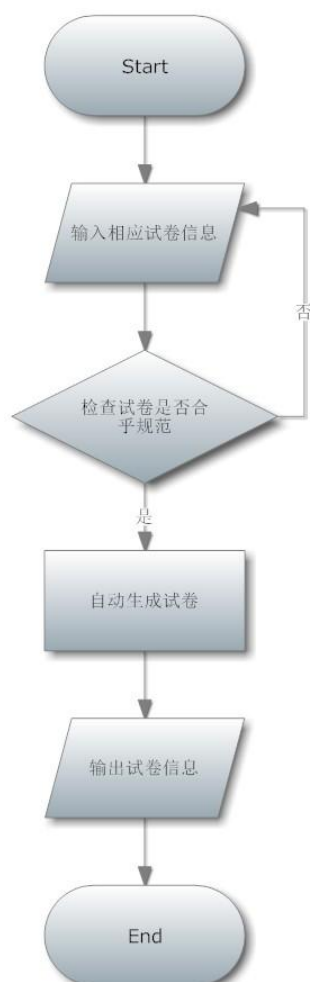
4.8.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
自动创建试	教师自动创	输入新创建	存储试题信	显示自动创	教师自动创



卷	建试卷界面	的试卷的题号信息, 获得输入信息并送至 Model 进行查找, 并且通过自动生成试卷地算法最终完成试卷的创建。得到结果送至 View	息, 并执行信息的判断方法	建试卷的结果	建试卷有两种结果: 成功; 失败
---	-------	--	---------------	--------	------------------

### 4.8.7 流程图

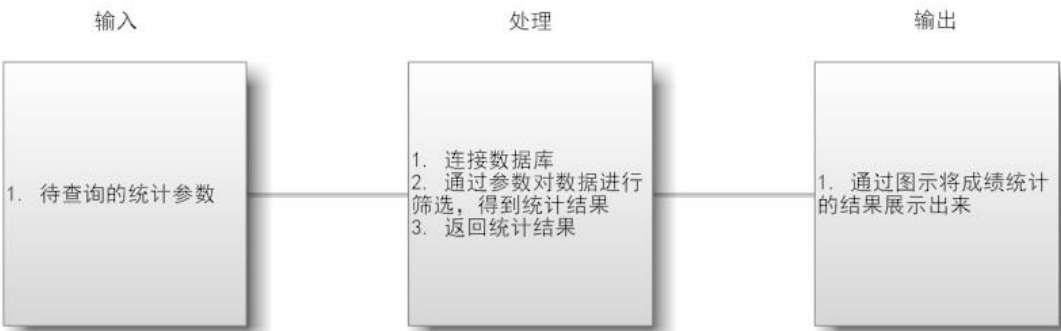


## 4.9 成绩统计

### 4.9.1 模块概述

在这个模块中教师可以按学生、试卷、题型、考察点，显示多种统计数据，提供直观图示，在这个基础之上进行自己对试卷以及学生学习情况的分析。

### 4.9.2 IPO 图



### 4.9.3 功能

按学生、试卷、题型、考察点，显示多种统计数据，并且提供直观图示。

### 4.9.4 输入项

名称	标识	类型和格式	输入方式
统计参数	Arguments	int(10)	手工输入

### 4.9.5 输出项

名称	标识	类型和格式	输出方式
成绩统计结果	StatisticalResult	图示	由脚本输出

### 4.9.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
成绩统计	教师成绩统计界面	输入统计参数, 获得输入信息并送至 Model 进行查找, 并且根据参数进行查询。得到结果送至 View	存储成绩统计信息, 并执行信息的判断方法	显示成绩统计的结果, 并且给出相应的直观图示	结果的表现形式可能会根据具体情况进行确定

### 4.9.7 流程图

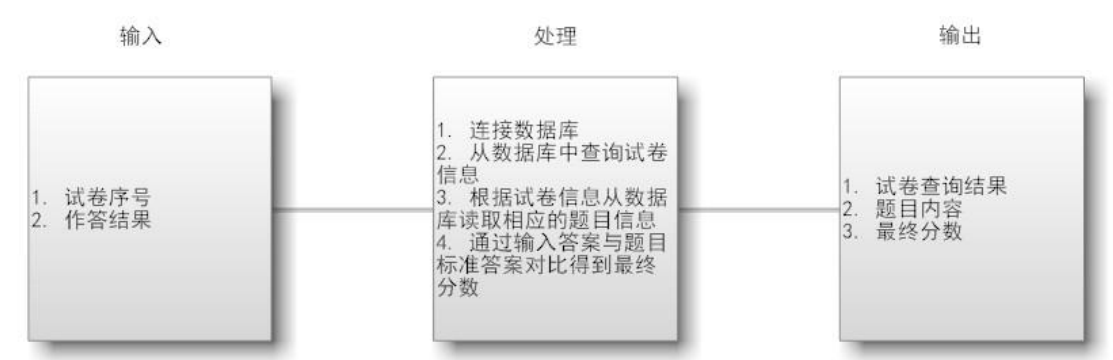


# 4.10 学生在线答题

## 4.10.1 模块概述

在这个模块中，学生登录到我们的在线测试子系统中，并且利用我们的在线测试子系统，通过预存的试卷、试题进行在线测试。在这个模块里，学生可以选择浏览并选择试卷，在线答题。测试开始后开始计时，中途退出无效，提交试卷后自动评分。

## 4.10.2 IPO 图



## 4.10.3 功能

学生浏览并选择试卷，在线答题。测试开始后开始计时，中途退出无效，提交试卷后自动评分。

## 4.10.4 输入项

名称	标识	类型和格式	输入方式
试题序号	PaperNo	int(10)	手工输入
作答结果	StudentAnswer	char(1)	手工输入

## 4.10.5 输出项

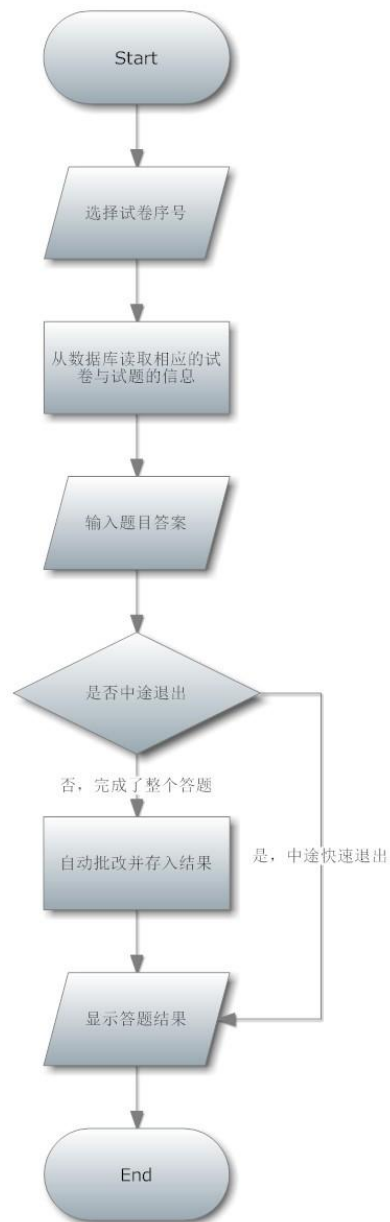
名称	标识	类型和格式	输出方式
试题序号	PaperNo	int(10)	由脚本输出
题目内容	QContent	varchar(100)	由脚本输出

在线答题结果	Score	int(10)	由脚本输出
--------	-------	---------	-------

4.10.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
在线答题	学生在线答题界面	输入相应的试卷选择，获得输入并送至 Model 进行查找，将试题送至 View 输出显示并且读取学生作答结果，最终自动评分	存储试卷、试题、学生成绩信息	显示试卷里的题目，并且给出评价结果	最后答题的结果存入数据库

### 4.10.7 流程图

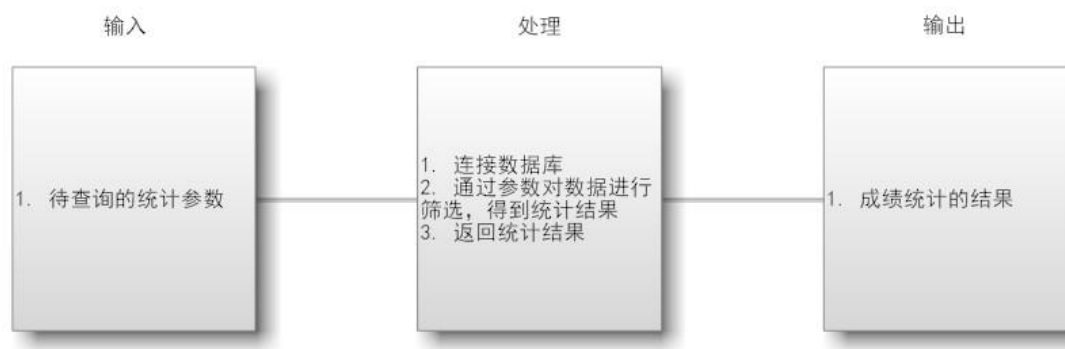


## 4.11 历史查询

### 4.11.1 模块概述

在我们的在线测试子系统中，除了教师可以对成绩进行查询和统计之外，学生也可察看自己在之前的在线测试中的历史成绩。

4.11.2 IPO 图



4.11.3 功能

学生察看历史成绩。

4.11.4 输入项

名称	标识	类型和格式	输入方式
统计参数	Arguments	int(10)	手工输入

4.11.5 输出项

名称	标识	类型和格式	输出方式
历史成绩查询结果	HistoryResult	int(10)	由脚本输出

4.11.6 设计方法（算法）

功能	视图	控制器	模型	结果视图	说明
学生历史成绩查询	学生历史成绩查询界面	输入统计参数, 获得输入信息并送至 Model 进行查找, 并且根据参数进行查询。得到结	存储成绩统计信息, 并执行信息的判断方法	显示成绩统计的结果	具体输入输出形式可能会根据具体情况确定

#### 4.11.7 流程图





# 五、用户界面设计

## 5.1 用户登录



## 5.2 题库管理界面



## 5.3 试卷生成界面



# 5.4 学生答题界面



# 5.5 成绩统计界面

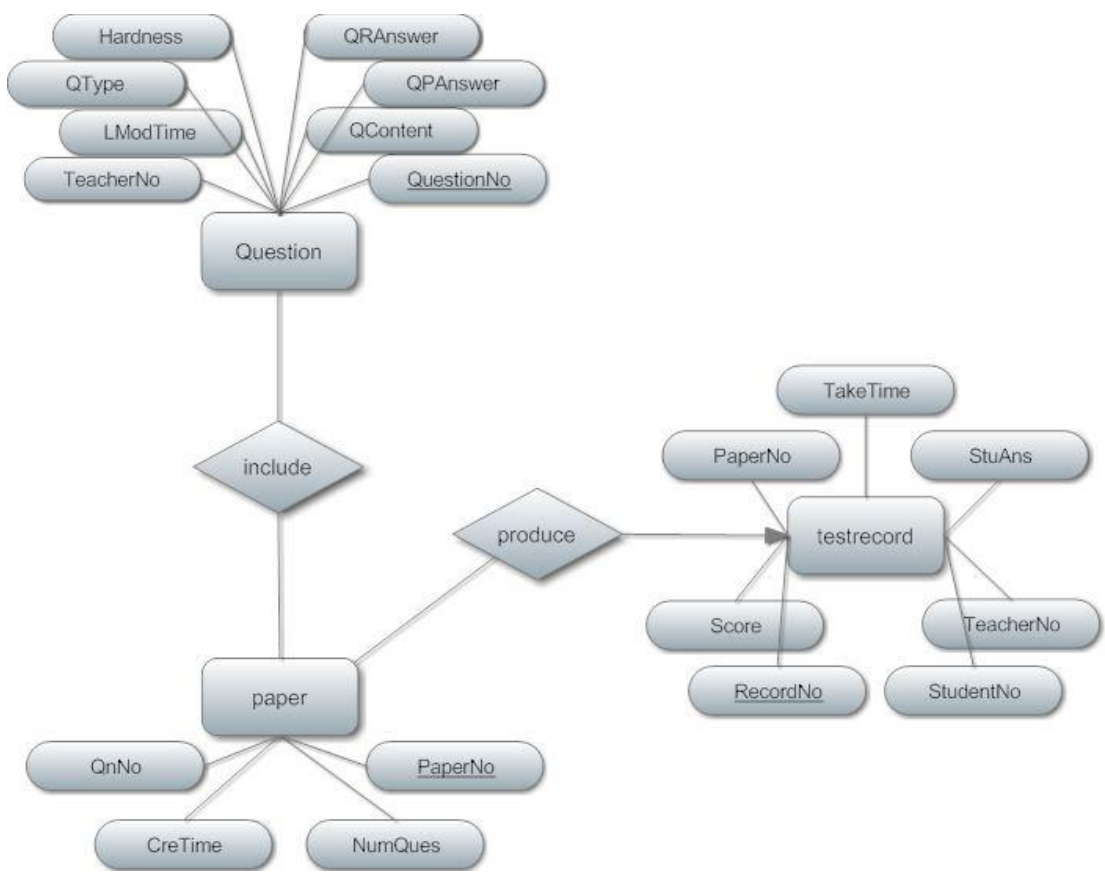


# 5.6 学生历史数据查询

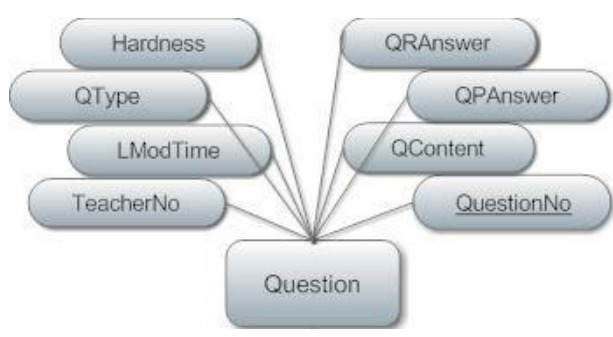


# 六、数据库设计

## 6.1 E-R 图



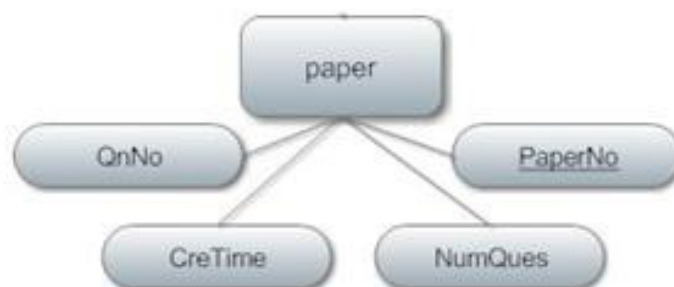
## 6.2 题目实体



#	字段	类型	整理	属性	空	默认	额外
1	QuestionNo	int(10)		UNSIGNED	否	无	AUTO_INCREMENT

D					NCREMENT	
2	QContent	varchar(100)	utf8_general_ci	否	无	
3	QAnswer	varchar(120)	utf8_general_ci	是	NULL	
4	QAnswer	char(1)	utf8_general_ci	否	无	
5	TeacherNo	int(11)	UNSIGNED	否	无	
6	QType	enum('Selection', 'Blank', 'TorF')	utf8_general_ci	否	Selection	
7	LModTime	timestamp	on update CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
8	Hardness	int(1)		否	0	

## 6.3 试卷实体



#	字段	类型	整理	属性	空	默认
1	PaperNo	int(10)		UNSIGNED	否	无
2	NumQues	int(2)		UNSIGNED	否	无
3	CreTime	timestamp			否	CURRENT_TIMESTAMP
4	QnNo	int(10)		UNSIGNED	否	无

## 6.4 考试成绩记录实体



#	字段	类型	整理	属性	空	默认
1	RecordNo	int(10)		UNSIGNED	否	无
2	StudentNo	int(11)		UNSIGNED	否	无
3	TeacherNo	int(11)		UNSIGNED	否	无
4	Score	int(10)		UNSIGNED	否	无
5	PaperNo	int(10)		UNSIGNED	否	无
6	StuAns	varchar(30)	utf8_general_ci		否	无
7	TakeTime	timestamp			否	CURRENT_TIMESTAMP

# 七、接口设计

## 7.1 基础信息管理子系统

在线测试子系统与基础信息管理子系统之间的接口主要就在于数据库方面。这里，根据我们之前大组开会协商的结果，基础信息管理子系统将会负责包括教师信息、学生信息等包含了相关的公用信息的基本数据库。于是在线测试子系统只要与它保持一致即可。

## 7.2 自动排课子系统

根据需求，在线测试子系统与自动排课子系统并没有直接的关联，所以也不会有冲突或者共享。所以不用考虑这两个子系统之间的接口设计。

## 7.3 选课子系统

在线测试子系统与选课子系统之间的联系在于数据库上。这里我们将以选课子系统为准进行数据库的统一。

## 7.4 资源共享子系统

根据需求，资源共享子系统与在线测试子系统并没有直接的关联，所以也不会有冲突或者共享。所以不用考虑这两个子系统之间的接口设计。

## 7.5 论坛交流子系统

根据需求，论坛交流子系统与在线测试子系统并没有直接的关联，所以也不会有冲突或者共享。所以不用考虑这两个子系统之间的接口设计。

## 7.6 成绩管理子系统

根据需求，成绩管理子系统与在线测试子系统并没有直接的关联，所以也不会有冲突或者共享。所以不用考虑这两个子系统之间的接口设计。但是如果成绩管理子系统有必要的话，我们会提供相应的数据。

# 八、系统开发难点分析

首先，在我们程序设计的现阶段，我们最大的难点考虑是如何进行相应的权限控制。比如教师和学生的权限是不同的，而不同权限的用户有着不同的作用。比如，教师能够向数据库中增加题目，但是学生就不能进行这些操作。这些权限的控制应当如何实现便成为了我们的系统的一个难点。这里我们需要在想办法完成权限的控制，从而实现在线测试子系统的功能。

此外，或许我们之后会遇到其他的问题，我们到时候也会再进行深入探讨。



# 九、故障处理

## 9.1 出错信息表

在这里，对于在线测试子系统中可能发生的一些错误，我们对一些可能发生率较高的错误进行了整理并且大致给出了一个相应的处理方法。当然，可能在后面的测试中还会发现其他的问题，我们也会对那些新产生的问题给出妥善的解决。

系统输出信息的形式	解决办法或相应对策
数据库连接失败	修改数据库配置，或者限制并发访问量
用户名密码错误	返回相应的信息给用户
服务器崩溃	定期对服务器进行相关维护工作
磁盘损坏	对磁盘及数据库进行备份。
非法指令	返回相应的处理信息

## 9.2 补救措施

首先，对于数据库，我们会采用定期备份的方法，防止因为硬盘损坏等原因导致数据丢失。并且我们会将数据在不同计算机上进行备份，以防止可能的数据丢失。

此外，我们会考虑对服务器进行相关的维护和优化，防止因为服务器的损坏导致程序无法正常运行。

最后，当程序无法自动处理相应的问题时，我们也会考虑人为介入，并且根据错误日志等进行错误排查，尽快恢复网站正常运行。

## 9.3 系统维护设计

在网站设计时，我们需要考虑使用一些相应的机制来尽可能的处理我们能够预见到的隐患，并且相应地进行处理，或者抛出对应的错误信息。

网站维护人员要能够使用相应权限对数据库进行管理，从而能够人为地干预数据库的异常运行，使之恢复正常工作。

对于外部访问，需要相应地控制其权限，限制其访问和操作仅局限在权限所允许的范围之内，从而防止可能的外来隐患。