

**软件设计**

**文档**

**目 录**

[第一章 软件系统总体结构 2](#_Toc23399)

[1.1 子模块的设计 3](#_Toc16715)

[1.1.1 模块功能 3](#_Toc6584)

[1.1.2 模块结构 4](#_Toc14358)

[1.1.3 输入输出 5](#_Toc21562)

[1.1.4 异常处理 9](#_Toc3970)

[1.2 接口设计 10](#_Toc3646)

[1.2.1 外部接口 10](#_Toc8511)

[1.2.2 内部接口 11](#_Toc6093)

[第二章 数据库设计 12](#_Toc19941)

[2.1 数据库说明 12](#_Toc9478)

[（1）实体 12](#_Toc8021)

[（2）联系 12](#_Toc11652)

[2.2 数据库逻辑结构 16](#_Toc15300)

[第三章 软件重用方案 19](#_Toc4232)

[第四章 设计类的重点服务 21](#_Toc21731)

# 第一章 软件系统总体结构

本系统采用B/S(Browser/Server)三层体系结构作为系统的总体结构，由浏览器Web服务器和数据库服务器组成，并综合运用HTML，CSS，JavaScript，Spring

Spring MVC,Mybatis等技术，由Web Server统一进行管理和发送，用户通过Web浏览器以HTTP协议向服务器发出请求，并接受和显示服务器提供的Web信息，具体如图1-1所示

本系统分为两大部分，教师版系统和学生版系统。教师版系统实现题目的增加、删除、修改、查询等功能，以及对于试题的抽取和试卷的批改，还可实现对学生所在的班级进行管理，对考试结果进行综合分析；学生版系统实现在线考试、刷题和个人成绩查询等功能。

系统总体结构图1-1所示：

|  |
| --- |
| 20190505203822380 |
| 图1-1 系统结构图 |

Web层采用SpringMVC框架，接收用户参数，封装数据，调用业务逻辑层完成处理，转发JSP页面完成显示，其中SpringMVC的组件有前端控制器、处理器映射器、处理器映射器、处理器适配器、视图解析器和视图渲染，其图解结构如图1-2所示：

|  |
| --- |
|  |
| 图1-2 SpringMVC结构图解 |

业务逻辑层通过Spring框架来构建，组合DAO层中的简单方法，形成复杂的功能（业务逻辑操作）

dao层使用MyBatis框架来构建，MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架，其主要就完成2件事情：

①封装JDBC操作

②利用反射打通Java类与SQL语句之间的相互转换

MyBatis的主要设计目的就是让我们对执行SQL语句时对输入输出的数据管理更加方便，所以方便地写出SQL和方便地获取SQL的执行结果才是MyBatis的核心竞争力。

## 1.1 子模块的设计

实质上，计算机科学与工程学院题库系统的综合性相对较强，复杂程度相对较高，可对现有软件进行充分利用，进行系统设计与规划。构建完善成熟的题库管理系统，其中涉及到以下内容，即前台网页界面、处理程序、MySQL后台数据库系统等，在网站页面中显示出以下内容，例如登录页面、课程信息、在线考试、在线作业、在线答疑等。处理程序其实也就是对用户提交表单与相关操作进行处理，存储在后台数据库的信息有用户数据、刷题数据、考试数据、班级数据和分数数据等。

### 1.1.1 模块功能

系统按照功能主要分为学生、老师、系统管理员三个模块，学生具有学生注册、学生登录、学生搜索题库、学生进行刷题、学生查看个人信息；老师具有老师注册登录、老师添加题目、老师设置试卷等功能；系统管理功能包括管理用户、管理科目、管理题目等功能，整个系统模块功能具有如图1-3所示：

|  |
| --- |
|  |
| 图1-3 系统模块功能 |

### 1.1.2 模块结构

系统总体结构可分为系统前台和系统后台两个功能模块。

其中，前台功能实现以下功能，用户注册、用户登录、课程选择、课程信息、在线考试、在线作业、在线答疑、题库、我的提交、我的题库和个人设置。系统前台功能如图1-4所示：

|  |
| --- |
|  |
| 图1-4 系统前台功能 |

系统后台功能实现以下功能，用户管理、题库管理、试卷管理、题型管理和系统设置。系统后台功能如图1-5所示：

|  |
| --- |
|  |
| 图1-5 系统后台功能 |

### 1.1.3 输入输出

输入输出（input/output，I/O）描述的是在计算机上输入输出数据的操作系统、程序或设备。一般的输入输出设备有打印机、硬盘、键盘和鼠标。实际上，有些设备只有输入功能，如键盘和鼠标；有些设备只有输出功能，如打印机；还有些设备具有输入输出2种功能，如硬盘、磁碟和可写性只读光盘（CD-ROM）。

本文主要从程序的输入输出项目来分析一下该系统的输入输出，该系统程序的输入项目如表1-1所示：

**表 1-1 输入项目表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 数据类型 | 输入方式 | 数据来源 |
| 1 | 手机号 | char(11) | 手动输入 | 用户 |
| 2 | 用户名 | varchar(30) | 手动输入 | 用户 |
| 3 | 密码 | char(32) | 手动输入 | 用户 |
| 4 | 学号 | varchar(12) | 手动输入 | 用户 |
| 5 | 题目相关信息 | varchar(50) | 手动输入 | 用户 |

该系统程序的输出项目如表1-2所示：

**表 1-2 输出项目表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 名称 | 数据类型 | 数据来源 |
| 1 | 成绩 | int(10) | 管理员 |
| 2 | 账号是否冻结（默认0未冻结密码） | varchar(30) | 管理员 |
| 3 | 状态（上架1 下架0） | char(32) | 用户、管理员 |

在数据输入输出的基础上对系统数据流向进行分析就比较容易了。首先，数据流程图（Data Flow Diagram，DFD/Data Flow Chart），是一种能全面地描述系统数据流程的主要工具，它用一组符号来描述整个系统中信息的全貌，综合地反映出信息在系统中的流动、处理和存储情况。

其次，它有两个特征：抽象性和概括性。抽象性指的是数据流程图把具体的组织机构、工作场所、物质流都去掉，只剩下信息和数据存储、流动、使用以及加工情况。概括性则是指数据流程图把系统对各种业务的处理过程联系起来考虑，形成一个总体。综合这两个特征对该系统的数据流向进行分析。

（1）顶层数据流图

|  |
| --- |
|  |
| 图1-6 系统顶层图 |

（2）用户登录数据流

|  |
| --- |
|  |
| 图1-7 题库管理系统访问数据流图 |

（3）查询操作数据流

|  |
| --- |
|  |
| 图1-8 题库管理系统数据查询流图 |

（4）试卷数据流

|  |
| --- |
|  |
| 图1-9 试卷数据流图 |
|  |
| 图1-9（续） 试卷数据流图 |

### 1.1.4 异常处理

异常处理（又称为错误处理）是编程语言或计算机硬件里的一种机制，用于处理软件或信息系统中出现的异常状况（即超出程序正常执行流程的某些特殊条件）。其功能提供了处理程序运行时出现的任何意外或异常情况的方法。本系统的异常处理机制如下：

（1）在应用程序遇到异常情况（如被零除情况或内存不足警告）时，就会产生异常。发生异常时，控制流立即跳转到关联的异常处理程序（如果存在）。

（2）如果给定异常没有异常处理程序，则程序将停止执行，并显示一条错误信息。

（3）可能导致异常的操作通过try关键字来执行。

（4）程序可以使用throw关键字显式地引发异常。

（5）异常对象包含有关错误的详细信息，其中包括调用堆栈的状态以及有关错误的文本说明。

（6）即使引发了异常，finally块中的代码也会执行，从而使程序可以释放资源。

（7）异常处理使用try、catch和finally关键字来尝试可能未成功的操作，处理失败，以及在事后清理资源。异常处理通常是防止未知错误产生所采取的处理措施。

采用异常处理机制的好处是开发者不用再绞尽脑汁去考虑各种错误，这为处理某一类错误提供了一个很有效的方法，使项目软件开发效率大大提高。

异常由公共语言运行库(CLR)、第三方库或使用throw关键字的应用程序代码生成。用户也可以用自定义的异常处理类来扩展php内置的异常处理类。以下的代码说明了在内置的异常处理类中，哪些属性和方法在子类中是可访问和可继承的。

## 1.2 接口设计

本系统软件的接口设计遵循六大基本原则：分别是：单一职责原则、依赖倒置原则、接口隔离原则、迪米特法则、里氏替换原则、开闭原则。这6个原则是建立稳定的，灵活和健壮设计的基础，但设计的时候还要根据实际情况考虑。

（1）单一职责原则(SRP)：该原则要求我们在设计类或者接口的时候。尤其在设计接口的时候把职责分清楚，通常一个职责不是单一的方法，是一类方法的组合。

（2）依赖倒置原则：为高层模块不应该依赖底层模块，他们都应该依赖抽象。抽象不能依赖细节。细节应该依赖抽象，这个原则要求设计的时候尽量用抽象（抽象类或者接口）把各个模块独立开来，实现解耦，使各模块相对独立。简单来说就是要用面向接口设计。

（3）接口隔离原则(ISP)：接口分两种，分别为类实例接口和类接口。这个原则依赖建立在最小的接口之上，依赖自己需要的接口。

（4）迪米特法则(LoD)：也称为最少知识原则。就是一个对象应该对其他对象有最少的了解。这个法则的本质就是解耦，解耦是有限度的不能为了解耦而解耦。

（5）里氏替换原则：（所有引用基类的地方，都能透明地使用其子类的对象。）简单来说就是父类出现的地方，替换为子类不会产出异常。里氏替换原则的好处就是增强程序的健壮性，保持程序的兼容性。

（6）开闭原则：软件的实体如类，模块和函数应该对扩展开发，对修改关闭。这个原则要求我们设计的系统扩展性好，因为需要不会一直不变的，我们需要应对的永远是变化。

### 1.2.1 外部接口

（1）用户界面：在界面设计上，应做到简单明了，易于操作，并且要注意到界面的布局，应突出的显示重要以及出错信息。外观上也要做到合理化，考虑到用户多对Windows风格较熟悉，所以该系统尽量向这一方向靠拢。

（2）软件与硬件接口：本系统设有人机操作界面，考虑到操作简单，易于管理方面，主要硬件与接口设备为pc、鼠标、键盘。而软件接口主要以Windows平台为基本平台。

### 1.2.2 内部接口

内部接口也被称为嵌套接口，这意味着在另一个接口内声明一个接口。例如，Entry接口声明在Map接口中。由于各模块之间相互独立又彼此关联，系统主要通过函数调用实现各部分连接。

# 第二章 数据库设计

数据库的设计关系到整个应用系统的运行效率，数据库设计得好，不仅有利于日常数据的维护更新，而且可以提高系统的运行效率，缩短数据查询响应周期，增加网站的流量。合理的数据库设计可以使围绕它支持的Web页面的Java代码简单化，易于实现，并且可以提高数据存储的效率，保证数据的完整一致。校园二手商品交易系统采用MySQL作为后台数据库开发工具。

## 2.1 数据库说明

概念模型用于信息世界的建模，与具体的DBMS无关。为了把现实世界中的具体事物抽象、组织为某一DBMS支持的数据模型。人们常常首先将现实世界抽象为信息世界，然后再将信息世界转换为机器世界。也就是说，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统和具体的DBMS，而是概念级的模型，然后再把模型转换为计算机上某一个DBMS支持的数据模型。实际上，概念模型是现实世界到机器世界的一个中间层次。

信息世界中包含的基本概念有实体和联系。

（1）实体

客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物，也可以是抽象的概念或联系。例如，一个学生、一门课、一个供应商、一个部门、一本书、一位读者等都是实体。

（2）联系

在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是组成实体的各属性之间的联系。两个实体型之间的联系可分为3类，一对一联系，(1:1)；一对多联系(1:n)；多对多联系(m:n)。

概念模型是对信息世界建模，所以概念模型应该能够方便、准确地表示信息世界中的常用概念。概念模型的表示方法很多，其中最为常用的是P.P.S.Chen于1976年提出的实体，联系方法(Entity-Relationship Approach)简记为E-R表示法)。该方法用E-R图来描述现实世界的概念模型，称为实体-联系模型，简称E-R模型。

根据数据流程分析，绘制计算机科学与工程学院题库管理系统的全局E-R模型如图2-1所示。

|  |
| --- |
| 未命名文件 (2) |
| 图2-1 系统全局E-R图 |

题库管理系统主要包含教师实体、学生实体、题库实体、试卷信息实体等，如题库实体的属性共包含试题编号、试题章节、试题难度、试题类型、试题分值、试题内容、试题答案和备注等属性。图2-2是题库实体属性图：

|  |
| --- |
|  |
| 图2-2 题库实体属性图 |

试卷信息实体主要包含：试卷编号、课程号、试题编号、试题章节、试题难

度、试题类型、试题分值、试题内容、试题答案和备注等属性。图 2-3 是试卷信息实体的属性图。

|  |
| --- |
|  |
| 图2-3 试卷实体属性图 |

学生实体主要包含：学号、姓名、性别、出生日期、身份证号码、邮箱

照片、备注等属性。图 2-4 是学生实体的属性图。

|  |
| --- |
| UTLWHPFHH]LM`{9~VO0@R{E |
| 表2-4 学生实体属性图 |

E-R 图是由实体型、属性和联系三个基本要素组成。在题库管理系统中，通过前期的需求分析及其数据库理论知识的理解，得出：

1. 一个专业可以包含多名授课教师，一位授课教师只能归属于一个专业，因此专业和授课教师之间具有一对多的联系。
2. 一位授课教师可以教多门课程（科目），一门课程也可以让多个授课教师去教，因此授课教师和科目之间具有多对多的联系。
3. 一个专业可以有多门课程，而一门课程可以归属于多个专业来管理，因此专业和科目之间具有多对多的联系。
4. 一门课程可以由多个章节构成，而一个章节只能归属于一门课程，因此章节和科目之间是一对多的联系。
5. 一个题型可以包含多个题目，而一个题目只能对应于一个题型，因此题型与题目之间是一对多的联系。
6. 一份试卷可以包含多个题型，而一个题型也可以在多个试卷中出现，因此试卷和题型之间是多对多的关系。
7. 一个科目可以有多份试卷，而一份试卷只能是一门课程所出，因此科目和试卷之间是一对多的关系。

题库与试卷信息之间的 E-R 图如图 2-5 所示：

|  |
| --- |
| 0QY[}O17CQWTL[22%0%0GYF |
| 图2-5 题库与试卷之间的E-R图 |

## 2.2 数据库逻辑结构

数据库逻辑设计主要是把数据库概念设计时设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构。它包括数据项、记录及记录间的联系、安全性和一致性约束等等。导出的逻辑结构是否与概念模式一致，从功能和性能上是否满足用户的要求，要进行模式评价。

通过对系统进行的需求分析，设计出了题库管理系统的主要数据库表。如表 2-1 所示的是学生信息的数据字典，学生信息表主要包含的字段有：学号、姓名、性别、出生年月日、身份证号码、照片、邮箱、备注等。其中学号是学生信息表的主键（主属性），其数据类型是 int 型，不能为空。

表2-1 学生信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生信息表 | | | | | |
| 序号 | 名称 | 允许为空 | 数据类型 | 最大长度 | 是否主键 |
| 1 | 学号 | No | char | 12 | 是 |
| 2 | 姓名 | Yes | varchar | 8 | 否 |
| 3 | 性别 | Yes | char | 2 | 否 |
| 4 | 出生年月日 | Yes | datetime | 8 | 否 |
| 5 | 身份证号码 | Yes | char | 18 | 否 |
| 6 | 照片 | Yes | char | 10 | 否 |
| 7 | 邮箱 | Yes | varchar | 15 | 否 |
| 8 | 备注 | Yes | varchar | 10 | 否 |

教师信息表主要包含的字段有：教师号、姓名、性别、出生年月日、工龄、

教研室、职称、照片、备注等。其中教师号是教师信息表的主属性（主键），其

数据类型是 int 型，不能为空，详细情况见表 2-2 所示：

表2-2 教师信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 教师信息表 | | | | | |
| 序号 | 名称 | 允许为空 | 数据类型 | 最大长度 | 是否主键 |
| 1 | 教师号 | No | char | 10 | 是 |
| 2 | 姓名 | Yes | varchar | 8 | 否 |
| 3 | 性别 | Yes | char | 2 | 否 |
| 4 | 出生年月日 | Yes | datetime | 8 | 否 |
| 5 | 教研室 | Yes | varchar | 8 | 否 |
| 6 | 工龄 | Yes | Int | 3 | 否 |
| 7 | 照片 | Yes | char | 10 | 否 |
| 8 | 邮箱 | Yes | varchar | 15 | 否 |
| 9 | 备注 | Yes | varchar | 10 | 否 |

试卷信息表主要包含的字段有：试卷编号、课程号、试题编号、试题章节、

试题难度、试题类型、试题分值、试题内容、试题答案和备注。其中试卷编号是

主键，其数据类型是 int 型，不能为空，试卷信息表的详细信息如表 4-3 所示。

表2-3 试卷信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 试卷信息表 | | | | | |
| 序号 | 名称 | 允许为空 | 数据类型 | 最大长度 | 是否主键 |
| 1 | 试卷编号 | No | Int | 8 | 是 |
| 2 | 课程号 | No | Int | 12 | 否 |
| 3 | 试题编号 | Yes | Int | 10 | 否 |
| 4 | 试题章节 | Yes | char | 20 | 否 |
| 5 | 试题难度 | Yes | varchar | 5 | 否 |
| 6 | 试题类型 | Yes | varchar | 8 | 否 |
| 7 | 试题分值 | Yes | double | 8 | 否 |
| 8 | 试题内容 | Yes | varchar | 1000 | 否 |
| 9 | 试题答案 | Yes | varchar | 1000 | 否 |
| 10 | 备注 | Yes | varchar | 10 | 否 |

题库管理系统的数据库主要包含学生信息表、题型表、案例分析题表、选课信息表等。表 2-4 是题库管理系统的数据库表的清单。

表2-4系统数据表清单

|  |  |  |
| --- | --- | --- |
| 序号 | 数据表名称 | 数据表用途 |
| 1 | 学生信息表 | 用于保存学生身份信息表 |
| 2 | 教师信息表 | 用于保存教师身份信息表 |
| 3 | 管理员信息表 | 用于存储管理员身份信息表 |
| 4 | 用户权限表 | 保存系统各个用户的权限 |
| 5 | 题库表 | 用于存储所有试卷的表 |
| 6 | 试卷信息表 | 用于存储已组成的试卷 |
| 7 | 单选题表 | 保存系统单项选择题的表 |
| 8 | 多选题表 | 保存系统多项选择题的表 |
| 9 | 填空题表 | 保存系统填空题的表 |
| 10 | 判断题表 | 保存系统判断题的表 |
| 11 | 简答题表 | 保存系统简答题的表 |
| 12 | 名词解释表 | 保存系统名词解释的表 |
| 13 | 案例分析题 | 保存系统案例分析的表 |
| 14 | 论述题 | 保存系统论述题的表 |
| 15 | 计算题 | 用于存储系统计算题型的表 |
| 16 | 科目表 | 用于存储科目信息的表 |
| 17 | 难易程度表 | 存储题目难易程度的表 |
| 18 | 章节表 | 用于存储各科题目章节 |
| 19 | 题型表 | 用于存储各种试题的题型 |
| 20 | 课程号表 | 学院课程号总表 |

对于系统逻辑结构设计的任务是将系统的概念结构设计阶段所设计好的概念模型（E-R 图）成功转化为用户所选用的 DBMS（数据库管理系统）所能够支持的数据模型相匹配的数据库设计。

题库管理系统数据库的核心是系统的数据表。系统中的每个数据表都是根据系统需求分析所取得的数据，在数据库的需求分析阶段根据实际数据之间的关系

而设计起来的数据表。表与表之间的关系也是根据学院教务处的实际业务流程所建立起来的。

## 2.3 数据库实施、运行和维护

在完成系统数据库的物理设计之后便是题库管理系统数据库的实施阶段，是

用 RDBMS（关系数据库管理系统）提供的数据定义语言和其他的应用程序将数据库的逻辑结构设计阶段的逻辑模型和物理结构阶段的物理模型描述出来，并将其转换为数据库管理系统可以接受的源程序，再进行调试产生目标模式，最后组织数据载入数据库的实施阶段。

题库管理系统的数据库实施阶段主要包含对数据的载入和应用程序的编码和调试这两项重要的工作。由于学院教务处的数据量大，数据来源比较庞大而复杂，导致其数据的组织方式、组织结构和数据格式都与本系统的数据库系统有相 当大的差距。为提高系统数据库实施阶段的效率和质量，在系统的需求分析阶段就必须取得学院教务处的基础数据，并根据教务处的实际业务流程，以及各个部门人员的交了与沟通的前提下，尽可能地将系统的数据进行标准化。

题库管理系统数据库的运行阶段首先经历的是系统的试运行阶段。该试运行

阶段的主要工作就是在题库管理系统里面运行数据库的应用程序，并对数据库执

行诸如增加、删除、修改、查询等操作，然后针对试运行阶段的检测数据测试系

统的各个功能，并分析其能否达到用户的实际要求，如果不能够满足需求，必须

对数据库进行修改、调整、测试，直到达到设计需求为止。

系统数据库的试运行成功后，数据库就可以进入正式运行阶段，这也标志着

数据的开发工作基本完成。但是由于正式运行阶段的数据格式、数据的运行环境

以及数据的储存结构可能会发生改变，这就使得数据库必须进入维护阶段。

为减少系统数据库实施和运行阶段的效率，系统数据库的维护之前最好能够

对系统数据库进行备份和还原。并且系统数据库的备份和还原可以避免操作人员

在使用本系统时候，若不慎将数据库的数据删除或者清空数据库等现象，同时也

达到了系统数据库的运行效率。

为保障本系统数据库的维护，设计了数据库备份与还原功能，可以在使用本系统时候时对本系统的数据库进行备份和恢复。因此，在使用本系统时，若不小心清空数据库或者是误删除了一条很重要的数据，这时只要通过数据的备份功能，就不必担心因数据丢失而造成不必要的损失

# 第三章 软件重用方案

从广义的角度来说，本系统在以下三个层次进行了重用，知识的重用，方法和标准的重用以及软件成分的重用。

知识的重用是如软件工程等相关知识的重用，应用相同的知识设计规划、建设不同的软件产品。方法和标准的重用是比如本产品是利用面向对象方法设计又或者是采用国家标准局制定的软件开发文档规范。

软件成分的重用进一步我们划分为三个级别，即代码重用，设计结果重用和分析结果重用。而本软件产品的重用主要在源代码级，这与我们的开发工具有很大的关系，我们选用的开发工具是eclipse和IDE，采用spring MVC框架。利用该框架可以快速开发出一个程序的框架，是一个灵活性较强的框架。

我们的重用设计是建立在MVC框架之上。我们分析可能重用部分的实例有两种方式，一种是继承类库中的构件要用到的基本功能的类。主要是一些界面元素，如菜单活框、列表框，这些构件在很多模块中都要用到，且处理逻辑大致相同，并进行扩充其处理逻辑，如增加输入合法性检测等，这样我们在使用这些经过扩充的构件时不必每个都去重复那些处理逻辑，大大提高了效率，而且，这些构件都是经过测试或其他人使用过的，质量也有保证。

另一种方式是我们的构件在类库找不到相似的类，我们将从头创建自己的类，但为了将自己创建的类，如用户权限处理排序打印预览等，也纳入MVC的框架，我们自己创建的类都继承框架的一个抽象类，这样做的好处是把自己创建的类纳入Controller类的层次化管理，且这也是为实现虚拟函数和动态联编方便。我们也为每个件的功能和接口建立了文档，供应用开发中使用。这样我们就在MVC的框架下，建立了我们自己的一个重用构件库。最后是在应用中重用构件库中的构件。当用户需求变化时，我们能用构件库快速重构我们的应用。

使用重用明显会将开发效率提高一个层次，它不仅提高了开发效率，还保证了应用的风格和质量。特别是对我们这种新手较多的团队开发，采用重用的方法，让有经验的成员负责整个应用的框架，让新手使用重用来创建应用，这非常有利于提高效率，在保证质量的同时也为新手提供一个循序渐进的学习机会，有利于新手的成长。当然使用重用容易，但自己建立重用构件库也是需要代价的，是一个需时间积累的过程，也有一定风险。我们在重用中也遇到了一些问题，一个比较突出的问题是我们的重用主要是在源代码级，因此我们的开发必须在MVC的框架下。

我们的重用主要是在源代码级，通过类的继承来实现。其实可重用的范围是很大的，如设计的重用，测试用列的重用，可运行的代码的重用等。我们想将来扩大重用的粒度，在框架基础上，进一步根据我们单位的软硬件环境定制出一个适用于我们系统的框架，如加进注册功能，数据库连接的功能等。这样，我们可直接重用这个框架，这可以极大地提高软件的开发和维护的效率。最后，我们还设想，将来应有专门的人员管理重用，把重用的维护和应用开发分开，责任明确这样可以更进一步地做好重用工作。

# 第四章 设计类的重点服务

本系统主要设计了三个大的关键类:student,teacher,admin,即学生类，老师类，管理类。基于这三个父类的基础上，继承更行，查询，修改等子类。

设计的类图如图4-1所示：

|  |
| --- |
|  |
| 图4-1 题库系统设计类图 |

# 系统详细设计

系统的各项需求分析与系统的数据库设计是完成整个管理信息系统的基础，

如何实现系统才是题库管理系统的关键。

## 5.1 登录验证模块

### 5.1.1 系统登录机制

所谓系统的登录是指当用户打开应用程序后，用一定的账号和密码登录进入

系统，在系统登录界面，根据用户对输入的身份、账号和密码进行判断，如果与

设置的相一致则正常启动题库管理系统，否则将退出系统。由于本系统主要涉及

学生（或考生）、教师和管理员三种权限，即三个部门，在登录系统时需要进行身份的识别。因此题库管理系统运用了用户登录机制，用户权限图如图 5-1 所示。

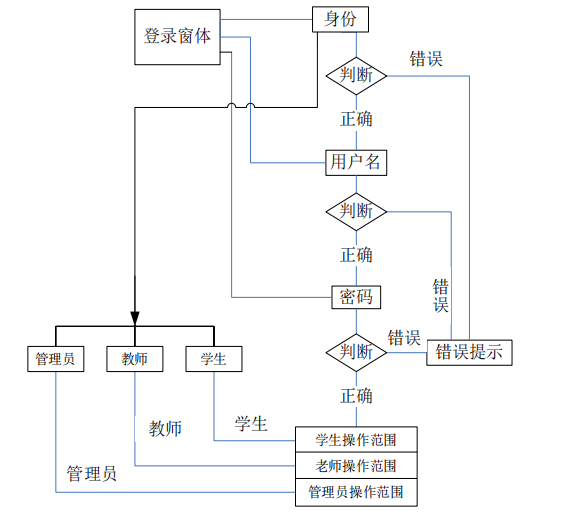


图5-1 身份权限图

登录系统时，以不同身份登录进入题库管理系统，将会具有不同的权限对系

统进行管理。这种登录机制有利于权限的分配。

### 5.1.2 密码修改机制

由于每个用户在使用时会有密码泄露等问题出现，修改密码主要是为了满足

用户当觉得自己的密码的保密性不够好，想对其进行修改。在本次设计要求密码

必须大于 3 个字符。修改密码的流程图为：

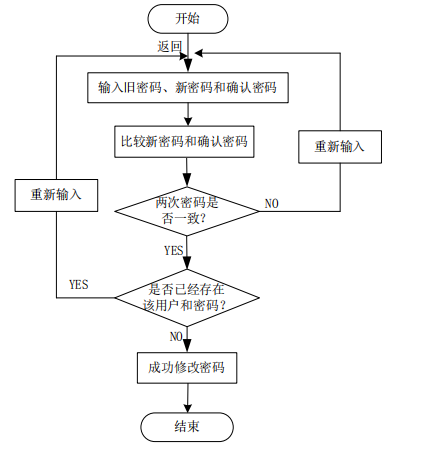


图5-2 修改密码的流程图

## 5.2 主要界面

题库管理系统登录界面，从创新的角度来讲，用户只需要输入用户名和密码，然后选择学生、教师和管理员三种身份中的一种就可以登录题库管理系统，系统会自动识别用户身份判断跳转的页面。从安全性的角度来讲，若用户名和密码不一致或者用户名不存在时，系统会提示错误，因为在设计时，每个页面中增加了对用户姓名的判断的代码：Session["UserName"].ToString()；如果不以注册用户的身份进入则不能访问任何页面。

## 5.3 辅助模块

科目设置、试题的难度设置以及密码修改等一起构成系统的辅助模块，这使

得题库管理系统的试题划分更详细、试卷的组成更具有针对性，试卷的价值更高。

### 5.3.1 设置试题难度

教师在题库管理系统中占据特别重要的地位，从教师操作系统的流程来说，

设置试题的难度系数是试题库的首要工作。

随着科技的发展，越来越多的人员从事过题库管理系统的设计与开发，但经

过前期的需求调研发现，有相当一部分系统在试题细化方面做的不够细致，一般

的题库管理系统只注重是否实现试题的录入、能否生成试卷。在实际的考试过程

中，会有一些特殊情况出现，比如试卷遗失是否有备份试卷；AB 卷的难易程度是否一致等。因此，从考试公平性的角度来讲，同类型的考试试卷的难易程度应该相同才能够体现考试的公平性，在现实的工作过程中，教师在出试卷的时候是极其复杂的事情，不但需要大量的查询资料，还得对平时的试题精选改编，这些都需要花费教师很多宝贵时间，诸如此类的繁琐性问题，就是题库管理系统需要解决的问题，本系统添加了试卷的难易程度、试卷的查询等更加人性化的服务特点。对于题库管理系统中的试题难易程度的设置，可以对试题进行难度系数的设

置，然后利用 Gridview 控件对难易程度进行相应的编辑、修改、删除等操作，从而实现功能需求。

### 5.3.2 添加科目

随着社会的发展，越来越多的新科目伴随着诞生，因此，在题库管理系统中，

需要有创建新科目的功能。本系统的的科目创建与其他的系统有很大的改进，以

往的题库管理系统只能单纯的添加科目的名称，然后保存就表示科目添加成功。

而本系统在此基础上，结合当今学院的科目设置原则，即科目所属的专业、所属系部等情况对科目的划分进行了改良，从而避免了不同专业的同名课程的试题紊乱的现象。系部选择和专业选择的信息的显示可以通过对两个下拉菜单

（DropDownList）进行数据绑定实现的，点击“添加科目”按钮可以进入科目添加

的编辑界面，并通过 Label 标签来显示科目添加是否成功录入。

### 5.3.3 科目管理

添加科目后，需要对已有的科目进行相应的管理，而本系统的科目管理主要

就是实现科目的筛选、查看、修改及删除等功能。因此科目管理界面主要包含两

个部分，一部分是检索筛选功能，另一部分是科目管理即能够查看、编辑科目。

第一，检索筛选功能。检索和筛选功能的目的主要是因为目前各个职业学校

的专业多，科目多，如果需要查询某一科目，单纯一页一页的查询，既浪费时间、

浪费精力，同时出错率还高。用 Gridview 控件对系部名称的识别显示出符合要求的信息，从而实现检索的功能，这样协助教师的教学工作。。

第二，科目管理功能。数据库中的科目的详细信息可以通过 GridView 进行数据绑定来显示科目的详细信息。操作点击删除选项，可以删除勾选的科目信息，

点击编辑选项，可以对系部名称、专业名称、科目名称进行更改。当每个页面的

信息数大于信息的额度时，可以对 GridView 进行了分页设置，并显示分页选项。

## 5.4 试题操作模块

由于在日常的考试中，最常用的考试题型有单选题、多选题、判断题、填空

题和简答题。

试题操作模块主要以数据结构课程为例，分别介绍单选题录入、判断题录入、

简答题录入、判断批量录入和试题维护等方面作为范例。

### 5.4.1 添加试题

图 5-3 和图 5-4 为单选题的录入界面。单选题的设计主要是明确题目的科目、章节、难度系数、选项和答案，可以在单选题的设计界面，用一个文字编辑器（FckEditor）来设计题目部分，由于现在有很多题目不是单纯的能够用文字形式就可以呈现的，还必须掺杂着表格或者图片才能够完全表达题目意思，用文字编辑器（FckEditor）来设计题目，可以解决图文混排的试题形式。另外在单选题的选项数量上页进行了相应的改良，在以往的 A,B,C,D 四个选项的基础上适当地增加E 选项，以达到适应出题方式的变化多样。但是在实际的应用过程中，可以不填写，导出该题答案时，系统也会自动判断该选项是否为空，如果不为空再输出内容，不会出现空选项的现象，对于其它四个选项，本系统对其文本框设置了验证控件（RegularExpressionValidator 控件），如果内容为空的保存则会提示错误。

对于单选表的详细设计，大家可以参考第三章的题库表的物理结构图。单选题和 多选题在设计上几乎没有区别，多选题只是在选项基础上增加了 F 选项，也是为了适应更广泛的用户群体。

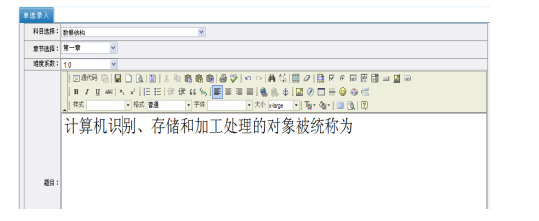


图5-3 单选题录入界面

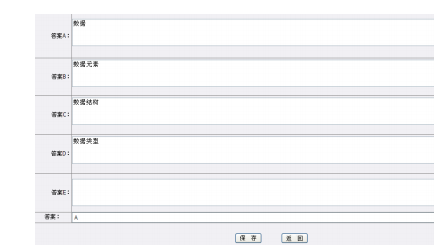


图5-4 单选题录入界面

图 5-5 所示为题库管理系统的判断题录入界面，科目为数据结构。了解数据

结构这个科目的人都知道，数据结构这个科目的试题比其他科目的试题更具有复

杂性，题目中不但有文字和图片，还有各种符号、表格，是题库管理系统中试题

较为繁琐的科目，因此将其作为测试程序的试题再好不过了。判断题界面与其他

各个题型设计上相同的，需要包含科目、章节、难度、题目部分和答案。采用

RadioButtonList 控件实现答案部分，答案的类型在数据库中为 bit 类型，也就是说 True 或者 False（或 0 和 1），而不是以正确的文字形式或者错误的文字形式存储，输入所有内容之后，单击“保存”按钮，就会出现如图 5-6 所示内容的对话框，该对话框采用 JavaScript 技术实现，文字“添加成功”是用户自定义的内容，点击“确定”之后，即完成了整个判断题的录入过程，这时候页面会自动刷新，如需录入新的试题，用户可以再次录入新的试题，在程序中应用 JavaScript 技术可以让界面变得更生动更美观。

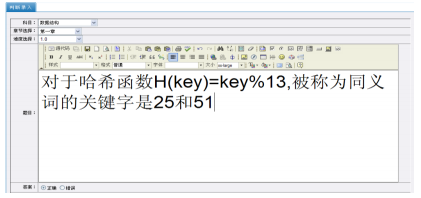


图5-5 判断题录入界面



图5-6 录入成功后弹出对话框

对于图文混排的数据库结构试题的录入，可以设计成如图 5-7 所示的界面结构。如图 5-8 所示试题录入内容，由于文字编辑器（FckEditor）不支持直接粘贴复制图片的功能，因此在录入该试题的时候，只能选题图片和文字两种形式录入，对于一般文字形式的试题可以直接粘贴复制过来，对于图片形式，可以通过单击图标后，出现“图像属性”窗口，用户可以选择点击“浏览服务器”和单击“上传”两种形式上传图片，待图片上传后会显示如图 5-7 所示的界面，然后就可以导出相应的试卷，具体输出样子会在试卷导出部分展示。

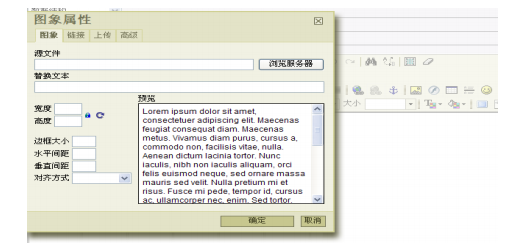


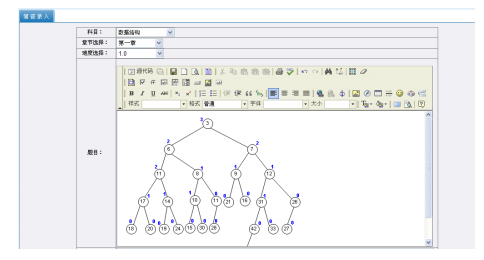
图5-7 简答题添加图片界面

图5-8 简答题录入界面

### 5.4.2 修改试题

单项选择题的维护界面如图 5-9所示，可以通过点击“详细”来修改用户需要

修改的题目。修改试题的实质主要是为了当教师出错题目，可以方便改正，另外

此项设计题目修改还可以实现试题的重复写入，用户可以进入待修改的页面或试

题的录入界面，然后根据需要对需要改动的地方进行调整，修改完成后单击“保存”按钮，这样可以节省大量的题目录入效率。同时也可以让数据库中的数据信息得 到更新，这样做一是为了美观二是实现起来简单。这一功能可以通过GridView 控件上的 HyperLinkField 属性来实现“详细”部分的页面链接。

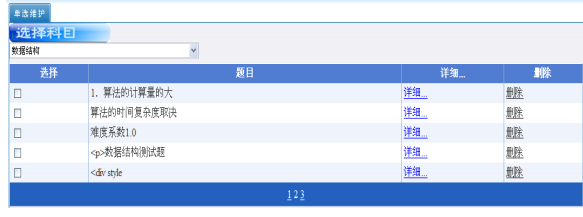


图5-9单选题维护界面

### 5.4.3 删除试题

在单选题的维护界面，可以看到“删除”，删除试题可以直接单击“删除”即可删除相应的题目，如图 5-9所示。在 5-9 图中可以看出，图中的一个下拉列表框（DropDownList）控件，主要是用于选择科目的，下拉列表框中的内容是与数据库中科目表里面的数据像绑定的，在此选择科目的目的主要是为了区分各个试题的类型，这样可以方便试题的查看与管理，直接提高教师的工作效率。对于实体的修改、查找、删除等功能主要是利用一个 GridView 控件绑定实现的，对于修改功能，前面已经介绍过了。这里主要介绍一下删除功能的实现过程。在每道试题的前面有个正方形的复选框，用户可以通过勾选试题前面的复选框然后点击删除链接即可删除该试题，这个删除只是系统对每道题的 ID 的识别，然后通过删除此题的 ID 即删除该试题所有信息，这里的删除操作只能单个删除，后面文章会讲解批量删除的实现。

### 5.4.4 批量导入试题

由于每门课程的判断题的数目都不止一个，有的科目甚至会有上百道判断题，

这样如果一题一题的录入会耗费大量的时间，题库管理系统的批量录入模块可以

解决单一录入试题的繁琐性，节约了广大的教师在录入试题的时间和精力。具体

的批量录入界面如图 5-10 所示。



图5-10 批量录入界面

每道试题的题型都有唯一的模板，而这些模板都是以 Excel 表的形式存在，可以将这些试题的模板储存到模板表中，用户可以下载或直接打开相应的模板，然后设计好相应的试题，这样可以通过“浏览”按钮来进行批量录入。

不光是判断题可以实现批量录入，对于所有的有相应模板可以编辑试题的题

目，都可以通过批量录入模块来提高系统的效率。试题批量录入的步骤如下：

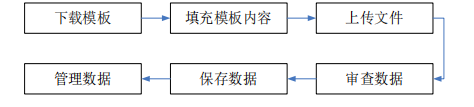


图5-10 试题批量录入步骤

## 5.5 试卷模块

试卷操作这一功能主要是对于管理员设计，在这个功能中，管理员可以对考

生上交的试卷进行查看、删除等操作，这些操作是不允许学生进行的，因为他们

没有权限，权限的设置主要是保障系统的安全和相关考生的权益。题库管理系统

的精髓和最终目标是组卷功能模块。与大多数的题库管理系统相比，本题库管理

系统在自动组卷的基础上，设计了“人工组卷”。人工组卷主要是为了避免机械化的自动组卷，同时也可以更加贴近授课教师的纸质试卷，确保试卷的质量的同时工作效率也得到了提升。

### 5.5.1 组卷

在组卷的环节中，对于教师来说是比较重要的，比如给学生进行复习时，可

以抽取以往考试的试题，即需要抽取现有的试卷；对于需要重新出试卷的可以通

过系统进行自动组卷；对于希望每道试题都是自己制定的试题可以通过人工组卷

的方式进行试卷的重组。

对于抽取已有的试卷，可以直接进入试卷管理界面，然后选择已经生成的试

卷，并点击“导出试卷”，即可导出已有的试卷，对于系统的自动组卷和人工组卷是本系统介绍的重点。

（1）自动组卷

随机组卷就是教师在考试科目、题目难度系数和章节选择确定的情况下，拟定各种题型的数目和分值，然后系统根据教师要求，随机组合成一份试卷。

自动组卷的流程图如图 5-11 所示：

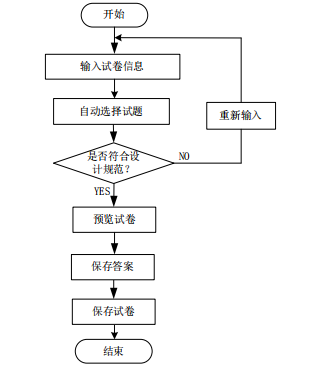


图5-11 自动组卷流程图

（2）人工组卷

相对于自动组卷，人工组卷差别主要在于：自动组卷或随机组卷只需要说明

试卷的题型，每种题型的数目和分值；二是不需要了解每道题的内容；而人工组

卷更能贴近课程的主要内容。教师在人工组卷时需要浏览所组成试卷的每种题

型。然后将需要的试题，在其前面的复选框上打钩，然后单击“保存”即可生产

一份试卷。图5-12就是人工组卷的流程图。

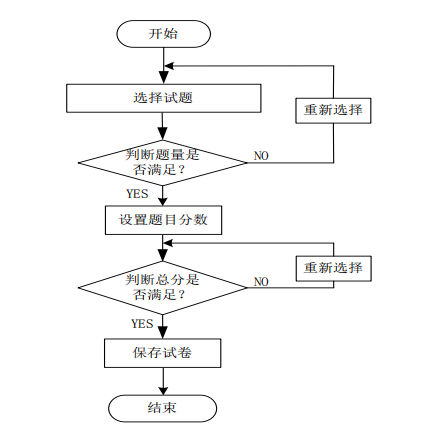


图5-12 人工组卷流程图

### 5.5.2 试卷管理界面

试卷管理界面如图 5-13，为了可以根据不同的考试科目来查看该科目的所有

试卷，在顶端设置了一个下拉列表框的查询项，考试中设计的科目信息、试卷的

名称和状态等信息进行更正、删除和登记操作叫做试卷的管理，通过管理你可以

查看、编辑和修改试卷中的详细的内容。这样的话就可以大量的通过软件高效的

管理试卷，避免一个个的依靠人力去查询和操作了，所以为教师增添这个功能。



图5-13 试卷管理界面