**目 录**

[**1 绪 论** 1](#_Toc40466471)

[1.1 研究的背景及意义 1](#_Toc40466472)

[1.1.1选题的背景 1](#_Toc40466473)

[1.1.2 国内外研究现状 1](#_Toc40466474)

[1.1.3 研究的意义 1](#_Toc40466475)

[1.2 系统目标 2](#_Toc40466476)

[**2 需求分析** 3](#_Toc40466477)

[2.1 业务需求 3](#_Toc40466478)

[2.2 功能需求 6](#_Toc40466480)

[2.2.1 角色分析 6](#_Toc40466481)

[2.2.2 功能需求 7](#_Toc40466482)

[2.3 非功能需求 26](#_Toc40466483)

[2.3.1 环境需求 26](#_Toc40466484)

[2.3.2 性能需求 27](#_Toc40466485)

[2.3.3 安全需求 27](#_Toc40466486)

[**3** **总体设计** 28](#_Toc40466487)

[3.1系统设计的原则 28](#_Toc40466488)

[3.2 系统体系结构设计 28](#_Toc40466489)

[3.3 系统功能模块设计 29](#_Toc40466490)

[**4 数据库设计** 30](#_Toc40466491)

[4.1 概念结构设计 30](#_Toc40466492)

[4.1.1 设计思路 30](#_Toc40466493)

[4.1.2 E-R图 30](#_Toc40466494)

[4.2 逻辑结构设计 34](#_Toc40466495)

[4.2.1 设计思路 34](#_Toc40466496)

[4.2.2 逻辑模型 34](#_Toc40466497)

[4.3 物理结构设计 38](#_Toc40466498)

[4.3.1 存取方式 38](#_Toc40466499)

[4.3.2 存储结构 38](#_Toc40466500)

[**5 界面设计** 39](#_Toc40466501)

[5.1 界面关系图或工作流图 39](#_Toc40466502)

[5.2 界面设计成果 40](#_Toc40466503)

[5.2.1 主界面 40](#_Toc40466504)

[5.2.2 子界面 41](#_Toc40466505)

[**6 详细设计** 45](#_Toc40466506)

[6.1 系统主要功能模块介绍 45](#_Toc40466507)

[6.2 个人信息管理 46](#_Toc40466508)

[6.2.1 功能结构设计 46](#_Toc40466509)

[6.2.2 类图设计 46](#_Toc40466510)

[6.2.3 顺序图设计 49](#_Toc40466511)

[6.2.4 核心处理流程设计 49](#_Toc40466512)

[6.3 用户信息管理 50](#_Toc40466513)

[6.3.1 功能结构设计 50](#_Toc40466514)

[6.3.2 类图设计 50](#_Toc40466515)

[6.3.3 顺序图设计 52](#_Toc40466516)

[6.3.4 核心处理流程设计 52](#_Toc40466517)

[6.4 考试信息管理 53](#_Toc40466518)

[6.4.1 功能结构设计 53](#_Toc40466519)

[6.4.2 类图设计 53](#_Toc40466520)

[6.4.3 顺序图设计 55](#_Toc40466521)

[6.4.4 核心处理流程设计 55](#_Toc40466522)

[6.5 成绩管理 56](#_Toc40466523)

[6.5.1 功能结构设计 56](#_Toc40466524)

[6.5.2 类图设计 56](#_Toc40466525)

[6.5.3 顺序图设计 58](#_Toc40466526)

[6.5.4 核心处理流程设计 58](#_Toc40466527)

[6.6 教学资料管理 59](#_Toc40466528)

[6.6.1 功能结构设计 59](#_Toc40466529)

[6.6.2 类图设计 59](#_Toc40466530)

[6.6.3 顺序图设计 61](#_Toc40466531)

[6.6.4 核心处理流程设计 61](#_Toc40466532)

[6.7 学生成长管理 62](#_Toc40466533)

[6.7.1 功能结构设计 62](#_Toc40466534)

[6.7.2 类图设计 62](#_Toc40466535)

[6.7.3 顺序图设计 64](#_Toc40466536)

[6.7.4 核心处理流程设计 64](#_Toc40466537)

[6.8 帖子管理 65](#_Toc40466538)

[6.8.1 功能结构设计 65](#_Toc40466539)

[6.8.2 类图设计 65](#_Toc40466540)

[6.8.3 顺序图设计 67](#_Toc40466541)

[6.8.4 核心处理流程设计 67](#_Toc40466542)

[6.9 评论管理 68](#_Toc40466543)

[6.9.1 功能结构设计 68](#_Toc40466544)

[6.9.2 类图设计 68](#_Toc40466545)

[6.9.3 顺序图设计 70](#_Toc40466546)

[6.9.4 核心处理流程设计 70](#_Toc40466547)

[**7 编码** 71](#_Toc40466548)

[7.1 代码实现与核心算法 71](#_Toc40466549)

[7.2 代码优化分析 75](#_Toc40466550)

[**8 测试** 76](#_Toc40466551)

[8.1 测试方案设计 76](#_Toc40466552)

[8.1.1 测试策略 76](#_Toc40466553)

[8.1.2 测试进度安排 76](#_Toc40466554)

[8.1.3 测试资源 76](#_Toc40466555)

[8.1.4 关键测试点 76](#_Toc40466556)

[8.2 测试用例构建 77](#_Toc40466557)

[8.2.1 测试用例编写约定 77](#_Toc40466558)

[8.2.2 测试用例设计 77](#_Toc40466559)

[8.2.3 关键测试用例 77](#_Toc40466560)

[8.2.4 测试用例维护 78](#_Toc40466561)

[**9 总结与展望** 79](#_Toc40466562)

[9.1 设计工作总结 79](#_Toc40466563)

[9.2 未来工作展望 79](#_Toc40466564)

[**谢 辞** 80](#_Toc40466565)

[**参考文献** 81](#_Toc40466566)

[**附录A 外文翻译—原文部分** 82](#_Toc40466567)

[**附录B 外文翻译—译文部分** 85](#_Toc40466568)

[**附录C 软件使用说明书** 88](#_Toc40466569)

[**附录D 主要源代码** 92](#_Toc40466570)

1 绪 论

## 1.1 研究的背景及意义

### 1.1.1选题的背景

世界各组织的管理方式从根本上改变是因为网络的快速发展，自二十世纪九十年代开始，可以通过互联网管理信息这一设想出现在我国的组织机构的计划中。当时妨碍了各大机构发展互联网的原因存在很多，例如机构的名誉非常不容易被认可证明、互联网的相关法律法规也非常欠缺、开发技术也不够成熟等。自从进入二十一世纪来到以后，我国快速发展了之前落后的经济，受到之前经济影响的机构管理被限制的困难被逐一攻克解决，国内各大机构都选择参加到电子信息化的管理形式中来。

近年来，在互联网的火速发展下，局域网的普及，互联网的建立，为本中等学校通用IS平台的建立提供了基础条件。中等学校通用IS平台与传统的中等学校信息管理方式相比，有着无法相比的优势，网络共享、传播速度快的特点，在某些程度上提高了中等学校信息的管理效率。

### 1.1.2 国内外研究现状

近年来，因为网络快速发展和世界的关于网络的科学技术快速发展的原因，互联网开始被广泛的认识了。使用计算机进行传统信息管理逐渐替代传统的管理方式也成为一种新的趋向。截止到现在，全球市值最高的十家公司里，有7家科技公司在管理信息的时候是通过使用电脑的，其中，利用以前传统方式管理的实体企业被发展较晚的中国公司科技公司跨越了，现如今的社会更像是人工智能开展，像无人驾驶，无人机等是一种史无前例的实验，未来将会把很多不可能的东西或想法变为可能，所有的企业在未来都将脱离不了互联网。虽然国际市场比国内市场大得多，但因为互联网的发展的原因，已经迈出了国门的会计管理和办公管理都达到了世界一流的水平。所以在以后，这个世界将会大大的改变我们现在对它的认知，而社会也将会被互联网改变。

世界上第一台计算机占地面积非常庞大，处理速度慢，只能用于科学计算，一次正常的开机都要花费许多许多的时间，并且必须被专业的操作人员使用，都是用代码实现算法，非常的复杂。自从Microsoft创建了操作系统以来，人们可已经能够直观的操作计算机，并且电脑的用途也变得多样化起来。计算机的发展为企业提高管理效率，宣传发布信息等提供了更好的网络商机。网络的发展都为中等学校通用IS平台的设计与实现提供了良好的基础，在网络和计算机的大力发展下，中等学校通用IS平台开始出现，中等学校通用IS平台是借助网络和计算机的无纸媒体，既节省了纸张的浪费，又保证了信息的实时性。

### 1.1.3 研究的意义

日前信息化建设快速发展的现在，很多中等学校的信息管理水平还停留在纸质通知、黑板通告和广播通报这些传统手段上，管理方式极有待提高，自主研发一款通用的适用于各中等学校的IS平台能提高学校信息管理的效率，不再通过纸质通知从而节省了资源，避免了浪费，同时传递消息速度快速，提高了工作效率。

本平台可以提供一种方便快捷的交流方式给师生和家长，而且教师还能通过立体的教学管理方式进行教学。学生和家长在任何时候任何地方都可以了解到自己或小孩的状态。家长只要通过平台而不用通过电话与老师联系就能掌握自己小孩的状态，这样的方式也可以减轻老师来自工作上的压力。

通过对多数学校的调查发现，已经建设并投入运行通用IS平台的学校廖廖无几，所谓的校园网也只是为了教务综合管理、图书资料的查阅或通知公告的发布，还没有起到办公、教学、服务的目的。

## 1.2 系统目标

本平台的目的在于建立一个易于操作、便于管理、高效稳定、安全可靠的中等学校通用IS平台，实现学校老师与学生和学生家长之间进行交流的服务。提高学校的信息管理效率，为满足管理员、老师、学生、学生家长等四者之间的需求。最终目标就是开发一个功能较为齐全，用户拥有良好的体验的中等学校通用IS平台。

对于老师这一个角色在本平台中具有个人信息的修改、查询学生的信息、上传学生考试的成绩、上传老师自己所教学科的教学资料、登记学生各个方面的表现、在论坛上进行交流等功能。

对于学生这一个角色在本平台中具有个人信息的修改、查询自己的考试成绩、下载老师上传的教学资料、查看老师登记的各个方面的表现、在论坛上进行交流等功能。

对于学生家长这一个角色在本平台具有个人信息的修改、查询自己孩子的个人信息、查询自己孩子的考试成绩、查看老师给自己孩子登记的各个方面的表现、在论坛上进行交流等功能。

对于管理员这一个角色在本平台具备管理用户、日志等功能。

# 2 需求分析

## 2.1 业务需求

1. 老师可上传教学资料，查看学生信息，上传学生成绩，记录学生的表现，在论坛发帖交流。
2. 学生能够查看考试的成绩分数、查看老师给登记的评语、下载老师发布的教学资料、在论坛上进行交流。
3. 家长能够查看自己小孩的个人信息、考试成绩、老师给登记的评语、在论坛上进行交流。
4. 管理员具备管理维护系统的功能。
5. 系统的响应速度应该合理，安全性应该较高。
6. 系统运行要稳定，要容易维护。

### 2.1.1主要业务流程

1. 用户个人信息修改活动图如下图2-1所示：

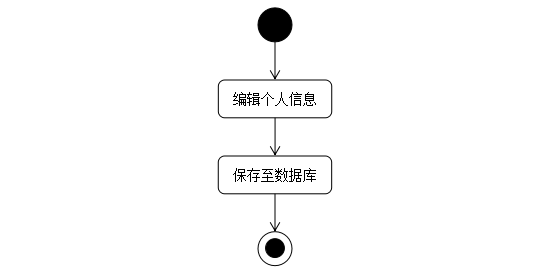


图2-1 个人信息修改活动图

1. 教师上传教学资料活动图如下图2-2所示：

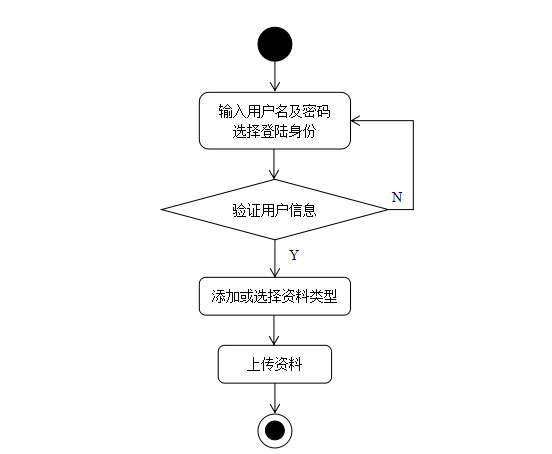


图2-2 上传教学资料活动图

1. 教师上传成绩活动图如下图2-3所示：

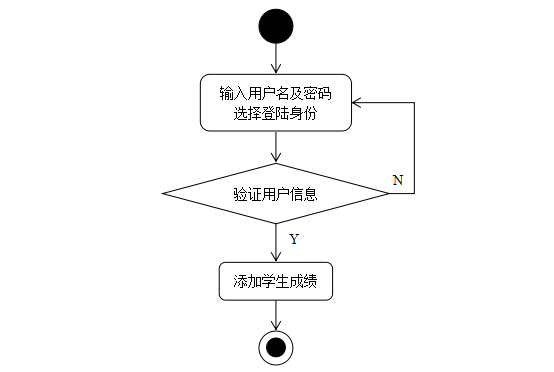


图2-3 上传成绩活动图

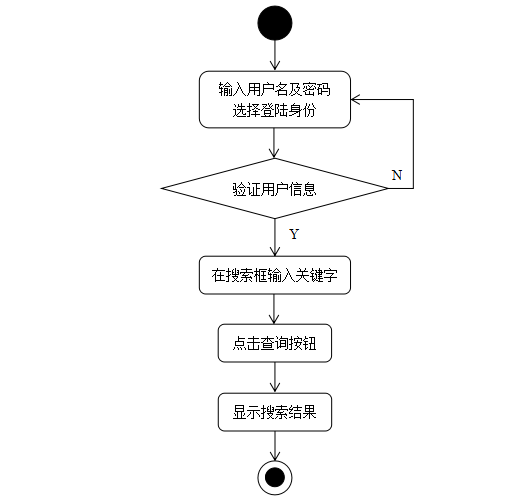
1. 用户搜索活动图如下图2-4所示： 

图2-4 搜索活动图

1. 用户回复活动图如下图2-5所示：

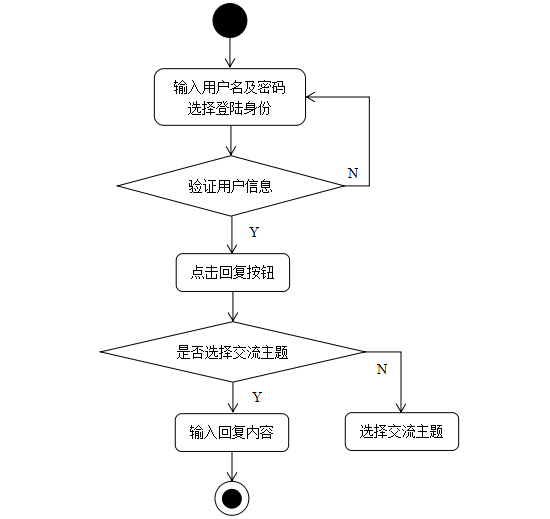


图2-5 回复活动图

## 2.2 功能需求

### 2.2.1 角色分析

本系统涉及到以下角色，如下表3-1所示：

表3-1 系统角色表

|  |  |
| --- | --- |
| **角色** | **职责或功能** |
| 老师 | 修改个人信息、查看学生信息、上传学生成绩、上传教学资料、记录评语、发帖交流 |
| 学生 | 修改个人信息、查询考试成绩、下载教学资料、查看评语、发帖交流 |
| 家长 | 个人信息的修改、查看自己孩子的个人信息、查看自己孩子的考试成绩、查询老师给自己小孩记录的评语、发帖交流 |
| 系统管理员 | 管理用户、系统数据等 |

### 2.2.2 功能需求

以下从用户角度出发，给出了平台的总体用例图，包含用户管理，考试管理，教学管理，论坛管理等用例，如下图2-6所示：

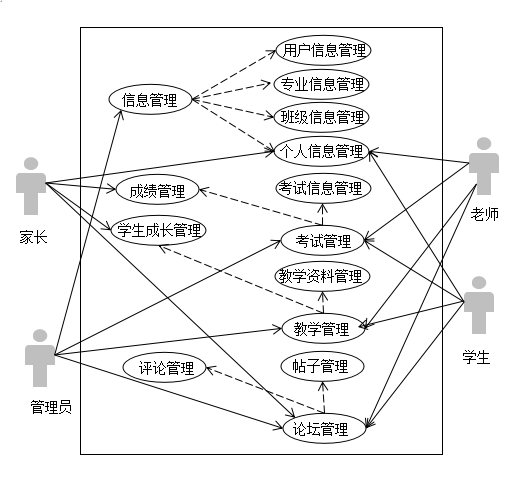


图2-6 系统总体用例图

2.2.2.1信息管理

信息管理包括四个模块：管理个人信息的模块，管理用户的模块，管理专业的模块,

班级信息管理模块。管理员可以通过平台更改其帐户的密码。管理员可以执行增、删、改、

查操作在用户、专业和班级信息模块。在信息管理这个模块，师生和家长这三类用户依据

自身情况更改信息和密码。该模块的用例图如下图2-7所示：

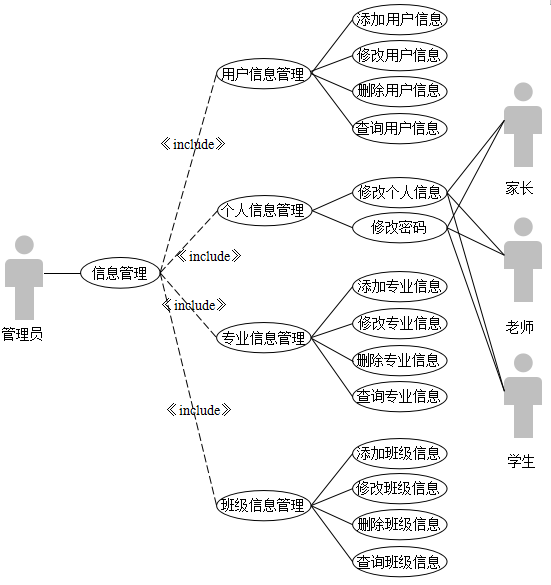


图2-7 信息管理用例图

修改个人信息用例分析表如下表2-2所示：

表2-2 修改个人信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改个人信息 |
| **标识符\*** | 01 |
| **用例描述** | 用户个人信息修改的过程 |
| **参与者表** | 管理员、家长、老师、学生 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |

续表2-2

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **基本操作流** | 1. 用户在登陆平台后，进入平台主界面 2. 点击左侧个人信息菜单栏，显示有两栏菜单：个人信息和修改密码 3. 点击个人信息进入页面，找到修改按钮后点击 4. 填写所需要修改的信息后点击保存 |

修改密码用例分析表如下表2-3所示：

表2-3 修改密码用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改密码 |
| **标识符\*** | 02 |
| **用例描述** | 用户依据情况修改密码的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 用户修改密码成功 |
| **基本操作流** | 1. 用户通过用户名和密码登陆平台后，进入平台主界面 2. 点击左侧个人信息菜单栏 3. 点击修改密码进入页面，填写原来的密码和两次新的密码，点击修改 |

添加用户信息用例分析表如下表2-4所示：

表2-4 添加用户信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加用户信息 |
| **标识符\*** | 03 |
| **用例描述** | 描述了管理员添加平台各类用户的信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |

续表2-4

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **基本操作流** | 1. 管理员点击人员信息进入页面 2. 点击“添加按钮” 3. 填写用户信息，点击保存 |

修改用户信息用例分析表如下表2-5所示：

表2-5 修改用户信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改用户信息 |
| **标识符\*** | 04 |
| **用例描述** | 描述了管理员修改用户信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员点击人员信息进入页面 2. 点击“修改按钮” 3. 填写用户信息，点击保存 |

删除用户信息用例分析表如下表2-6所示：

表2-6 删除用户信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除用户信息 |
| **标识符\*** | 05 |
| **用例描述** | 描述了管理员删除用户的流程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员在左侧人员信息菜单栏下点击任一用户信息进入页面 2. 选择想要删除的用户，点击“删除按钮” 3. 填写用户信息，点击保存 |

查询用户信息用例分析表如下表2-7所示：

表2-7 查询用户信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查询用户信息 |
| **标识符\*** | 06 |
| **用例描述** | 描述了管理员查询用户信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员通过用户名和密码登录本平台 2. 在左侧人员信息菜单栏下点击学生信息、老师信息或家长信息，进入相关页面查询信息 |

添加专业信息用例分析表如下表2-8所示：

表2-8 添加专业信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加专业信息 |
| **标识符\*** | 07 |
| **用例描述** | 描述了管理员添加专业信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员进入页面 2. 点击添加，弹出信息窗口 3. 填写信息，点击保存 |

修改专业信息用例分析表如下表2-9所示：

表2-9 修改专业信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改专业信息 |
| **标识符\*** | 08 |

续表2-9

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例描述** | 描述了管理员修改专业信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员点击专业信息进入页面 2. 单击修改 3. 填写专业信息，点击保存 |

删除专业信息用例分析表如下表2-10所示：

表2-10 删除专业信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除专业信息 |
| **标识符\*** | 09 |
| **用例描述** | 描述了管理员删除专业信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员在左侧专业信息菜单栏下点击进入页面 2. 选择想要删除的专业，点击“删除按钮” |

查询专业信息用例分析表如下表2-11所示：

表2-11 查询专业信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查询专业信息 |
| **标识符\*** | 10 |
| **用例描述** | 描述了管理员查询专业信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |

续表2-11

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员登录平台进入主页 2. 点击专业信息菜单栏下的专业信息进入页面 |

添加班级信息用例分析表如下表2-12所示：

表2-12 添加班级信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加班级信息 |
| **标识符\*** | 11 |
| **用例描述** | 描述了管理员添加班级信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员点击班级信息进入页面 2. 点击添加 3. 在弹出的窗口填写班级信息，点击保存 |

修改班级信息用例分析表如下表2-13所示：

表2-13 修改班级信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改班级信息 |
| **标识符\*** | 12 |
| **用例描述** | 描述了管理员修改班级信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员在左侧班级信息菜单栏下点击进入页面 2. 点击“修改按钮” 3. 填写班级信息，点击保存 |

删除班级信息用例分析表如下表2-14所示：

表2-14 删除班级信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除班级信息 |
| **标识符\*** | 13 |
| **用例描述** | 描述了管理员删除班级信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员点击班级信息进入页面 2. 选择班级，点击“删除按钮” |

查询班级信息用例分析表如下表2-15所示：

表2-15 查询班级信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查询班级信息 |
| **标识符\*** | 14 |
| **用例描述** | 描述了管理员查询班级信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员登录平台进入主页 2. 点击班级信息，进入页面查询信息 |

2.2.2.2考试管理

考试管理包括考试信息管理和成绩管理。管理员在此模块可以对考试信息进行增加操作、修改操作、删除操作、查看操作，以及查看老师上传的学生成绩；老师在此模块可以查看管理员发布的考试信息，以及对成绩进行增加操作、修改操作、删除操作、查看操作；学生在此模块可以查看管理员发布的考试信息以及老师上传的考试成绩；家长在此模块可以查看自己孩子的考试成绩。

该模块的用例图如下图2-8所示：

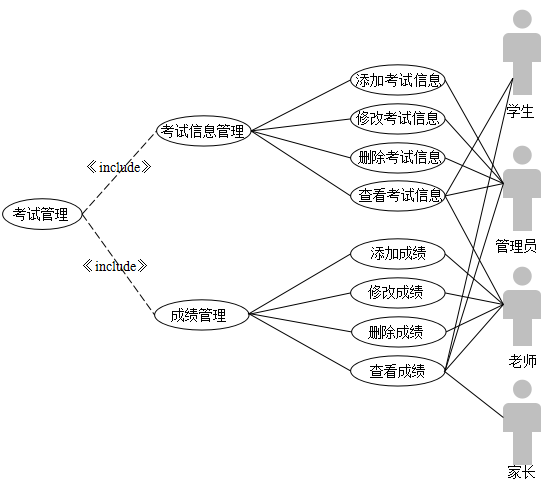


图2-8 考试管理用例图

添加考试信息用例分析表如下表2-16所示：

表2-16 添加考试信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加考试信息 |
| **标识符\*** | 15 |
| **用例描述** | 描述了管理员添加考试信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员在左侧学生成绩菜单栏下点击考试信息进入页面 2. 点击“添加按钮” 3. 填写考试信息，点击保存 |

修改考试信息用例分析表如下表2-17所示：

表2-17 修改考试信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改考试信息 |
| **标识符\*** | 16 |
| **用例描述** | 描述了管理员修改考试信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员点击考试信息进入页面 2. 点击“修改按钮” 3. 填写考试信息，点击保存 |

删除考试信息用例分析表如下表2-18所示：

表2-18 删除考试信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除考试信息 |
| **标识符\*** | 17 |
| **用例描述** | 描述了管理员删除考试信息的过程 |
| **参与者表** | 管理员 |
| **状态\*** | 进行中 |
| **前置条件** | 管理员已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 管理员在左侧学生成绩菜单栏下点击考试信息进入页面 2. 选择想要删除的考试信息，点击“删除按钮” |

查看考试信息用例分析表如下表2-19所示：

表2-19 查看考试信息用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查看考试信息 |
| **标识符\*** | 18 |
| **用例描述** | 描述了本平台用户查看考试信息的过程 |

续表2-19

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **参与者表** | 管理员、老师、学生 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 点击学生成绩菜单栏下的考试信息进入页面 |

添加成绩用例分析表如下表2-20所示：

表2-20 添加成绩用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加成绩 |
| **标识符\*** | 19 |
| **用例描述** | 描述了老师添加学生成绩的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击学生成绩进入页面 2. 点击添加 3. 填写信息，点击保存 |

修改成绩用例分析表如下表2-21所示：

表2-21 修改成绩用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改成绩 |
| **标识符\*** | 20 |
| **用例描述** | 描述了老师修改学生成绩信息的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |

续表2-21

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **基本操作流** | 1. 老师进入学生成绩页面 2. 点击修改 3. 填写信息，点击保存 |

删除成绩用例分析表如下表2-22所示：

表2-22 删除成绩用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除成绩 |
| **标识符\*** | 21 |
| **用例描述** | 描述了老师删除成绩信息的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击学生成绩进入页面 2. 选择学生成绩，点击删除 |

查看成绩用例分析表如下表2-23所示：

表2-23 查看成绩用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查看成绩 |
| **标识符\*** | 22 |
| **用例描述** | 描述了本平台用户查看成绩的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户登录平台进入主页 2. 点击学生成绩菜单栏下的学生成绩进入页面 |

2.2.2.3教学管理

教学管理包括教学资料管理和学生成长管理。管理员在此模块可以查看教学资料以及学生的评语；老师在此模块可以对教学资料进行管理：增加教学资料、修改教学资料、删除教学资料、查看教学资料，以及对学生成长进行管理：增加评语、修改评语、删除评语、查看评语；学生在此模块可以查看老师上传的教学资料以及老师写给自己的评语；家长在此模块可以查看老师登记小孩的评语。该模块的用例图如下图2-9所示：

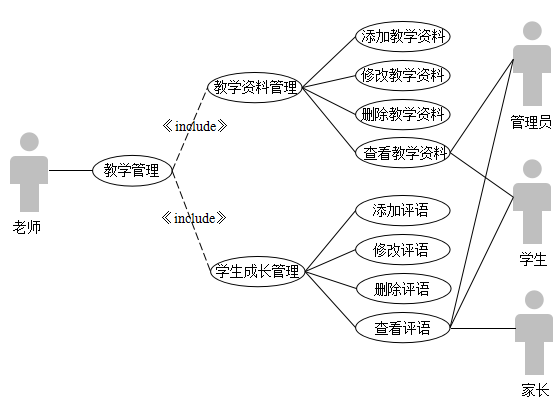


图2-9 教学管理用例图

添加教学资料用例分析表如下表2-24所示：

表2-24 添加教学资料用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加教学资料 |
| **标识符\*** | 23 |
| **用例描述** | 描述了老师教学资料的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |

续表2-24

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **基本操作流** | 1. 老师点击左侧教学资料菜单栏下教学资料的进入页面 2. 点击“添加按钮” 3. 填写修改教学资料信息，点击保存 4. 选择添加的教学资料，点击上传 |

修改教学资料用例分析表如下表2-25 所示：

表2-25 修改教学资料用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改教学资料 |
| **标识符\*** | 24 |
| **用例描述** | 描述了老师修改教学资料的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击教学资料进入页面 2. 选择教学资料信息，点击上传资料 |

删除教学资料用例分析表如下表2-26所示：

表2-26 删除教学资料用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除教学资料 |
| **标识符\*** | 25 |
| **用例描述** | 描述了老师使用本平台删除教学资料的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击教学资料进入页面 2. 勾选一条或多条教学资料，点击删除 |

查看教学资料用例分析表如下表2-27所示：

表2-27 查看教学资料用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查看教学资料 |
| **标识符\*** | 26 |
| **用例描述** | 描述了用户查看教学资料的过程 |
| **参与者表** | 管理员、老师、学生 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户用用户名和密码，选择权限登录平台进入主页 2. 点击教学资料菜单栏下的教学资料进入页面查询信息 |

添加评语用例分析表如下表2-28所示：

表2-28 添加评语用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加评语 |
| **标识符\*** | 27 |
| **用例描述** | 描述了老师通过本平台添加评语的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击学生成长进入页面 2. 点击添加 3. 填写信息，点击保存 |

修改评语用例分析表如下表2-29所示

表2-29 修改评语用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改评语 |
| **标识符\*** | 28 |
| **用例描述** | 描述了老师通过本平台修改评语的过程 |
| **参与者表** | 老师 |

续表2-29

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击学生成长进入页面 2. 选择一条记录，点击修改 3. 填写新的评语，点击保存 |

删除评语用例分析表如下表2-30所示

表2-30 删除评语用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除评语 |
| **标识符\*** | 29 |
| **用例描述** | 描述了老师删除评语的过程 |
| **参与者表** | 老师 |
| **状态\*** | 进行中 |
| **前置条件** | 老师已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 老师点击左侧学生成长进入页面 2. 选择想要删除的评语，点击“删除按钮” |

查看评语用例分析表如下表2-31所示

表2-31 查看评语用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查看评语 |
| **标识符\*** | 30 |
| **用例描述** | 描述了本平台用户查看评语的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |

续表2-31

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **基本操作流** | 1. 用户登录平台进入主页 2. 点击学生成长进入页面 |

2.2.2.4论坛管理

论坛管理包括帖子管理和评论管理。管理员在此模块可以加、改、删、查交流贴，以及对交流贴进行评论；师生和家长在这个模块可以添加、修改和查看帖子以及对帖子添加评论。该模块的用例图如下图2-10所示：

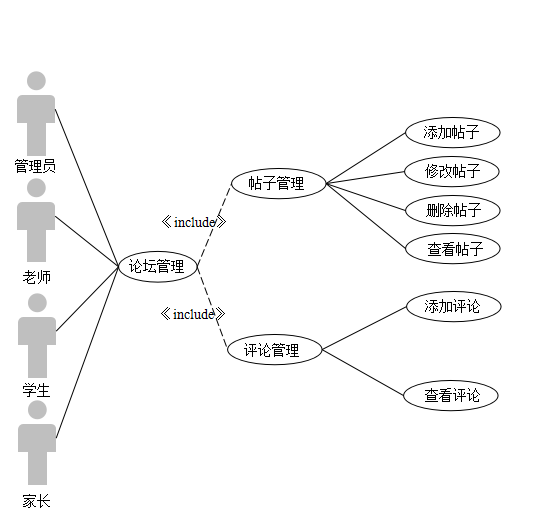


图2-10 论坛管理用例图

添加帖子用例分析表如下表2-32所示：

表2-32 添加帖子用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加帖子 |
| **标识符\*** | 31 |
| **用例描述** | 描述了用户添加主题帖的过程 |
| **参与者表** | 管理员、老师、学生、家长 |

续表2-32

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户点击交流信息进入页面 2. 点击添加 3. 填写想要发布的帖子信息，点击保存 |

修改帖子用例分析表如下表2-33所示：

表2-33 修改帖子用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 修改帖子 |
| **标识符\*** | 32 |
| **用例描述** | 描述了用户修改主题帖的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户点击交流信息进入页面 2. 选择一个帖子，点击“修改按钮”” 3. 填写想要修改的帖子信息，点击保存 |

删除帖子用例分析表如下表2-34所示：

表2-34 删除帖子用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 删除帖子 |
| **标识符\*** | 33 |
| **用例描述** | 描述了管理员删除帖子的过程 |
| **参与者表** | 管理员、老师、家长、学生 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |

续表2-34

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **基本操作流** | 1. 用户点击论坛管理菜单栏下的交流信息进入页面 2. 选择想要删除的帖子，点击“删除按钮” |

查看帖子用例分析表如下表2-35所示：

表2-35 查看帖子用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查看帖子 |
| **标识符\*** | 34 |
| **用例描述** | 描述了本平台用户查看帖子的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户登录平台进入主页 2. 点击相关帖子的标题 |

添加评论用例分析表如下表2-36所示：

表2-36 添加评论用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 添加评论 |
| **标识符\*** | 35 |
| **用例描述** | 描述了本平台用户添加评论的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户点击全部交流进入页面 2. 填写想要评论的内容，点击保存 |

查看评论用例分析表如下表2-37所示：

表2-31 查看评论用例描述

|  |  |
| --- | --- |
| **描述项** | **说明** |
| **用例名称** | 查看评论 |
| **标识符\*** | 36 |
| **用例描述** | 描述了本平台用户查看评论的过程 |
| **参与者表** | 管理员、老师、学生、家长 |
| **状态\*** | 进行中 |
| **前置条件** | 用户已登录平台 |
| **后置条件** | 平台给出操作成功提示 |
| **基本操作流** | 1. 用户进入前台有交流贴的页面 2. 查看帖子下方 |

## 2.3 非功能需求

### 2.3.1 环境需求

本平台基于SSM框架进行开发，由于JVM具有跨平台的特性，因此使得本平台可以直接在三大操作系统——Windows、Linux、MacOS进行部署运行，不需要大量的修改代码适应不同平台，本平台对运行环境的配置要求比较低，在后期的运行维护上可以不需要比较多的人力成本，因此具有良好的经济可行性。

本平台具体运行环境如下表2-33所示：

表2-33 运行环境配置要求表

|  |  |
| --- | --- |
| **服务端运行环境** | |
| **项目** | **配置要求** |
| 操作系统 | Windows 7及以上版本或Linux或MacOS |
| JDK | 1.8 |
| 应用服务器 | Apache Tomcat 8.5.29 |
| 应用数据库 | MySQL 5.6.19及以上版本 |
| **客户端运行环境** | |
| **项目** | **配置要求** |
| 操作系统 | Windows 7及以上版本或Linux或MacOS |
| 浏览器 | IE、360浏览器、火狐浏览器、Chrom浏览器等 |
| 分辨率 | 最佳效果为1024×768像素 |

### 性能需求

1. 系统的安全性：使用本系统的用户需要登录。
2. 系统的可升级性：系统的容纳信息一定是持续增长，到时系统可以升级，接受信

息的更新发展。

1. 系统的保密性：系统经过一段时间的使用后，将会存在很多很多的秘密的数据，

而这些数据是不能够被泄露出去的，如果被泄露了，使用了系统的用户将会有损失，故而开发系统时，开发人员应该注意系统的保密性是不是合格的。

1. 系统的可靠性：系统不可以有病毒类代码，必须有拦截器，验证方法，对里面信

息的保护措施，这样才让人用的放心。

1. 系统的简单性：系统应该设计成用户容易使用的系统，不需要努力学习培训，缩

短用户熟悉系统的进程。

1. 系统的一致性：用户会存入数据到系统中，在使用系统时，所以已经存入的数据

要和用户存进的数据保持一致，不能有错误的数据出现。用户还会有更新数据的需求，更新时能及时更新。还可以单独保存数据，不可以把还要用的数据删除时连带删除还要使用的数据。

1. 系统的稳定性：开发的中等学校通用IS平台要求运行稳定，运行过程中无界面

不清楚、字体模糊等现象。

1. 系统的存储性：因为是中等学校通用IS平台，所以就会在数据库要求上比较严

格，录入大量繁琐复杂的信息，这就需要一个可以容纳非常多的数据的数据库。

### 2.3.3 安全需求

1. 容错性

用户可能会因为对系统不熟悉的原因导致在输入数据的时候或输错，或者进行某

项操作时操作失败，所以系统应该有一定的容错机制，然后给用户相应的提示，教用户正确使用系统。

1. 系统的外在环境安全

系统还要考虑外面的环境是否是安全的，可以通过使用一些措施来使得系统在一

个安全的环境下，从而确定保护了数据的安全性，不让它泄露出系统。

1. 系统内部安全

给用户分派不用的权限使用系统。

# 3 总体设计

## 3.1系统设计的原则

在充分考虑到本平台的总体目标以及平台的运行环境的前提下，本平台的设计与实现将遵循以下列出的原则：

1. 实用性原则。一个系统的好坏可以通过用户的体验以及功能是否齐全来体现，本平台尽量能够紧密结合用户所提出的实际业务需求，用来解决学校信息管理效率比较低下的弊端。
2. 经济可行性原则。满足系统运行和性能要求是通过采用基于SSM框架，Tomcat应用程序服务器和MySQL数据库，运用了这两个服务器可以使得维护系统需要的成本不需要很大，所以实现了经济可行性的原则。
3. 简单性原则：系统设计的应该简单容易上手，设计的各种功能操作应该直观，不需要努力学习培训，减轻了用户使用该系统的时间。

## 3.2 系统体系结构设计

B/S结构是这一段时间非常流行的一种结构，它可以帮助开发人员不用编写大量大段的代码，只需要实现业务逻辑方面的功能，使得系统维护简单。用户也不需要安装客户端才能使用，只需要拥有一个浏览器，然后安装一个数据库就可以，用户使用的浏览器和服务器连接的数据库进行动态的数据交换。很多需要下载客户端才能实现的功能只采用B/S架构开发的系统也同样能实现，而且还可以节约开发者许多不需要的成本。B/S结构最好的一个优点是不用特地安装专业的软件，只要有互联网就可以访问系统。在后期的维护上面就不用花费力气，通过在客户端就能使得一些问题得到解决，而不用通过服务器，这样开发出来的系统在用户交互性上面也能更好。

本平台按照MVC的三层结构进行开发，三层结构分别为数据层、表现层、业务层。其中数据层主要是实现访问数据库操作的方法的封装，赖于MyBatis框架；业务层的功能主要是处理业务逻辑相关的问题，显示层是显示数据的；表现层是用户看见的UI界面，基于Spring框架搭建而成。

## 3.3 系统功能模块设计

本系统的系统功能模块图如图3-3所示：

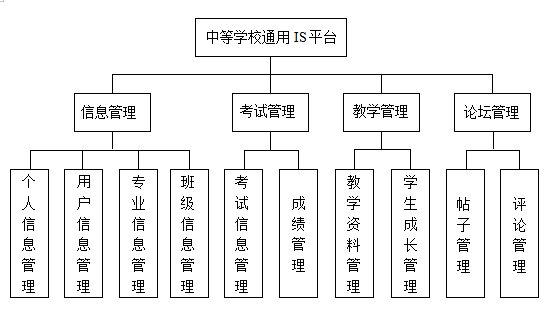


图3-3系统功能模块图

如上图3-3所示，本平台包括信息管理、考试管理、教学管理、论坛管理共四个模块。其中信息管理模块又包括个人信息管理、用户信息管理、专业信息管理和班级信息管理；考试管理包括考试信息管理和成绩管理；教学管理包括教学资料管理和学生成长管理；论坛管理包括帖子管理和评论。

# 4 数据库设计

## 4.1 概念结构设计

### 4.1.1 设计思路

一个好的数据库可以与程序开发的优劣有关。在设计数据库时，表和表之间的关系有多种，系统进行开发需要设计的数据表中的内容的信息与信息之间也有着密不可分的关系。

本中等学校通用IS平台使用的是MySQL数据库，体积小，不占地方，而且数据存储快，因为中等学校通用IS平台，主要的就是对信息的管理，信息内容比较多，这就需好好的设计一个好的数据库，分类要清楚，不能添加信息的时候，造成信息太过混乱，设计好的数据库首先需要理顺各个实体之间的关系，下面用E-R图来具体表示。

### 4.1.2 E-R图

系统总体E-R图如下图4-1所示：

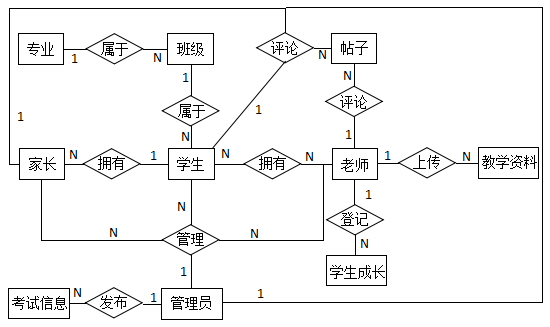


图4-1 系统E-R图

1. 管理员实体图，如下图4-2所示：

实体：管理员

属性：管理员编号、管理员名称、密码。

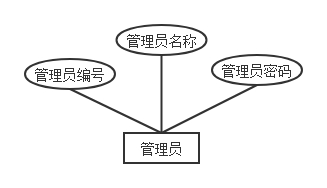


图4-2 管理员实体图

1. 老师实体图，如下图4-3所示：

实体：老师

属性：编号、姓名、密码、性别、年龄、电话、专业编号、专业名称。

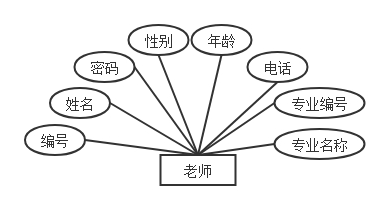


图4-3 老师实体图

1. 学生实体图，如下图4-4所示：

实体：学生

属性：编号、姓名、密码、性别、年龄、电话、专业编号、专业名称、班级编号、班级名称。

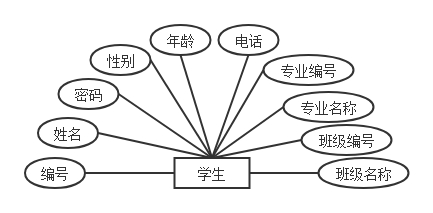


图4-4 学生实体图

1. 家长实体图，如下图4-5所示：

实体：家长

属性：编号、姓名、密码、性别、年龄、电话、学生编号、学生姓名。

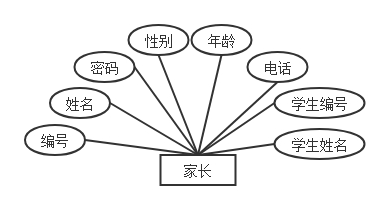


图4-5 家长实体图

1. 专业实体图，如下图4-6所示：

实体：专业

属性：专业编号、专业名称、专业备注。

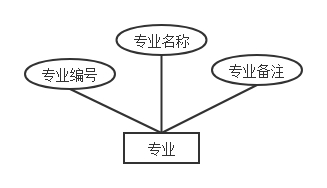


图4-6 专业实体图

1. 班级实体图，如下图4-7所示：

实体：班级

属性：包括编号、名称、专业编号、专业名称。

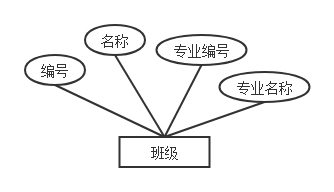


图4-7 班级实体图

1. 学生成长实体图，如下图4-8所示：

实体：学生成长

属性：编号、内容、类型、老师编号、老师姓名、学生编号、学生姓名。

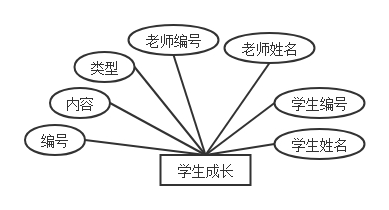


图4-8 学生成长实体图

1. 教学资料实体图，如下图4-9所示：

实体：教学资料

属性：编号、名称、内容、老师编号、老师姓名。

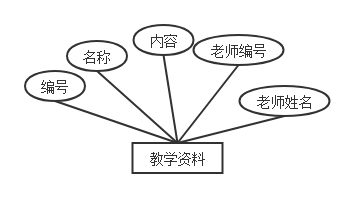


图4-9 教学资料实体图

1. 帖子实体图，如下图4-10所示：

实体：帖子

属性：编号、标题、内容、发布人编号、发布人姓名。

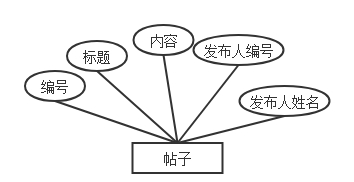


图4-10 帖子实体图

1. 评论实体图，如下图4-11所示：

实体：评论

属性：编号、内容、时间、帖子编号、帖子名称。

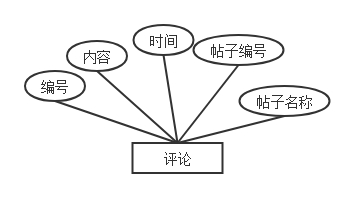


图4-11 评论实体图

1. 考试信息实体图，如下图4-12所示：

实体：考试信息

属性：编号、名称、时间、地址。

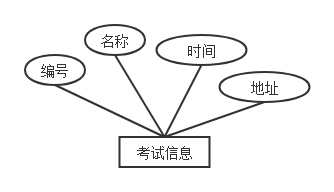


图4-12 考试信息实体图

## 4.2 逻辑结构设计

### 4.2.1 设计思路

根据上一小节概念结构设计出的E-R图以及属性图，遵循一个实体一张表，多对多关系要建立一张联系表的方案将关系模式转换为二维表的逻辑结构。

### 4.2.2 逻辑模型

1. E-R图转换为关系表
2. 管理员信息表（编号，姓名，密码）
3. 老师信息表（编号，姓名，密码，性别，年龄，电话，专业编号，专业名称）
4. 学生信息表（编号，姓名，密码，性别，年龄，电话，专业编号，专业名称，班级编号，班级名称）
5. 家长信息表（编号，姓名，密码，性别，年龄，电话，学生编号，学生姓名）
6. 专业信息表（编号，名称，备注）
7. 班级信息表（编号，名称，专业编号，专业名称）
8. 学生成长信息表（编号，内容，类型，老师编号，老师姓名，学生编号，学生姓名）
9. 教学资料信息表（编号，名称，内容，老师编号，老师姓名）
10. 帖子信息表（编号，标题，内容，发布人编号，发布人，姓名）
11. 评论信息表（编号，内容，时间，帖子编号，帖子标题）
12. 考试信息表（编号，名称，时间，地址）
13. 数据表格
    1. 管理员信息表如下表4-1所示：

表4-1 管理员信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 姓名 | 字符型 | 50 | 否 |
| 3 | 密码 | 字符型 | 50 | 否 |

* 1. 老师信息表如下表4-2所示：

表4-2 老师信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 姓名 | 字符型 | 50 | 否 |
| 3 | 密码 | 字符型 | 50 | 否 |
| 4 | 性别 | 整型 | 11 | 否 |
| 5 | 年龄 | 整型 | 11 | 否 |
| 6 | 电话 | 字符型 | 50 | 否 |
| 7 | 专业编号 | 整型 | 50 | 否 |
| 8 | 专业名称 | 字符型 | 50 | 否 |

* 1. 学生信息表如下表4-3所示：

表4-3 学生信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 姓名 | 字符型 | 50 | 否 |
| 3 | 密码 | 字符型 | 50 | 否 |
| 4 | 性别 | 整型 | 11 | 否 |
| 5 | 年龄 | 整型 | 11 | 否 |

续表4-3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 6 | 电话 | 字符型 | 50 | 否 |
| 7 | 专业编号 | 整型 | 11 | 否 |
| 8 | 专业名称 | 字符型 | 50 | 否 |
| 9 | 班级编号 | 整型 | 11 | 否 |
| 10 | 班级名称 | 字符型 | 50 | 否 |

* 1. 家长信息表如下表4-4所示：

表4-4 家长信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 姓名 | 字符型 | 50 | 否 |
| 3 | 密码 | 字符型 | 50 | 否 |
| 4 | 性别 | 整型 | 11 | 否 |
| 5 | 年龄 | 整型 | 11 | 否 |
| 6 | 电话 | 字符型 | 50 | 否 |
| 7 | 学生编号 | 整型 | 11 | 否 |
| 8 | 学生名称 | 字符型 | 50 |  |

* 1. 专业信息表如下表4-5所示：

表4-5 专业信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 名称 | 字符型 | 50 | 否 |
| 3 | 备注 | 字符型 | 255 | 否 |

* 1. 班级信息表如下表4-6所示：

表4-6 班级信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 名称 | 字符型 | 255 | 否 |
| 3 | 专业编号 | 整型 | 11 | 否 |
| 4 | 专业名称 | 字符型 | 255 | 否 |

* 1. 学生成长信息表如下表4-7所示：

表4-7 学生成长信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 内容 | 字符型 | 50 | 否 |
| 3 | 类型 | 字符型 | 50 | 否 |

续表4-7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 4 | 老师编号 | 整型 | 11 | 否 |
| 5 | 老师姓名 | 字符型 | 50 | 否 |
| 6 | 学生编号 | 整型 | 11 | 否 |
| 7 | 学生姓名 | 字符型 | 50 | 否 |

* 1. 教学资料信息表如下表4-8所示：

表4-8 教学资料信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 名称 | 字符型 | 50 | 否 |
| 3 | 内容 | 字符型 | 50 | 否 |
| 4 | 老师编号 | 整型 | 11 | 否 |
| 5 | 老师姓名 | 字符型 | 50 | 否 |

* 1. 帖子信息表如下表4-9所示：

表4-9 帖子信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 标题 | 字符型 | 50 | 否 |
| 3 | 内容 | 字符型 | 255 | 否 |
| 4 | 发布人编号 | 整型 | 11 | 否 |
| 5 | 发布人姓名 | 字符型 | 50 | 否 |

* 1. 评论信息表如下表4-10所示：

表4-10 评论信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 内容 | 字符型 | 255 | 否 |
| 3 | 时间 | 日期 | 50 | 否 |
| 4 | 帖子编号 | 整型 | 11 | 否 |
| 5 | 帖子名称 | 字符型 | 50 | 否 |

* 1. 考试信息表如下表4-11所示：

表4-11考试信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **字段名称** | **数据类型** | **字段长度** | **是否主键** |
| 1 | 编号 | 整型 | 11 | 是 |
| 2 | 名称 | 字符型 | 50 | 否 |
| 3 | 时间 | 日期 | 50 | 否 |
| 4 | 地址 | 字符型 | 50 | 否 |

## 4.3 物理结构设计

### 4.3.1 存取方式

通过对本系统的数据库设计分析，采用索引存取的方式对数据进行存取，在每个表中设置主键字段和经常需要查找的字段的B+树索引，提高数据库的查找效率。本系统采用MySQL数据库进行存储。

### 4.3.2 存储结构

针对数据库系统的数据结构，采用索引顺序存储、数据分离和缓存优化技术。系统中数据库表的存储采用顺序存储，并对行字段建立相应的索引，将频繁查询的表的数据分开，加快系统的查询速度，将一些不经常更新的查询结果存储在缓存数据库中，节省了重复查询的时间开销。

5 界面设计

## 5.1 界面关系图或工作流图

本系统的界面关系图如下图5-1所示：

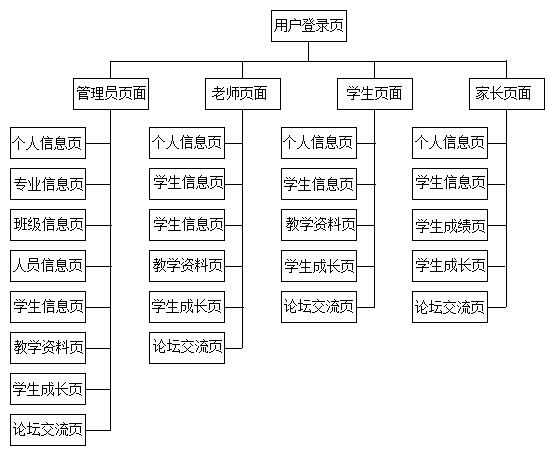


图5-1 界面关系图

## 5.2 界面设计成果

### 主界面

1. 前台主界面

前台主要是展示关于该学校的一些相关信息，以及系统用户可以在这交流，如下图5-2所示：



图5-2前台主界面

1. 用户主界面

用户登录系统后进入主界面，用户可以选择菜单列表下的想要进行操作的界面，如下图5-3所示：

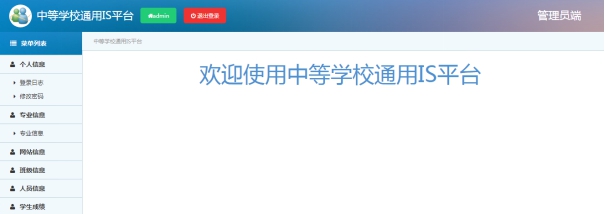


图5-3用户主界面

### 5.2.2 子界面

1. 个人信息界面

用户登录系统后，可以查看自己的个人信息，以及可以对个人信息进行修改，如下图5-4所示：

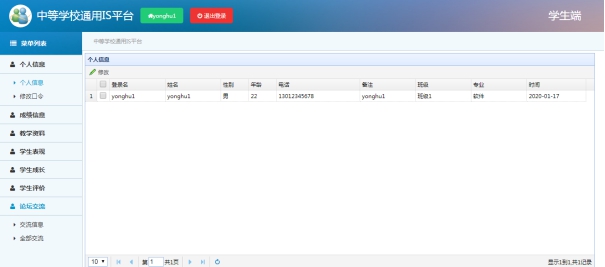


图5-4个人信息界面

1. 考试信息界面

管理员登录系统后进入考试信息页面可以添加、修改、删除考试信息，老师和学生可以查看相关的考试信息，如下图5-5所示：

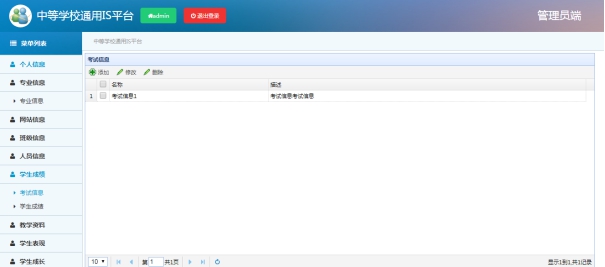


图5-5考试信息界面

1. 学生成绩界面

老师登录系统后进入学生成绩界面可以添加、修改、删除考试成绩，管理员和学生可以查看相关的考试成绩，如下图5-6所示：

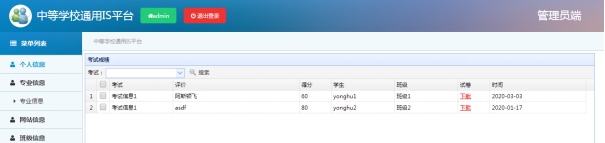


图5-6学生成绩界面

1. 教学资料界面

老师登录系统后进入教学资料页面可以添加、修改、删除教学资料，管理员和学生可以查看相关的教学资料，学生还可下载老师添加的教学资料，如图5-7所示：

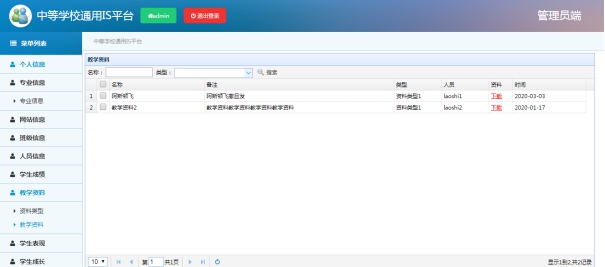


图5-7教学资料界面

1. 学生成长界面

老师登录系统后进入学生成长页面可以添加、修改、删除对学生登记的评语，管理员、学生和家长可以查看相关的评语，如图5-8所示：



图5-8学生成长界面

1. 帖子界面

用户已经发出去的帖子的界面，用户可以点击查看就可以看见帖子的详细内容，如图5-9所示：

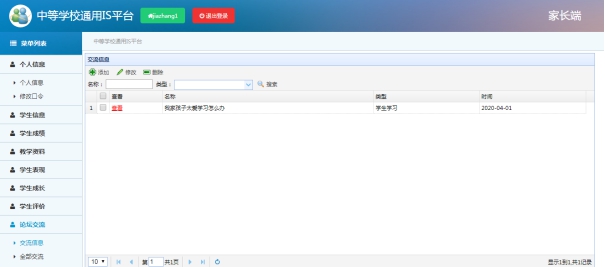


图5-9帖子界面

1. 评论界面

用户登录系统后可以点击感兴趣的帖子查看它底下的评论，也可以添加评论进行交流，如图5-10所示：



图5-10评论界面

6 详细设计

## 6.1 系统主要功能模块介绍

系统总包图如下图6-1所示：

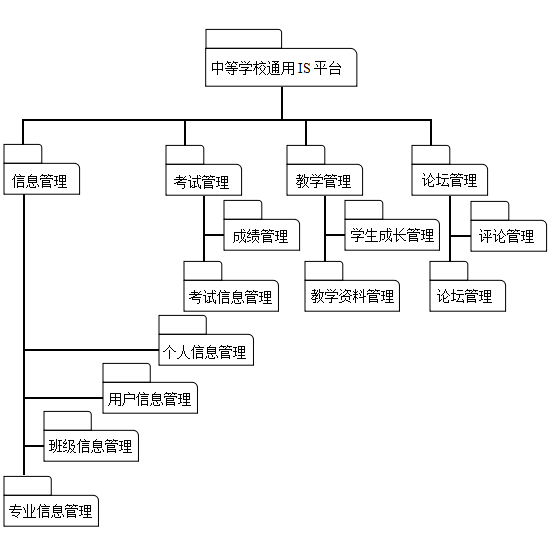


图6-1 系统总包图

如上图所示，系统包括信息管理、考试管理、教学管理、论坛管理四个模块。其中信息管理模块又包括个人信息管理、用户信息管理、专业信息管理和班级信息管理；考试管理包括考试信息管理和成绩管理；教学管理包括教学资料管理和学生成长管理；论坛管理包括帖子管理和评论。

## 6.2 个人信息管理

### 6.2.1 功能结构设计

个人信息管理包图如下图6-2所示：

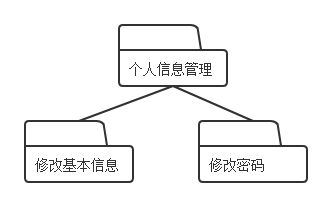


图6-2 个人信息管理包图

1. 修改基本信息：用户修改信息使用的功能点。
2. 修改密码：用户更换密码时使用的功能点。

### 6.2.2 类图设计

个人信息管理主要涉及User实体类、userMapper类、userDao类、userService类、userAction类，userMapper类、userDao类、userService类、userAction类，均依赖User实体类。

个人信息管理的关键类如下图6-3所示：

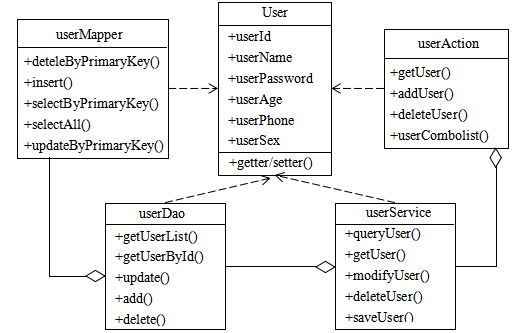


图6-3 个人信息管理类图

userMapper类描述如下表6-1所示：

表6-1 userMapper类描述

|  |  |  |
| --- | --- | --- |
| userMapper类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | delete () | 删除用户 |
| insert() | 添加用户 |
| selectUser () | 查找用户信息 |
| selectAll() | 查询所有用户信息 |
| updateUser() | 修改用户信息 |

userDao类描述如下表6-2所示：

表6-2 userDao类描述

|  |  |  |
| --- | --- | --- |
| userDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | add() | 添加用户 |
| getUser () | 获取用户 |
| update() | 更新用户信息 |
| delete() | 删除用户 |
| getList () | 获得用户列表 |

userService类描述如下表6-3所示：

表6-3 userService类描述

|  |  |  |
| --- | --- | --- |
| userService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | queryUser() | 查询用户 |
| getUser() | 获取用户 |
| modifyUser() | 修改用户 |
| deleteUser() | 删除用户 |
| saveUser() | 保存用户 |

userAction类描述如下图6-4所示：

表6-4 userAction类描述

|  |  |  |
| --- | --- | --- |
| userAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | get() | 获得用户信息 |
| add() | 添加用户信息 |
| delete() | 删除用户信息 |
| List() | 获得用户列表 |

User类描述如下图6-5所示：

表6-5 user类描述

|  |  |  |
| --- | --- | --- |
| user类 | | |
| 类的描述：用户实体类 | | |
| 属性  描  述 | userId | 用户Id |
| userName | 用户名称 |
| userpassword | 用户密码 |
| userAge | 用户年龄 |
| userPhone | 用户电话 |
| UserSex | 用户性别 |
| 操作描述 | Getter/setter() | getter/setter()方法 |

### 6.2.3 顺序图设计

个人信息修改时序图如下图6-4所示：

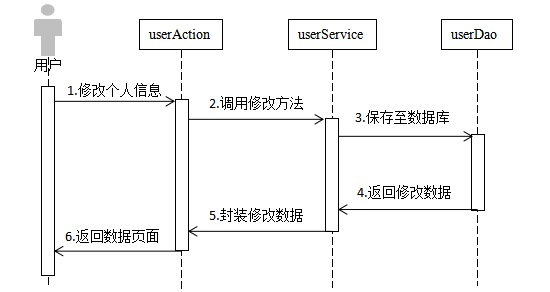


图6-4 个人信息修改时序图

### 6.2.4 核心处理流程设计

个人信息管理流程图如下图6-5所示：

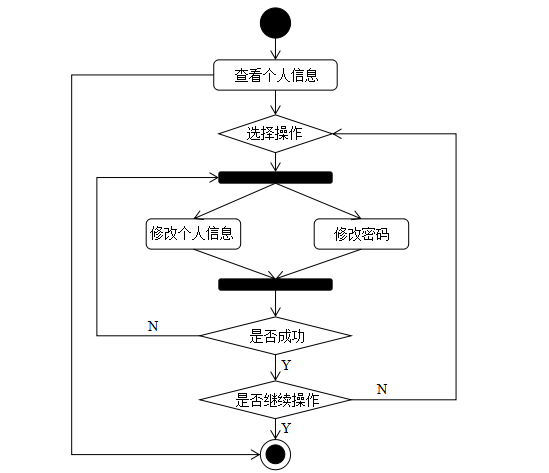


图6-5 个人信息管理

## 6.3 用户信息管理

### 6.3.1 功能结构设计

用户信息管理包图如下图6-6所示：

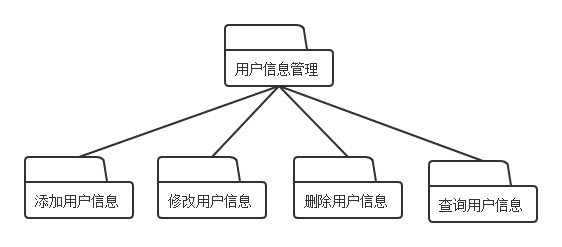


图6-6 用户信息管理包图

1. 添加用户信息：管理员录入需要使用系统的用户信息。
2. 修改用户信息：管理员修改已录入的用户信息。
3. 删除用户信息：管理员删除已录入的用户信息。
4. 查询用户信息：管理员查询需要查找的用户信息。

### 6.3.2 类图设计

个人信息管理主要涉及User实体类、userMapper类、userDao类、userService类、userAction类，userMapper类、userDao类、userService类、userAction类，均依赖User实体类。个人信息管理的关键类如下图6-7所示：

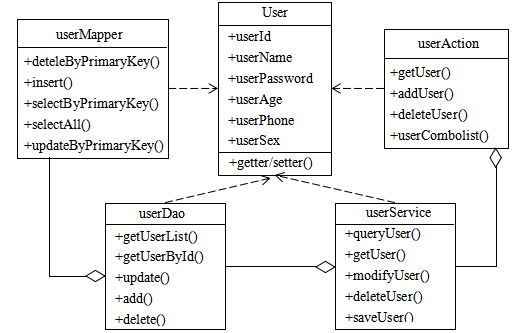


图6-7 用户信息管理类图

userDao类描述如下表6-6所示：

表6-6 userDao类描述

|  |  |  |
| --- | --- | --- |
| userDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | add() | 添加用户 |
| getById() | 利用Id获取用户 |
| update() | 修改用户信息 |
| delete() | 删除用户信息 |
| getList () | 获得用户列表 |

userService类描述如下表6-7所示：

表6-7 userService类描述

|  |  |  |
| --- | --- | --- |
| userService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | query() | 查询用户 |
| get() | 获取用户信息 |
| modify() | 修改用户信息 |
| delete() | 删除用户信息 |
| save() | 保存用户信息 |

userAction类描述如下表6-8所示：

表6-8 userAction类描述

|  |  |  |
| --- | --- | --- |
| userAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | get() | 获得用户信息 |
| add() | 添加用户信息 |
| delete() | 删除用户信息 |
| List() | 获得用户列表 |

### 6.3.3 顺序图设计

删除用户信息时序图如图6-8所示：

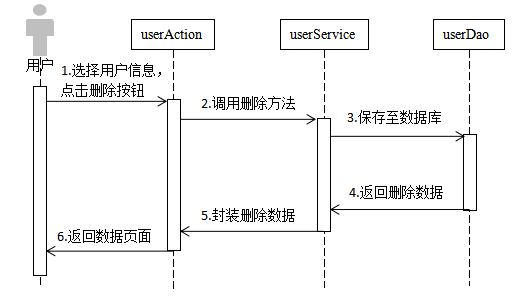


图6-8 修改用户信息时序图

### 6.3.4 核心处理流程设计

用户信息管理流程图如下图6-9所示：

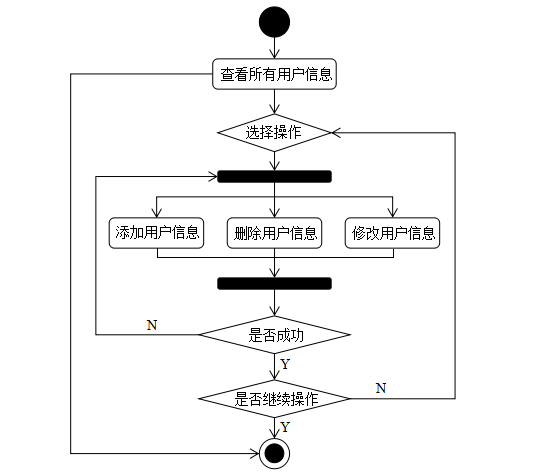


图6-9 用户信息管理流程图

## 6.4 考试信息管理

### 6.4.1 功能结构设计

考试信息管理包图如下图6-10所示：

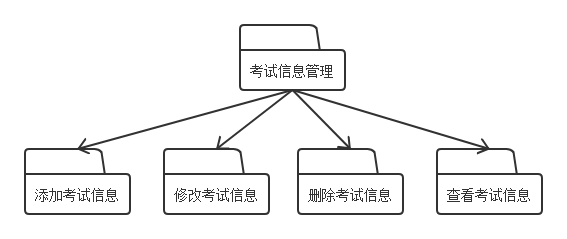


图6-10 考试信息管理包图

1. 添加考试信息：管理员通过平台添加考试信息，如考试时间、地点等。
2. 修改考试信息：管理员对已添加的考试信息进行修改。
3. 删除考试信息：管理员对已添加的考试信息进行删除。
4. 查看考试信息：管理员、老师和学生通过平台查看管理员添加的考试信息。

### 6.4.2 类图设计

考试信息管理主要涉及Exam实体类、examMapper类、examDao类、examService类、examAction类，examMapper类、examDao类、examService类、examAction类，均依赖Exam实体类。

考试信息管理的关键类如下图6-11所示：

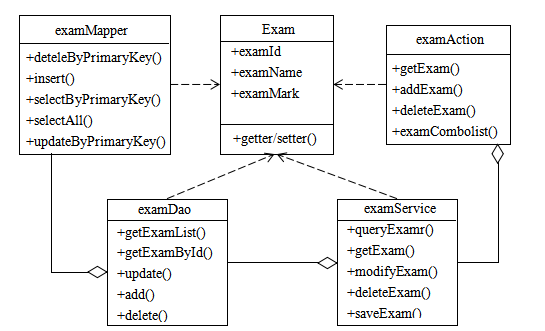


图6-11 考试信息管理类图

examDao类描述如下表6-9所示：

表6-9 examDao类描述

|  |  |  |
| --- | --- | --- |
| examDao类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | getList () | 获得考试列表 |
| getById() | 通过Id获取考试 |
| update() | 修改考试信息 |
| delete() | 删除考试信息 |
| add() | 添加考试信息 |

examService类描述如表6-10所示：

表6-10 examService类描述

|  |  |  |
| --- | --- | --- |
| examService类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | queryExam() | 查询考试 |
| getExam() | 获取考试信息 |
| modifyExam() | 修改考试信息 |
| deleteExam() | 删除考试信息 |
| saveExam() | 保存考试信息 |

examAction类描述如表6-11所示：

表6-11 examAction类描述

|  |  |  |
| --- | --- | --- |
| examAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | getExam() | 获取考试信息 |
| addExam() | 添加考试信息 |
| deleteExam() | 删除考试信息 |
| examComboList() | 获取考试信息列表 |

### 6.4.3 顺序图设计

添加考试信息时序图如下图6-12所示：

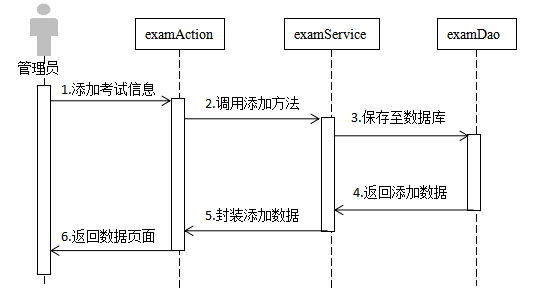


图6-12 添加考试信息时序图

### 6.4.4 核心处理流程设计

考试信息管理流程图如下图6-13所示：

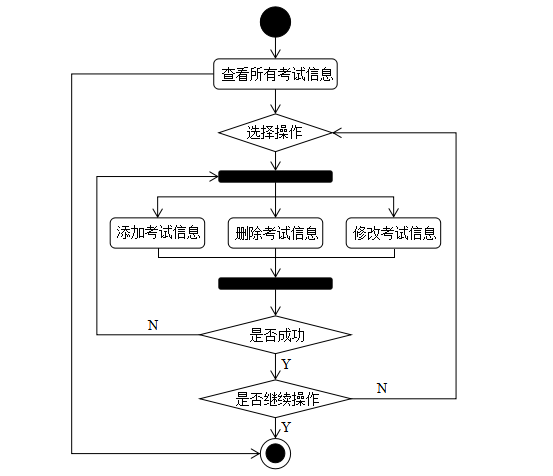


图6-13 考试信息管理流程图

## 6.5 成绩管理

### 6.5.1 功能结构设计

成绩管理包图如下图6-14所示：

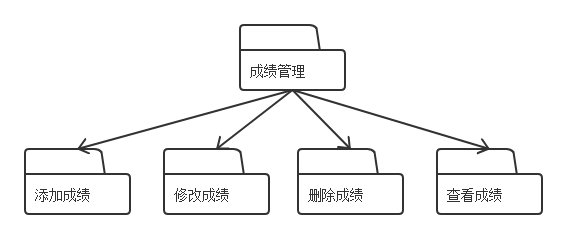


图6-14 成绩管理包图

1. 添加成绩：老师添加学生的考试成绩，以便学生和家长查看。
2. 修改成绩：老师对已添加的成绩进行修改。
3. 删除成绩：老师对已添加的成绩进行删除。
4. 查看成绩：管理员、老师、学生和家长可以查看老师添加的考试成绩，管理员可以查看全部学生的，学生和家长仅能查看该学生的成绩。

### 6.5.2 类图设计

成绩管理主要涉及grade实体类、gradeMapper类、gradeDao类、gradeService类、gradeAction类，gradeMapper类、gradeDao类、gradeService类、gradeAction类，均依赖Ugrade实体类。

个人信息管理的关键类如下图6-15所示：

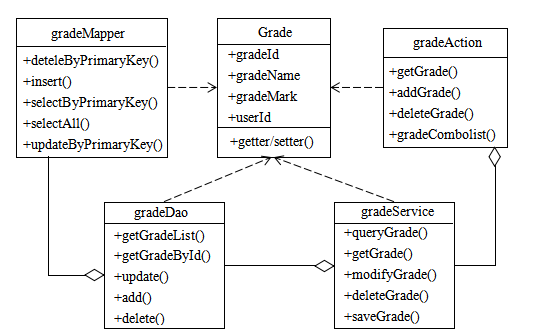


图6-15 成绩管理类图

gradeDao类描述如下表6-12所示：

表6-12 gradeDao类描述

|  |  |  |
| --- | --- | --- |
| gradeDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | getGradeList () | 获得成绩列表 |
| getById() | 利用Id获得成绩 |
| update() | 修改成绩信息 |
| delete() | 删除成绩信息 |
| add() | 添加成绩信息 |

gradeService类描述如下表6-13所示：

表6-13 gradeService类描述

|  |  |  |
| --- | --- | --- |
| gradeService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | saveGrade() | 保存成绩信息 |
| getGrade() | 获取成绩信息 |
| modifyGrade() | 修改成绩信息 |
| deleteGrade() | 删除成绩信息 |
| queryGrade() | 查询成绩信息 |

gradeAction类描述如下表6-14所示：

表6-14 gradeAction类描述

|  |  |  |
| --- | --- | --- |
| gradeAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | getGrade() | 获取成绩 |
| addGrade() | 添加成绩信息 |
| deleteGrade() | 删除成绩信息 |
| gradeComboList() | 获取成绩信息列表 |

### 6.5.3 顺序图设计

学生查询成绩时序图如下图6-16所示：

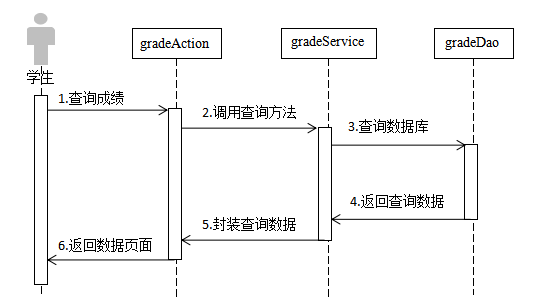


图6-16 查询成绩时序图

### 6.5.4 核心处理流程设计

成绩管理流程图如下图6-17所示：

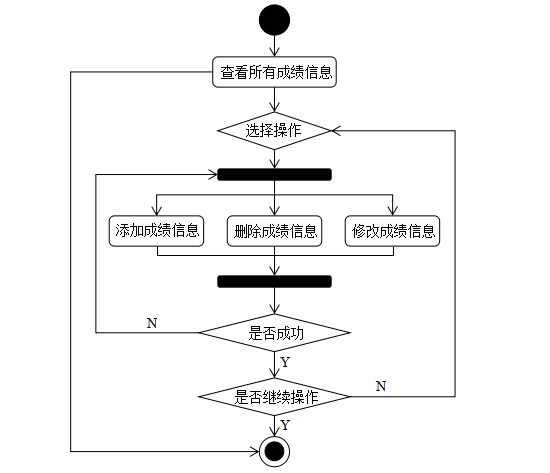


图6-17 成绩管理流程图

## 6.6 教学资料管理

### 6.6.1 功能结构设计

教学资料管理包图如下图6-18所示：

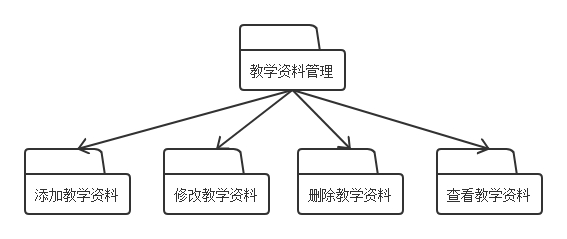


图6-18 教学资料管理包图

1. 添加教学资料：老师添加想要发布的教学资料，以便学生查看。
2. 修改教学资料：老师对已添加的教学资料进行修改。
3. 删除教学资料：老师对已添加的教学资料进行删除。
4. 查看教学资料：管理员、老师、学生可以查看老师添加的教学资料。

### 6.6.2 类图设计

教学资料管理主要涉及document实体类、documentMapper类、documentDao类、documentService类、documentAction类，documentMapper类、documentDao类、documentService类、documentAction类，均依赖document实体类。

教学资料管理的关键类如下图6-19所示：

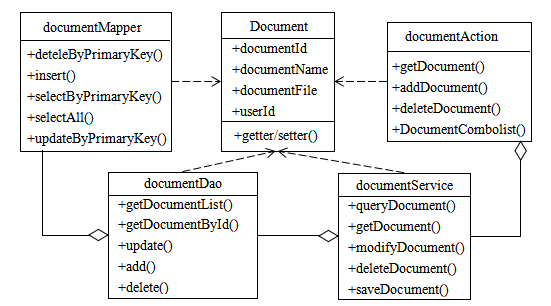


图6-19 教学资料管理类图

documentDao类描述如下表6-15所示：

表6-15 documentDao类描述

|  |  |  |
| --- | --- | --- |
| documentDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | getList () | 获得教学资料列表 |
| getById() | 利用Id获取教学资料信息 |
| update() | 修改教学资料信息 |
| delete() | 删除教学资料信息 |
| add() | 添加教学资料信息 |

documentService类描述如下表6-16所示：

表6-16 documentService类描述

|  |  |  |
| --- | --- | --- |
| documentService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | query() | 查询教学资料信息 |
| get() | 获取教学资料信息 |
| modify() | 修改教学资料信息 |
| delete() | 删除教学资料信息 |
| save() | 保存教学资料信息 |

documentAction类描述如下表6-17所示：

表6-17 documentAction类描述

|  |  |  |
| --- | --- | --- |
| documentAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | get() | 获取教学资料信息 |
| add() | 添加教学资料信息 |
| delete() | 删除教学资料信息 |
| ComboList() | 获取教学资料信息列表 |

### 6.6.3 顺序图设计

学生查询教学资料时序图如下图6-20所示：

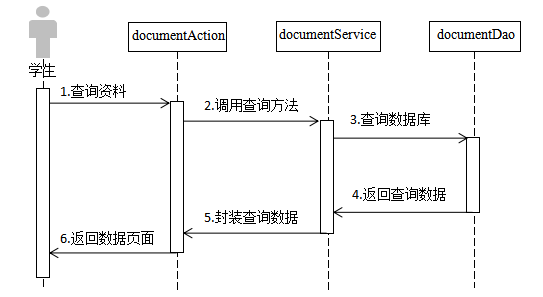


图6-20 查询教学资料时序图

### 6.6.4 核心处理流程设计

教学资料管理流程图如下图6-21所示:

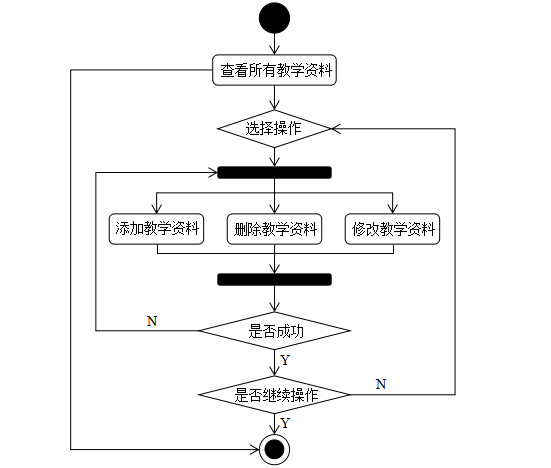


图6-21 教学资料管理流程图

## 6.7 学生成长管理

### 6.7.1 功能结构设计

学生成长管理包图如下图6-22所示：

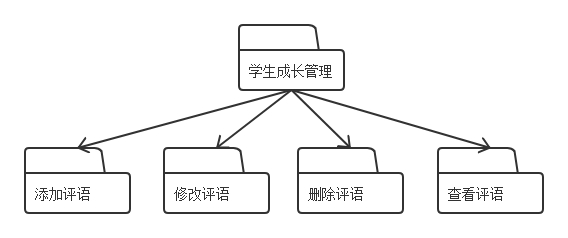


图6-22 学生成长管理包图

1. 添加评语：老师添加对学生各个方面记录的评语。
2. 修改评语：老师可以修改对学生的评语，如评语内容有误。
3. 删除评语：如果老师错误的记录的学生的评语可以进行删除。
4. 查看评语：管理员和老师可以查看老师记录的评语，学生和家长仅能查看该学生的评语。

### 6.7.2 类图设计

学生成长管理主要涉及behave实体类、behaveMapper类、behaveDao类、behaveService类、behaveAction类，behaveMapper类、behaveDao类、behaveService类、behaveAction类，均依赖behave实体类。

学生成长管理的关键类如下图6-23所示：

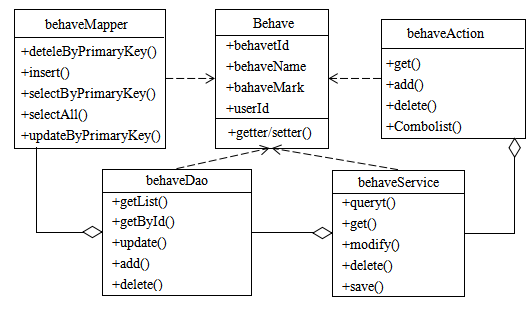


图6-23 学生成长管理类图

behaveDao类描述如下表6-18所示：

表6-18 behaveDao类描述

|  |  |  |
| --- | --- | --- |
| behaveDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | add() | 添加评语 |
| getById() | 通过评语Id获取评语信息 |
| update() | 修改评语信息 |
| delete() | 删除评语信息 |
| getList () | 获得评语列表 |

behaveService类描述如下表6-19所示：

表6-19 behaveService类描述

|  |  |  |
| --- | --- | --- |
| behaveService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | query() | 查询评语信息 |
| get() | 获取评语信息 |
| modify() | 修改评语信息 |
| delete() | 删除评语信息 |
| save() | 保存评语信息 |

behaveAction类如下表6-20所示：

表6-20 behaveAction类描述

|  |  |  |
| --- | --- | --- |
| behaveAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | get() | 获取评语信息 |
| add() | 添加评语信息 |
| delete() | 删除评语信息 |
| ComboList() | 获取评语信息列表 |

### 6.7.3 顺序图设计

学生查询评语时序图如下图6-24所示：

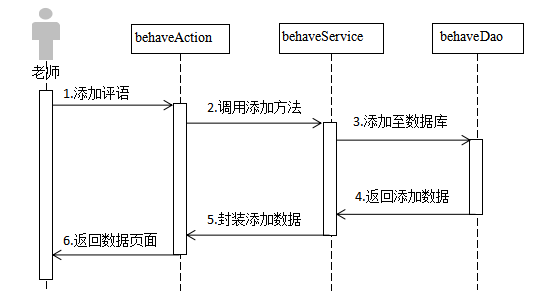


图6-24 查询成长记录时序图

### 6.7.4 核心处理流程设计

学生成长管理流程图如下图6-25所示：

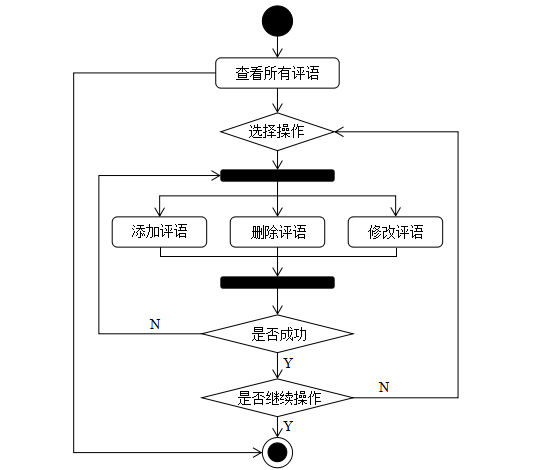


图6-25 学生成长管理流程图

## 6.8 帖子管理

### 6.8.1 功能结构设计

帖子管理包图如下图6-26所示：

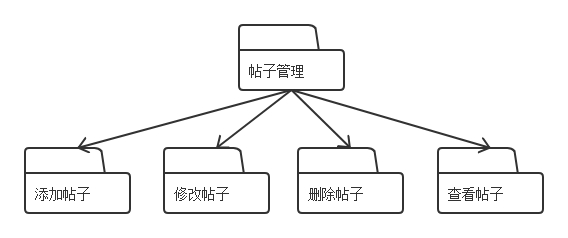


图6-26 帖子管理包图

1. 添加帖子：用户在想要向别的用户请教时可以发布帖子。
2. 修改帖子：用户可以修改已经添加的帖子，如标题不恰当，内容不对。
3. 删除帖子：用户可以删除曾经添加的帖子。
4. 查看帖子：用户可以查看平台所有的帖子。

### 6.8.2 类图设计

帖子管理主要涉及article实体类、articleMapper类、articleDao类、articleService类、articleAction类，articleMapper类、articleDao类、articleService类、articleAction类，均依赖article实体类。

帖子管理的关键类如下图6-27所示：

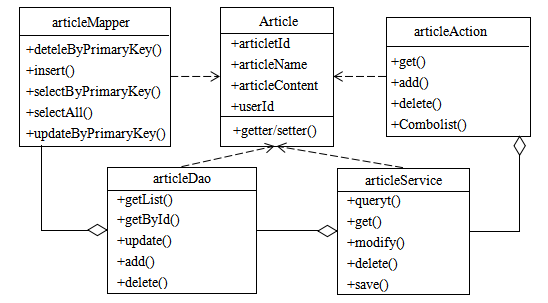


图6-27 帖子管理类图

articleDao类描述如下表6-21所示：

表6-21 articleDao类描述

|  |  |  |
| --- | --- | --- |
| articleDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | getList () | 获得帖子列表 |
| getById() | 通过帖子Id获取帖子 |
| update() | 修改帖子信息 |
| delete() | 删除帖子信息 |
| add() | 添加帖子信息 |

articleService类描述如下表6-22所示：

表6-22 articleService类描述

|  |  |  |
| --- | --- | --- |
| articleService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | query() | 查询帖子信息 |
| get() | 获取帖子信息 |
| modify() | 修改帖子信息 |
| delete() | 删除帖子信息 |
| save() | 保存帖子信息 |

articleAction类描述如下表6-23所示：

表6-23 articleAction类描述

|  |  |  |
| --- | --- | --- |
| articleAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | get() | 获取帖子信息 |
| add() | 添加帖子信息 |
| delete() | 删除帖子信息 |
| ComboList() | 获取帖子信息列表 |

### 6.8.3 顺序图设计

管理员删除帖子时序图如下图6-28所示：

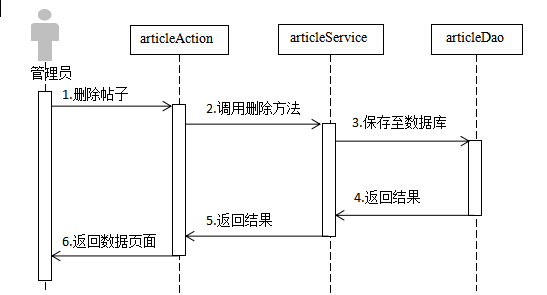


图6-28 删除帖子时序图

### 6.8.4 核心处理流程设计

帖子管理流程图如下图6-29所示：

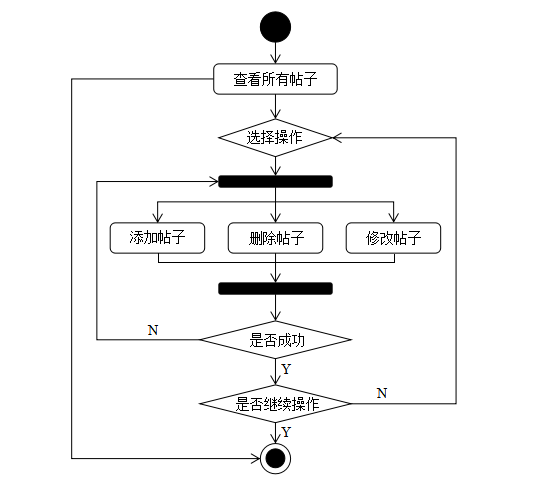


图6-29 帖子管理流程图

## 6.9 评论管理

### 6.9.1 功能结构设计

评论管理包图如下图6-30所示：

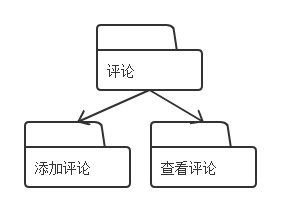


图6-30 评论管理包图

1. 添 加 评 论 ： 用 户 可 以 在 平 台 所 有 的 帖 子 下 发 布 评 论 。
2. 查看评论：用户可以查看帖子下面的评论。

### 6.9.2 类图设计

评论管理主要涉及comment实体类、commentMapper类、commentDao类、commentService类、commentAction类，commentMapper类、commentDao类、commentService类、commentAction类，均依赖comment实体类。

评论管理的关键类如下图6-31所示：

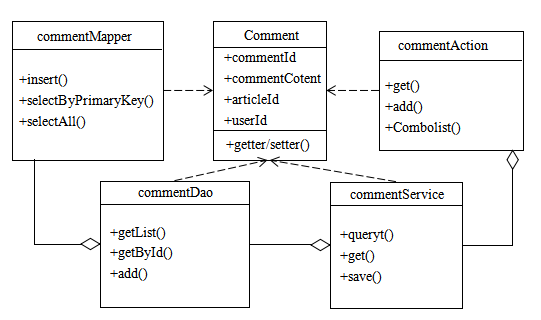


图6-31 评论管理类图

commentDao类描述如下表6-24所示：

表6-24 commentDao类描述

|  |  |  |
| --- | --- | --- |
| commentDao类 | | |
| 类的描述：负责封装对数据库操作的方法 | | |
| 操作描述 | add() | 添加评论 |
| getById() | 通过评论Id获取评论 |
| getList () | 获得评论列表 |

commentService类描述如下表6-25所示：

表6-25 commentService类描述

|  |  |  |
| --- | --- | --- |
| commentService类 | | |
| 类的描述：负责业务逻辑应用设计 | | |
| 操作描述 | query() | 查询评论信息 |
| get() | 获取评论信息 |
| save() | 保存评论信息 |

commentAction类描述如下表6-26所示：

表6-26 commentAction类描述

|  |  |  |
| --- | --- | --- |
| commentAction类 | | |
| 类的描述：负责请求转发 | | |
| 操作描述 | get() | 获取评论信息 |
| add() | 添加评论信息 |
| ComboList() | 获取评论信息列表 |

### 6.9.3 顺序图设计

家长添加评论时序图如下图6-32所示：

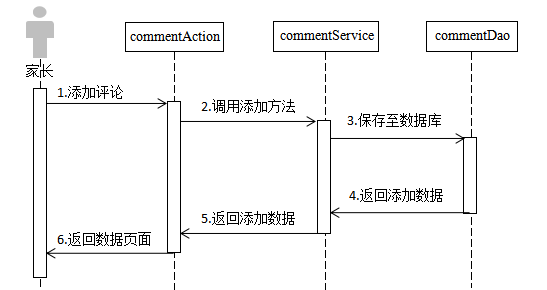


图6-32 添加评论时序图

### 6.9.4 核心处理流程设计

评论管理流程图如下图6-33所示：

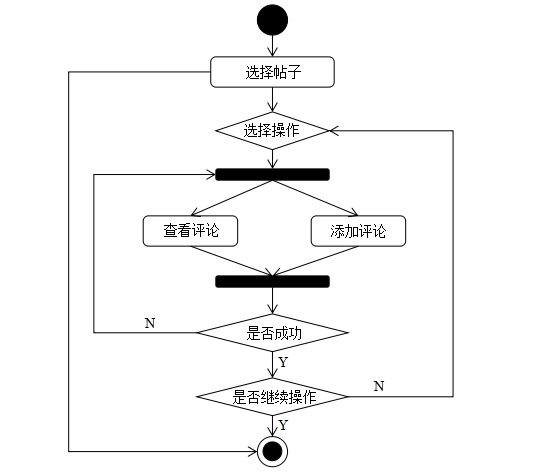


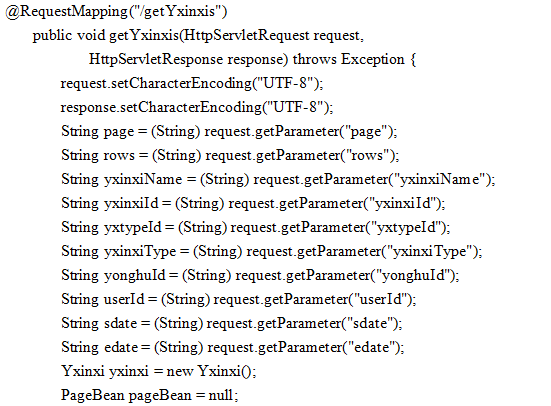
图6-33 评论管理流程图

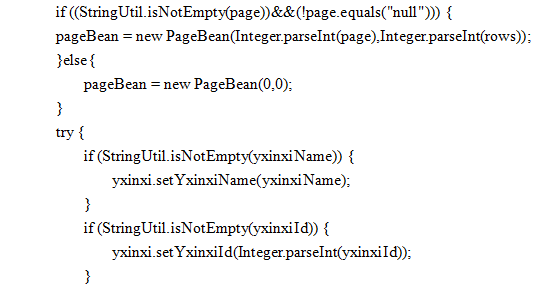
7 编码

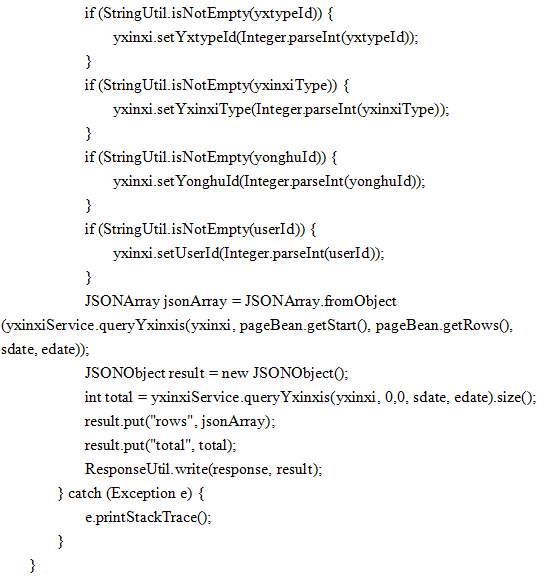
## 7.1 代码实现与核心算法

本段代码的主要目的是处理用户对教学资料的一系列的增加、删除、修改、查询等操作。本系统的基本功能都大部分是这四种操作，因此选取了一个作为代表。

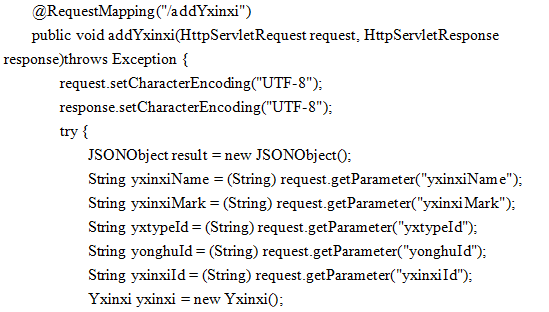
1. 本段代码主要是用户在教学资料页面将数据库中存在的教学资料信息显示在前台页面中。

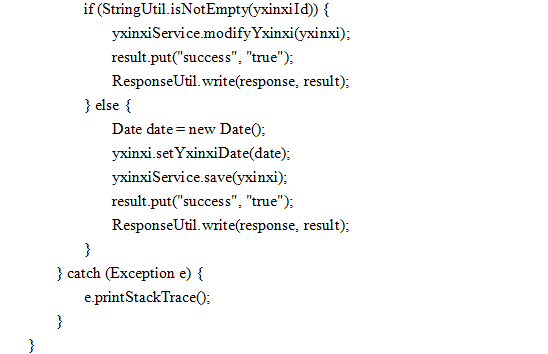
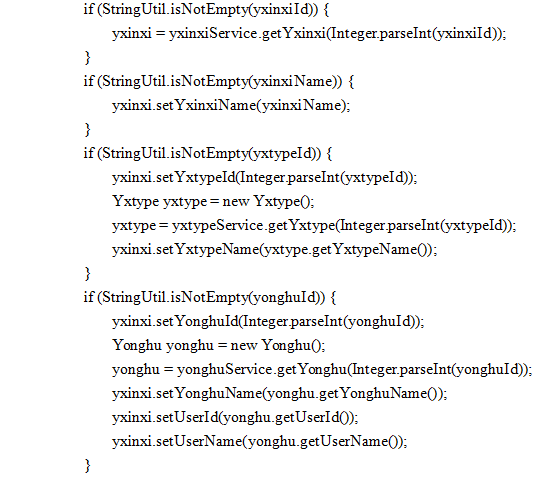




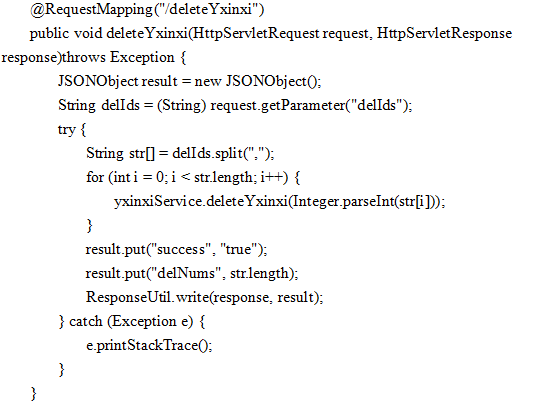


1. 本段代码主要是实现对教学资料进行删除的操作，可以选择一条或多条进行删除。

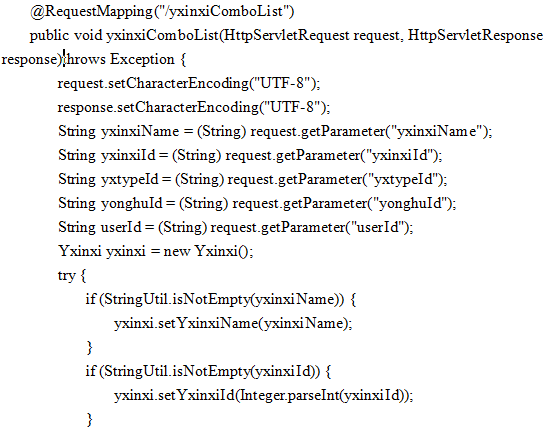


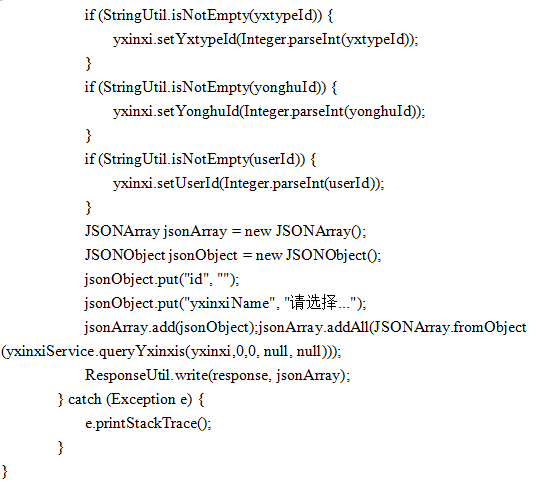


1. 本段代码是老师通过教学资料的Id删除教学资料



1. 本段代码是在搜索或者其他需要选择教学资料是提供教学资料列表，而不用手动输入。





## 7.2 代码优化分析

对于用户角色涉及的信息页面，根据其菜单栏分为个人信息、考试信息、成绩、教学资料、学生成长以及论坛，这些会在不同的角色的页面上都出现，因此本系统将公用的页面部分编写成子模板，在每个页面进行引用，这样可以减少重复代码，优化系统结构，同时提升运行速度。

8 测试

## 8.1 测试方案设计

### 8.1.1 测试策略

一个系统设计好后，就会进入测试阶段，测试的目标是检验设计好的网站是否可以正常无误的运行，尽可能的发现网站的问题，已使后期网站投入使用后网站尽少出错。

中等学校通用IS平台完成了开发之后，需要进行测试，这个测试步骤就是为了检验程序是不是有什么问题存在，主要作用还是发现程序的问题，测试过程中会采用测试经常用到的方法以及手段来挖掘问题，这样做的目的就是保障程序的质量，让用户使用开发出来的程序没有任何顾虑，能够安心运行程序，操作程序对应功能，另外测试期间还可以大致浏览程序的各个界面设计是不是很简洁大方，颜色搭配以及功能布局有没有凌乱的感觉，程序各个部分的功能全部都要反复操作，一旦哪个部分操作逻辑存在问题，或者是功能不能达到用户需求，就需要开发者赶紧纠正，及时更改程序内容，使之最终能实现用户需要的功能，满足用户基本功能需求。关于中等学校通用IS平台的测试，使用了测试用得最多的测试方法，即黑盒测试进行中等学校通用IS平台的测试。

### 8.1.2 测试进度安排

本平台的测试进度计划安排如下：

1. 编写需要进行测试的测试用例
2. 用编写好的测试用例进行测试并记录
3. 测试系统的可操作性和容错性

### 8.1.3 测试资源

为使本系统达到满足测试任务的要求，使用了如下测试资源如下表8-1所示：

表8-1测试资源表

|  |  |  |
| --- | --- | --- |
| **硬件环境** | | |
| **资源类型** | **资源描述** | **数量** |
| 计算机硬件 | Intel I5 8G内存 1T硬盘 | 1 |
| 操作系统 | Windows 7 1366Í768分辨率 | 1 |
| JDK | JDK 1.8 | 1 |
| 浏览器 | 2345浏览器/Firefox 浏览器/360极速浏览器/Chrome浏览器 | 3 |
| **软件环境** | | |
| **资源类型** | **软件** | |
| 应用数据库 | MySQL 5.6.19 | |
| 应用服务器 | Tomcat 8.5.29 | |
| 运行软件 | MyEclipse8.5 | |

### 8.1.4 关键测试点

1. 用户是否实现自己功能范围内的操作
2. 前台页面和后台用户页面的信息显示是否正确
3. 模糊查询是否可以
4. 搜索功能的返回结果是否正确

## 8.2 测试用例构建

### 8.2.1 测试用例编写约定

1. 测试用例与开发的对应约定如表8-1所示：

表 8-1 测试用例与开发约定

|  |  |  |
| --- | --- | --- |
| **开发阶段** | **依据文档** | **编写的用例** |
| 需求分析阶段 | 需求分析文档 | 对应系统测试的用例 |
| 系统设计阶段 | 系统设计文档 | 对应集成测试的用例 |
| 详细设计阶段 | 详细设计文档 | 对应单元测试的用例 |

1. 测试用例类型约定如表8-2所示：

表 8-2 测试用例类型约定

|  |  |  |  |
| --- | --- | --- | --- |
| **测试用例** | **对应测试类型** | **测试覆盖率（测试人员）** | **测试覆盖率（其他人员）** |
| 功能测试用例 | 主要包括中等学校通用IS平台运行过程的容错性测试、可操作性测试以及功能测试 | 85%~100% |  |
| 用户页面测试用例 | 前台页面测试、后台用户信息页面以及少量功能测试 | 100% | 85% |

### 8.2.2 测试用例设计

基于本平台的基本功能制定出测试用例中的执行步骤和预期结果，同时主要采用黑盒测试方法，对平台功能的输入和输出进行制约和检查，在此过程中，主要用到的测试用例设计方法有：

1. 等价类划分法
2. 边界值法
3. 场景法

### 8.2.3 关键测试用例

本系统使用者主要分为四类，分别为管理员、老师、学生和其家长，四类用户的基本功能大致相同，只是权限不一样，因此本文从中选取了一个角色进行功能测试，如下表8-3所示：

表8-3 老师角色测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试目的** | | 测试老师角色的基本功能 | |
| **预置条件** | | 已经成功登录此平台 | |
| **页面** | **测试步骤** | | **预期结果** |
| 个人信息 | 点击修改，弹出填写信息列表，填写用户信息，点击保存 | | 提示修改成功 |
| 点击修改密码，填写原密码和两次新密码，点击修改 | | 提示密码修改成功 |
| 学生信息 | 在搜索框中输入关键字或者选择性别（可多个条件查询），点击搜索 | | 显示含有关键字或者所选择的性别的学生 |

续表8-3

|  |  |  |  |
| --- | --- | --- | --- |
| **页面** | **测试步骤** | | **预期结果** |
| 学生成绩 | 点击添加，填写信息，点击保存 | | 提示添加成功 |
| 点击修改，填写修改信息，点击保存 | | 提示修改成功 |
| 选择一条或者多条信息，点击删除 | | 提示删除成功 |
| 教学资料 | 点击添加，填写信息，点击保存 | | 提示添加成功 |
| 点击修改，填写修改信息，点击保存 | | 提示修改成功 |
| 选择一条或者多条信息，点击删除 | | 提示删除成功 |
| 选择一条信息，点击上传资料，选择文件，点击保存 | | 提示上传成功 |
| 在搜索框中输入关键字或者选择性别（可多个条件查询），点击搜索 | | 显示含有关键字或者所选择的性别的学生 |
| 学生成长 | 点击添加，填写信息，点击保存 | | 提示添加成功 |
| 点击修改，填写修改信息，点击保存 | | 提示修改成功 |
| 选择一条或者多条信息，点击删除 | | 提示删除成功 |
| 在搜索框中输入关键字或者选择性别（可多个条件查询），点击搜索 | | 显示含有关键字或者所选择的性别的学生 |
| 论坛管理 | 点击添加，填写信息，点击保存 | | 提示添加成功 |
| 点击修改，填写修改信息，点击保存 | | 提示修改成功 |
| 选择一个帖子，点击评论，填写评论内容，点击保存 | | 可在该帖子下看机该评论 |
| **测试状态** | | 上述操作均达到“测试”要求 | |
| **测试结果** | | 经过测试，该模块功能正常，能够对用户的正常操作进行处理，也可以对错误的操作进行阻止，并提示用户错误信息，达到了预期目标。 | |

### 8.2.4 测试用例维护

1. 删除测试用例中多余的用例

如果存在两个及以上的测试用例对同一功能针对一组相同的输入输出信息测试，那么这些测试用例是多余的。需要制定一定的时间整理测试用例，然后将多余的测试用例删除。

1. 增添测试用例中没有的用例

如果，某个功能在已经存在的测试用例中没有测试用例来进行测试，那么应该编写新的测试用例对此功能进行功能测试，并将新的测试用例整合到已经存在的测试用例中。

9 总结与展望

## 9.1 设计工作总结

随着互联网的发展，学校信息管理水平也相对应的提高起来，因此开发一个中等学校通用IS平台是非常有必要的，本文根据平台项目的要求按照软件开发的步骤和方法进行了软件的需求分析，系统的设计和实现以及对系统进行各项测试，最终达到了预期的目的和要求。

本文主要包含以下几个方面：

1. 中等学校通用IS平台的需求分析，包括功能需求和非功能需求。
2. 信息管理、考试管理、教学管理、论坛管理的数据库设计、界面设计、功能模块的详细设计和编码实现。
3. 平台各个模块的测试用例的设计和测试。

在开发平台的整个过程中，我对Spring MVC有了更加深的体会，同时也加强了我对软件开发流程的了解以及编写代码的能力。通过该平台的数据库设计，让我掌握了设计良好数据库的能力，能够通过项目的需求设计出一张张有关联的表。通过此次的毕业论文工作，相信会给我以后进行软件开发有很大的帮助。

## 9.2 未来工作展望

虽然本文中的中等学校通用IS平台已经实现了基本的功能，但在本平台中还存在一些设计缺点和不足，想要真正投入使用肯定还不行，需要进一步加以完善。本平台的功能还比较简单，模块还可以添加，如宿舍管理、报修管理等模块，只用一个平台就可满足中等学校大部分人的需求，这样的话学校管理起来也更加方便及时。

# **谢 辞**

参考文献

[1] 张孝祥、徐明华, JAVA基础与案例开发详解, 清华大学出版社, 2009.

[2] 康牧，JSP动态网站开发实用教程，清华大学出版社，2009.

[3] 刘亚宾，精通Eclipse--JAVA技术大系，电子工业出版社，2005.

[4] 王玉英. 基于JSP的SQL数据库访问技术[J]. 现代计算机：专业版, 2010, 19(14):63-66

[5] 赵钢. JSP Servlet+EJB的Web模式应用研究[J]. 电子设计工程, 2013, 21(13):47-49

[6] 肖英. 解决JSP/Servlet开发中的中文乱码问题[J]. 科技传播, 2011, (1)11-25

[7] 黄艳峰. 在Java语言中实施“案例教学”的研究与探索[J]. 电脑知识与技术, 2010, 6(5):1148-1149

[8] 赵钢. JSP Servlet+EJB的Web模式应用研究[J]. 电子设计工程, 2013, 21(13):47-49

[9] 肖英. 解决JSP/Servlet开发中的中文乱码问题[J]. 科技传播, 2011, (1)11-25

[10]邓子云.JSP网络编程从基础到实践[M].北京:电子工业出版社,2009.

[11] Todd Cook. JSP从入门到精通[W].北京：电子工业出版社，2003·

[12] Bruce Eckel. Thinking in java(English Edition). 北京：机械工业出版社,2006·

[13] 赵安君,曾应员,徐邦海，常春藤.网络安全技术于应用[M].北京:人民邮电出版社,2007.

[14] 范明,叶阳东,邱保志,职为梅.数据库原理教程[M].北京:科学出版社,2008.

[15] 孙涌.现代软件工程[M].北京:北京希望电子出版社,2003.

[16] 贾晓芳，沈泽刚.Java Web应用开发中的常见乱码形式及解决方法[J]. 软件导刊，2017,（04）

[17]籍慧文.Web应用开发中JAVA编程语言的应用探讨[J].科技创新与应用，2017, (07)

[18]苟文博，于强.基于MySQL的数据管理系统设计与实现[J].电子设计工程，2017,（06）

[19]杜丁超.计算机软件Java编程特点及其技术分析[J].电脑知识与技术,2017,13(36)

[20]蔡创.计算机软件开发中JAVA编程语言的分析和思考[J].信息技术与信息化,2017(12)

[21]张永强. 计算机软件Java编程特点及其技术分析[J]. 计算机产品与流通,2019(01):23.

[22]王越. JAVA编程语言在计算机软件开发中的应用[J]. 电子技术与软件工程,2019(01):35.

[23]叶欣,李建佳,温煜,赖舒婷. 基于Java语言的医疗质量管理系统设计与实现[J]. 中国数字医学,2018,13(12):54-55+104.

[24]曹文渊. JAVA语言在计算机软件开发中的应用[J]. 电子技术与软件工程,2019(02):53-54.

[25]倪海顺. 计算机软件开发的Java编程语言应用探讨[J]. 信息与电脑(理论版),2019(02):60-61.

[26] Hsiao I H, Sosnovsky S, Brusilovsky P. Guiding students to the right questions: adaptive navigation support in an e-learning system for Java programming[J]. Journal of Computer Assisted Learning, 2010, 26(4):270-283·

[27] Vegh A. MYSQL Database Server[M]. Web Development with the Mac®. Wiley Publishing, Inc., 2011,179-194·

[28] Rasoolzadegan A, Barforoush A A. Reliable yet flexible software through formal model transformation (rule definition)[J]. Knowledge & Information Systems, 2014, 40(1):79-126·

[29] Rongrong Gong, Shijian Luo, Ji He.Use Case Based Innovative Design of E-commerce Website ，Journal of Computer Science and Technology 2012，5：32-35

[30] Jan Hatzius，Goldman Sachs Investment Research. Internet B2B eCommerce B2B:2B or Not 2B? Version1.1,2010,12: 68-71.

附录A 外文翻译—原文部分

The Source Of Article: Spring MVC Framework for Web 2.0, Thakare R V , Kakade S , Sapre B , et al , 1(3):242-247

An application as having three main layers: presentation (UI), application logic, and resource management. In MVC, the presentation layer is split into controller and view. The most important separation is between presentation and application logic. The View/Controller split is less so. MVC encompasses more of the architecture of an application than is typical for a design pattern. Hence the term architectural pattern may be useful, or perhaps an aggregate design pattern.

*Model*

The domain-specific representation of the information on which the application operates. The model is another name for the application logic layer (sometimes also called the domain layer). Application (or domain) logic adds meaning to raw data (e.g., calculating if today is the user’s birthday, or the totals, taxes and shipping charges for shopping cart items). Many applications use a persistent storage mechanism (such as a database) to store data. MVC does not specifically mention the resource management layer because it is understood to be underneath or encapsulated by the Model.

*View*

Renders the model into a form suitable for interaction, typically a user interface element. MVC is often seen in web applications, where the view is the HTML page and the code which gathers dynamic data for the page.

*Controller*

Processes and responds to events, typically user actions, and may invoke changes on the model and view. Though MVC comes in different flavours, the control flow generally works as follows:

1. The user interacts with the user interface in some way (e.g., user presses a button).

2. A controller handles the input event from the user interface, often via a registered handler or callback.

3. The controller accesses the model, possibly updating it in a way appropriate to the user’s action (e.g., controller updates user’s shopping cart). Complex controllers are often structured using the command pattern to encapsulate actions and simplify extension.

4. A view uses the model to generate an appropriate user interface (e.g., view produces a screen listing the shopping cart contents). The view gets its own data from the model. The model has no direct knowledge of the view. (However, the observer pattern can be used to allow the model to indirectly notify interested parties, potentially including views, of a change.)

5. The user interface waits for further user interactions, which begins the cycle anew.

*A. Implementation Strategy*

The Spring Web model-view-controller (MVC) framework is designed around a DispatcherServlet that dispatches requests to handlers, with configurable handler mappings, view resolution, locale and theme resolution as well as support for uploading files.

*B. Features of Spring Web MVC*

Clear separation of roles. Each role -- controller, validator, command object, form object, model object, DispatcherServlet, handler mapping, view resolver, and so on can be fulfilled by a specialized object.

Powerful and straightforward configuration of both framework and application classes as JavaBeans. This configuration capability includes easy referencing across contexts, such as from web controllers to business objects and validators.

Adaptability, non-intrusiveness, and flexibility. Define any controller method signature you need, possibly using one of the parameter annotations (such as @RequestParam, @RequestHeader, @PathVariable,and more) for a given scenario.

Reusable business code, no need for duplication. Use existing business objects as command or form objects instead of mirroring them to extend a particular framework base class.

Customizable binding and validation. Type mismatches as application-level validation errors that keep the offending value, localized date and number binding, and so on instead of String-only form objects with manual parsing and conversion to business objects.

Customizable handler mapping and view resolution. Handler mapping and view resolution

strategies range from simple URL-based configuration, to sophisticated, purpose-built resolution strategies. Spring is more flexible than web MVC frameworks that mandate a particular technique.

Flexible model transfer. Model transfer with a name/value Map supports easy integration with any view technology.

Customizable locale and theme resolution, support for JSPs with or without Spring tag library, support for JSTL, support for Velocity without the need for extra bridges, and so on.

A simple yet powerful JSP tag library known as the Spring tag library that provides support for features such as data binding and themes. The custom tags allow for maximum flexibility in terms of markup code.

A JSP form tag library, introduced in Spring 2.0, that makes writing forms in JSP pages much easier.

Beans whose lifecycle is scoped to the current HTTP request or HTTP Session. This is not a specific feature of Spring MVC itself, but rather of the WebApplicationContext container(s) that Spring MVC uses.

*C. Pluggability of other MVC Implementation*

If you do not want to use Spring's web MVC, but intend to leverage other solutions that Spring offers, you can integrate the web MVC framework of your choice with Spring easily. Simply start up a Spring root application context through its ContextLoaderListener, and access it through its

ServletContext attribute (or Spring's respective helper method) from within a Struts or WebWork action. No "plug- ins" are involved, so no dedicated integration is necessary. From the web layer's point of view, you simply use Spring as a library, with the root application context instance as the

entry point.

Your registered beans and Spring's services can be at your fingertips even without Spring's Web MVC. Spring does not compete with Struts or WebWork in this scenario. It simply addresses the many areas that the pure web MVC frameworks do not, from bean configuration to data access

and transaction handling. So you can enrich your application with a Spring middle tier and/or data access tier, even if you just want to use, for example, the transaction abstraction with JDBC or Hibernate.

Spring's web MVC framework is, like many other web MVC frameworks, request-driven, designed around a central servlet that dispatches requests to controllers and offers other functionality that facilitates the development of web applications. Spring's DispatcherServlet however, does morethan just that. It is completely integrated with the Spring IoC container and as such allows you to use every other feature that Spring has.

The pattern-savvy reader will recognize that the DispatcherServlet is an expression of the “Front Controller” design pattern (this is a pattern that Spring Web MVC shares with many other leading web frameworks).

The DispatcherServlet is an actual Servlet (it inherits from the HttpServlet base class), [3] and as such is declared in the web.xml of your web application. You need to map requests that you want the DispatcherServlet to handle, by using a URL mapping in the same web.xml file. This is standard J2EE servlet configuration; the following example shows such a DispatcherServletdeclaration and mapping.

In the preceding example, all requests ending with .form will be handled by the exampleDispatcherServlet. This is only the first step in setting up Spring Web MVC. You now need to configure the various beans used by the Spring Web MVC framework. Upon initialization of a DispatcherServlet, the framework looks for a file named [servlet-name]-servlet.xml in the WEB-INF directory of your web application and creates the beans defined there, overriding the definitions of any beans defined with the same name in the global scope.

The Spring DispatcherServlet uses special beans to process requests and render the appropriate views. These beans are part of Spring Framework. You can configure them in the WebApplicationContext, just as you configure any other bean. However, for most beans, sensible defaults are provided so you initially do not need to configure them. These beans are described in the following table.

After you set up a DispatcherServlet, and a request comes in for that specific DispatcherServlet, the DispatcherServlet starts processing the request as follows:

1. The WebApplicationContext is searched for and bound in the request as an attribute that the controller and other elements in the process can use. It is bound by default under the key DispatcherServlet.WEB\_APPLICATION\_CONTEXT\_ATTRIBUTE.

2. The locale resolver is bound to the request to enable elements in the process to resolve the locale to use when processing the request (rendering the view, preparing data, and so on). If you do not need locale resolving, you do not need it.

3. The theme resolver is bound to the request to let elements such as views determine which theme to use. If you do not use themes, you can ignore it.

4. If you specify a multipart file resolver, the request is inspected for multiparts; if multiparts are found, the request is wrapped in a MultipartHttpServletRequest for further processing by other elements in the process.

5. An appropriate handler is searched for. If a handler is found, the execution chain associated with the handler (preprocessors, postprocessors, and controllers) is executed in order to prepare a model or rendering.

6. If a model is returned, the view is rendered. If no model is returned, (may be due to a preprocessor or postprocessor intercepting the request, perhaps for security reasons), no view is rendered, because the request could already have been fulfilled.

附录B 外文翻译—译文部分

文章来源：Spring MVC Framework for Web 2.0[J], Thakare R V , Kakade S , Sapre B , et al , 1(3):242-247

应用程序有三个主要层：表现层（UI）、应用程序逻辑和资源管理。在MVC中，表现层分为控制器和视图。最重要的分离是表示和应用程序逻辑之间的分离。视图/控制器拆分较少。与设计模式相比，MVC包含了更多的应用程序架构。因此，术语架构模式可能是有用的，或者可能是一个聚合的设计模式。

1. 模型

应用程序在其上运行的信息的特定于域的表示形式。 该模型是应用程序逻辑层（有时也称为域层）的另一个名称。 应用程序（或域）逻辑为原始数据增加了含义（例如，计算今天是否是用户的生日，或计算购物车商品的总额，税金和运费）。 许多应用程序使用持久性存储机制（例如数据库）来存储数据。 MVC没有特别提及资源管理层，因为它被理解为在模型的下方或封装。

1. 视图

将模型渲染为适合交互的形式，通常是用户界面元素。 MVC通常出现在Web应用程序中，其中的视图是HTML页面和为页面收集动态数据的代码。

1. 控制器

处理并响应事件（通常是用户操作），并可以在模型和视图上调用更改。 尽管MVC具有不同的风格，但控制流程通常如下：

1. 用户以某种方式与用户界面进行交互（例如，用户按下按钮）。
2. 控制器通常通过注册的处理程序或回调来处理来自用户界面的输入事件。
3. 控制器访问模型，并可能以适合用户操作的方式对其进行更新（例如，控制器更新用户的购物车）。 复杂的控制器通常使用命令模式进行结构化，以封装动作并简化扩展。
4. 视图使用该模型生成适当的用户界面（例如，视图产生列出购物车内容的屏幕）。 视图从模型中获取自己的数据。 该模型没有直接的视图知识（但是，可以使用观察者模式来允许模型将更改间接地通知相关方，可能包括视图。）
5. 用户界面等待其他用户交互，从而重新开始循环。

1. 实施策略

Spring Web模型-视图-控制器（MVC）框架是围绕DispatcherServlet设计的，该框架将请求分配给处理程序，并具有可配置的处理程序映射，视图分辨率，区域设置和主题分辨率以及对上传文件的支持。

1. Spring Web MVC的功能
2. 角色分离清晰。每个角色-控制器，验证器，命令对象，表单对象，模型对象，DispatcherServlet，处理程序映射，视图解析器等等，都可以由专门的对象来实现。
3. 框架和应用程序类作为JavaBean的强大而直接的配置。 此配置功能包括跨上下文的轻松引用，例如从Web控制器到业务对象和验证器。
4. 适应性，非侵入性和灵活性。 定义给定方案所需的任何控制器方法签名，可能使用参数注释之一（例如@ RequestParam，@ RequestHeader，@ PathVariable等）。
5. 可重用的业务代码，无需重复。使用现有业务对象作为命令或表单对象，而不是对其进行镜像以扩展特定的框架基类。
6. 可定制的绑定和验证。类型不匹配是应用程序级验证错误，它保留有问题的值，本地化的日期和数字绑定等，而不是通过手动解析并转换为业务对象的纯字符串形式的对象。
7. 可自定义的处理程序映射和视图分辨率。处理程序映射和视图解析策略的范围从基于URL的简单配置到专门针对目标的复杂解析策略。Spring比要求特定技术的Web MVC框架更加灵活。
8. 灵活的模型传输。使用名称/值进行模型传递Map支持与任何视图技术轻松集成。
9. 可自定义的区域设置和主题解析，支持带有或不带有Spring标记库的JSP，支持JSTL，无需任何桥接就支持Velocity等。
10. 一个简单但功能强大的JSP标签库，称为Spring标签库，它为诸如数据绑定和主题之类的功能提供支持。自定义标签在标记代码方面提供了最大的灵活性。
11. 在Spring 2.0中引入的JSP表单标签库使在JSP页面中编写表单变得更加容易。
12. 生命周期仅限于当前HTTP请求或HTTP会话的Bean。这不是Spring MVC本身的特定功能，而是Spring MVC使用的WebApplicationContext容器。
13. 其他MVC实现的可插拔性

如果您不想使用Spring的Web MVC，但打算利用Spring提供的其他解决方案，则可以轻松地将自己选择的Web MVC框架与Spring集成在一起。只需通过ContextLoaderListener启动Spring根应用程序上下文，然后通过其访问Struts或WebWork操作中的ServletContext属性（或Spring的相应辅助方法）。不涉及“插件”，因此不需要专门的集成。从Web层的角度来看，您只需将Spring用作库，将根应用程序上下文实例用作入口点。

即使没有Spring的Web MVC，您注册的Bean和Spring的服务也可以唾手可得。在这种情况下，Spring无法与Struts或WebWork竞争。它只是解决了纯Web MVC框架所没有的许多领域，从bean配置到数据访问和交易处理。 因此，即使您只想使用JDBC或Hibernate等事务抽象，您也可以使用Spring中间层和/或数据访问层来丰富您的应用程序。

与许多其他Web MVC框架一样，Spring的Web MVC框架是由请求驱动的，围绕中央Servlet设计，该Servlet将请求分发给控制器并提供其他功能来促进Web应用程序的开发。但是，Spring的DispatcherServlet不仅能做到这一点。它与Spring IoC容器完全集成在一起，因此，您可以使用Spring具有的所有其他功能。

精通模式的读者将认识到DispatcherServlet是“ Front Controller”设计模式的一种表达（这是Spring Web MVC与许多其他领先的Web框架共享的模式）。

DispatcherServlet是一个实际的Servlet（它继承自HttpServlet基类），[3]并因此在您的Web应用程序的web.xml中声明。您需要通过在同一web.xml文件中使用URL映射来映射希望DispatcherServlet处理的请求。这是标准的J2EE Servlet配置。以下示例显示了这样的DispatcherServlet声明和映射。

在前面的示例中，所有以.form结尾的请求都将由exampleDispatcherServlet处理。 这只是设置Spring Web MVC的第一步。现在，您需要配置Spring Web MVC框架使用的各种bean。在全球范围内，在初始化DispatcherServlet时，框架会在Web应用程序的WEB-INF目录中查找名为[servlet-name] -servlet.xml的文件，并创建在其中定义的Bean，覆盖所有用相同名称定义的Bean的定义。

Spring DispatcherServlet使用特殊的bean来处理请求并呈现适当的视图。 这些bean是Spring Framework的一部分。您可以在WebApplicationContext中配置它们，就像配置任何其他bean一样。但是，对于大多数bean，提供了合理的默认值，因此您最初不需要配置它们。下表描述了这些bean。

设置DispatcherServlet并收到针对该特定DispatcherServlet的请求之后，DispatcherServlet开始按以下方式处理请求：

1. 搜索WebApplicationContext并将其绑定为请求中的属性，控制器和流程中的其他元素可以使用该属性。默认情况下，它绑定在键DispatcherServlet.WEB\_APPLICATION\_CONTEXT\_ATTRIBUTE下。
2. 语言环境解析器绑定到请求，以使流程中的元素能够解析在处理请求（呈现视图，准备数据等）时要使用的语言环境。如果不需要区域设置解析，则不需要它。
3. 主题解析器绑定到请求，以使诸如视图之类的元素确定要使用的主题。如果不使用主题，则可以将其忽略。
4. 如果指定了多部分文件解析器，则检查请求中是否有多部分；否则，将检查请求。如果找到多部分，则将该请求包装在MultipartHttpServletRequest中，以供流程中的其他元素进一步处理。
5. 搜索适当的处理程序。如果找到处理程序，则执行与处理程序（预处理器，后处理器和控制器）关联的执行链，以准备模型或渲染。
6. 如果返回模型，则呈现视图。如果未返回任何模型（可能是由于安全性原因，可能是由于预处理器或后处理器拦截了该请求），则不会呈现任何视图，因为该请求可能已被满足。

附录C 软件使用说明书

1. 前台可以任意用户查看，如新闻信息、交流信息等，如图1所示：



图1 前台界面图

1. 用户登录：用户通过用户名和密码登录，在登录前需要选择相应的身份才能登录，如图2所示：



图2 用户登录界面

1. 个人信息：可以查看登录的用户信息，点击修改可以修改用户信息，如图3所示：

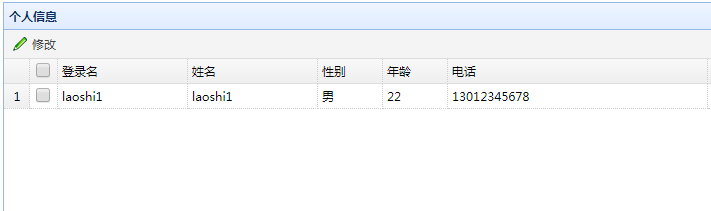


图3 个人信息界面

1. 修改密码：先填写原密码，再填写两遍新密码，点击修改，如图4所示：



图4 修改密码界面

1. 网站信息包括新闻信息：管理员可以添加、修改、删除新闻，如图5所示：

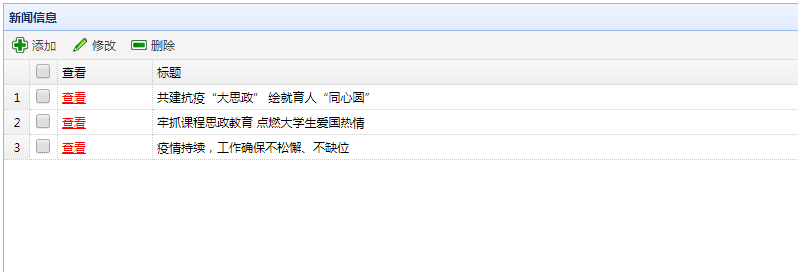


图5 新闻信息界面

1. 专业班级包括专业信息和班级信息：管理员可以对专业信息和班级信息进行添加、修改、删除操作，如图6所示：

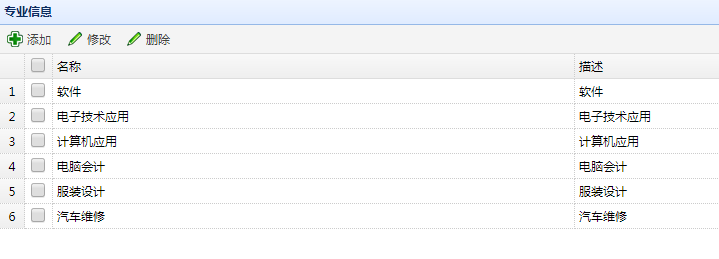


图6 专业信息界面

1. 人员信息包括老师、学生、家长的信息：管理员可以对人员信息进行添加、修改、删除操作，如图7所示：

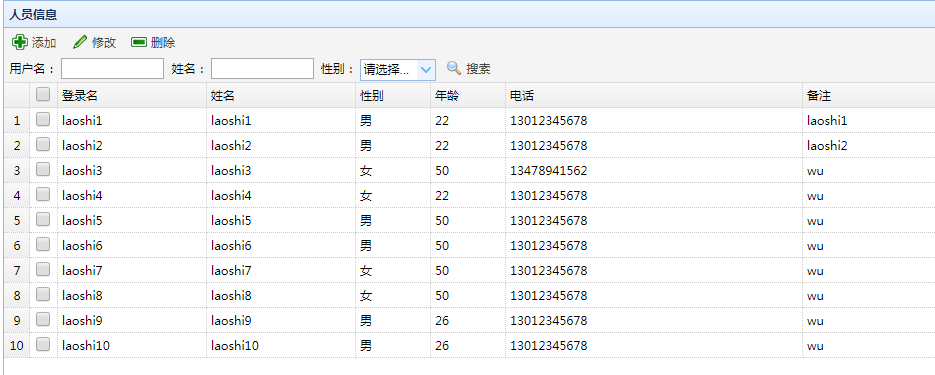


图7 人员信息界面

1. 学生成绩包括考试信息和学生的成绩：管理员可以对考试信息进行添加、修改、删除操作；可以查看学生成绩。老师可以对学生成绩进行添加、修改、删除操作；可以查看考试信息。学生和家长可以查看考试信息和学生成绩如图8和图9所示：

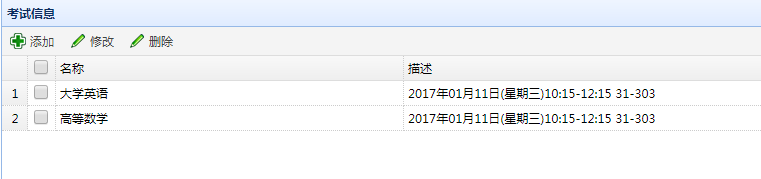


图8 专业信息界面

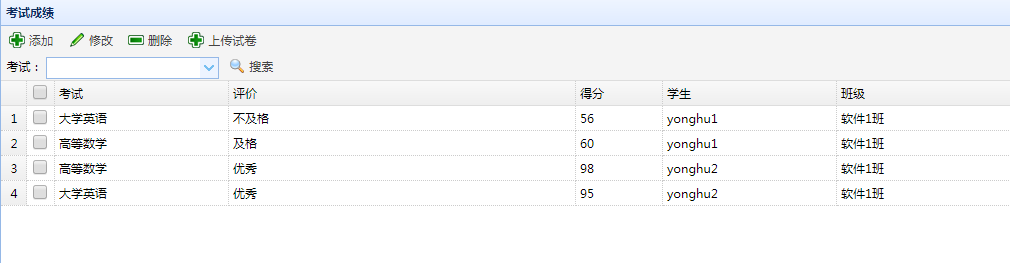


图9 考试成绩界面

1. 教学资料：老师可以对教学资料进行添加、修改、删除操作；学生可以下载老师上传的教学资料，如图10所示：

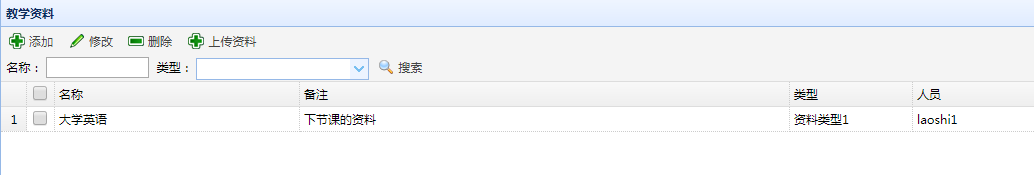


图10 教学资料界面

1. 学生成长：老师可以对学生成长（记录学生的日常表现）进行添加、修改、删除操作，学生和家长可以查看老师记录的相应学生的日常表现，如图11所示：

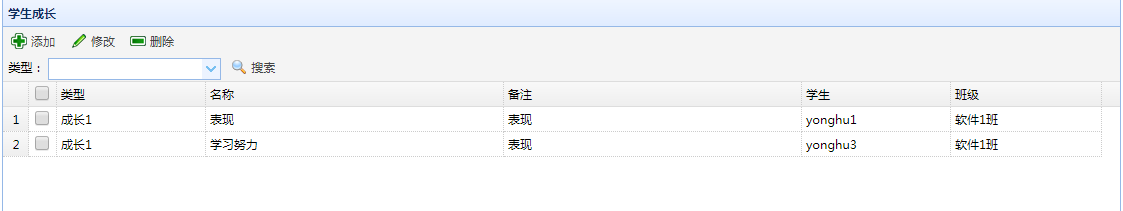


图11 学生成长界面

1. 论坛交流：用户可以对帖子进行添加、修改、删除操作；在相应的帖子可以添加和查看评论，如图12和13所示：

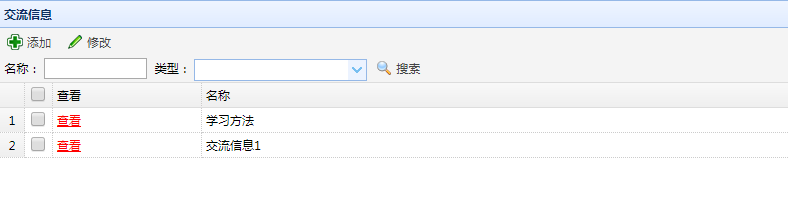


图12 交流信息界面

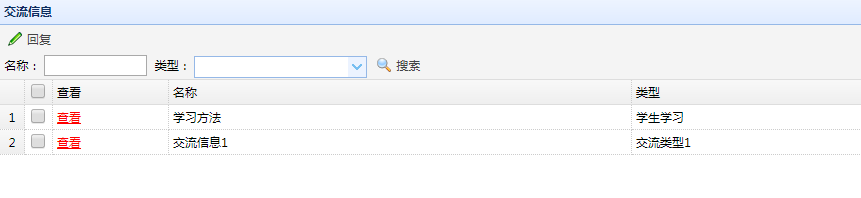
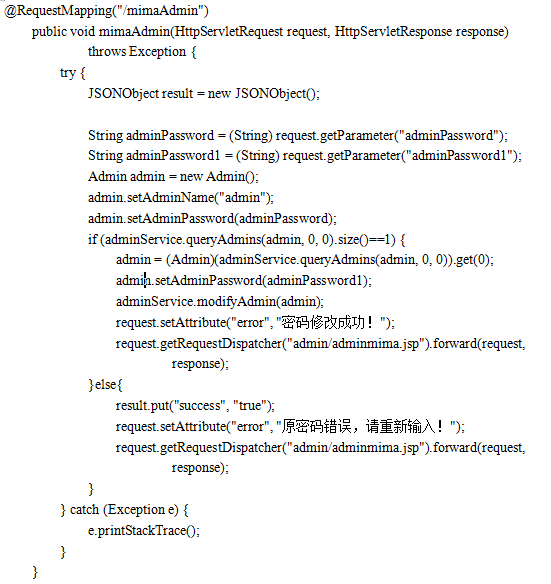


图12 交流信息界面

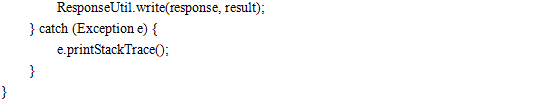
附录D 主要源代码

1. 管理员修改密码

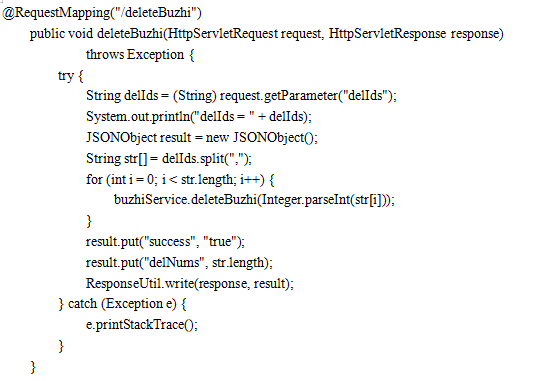


1. 添加专业信息





1. 删除班级信息



1. 获取用户列表



