**(2014级)**

**学生信息管理系统设计说明书**

|  |  |  |
| --- | --- | --- |
| **学 院、系：** | **信息工程学院** | |
| **专 业：** | **计算机软件技术** | |
| **学 生 姓 名：** | **李金军** | |
| **指导教师姓名：** | **易晓鹏** | **职称** |
| **联名指导老师**  **最终评定成绩：** | **职称**  **100** | |

2016年11月

摘要 本系统依据开发要求主要应用于教育系统，完成对日常的教育工作中学生成绩档案的数字化管理。开发本系统可使学院教职员工减轻工作压力，比较系统地对教务、教学上的各项服务和信息进行管理，同时，可以减少劳动力的使用，加快查询速度、加强管理，以及国家各部门关于信息化的步伐，使各项管理更加规范化。

本系统在开发过程中，注意使其符合操作的业务流程，并力求系统的全面性、通用性，使得本系统不只适用于一家教育机构。在开发方法的选择上，选择了生命周期法与原型法相结合的方法，遵循系统调查研究、系统分析、系统设计和系统实施四个主要阶段进行设计，而在具体的设计上，采取了演化式原型法，随着用户的使用及对系统了解的不断加深，对某一部分或几部分进行重新分析、设计、实施。本论文主要从系统分析、系统设计、系统实施与使用等几个方面进行介绍。

关键字：信息管理系统 信息化 数据库

Abstract Development of this system is based on requirements is mainly used in the educational system, completed routine education of students ' achievement in digital management of archives. Development of this system allows College faculty and alleviate the work pressure, more academic, teaching, the system of the management services and information, and can reduce the use of labour, speeding up the query speed, strengthen management, and State departments on the pace of information technology, make the management more standardized.

This system during the development process, attention to the requirements of the operation, and to seek business process system of comprehensive, universal, so that the system not only for an educational institution. In the development of the method chosen, select a life cycle approach to the prototype method, the method of combining research, follow the system analysis, system design and system implementation phase 4 of the main design, and in particular on the design, evolution has taken law-type prototype, as the user's use and understanding of the system, and the deepening of a part or parts to analyze, design and implementation. This paper mainly from system analysis, system design, implementation and use of an introduction.

Keywords: information management system Information Database

目 录

**[第一章 引言](#_Toc405551538)** [1](#_Toc405551538)

**[第二章 系统分析](#_Toc405551539)** [2](#_Toc405551539)

[2.1可行性研究 2](#_Toc405551540)

[2.2系统可行性分析 2](#_Toc405551541)

[2.3用户需求分析 2](#_Toc405551542)

[2.4学生信息管理系统目标 2](#_Toc405551543)

[2.5硬件开发环境研究分析及运行环境 2](#_Toc405551544)

**[第三章 系统设计](#_Toc405551545)** [3](#_Toc405551545)

[3.1系统模块设计 3](#_Toc405551546)

[3.2系统功能结构图 3](#_Toc405551547)

[3.3系统功能模块结构E-R图设计 3](#_Toc405551548)

[3.4 数据库逻辑设计 3](#_Toc405551549)

**[第四章 系统的实现](#_Toc405551550)** [4](#_Toc405551550)

[4.1用户登陆界面 4](#_Toc405551551)

[4.2用户管理界面 4](#_Toc405551552)

[4.3教师信息管理界面 4](#_Toc405551553)

[4.4班级管理界面 4](#_Toc405551554)

[4.5学生信息管理界面 4](#_Toc405551555)

[4.6成绩管理界面 4](#_Toc405551556)

[4.7课程管理界面 4](#_Toc405551557)

[4.8数据库连接配置 4](#_Toc405551558)

[结论 5](#_Toc405551559)

[致谢语 6](#_Toc405551560)

[参考文献 7](#_Toc405551561)

**第一章 引言**

学生信息管理系统（Student Information Management System）,以下简称SIMS，是针对学校人事处的大量业务处理工作而开发的管理软件，是典型的管理信息系统（Management Information System）。它是一个教育单位不可缺少的部分,它的内容对于学校的决策者和管理者来说都至关重要,它能够为用户提供充足的信息和快捷的查询手段。能有效的帮助学校和老师掌握学生的情况，为学生提供成绩跟课程方面的查询。在传统模式下利用人工进行学生信息管理，存在着较多的缺点，如：效率底，保密性差，时间一长将产生大量的文件和数据，更不便于查找，更新，维护等。诸如这些情况，令学校管理者对学生的信息管理带来了很大困难，严重影响了教育工作者的工作效率。随着科学技术的不断提高,计算机科学日渐成熟, 使用日趋成熟的计算机技术来代替传统的人工模式，来实现学生信息的现代化管理，其强大的功能已为人们深刻认识,它已进入人类社会的各个领域并发挥着越来越重要的作用。作为计算机应用的一部分,使用计算机对学生信息进行管理,具有着手工管理所无法比拟的优点，例如:检索迅速、查找方便、易修改、可靠性高、存储量大、数据处理快捷、保密性好、寿命长、成本低、便于打印等。这些优点能够极大地提高学生信息管理的效率,也是学校的科学化、正规化管理,与世界接轨的重要条件。用计算机制作的学生信息管理系统还可以通过功能强大的Internet网及时的向学生的家长传递该生在校的最新成绩，有助于学校与家长的信息互动，为更好地把握学校的教育方针一定的促进作用！因此，开发这样一套管理软件成为很有必要的事情。

**第二章 系统分析**

# **2.1可行性研究**

21世纪以来，人类经济高速发展，人们发生了日新月异的变化，特别是计算机的应用及普及到经济和社会生活的各个领域。使原本的旧的管理方法越来越不适应现在社会的发展。许多人还停留在以前的手工操作。这大大地阻碍了人类经济的发展。为了适应现代社会人们高度强烈的时间观念，学生信息管理系统软件为学校办公室带来了极大的方便。

# **2.2系统可行性分析**

本系统是用C#语言开发，并使用SQL Server 2008作为数据库管理学生信息管理系统。在计算机高度普及的当前社会状况下，这个系统的可行性主要体现在以下的几个方面：

1.技术可行性

可运用ADO.NET窗体技术进行开发，并结合适当的操作平台（Windows XP），可进行例如登陆界面的窗体开发，通过C#语言进行程序代码的设计，用SQL Server 2008 作为数据库。

2.经济可行性

现在，计算机的价格已经十分低廉，性能却有了长足的进步。而本系统的开发，为学校的工作效率带来了一个质的飞跃，为此主要表现有以下几个方面：

a.本系统的运行可以代替人工进行许多繁杂的劳动；

b.本系统的运行可以节省许多资源；

c.本系统的运行可以大大的提高学校的工作效率；

d.本系统可以使敏感文档更加安全，等等。

3.操作可行性

本系统为一个小型的学生信息管理系统，所耗费的资源非常的小，学校的电脑无论是硬件还是软件都能够满足条件，因此，本系统在运行上是可行的。

# **2.3用户需求分析**

1.需求分析说明

(1)采用现有的资源，先进的管理系统开发方案，充分利用学校现有的资源，减少开发中的时间和财力、物力、提高系统开发的水平和应用效果。

(2)系统满足学校的需求，例如学生信息的录入、查询、更新等。学生录入与排名

(3)系统具备数据库维护功能，及时根据用户需求进行数据添加、删除、修改等操作。

2.学校工作流程分析

学校工作总体规划由教务人员在学生信息管理系统中完成对运行教务处所需的基本数据的维护，包括这些信息的增加、修改及对各项信息的变动都将在这进行操作。

新的学年，教务人员首先加入年级信息，然后编排班级，再对来校学生进行基本的信息录入，新生入学后由教务人员在学籍系统中完成新学生信息的维护。在每个学期开始，教务处根据班级的情况，以班为单位，为每个班级安排一个班主任及对此年级安排一个年级组长，并对各科老师进行安排。

每举行一次考试后由任课老师对成绩进行录入，班主任对本班的成绩汇总。并进行排名，然后年级组长再进行汇总，并对本年级各科成绩及总成绩进行排名。

教务处、年级组长、班主任及任课老师跟据实际情况对录入的成绩进行维护，各位同学对以上录入的信息可以跟据自己的需要进行适当的查询。

3.性能需求分析

安全性：系统中涉及到大量学生档案、成绩的信息资料以及大量课程信息，因此要具有良好的安全性，以保证学校信息的安全。

实时性：系统中存储的大量信息，为使学校能够更好地了解学生的相关信息，系统需要及时更新，并且系统响应时间要短，以提高工作效率。

可扩展性：系统采用模块化程序设计的方法，使用B/S体系结构，既便于系统功能的各种组合和修改，又便于未参与开发的技术维护人员补充，维护。

可操作性：该系统采用控件式的页面布局，使得各种操作更加简便，许多选项只需要点击鼠标就可以完成。

# **2.4学生信息管理系统目标**

（1）节约资源，提高学籍信息的精确度

本系统能减少很多不必要的资源，不用象以前那样用冗余的纸张式的管理。大节省了学校能源。并且计算机的存储与快速查询功能大大提高了学籍管理的效率，并且还提高了学籍信息管理的精确度。

（2）方便快速操作，精减人员，节约开支

方便快速的操作，可减少学籍信息管理的漏洞，又减少因工作的冗余出现的错误，并且操作非常简单，可减少许多不很必要的人员，这无论从物质上还是工作人员的工资上都为学校节约了开支。为学校增加了财富。

（3）数据库分析

用户需求具体有学籍管理系统提供保存、更新、查询、维护，这就需求数据库结构能充分满足各种信息的输入与输出，实现有组织地、动态地存储大量关联数据，方便用户访问系统中的数据，它与文件系统的重要区别是数据的充分共享，交叉访问，与应用程序的高度独立性。

# **2.5硬件开发环境研究分析及运行环境**

（1）硬件要求

CPU：300MHz以上的处理器。

内存：4GB，推荐8GB。

硬盘：1GB以上剩余空间。

显示像素：最低800\*600，最佳效果1024\*768。

（2）运行环境Windows7或更高版本的操作系统

数据库系统：Microsoft SQL Server 2008

（3）开发环境

开发工具：Microsoft Visual Studio 2010

数据库系统：Microsoft SQL Server 2008

**第三章 系统设计**

**3.0 三层架构**

[三层架构](http://baike.baidu.com/view/687468.htm" \t "http://baike.baidu.com/_blank)(3-tier architecture) 通常意义上的三层架构就是将整个业务应用划分为：界面层（User Interface layer）、业务逻辑层（Business Logic Layer）、数据访问层（Data access layer）。区分层次的目的即为了“[高内聚低耦合](http://baike.baidu.com/view/3082578.htm" \t "http://baike.baidu.com/_blank)”的思想。在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构。微软推荐的分层式结构一般分为三层，从下至上分别为：数据访问层、业务逻辑层（又或称为领域层）、表示层。

**表示层**

位于最外层（最上层），最接近用户。用于显示数据和接收用户输入的数据，为用户提供一种交互式操作的界面。

**业务逻辑层**

业务逻辑层（Business Logic Layer）无疑是系统架构中体现核心价值的部分。它的关注点主要集中在业务规则的制定、业务流程的实现等与业务需求有关的系统设计，也即是说它是与系统所应对的领域（Domain）逻辑有关，很多时候，也将业务逻辑层称为领域层。例如Martin Fowler在《Patterns of Enterprise Application Architecture》一书中，将整个架构分为三个主要的层：表示层、领域层和数据源层。作为领域驱动设计的先驱Eric Evans，对业务逻辑层作了更细致地划分，细分为应用层与领域层，通过分层进一步将应用逻辑与领域逻辑的解决方案分离。

业务逻辑层在体系架构中的位置很关键，它处于数据访问层与表示层中间，起到了数据交换中承上启下的作用。由于层是一种弱耦合结构，层与层之间的依赖是向下的，底层对于上层而言是“无知”的，改变上层的设计对于其调用的底层而言没有任何影响。如果在分层设计时，遵循了面向接口设计的思想，那么这种向下的依赖也应该是一种弱依赖关系。因而在不改变接口定义的前提下，理想的分层式架构，应该是一个支持可抽取、可替换的“抽屉”式架构。正因为如此，业务逻辑层的设计对于一个支持可扩展的架构尤为关键，因为它扮演了两个不同的角色。对于数据访问层而言，它是调用者；对于表示层而言，它却是被调用者。依赖与被依赖的关系都纠结在业务逻辑层上，如何实现依赖关系的解耦，则是除了实现业务逻辑之外留给设计师的任务。

**数据访问层**

数据访问层：有时候也称为是持久层，其功能主要是负责数据库的访问，可以访问数据库系统、二进制文件、文本文档或是XML文档。

简单的说法就是实现对数据表的Select，Insert，Update，Delete的操作。如果要加入ORM的元素，那么就会包括对象和数据表之间的mapping，以及对象实体的持久化。

# **3.1系统模块设计**

**账号信息管理模块：**1.管理员可以查询用户信息、密码管理、设置用户权限。2.教师密码管理。3.学生密码管理。

**教师信息管理模块：**1.管理员可以增加、删除、查询和修改教师信息以及添加教师职务。2.学生、教师可以查询教师信息。

**班级信息管理模块：**1.管理员可以增加、删除、修改、查询班级信息。2.教师查询班级信息。

**学生信息管理模块：**1. 管理员可以对学生进行添加（包过基本信息）、删除、查看、修改信息。 2. 教师、学生可以查询学生信息

**成绩信息管理模块：**1.管理员可以录入、删除、修改、查询成绩。2.教师查询成绩。3.学生查询成绩。

**课程信息管理模块：**1. 管理员可以对选修课程进行添加、删除、统计每门课程选修的人数以及查找单个学生所选的课程。2. 教师、学生可以查看课程信息。

# **3.2 系统功能结构图**

根据学生信息管理系统的实际需求，可以将学生信息管理系统划分为账号管理、教师信息管理、学生信息管理、班级管理、成绩管理、课程管理6个部分各个部分的具体功能的系统功能结构图如图3-1所示。



图3-1系统功能结构图

# **3.3系统功能模块结构E-R图设计**

1.账户管理E-R图

账户管理

用户名

密码

权限

用户id

删除标识

图3-2帐户管理E-R图

2.教师管理E-R图

教师管理

电话号码

住址

备注

删除标识

教师编号

教师姓名

性别

教师id

出生日期

学历

职称

图3-3教师管理E-R图

3.学生管理E-R图

学生管理

电话号码

住址

备注

删除标识

学生编号

学生姓名

班级id

学生id

性别

出生日期

入学时间

图3-4学生管理E-R图

4.班级管理E-R图

班级管理

班级编号

班级名称

备注

删除标识

班级id

图3-5班级管理E-R图

5.成绩管理E-R图

成绩管理

学生id

课程id

成绩

成绩id

备注

图3-6成绩管理E-R图

6.课程管理E-R图

课程管理

课程id

课程名称

学分

删除标识

课程编号

图3-7课程管理E-R图

# **3.4数据库逻辑设计**

本系统使用的数据库管理系统为SQL Server 2008。数据库中的表有账户管理表、教师信息表、学生信息表、班级信息表、成绩信息、课程信息表、班级课程表、班级教师表、教师课程表等。

1. 账户管理表User，如表3-1

表3-1 User

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 中文名称 | 类型 | 说明 |
| UserId | 用户id | int |  |
| UserName | 用户名 | string |  |
| PassWord | 密码 | string |  |
| Rights | 权限 | int |  |
| Type | 删除标识 | int | 为0时表示该生被删除 |

（2）教师信息表Teacher，如表3-2

表3-2 Teacher

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 中文名称 | 类型 | 说明 |
| TeaId | 教师Id | int |  |
| TeaNo | 教师编号 | string |  |
| TeaName | 教师姓名 | string |  |
| Sex | 性别 | string |  |
| BirthDate | 出生日期 | DateTime |  |
| Degree | 学历 | string |  |
| Rank | 职称 | String |  |
| TeaTel | 电话号码 | string |  |
| TeaAddress | 住址 | string |  |
| Remark | 备注 | string |  |
| Type | 删除标识 | int | 当标识为0时表示该教师已被删除 |

1. 学生信息表Student，如表3-3

表3-3 Student

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 中文名称 | 类型 | 说明 |
| StuId | 学生Id | int |  |
| StuNo | 学生编号 | string |  |
| StuName | 学生姓名 | String |  |
| ClassId | 班级Id | int |  |
| Sex | 性别 | String |  |
| BirthDate | 出生日期 | DateTime |  |
| EntranceTime | 入学时间 | datatime |  |
| StuTel | 电话号码 | string |  |
| StuAddress | 住址 | string |  |
| Remark | 备注 | string |  |
| Type | 删除标识 | int | 为0时表示该生被删除 |

1. 班级信息表Class，如表3-4

表3-4 Class

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 中文名称 | 类型 | 说明 |
| ClassId | 班级Id | int |  |
| ClassNo | 班级编号 | string |  |
| ClassName | 班级名称 | string |  |
| Remark | 备注 | String |  |
| Type | 删除标识 | int | 当标识为0时表示该班级已被删除 |

1. 成绩信息表Score，如表3-5

表3-5 Score

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 是否为空（Y/N） | 关系(主键或外键) | 约束和默认值 |
| ScoreId | 成绩Id | int | 否 | 主键 | 自增 |
| StuId | 学生Id | Int | 否 | 外键 |  |
| CourseId | 课程Id | int | 否 | 外键 |  |
| Score | 成绩 | float | 是 |  |  |
| Remark | 备注 | string | 是 |  |  |

1. 课程信息表Course，如表3-6

表3-6 Course

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 是否为空（Y/N） | 关系(主键或外键)  注：如果是外键，则写上外键表名 | 约束和默认值 |
| CourseId | 课程id | int | 否 | 主键 | 自增 |
| CourseNo | 课程编号 | varchar | 否 |  |  |
| CourseName | 课程名称 | varchar | 否 |  |  |
| Points | 学分 | dec | 否 |  |  |
| type | 删除标识 | int | 否 |  |  |

1. 班级课程表Class\_course，如表3-7

表3-7 Class\_course

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 是否为空 | 关系(主键或外键)  注：如果是外键，则写上外键表名 | 约束和默认值 |
| Cl\_CoId | ID | Int | 否 | 主键 | 自增 |
| ClassId | 班级编号 | nvarchar | 否 | 外键，Class |  |
| CourseId | 课程编号 | nvarchar | 否 | 外键，Course |  |
| type | 删除标识 | Int | 否 |  |  |

1. 班级教师表Class\_Teacher，如表3-8

表3-8 Class\_Teacher

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 是否为空 | 关系(主键或外键)  注：如果是外键，则写上外键表名 | 约束和默认值 |
| Tea\_ClId | ID | int | 否 | 主键 | 自增 |
| ClassId | 班级编号 | nvarchar | 否 | 外键，Class |  |
| TeaId | 教师编号 | nvarchar | 否 | 外键，Teacher |  |
| type | 删除标识 | Int | 否 |  |  |

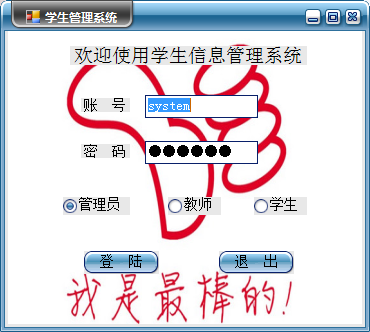
1. 教师课程表Teacher\_course，如表3-9

表3-9 Teacher\_course

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 中文名 | 数据类型 | 是否为空 | 关系(主键或外键)  注：如果是外键，则写上外键表名 | 约束和默认值 |
| Tea\_CoId | ID | int | 否 | 主键 | 自增 |
| CourseId | 课程编号 | nvarchar | 否 | 外键，Course |  |
| TeaId | 教师编号 | nvarchar | 否 | 外键，Teacher |  |
| type | 删除标识 | Int | 否 |  |  |

1. **系统的实现**

**4.1 用户登录界面**



登录界面图 4.1

本界面实现管理员、教师、学生等登陆系统功能，系统默认管理员账户system，密码为521521，输入账户和密码点击登录即可进入系统，设计代码基于C#源码设计。

用户登录主要代码:

public partial class LoginInterfaceForm : Form

{

//定义新皮肤

Sunisoft.IrisSkin.SkinEngine skin = new Sunisoft.IrisSkin.SkinEngine();

public LoginInterfaceForm()

{

InitializeComponent();

//选择皮肤文件

skin.SkinFile = System.Environment.CurrentDirectory + "\\skin\\" + "Wave.ssk";

//激活当前皮肤

skin.Active = true;

}

private void LoginInterfaceForm\_Load(object sender, EventArgs e)

{

}

private void btnLogin\_Click(object sender, EventArgs e)

{

string strErr = "";

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Login.Login objLogin = new Manage.Login.Login(strConnection);

if (rabAdmin.Checked == true)

{

Model.Users.Users objUsers = new Model.Users.Users();

objUsers.UserNo= txtUserNo.Text.Replace("'", "''");

objUsers.UserNo = objUsers.UserNo.Trim();

objUsers.PassWord = txtPassWord.Text.Replace("'", "''");

objUsers.PassWord = objUsers.PassWord.Trim();

//以管理员身份登录

//判断输入的用户名和密码是否正确,正确进入 MainForm界面,否则提示您输入的账号或密码错误，请重新输入

if (txtUserNo.Text == "system" && txtPassWord.Text == "521521")

{

objUsers.Rights = 1;

MainForm objEnter = new MainForm(objUsers.UserNo, objUsers.PassWord, objUsers.Rights);

this.Visible = false;

objEnter.Show();

}

else

{

MessageBox.Show("您输入的账号或密码错误，请重新输入！");

txtUserNo.Text = "";

txtPassWord.Text = "";

return;

}

}

////以教师身份登录

else if (rabTea.Checked == true)

{

Model.Users.Users objUsers = new Model.Users.Users();

objUsers.UserNo = txtUserNo.Text.Replace("'", "''");

objUsers.UserNo = objUsers.UserNo.Trim();

objUsers.UserNo = objUsers.UserNo.Replace("%", @"\%");

objUsers.PassWord = txtPassWord.Text.Replace("'", "''");

objUsers.PassWord = objUsers.PassWord.Trim();

objUsers.PassWord = objUsers.PassWord.Replace("%", @"\%");

objUsers.Rights = 2;

int iRent = objLogin.JudgeExist(objUsers.UserNo, objUsers.PassWord,objUsers.Rights, out strErr);

if (iRent == 1)

{

MainForm objEnter = new MainForm(objUsers.UserNo, objUsers.PassWord, objUsers.Rights);

objEnter.Show();

this.Visible = false;

}

else

{

MessageBox.Show("您输入的账号或密码错误，请重新输入！");

txtUserNo.Text = "";

txtPassWord.Text = "";

return;

}

}

else

{

// //以学生方式登录

Model.Users.Users objUsers = new Model.Users.Users();

objUsers.UserNo = txtUserNo.Text.Replace("'", "''");

objUsers.UserNo = objUsers.UserNo.Trim();

objUsers.PassWord = txtPassWord.Text.Replace("'", "''");

objUsers.PassWord = objUsers.PassWord.Trim();

objUsers.Rights = 3;

int iRent = objLogin.JudgeExist(objUsers.UserNo, objUsers.PassWord, objUsers.Rights, out strErr);

if (iRent == 1)

{

MainForm objEnter = new MainForm(objUsers.UserNo, objUsers.PassWord, objUsers.Rights);

objEnter.Show();

this.Visible = false;

}

else

{

MessageBox.Show("您输入的账号或密码错误，请重新输入！");

txtUserNo.Text = "";

txtPassWord.Text = "";

return;

}

}

}

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

**4.3 系统主界面**



本界面实现对六大模块(账户管理、教师管理、班级管理、学生管理、成绩管理、课程管理)以及用户的相关信息进行展示。

主界面主要代码：

private void MainForm\_Load(object sender, EventArgs e)

{

label2.Text = "计算机名:" + Environment.MachineName;

label4.Text = "当前画质:" + SystemInformation.VirtualScreen.Width + "\*" + SystemInformation.VirtualScreen.Height;

lblUserNo.Text = strUserNo;

if (iRight == 1)

{

lblUserMember.Text = "管理员";

}

else if (iRight == 2)

{

lblUserMember.Text = "教师";

}

else

{

lblUserMember.Text = "学生";

menuClass.Enabled = false;

pictureBox3.Enabled = false;

}

}

private void menuClass\_Click(object sender, EventArgs e)

{

Class.ClassManageForm objClass = new StudentManage.Class.ClassManageForm(iRight, strUserNo);

objClass.ShowDialog();

}

**4.4 账户信息管理界面**



账户信息管理图4.4

该界面可实现管理员查询用户（教师和学生），修改信息和删除用户

账户管理主代码：

1.查询用户：

//查询用户

private void btnSearch\_Click(object sender, EventArgs e)

{

string strUserNo = txtUserNo.Text.Replace(@"\", @"\\");

strUserNo = strUserNo.Replace("%", @"\%");

strUserNo = strUserNo.Replace("\_", @"\\_");

strUserNo = strUserNo.Replace("'", @"\''");

strUserNo = strUserNo.Trim();

string strUserName = txtUserName.Text.Replace(@"\", @"\\");

strUserName = strUserName.Replace("%", @"\%");

strUserName = strUserName.Replace("\_", @"\\_");

strUserName = strUserName.Replace("'", @"\''");

strUserName = strUserName.Trim();

//字符串连接

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Login.Login objlogin = new Manage.Login.Login(strConnection);

string strErr = "";

//根据用户账号或是用户名称来查询

if (strUserNo == "" && strUserName == "")

{

BindDataGridView();

}

else

{

DataTable objDataTable = new DataTable();

int iRent = objlogin.SearchUserBy(strUserNo, strUserName, out objDataTable, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

}

else

{

if (objDataTable.Rows.Count == 0)

{

MessageBox.Show("没有您要查询的记录");

}

else

{

dataGridView.DataSource = objDataTable;

}

}

txtUserNo.Text = "";

txtUserName.Text = "";

}

}

2.用户修改：

//修改用户名称 用户密码

private void btnSubmit\_Click(object sender, EventArgs e)

{

string strErr = "";

string strUserNo;

Model.Users.Users objUsers = new Model.Users.Users();

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Login.Login objLogin = new Manage.Login.Login(strConnection);

strUserNo = objUsers.UserNo = txtUserNo.Text.Replace("'", "''");

objUsers.UserNo = objUsers.UserNo.Trim();

objUsers.UserName = txtUserName.Text.Replace("'", "''");

objUsers.UserName= objUsers.UserName.Trim();

objUsers.PassWord = txtPWD.Text.Replace("'", "''");

objUsers.PassWord= objUsers.PassWord.Trim();

objUsers.Rights = int.Parse(txtRights.Text.ToString());

if (objUsers.UserNo == "" || objUsers.UserName == "")

{

MessageBox.Show("请将信息填写完整！");

}

else

{

if (userno == objUsers.UserNo)

{

int iRent = objLogin.ModifyUser(iRights,strUserNo, userno, objUsers, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("修改成功！");

this.Visible = false;

}

}

else

{

int iExist = objLogin.JudgeUserNo(objUsers.UserNo);

if (iExist == 0 )

{

int iRent = objLogin.ModifyUser(iRights,strUserNo,userno,objUsers, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("修改成功！");

this.Visible = false;

}

}

else

{

MessageBox.Show("此帐号已存在，修改失败");

txtUserNo.Text = userno;

}

3.用户删除：

//删除用户

private void menuDelUser\_Click(object sender, EventArgs e)

{

if (dataGridView.SelectedRows.Count == 0)

{

MessageBox.Show("请先选中要删除的行");

}

else

{

string strErr = "";

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Login.Login objlogin = new Manage.Login.Login(strConnection);

for (int i = 0; i < dataGridView.SelectedRows.Count; i++)

{

int iRights = int.Parse(dataGridView.SelectedRows[i].Cells["权限"].Value.ToString());

string strUserNumble = dataGridView.SelectedRows[i].Cells["帐号"].Value.ToString();

int iRent = objlogin.DelUser(iRights, strUserNumble, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

}

BindDataGridView();

}

}

**4.5 教师信息管理**



教师信息管理图4.5

管理员登录系统后点击添加教师后可对教师信息管理界面增加教师信息，不仅提高工作人员的工作效率，而且增加信息录入准确性。主要代码为：

private void btnSave\_Click(object sender, EventArgs e)

{

string strErr = "";

Model.Teacher.Teacher objTeacher = new Model.Teacher.Teacher();

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Teacher.Teacher objTea = new Manage.Teacher.Teacher(strConnection);

objTeacher.TeaNo = txtTeaNo.Text.Replace("'", "''");

objTeacher.TeaNo = objTeacher.TeaNo.Trim();

objTeacher.TeaName = txtTeaName.Text.Replace("'", "''");

objTeacher.TeaName = objTeacher.TeaName.Trim();

if (objTeacher.TeaNo == "" || objTeacher.TeaName == "")

{

MessageBox.Show("请将信息填写完整！");

}

else

{

if (rabMan.Checked == true)

{

objTeacher.Sex = "男";

}

else

{

objTeacher.Sex = "女";

}

objTeacher.Age = int.Parse(nudAge.Value.ToString());

objTeacher.Rank = cmbRank.SelectedItem.ToString();

objTeacher.Degree = cmbDegree.SelectedItem.ToString();

objTeacher.TeaAddress = txtAddress.Text.Replace("'", "''");

objTeacher.TeaTel = txtTel.Text.Replace("'", "''");

objTeacher.Remark = txtRemark.Text.Replace("'", "''");

int iTeaNo = objTea.JudgeTeaNo(objTeacher.TeaNo); //判断教师编号间是否重复

int iR = objTea.JudgeTeaNoLikeStuNo(objTeacher.TeaNo); //判断教师编号与学生编号是否重复

if (iTeaNo == 0)

{

MessageBox.Show("此编号的教师已经存在，添加失败");

}

else if (iR == 0)

{

MessageBox.Show("存在一个此编号的学生，教师编号不能和学生编号相同");

}

else

{

int iRent = objTea.AddTea(objTeacher, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("添加成功");

this.Visible = false;

}

**4.6 班级信息管理界面**



班级信息管理图4.6

管理员登录系统后点击添加班级信息后可增加班级信息，不仅提高工作人员的工作效率，而且方便管理和维护。主要代码为：

if (objClass.ClassName == "" || objClass.ClassNo == "")

{

MessageBox.Show("请将信息填写完整！");

}

else

{

int i = objclass.JudgeClassNo(objClass);

int j = objclass.JudgeClassName(objClass);

if (i == 0 && j == 0)

{

int iRent = objclass.AddClass(objClass, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("添加成功");

this.Visible = false;

}

}

else if (i == 1 && j == 0)

{

MessageBox.Show("当前班级已经存在，新建失败");

}

else if (i == 0 && j == 1)

{

MessageBox.Show("当前班级名称已经存在，新建失败");

}

else

{

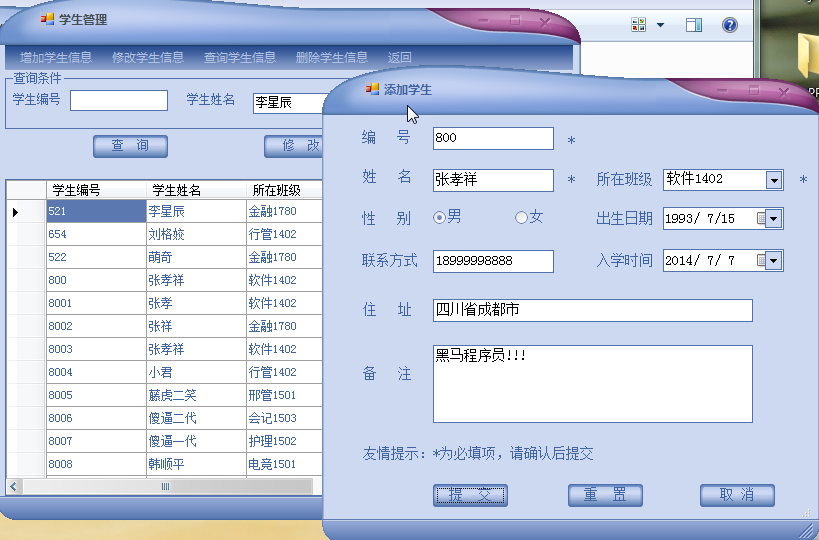
MessageBox.Show("班级编号和班级名称都已经存在，新建失败");

}

}

}

**4.7 学生信息管理界面**



学生信息管理图4.7

管理员登录系统后点击添加学生即可进行学生信息添加，既方便又快捷和高效率。主要代码为：

private void btnSave\_Click(object sender, EventArgs e)

{

string strErr = "";

Model.Student.Student objStudent = new Model.Student.Student();

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Student.Student objStu = new Manage.Student.Student(strConnection);

int Ar = objStu.JudgeClassExist();

if (Ar == 0)

{

MessageBox.Show("不存在班级,将不能添加学生");

}

else

{

objStudent.StuNo = txtStuNo.Text.Replace("'", "''");

objStudent.StuNo = objStudent.StuNo.Trim();

objStudent.StuName = txtStuName.Text.Replace("'", "''");

objStudent.StuName = objStudent.StuName.Trim();

if (objStudent.StuNo == "" || objStudent.StuName == "")

{

MessageBox.Show("请将信息填写完整！");

}

else

{

if (rabWoman.Checked == true)

{

objStudent.Sex = "女";

}

else

{

objStudent.Sex = "男";

}

objStudent.BirthDate = dtpBirthDate.Value;

objStudent.EntranceTime = dtpEntranceTime.Value;

//objStudent.StuTel = txtStuTel.Text.ToString();

//objStudent.StuTel = txtStuTel.Text.Replace("'", "''");

if (this.txtStuTel.Text != "")

{

string patten = @"(13\d{9}(;13\d{9})\*)|(15\d{9}(;15\d{9})\*)|(\(\d{3,4}\)|\d{3,4}-)?\d{7,8}(;(\(\d{3,4}\)|\d{3,4}-)?\d{7,8})\*";

Regex r = new Regex(patten);

Match m = r.Match(txtStuTel.Text);

if (!m.Success)

{

MessageBox.Show("请输入正确的电话号码");

this.txtStuTel.Text = "";

this.txtStuTel.Focus();

return;

}

else

{

objStudent.StuTel = txtStuTel.Text.ToString();

}

}

objStudent.StuAddress = txtAddress.Text.Replace("'", "''");

objStudent.Remark = txtRemark.Text.Replace("'", "''");

objStudent.ClassId = int.Parse(cmbClassId.SelectedValue.ToString());

//int Ar= objStu. JudgeClassExist();

int iExist = objStu.JudgeStuNo(objStudent.StuNo);

int R = objStu.JudgeStuNoLikeTeaNo(objStudent.StuNo);

if (iExist == 1)

{

MessageBox.Show("此编号的学生已经存在，请重新为此学生编号");

}

else if (R == 1)

{

MessageBox.Show("存在一个此编号的教师，学生编号不能和教师编号相同");

}

else

{

int iRent = objStu.AddStu(objStudent, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("添加学生信息成功,同时系统为此学生创建账号成功");

this.Visible = false;

}

**4.8 成绩信息管理界面**



成绩信息管理图4.7

其界面为添加成绩信息，管理员可以对学生成绩信息添加操作，不仅节省时间，而且节约资源。主要代码为：

//添加成绩信息

private void btnSave\_Click(object sender, EventArgs e)

{

string strErr = "";

Model.Class.Class objClass1 = new Model.Class.Class();

Model.Student.Student objStudent1 = new Model.Student.Student();

Model.Course.Course objCourse1 = new Model.Course.Course();

objClass1.ClassNo = cmbClassName.SelectedValue .ToString();

objStudent1.StuId = int.Parse(cmbStuNo.SelectedValue .ToString());

objCourse1.CourseId = int.Parse(cmbCourseName.SelectedValue .ToString());

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Score.Score objSco = new Manage.Score.Score(strConnection);

int i = objSco.Judge(objStudent1, objCourse1);　　　　// 判断是否存在所选的成绩，如果为0，表示不存在记录，可以增加。为1表示存在记录，不能重复插入成绩。

if (i == 0)

{

Model.Score.Score objScore = new Model.Score.Score();

Model.Student.Student objStudent = new Model.Student.Student();

Model.Course.Course objCourse = new Model.Course.Course();

objScore.CourseId = int.Parse(cmbCourseName.SelectedValue.ToString());

objScore.StuId = int.Parse(cmbStuNo.SelectedValue.ToString());

objScore.score = float.Parse(nudScore.Value.ToString());

objScore.StuName = txtStuName.Text;

int iRent = objSco.AddScore(objScore, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("添加成功");

this.Visible = false;

BindClassId();

BindStuId();

BindCourseId();

}

}

else

{

MessageBox.Show("不能重复插入分数");

}

}

**4.8 课程信息管理界面**



课程信息管理图4.8

该界面主要实现添加课程信息，管理员根据学生们的学习情况可以对学生们学习的课程进行必要添加，既便于储存，也便于操作和管理。主要代码为：

private void btnSubmit\_Click(object sender, EventArgs e)

{

string strConnection = System.Configuration.ConfigurationManager.ConnectionStrings["StudentManager"].ConnectionString;

Manage.Course.Course objManageCourse = new Manage.Course.Course(strConnection);

string strErr = "";

Model.Course.Course objCourse = new Model.Course.Course();

objCourse.CourseNo = txtCourseNo.Text.Replace("'", "''");

objCourse.CourseNo = objCourse.CourseNo.Trim();

objCourse.CourseName = txtCourseName.Text.Replace("'", "''");

objCourse.CourseName = objCourse.CourseName.Trim();

//objCourse.Points = float.Parse(this.txtPoints.Text.ToString().Replace("'", "''"));

//objCourse.Points = objCourse.Points.Trim();

objCourse.Remark = txtRemark.Text.Replace("'", "''");

objCourse.Remark = objCourse.Remark.Trim();

if (this.txtPoints.Text != "")

{

try

{

objCourse.Points = float.Parse(this.txtPoints.Text.ToString().Replace("'", "''"));

if (this.txtPoints.Text.IndexOf('.') == 0)

{

MessageBox.Show("小数点之前没有数字");

return;

}

if (this.txtPoints.Text.Length - this.txtPoints.Text.IndexOf('.') - 1 == 0)

{

MessageBox.Show("小数点后没有数字");

return;

}

else

{

objCourse.Points = float.Parse(this.txtPoints.Text.ToString().Replace("'", "''"));

}

if (objCourse.Points < 0)

{

MessageBox.Show("请输入正数");

return;

}

}

catch

{

MessageBox.Show("请输入正确的学分");

this.txtPoints.Text = "";

this.txtPoints.Focus();

return;

}

}

//objCourse.CourseNo = txtCourseNo.Text.Replace("'", "''");

//objCourse.CourseNo = objCourse.CourseNo.Trim();

//objCourse.CourseName = txtCourseName.Text.Replace("'", "''");

//objCourse.CourseName = objCourse.CourseName.Trim();

//objCourse.Points = float.Parse(this.txtPoints.Text.ToString().Replace("'", "''"));

//objCourse.Points = objCourse.Points.Trim();

//objCourse.Remark = txtRemark.Text.Replace("'", "''");

//objCourse.Remark = objCourse.Remark.Trim();

if (txtCourseName.Text == "" || txtCourseNo.Text=="")

{

MessageBox.Show("请将信息填写完整！");

}

else

{

int iCourseNo1 = objManageCourse.JudgeCourseNo(objCourse.CourseNo, out strErr);

int iCourseNo2 = objManageCourse.JudgeCourseName(objCourse.CourseName, out strErr);

if (iCourseNo1 == 0 && iCourseNo2== 0)

{

int iRent = objManageCourse.AddCourse(objCourse, out strErr);

if (iRent == 0)

{

MessageBox.Show(strErr);

return;

}

else

{

MessageBox.Show("添加成功");

this.Visible = false;

//BindDataGridView();

}

}

else if (iCourseNo1 == 1 && iCourseNo2== 0)

{

MessageBox.Show("此课程编号已经存在，添加失败");

}

else if (iCourseNo1 == 0 && iCourseNo2 ==1)

{

MessageBox.Show("此课程名称已经存在，添加失败");

}

else

{

MessageBox.Show("当前课程编号和课程名称都已存在，添加失败");

}

}

}

**4.8 数据库连接配置**

为实现以上界面对数据库的连接访问，需要App.config的配置：

<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<connectionStrings>

<add name="StudentManager" connectionString="Data Source=.;database=stumanage;User ID=tian;Password=123456"/>

</connectionStrings>

</configuration>

**结 论**

学生信息管理系统为其他系统(如学校图书管理系统、学校档案管理系统、教学管理系统、总务后勤管理系统等)提供学生的基本信息，同时它也需要教学管理系统提供课程设这数据等。这些系统在具体应用中构成一个大系统，相互调用对方的数据。本设计从功能上划分成了以下几大模块：账户管理模块，教师管理模块，班级管理模块，学生信息管理模块，成绩管理模块，课程信息管理模块六大模块。本系统功能适用于大中专及高校用户使用，具有很大的扩展空间及发展空间。

由以上分析，我们可以充分认识到设计并开发一个学生信息管理系统的必要性和迫切性，本次设计和开发的学生信息管理系统应符合学校学生信息管理的规定，满足对学校学生信息管理的需要，并达到操作过程中的直观，方便，实用，安全等要求。系统采用模块化程序设计的方法，既便于系统功能的组合和修改，又便于未参与系统开发的技术维护人员补充和维护。同时通过教育网络或互联网可以把整个学校的学生信息汇聚起来，为学校教育主管部门提供全面及时的学生信息，方便管理，同时亦可通过校园网扩展各类信息化应用，共享学生信息。

在几个星期的辛苦努力下，学生信息管理系统的基本功能已经实现。虽然时间很短暂，但却是对几年学习中理论知识和实践相结合的一次综合检验。通过这次毕业设计，收到了比以往理论课程还要大的收益，虽然由于时间仓促及本人的能力有限，系统还有很多不尽人意的地方：比如说界面不够美观；有些功能还不够完善和强大；代码的重用性不够高；一些细节的问题还没有解决。这些都需要平时经验的积累和对技术的熟练掌握，希望在以后的工作学习中能有进一步的提高。

**致谢语**

在系统开发的几个星期里,我受到了易晓鹏老师的细心指导,老师严谨的指导态度与深厚的理论知识都让我受益非浅,从他身上我学到了很多的东西,无论是理论还是实践都使我的知识有很大的提高。借此我特提出感谢。

同时我还感谢母校的老师,谢谢他们为本系统提供的帮助。让我能够顺利地完成本系统的开发。也感谢那些对我们项目组的小伙伴们,没有他们就不会有现在的系统。

**参考文献**

[1] 陈萱华．如何利用ASP.NET技术访问数据库．计算机与现代化,2003年

[2] 周爱民.JavaScript语言精髓与编程实践.电子工业出版社,2012年

[3] 朱涛，李云云．基于ASP.NET技术的WEB数据库分页显示,2005年

[4] 郑阿奇.SQL Server 2008教程.电子工业出版社,2011年

[5] 余金山．ASP.NET 2.0+SQL Server 2008企业项目开发与实战．北京：电子工业出版社,2008

[6] 朱玉超，鞠艳，王代勇．ASP.NET 项目开发教程[M] ．北京：电子工业出版社,2008

[7] 张海藩.软件工程[M].人民邮电出版社,2005.9

[8] 刘翊等.ERP原理与应用[M]. 北京:清华大学出版社, 2011年3月

[9] 杨晓光.面向对象程序设计(C#版)[M].北京:清华大学出版社,2010.8

[10] 窦万峰等.软件工程方法与实践. 北京：机械工业出版,2009年5月

[11].刘丽《数据库原理与应用—SQL Server 2008》.中国水利水电出版社

[12].王彤宇《管理信息系统》.中国水利水电出版社

[13].李挥剑《C# WinForm 实践开发教程》.中国水利水电出版社

[14].杨克玉、阮进军《C#程序设计》.中国水利水电出版社

[15].刘先省《Visual C#程序设计教程》.北京机械工业出版