# 目 录

[第1章 概 述 1](#_Toc140482942)

[1.1 项目的目的和意义 1](#_Toc140482943)

[1.2 技术开发基础 1](#_Toc140482944)

[1.3系统基本功能 2](#_Toc140482945)

[第2章 系统设计 3](#_Toc140482946)

[2.1 系统功能模块划分 3](#_Toc140482947)

[2.2 数据库设计 5](#_Toc140482948)

[第3章 系统的详细设计及实现 9](#_Toc140482949)

[3.1 公共类 9](#_Toc140482950)

[3.2 用户注册与修改密码 11](#_Toc140482951)

[3.3 系统登录与主窗体 15](#_Toc140482952)

[3.4 用户信息管理 17](#_Toc140482953)

[3.5 成绩信息管理 23](#_Toc140482954)

[3.6 论文发表情况管理 26](#_Toc140482955)

[3.7 实习情况管理 28](#_Toc140482956)

[3.8 专业信息管理 29](#_Toc140482957)

[3.9 导师信息管理 31](#_Toc140482958)

[3.10 研究生信息管理 32](#_Toc140482959)

[3.11 课程信息管理 34](#_Toc140482960)

[3.12 党员信息管理 36](#_Toc140482961)

[第4章 系统测试 38](#_Toc140482962)

[4.1 系统测试的原则 38](#_Toc140482963)

[4.2 测试结果 38](#_Toc140482964)

[结 论 41](#_Toc140482965)

[参考文献 42](#_Toc140482966)

[附录1团队开发会议记录1 43](#_Toc140482967)

[附录2团队开发会议记录2 44](#_Toc140482968)

## 1.1 项目的目的和意义

随着全国各个高校对研究生招生规模的不断扩大，研究生的数量急剧增加，这个过程必然伴随着巨大的信息量。为了加强各高校对研究生信息的管理，运用科学的管理方法非常重要。

现代早已成为一个信息化的社会，研究生信息管理系统也是一个高等教育单位不可或缺的部分，它的内容对于学校的决策者和管理者来说都至关重要，所以研究生管理系统应该能够为用户提供丰富的信息和快捷、高效的查询手段，极大地方便信息的查找、更新和维护。采用研究生信息管理系统不仅可以节省人力、物力，而且可以增强学校资料的安全性，提高学校的管理能力。目的是为学校办公管理提供一个快速、简单规范的管理平台，使学院在学生信息管理方面更加规范化、便捷化。

开发一套功能完善的研究生信息管理系统有利于满足现代化教育的教学管理需要，为高等院校的现代教育应用提供一个平台，提高研究生教育管理质量，提高研究生管理信息的整体效率。应用该系统，可以极大地提高研究生、教师、管理人员等使用者的工作效率，对于学生信息的查找、更新和维护提供了极大地便利。

## 1.2 技术开发基础

### 1.2.1 面向对象技术

面向对象方法是基于客观世界的对象模型化的软件开发方法。它的一条基本原则是计算机程序由单个能够起到子程序作用的单元或对象组合而成。面向对象技术达到了软件工程的三个主要目标：重用性、灵活性和扩展性。OOP=对象+类+继承+多态+消息，其中核心概念是类和对象。在面向对象的设计方法中，对象和传递消息分别是表现事物及事物间相互联系的概念。类和继承是适应人们一般思维方式的描述范式。方法是允许作用于该类对象上的各种操作。

面向对象程序设计方法是尽可能模拟人类的思维方式，使得软件的开发方法与过程尽可能接近人类认识世界、解决现实问题的方法和过程，也即使得描述问题的问题空间与问题的解决方案空间在结构上尽可能一致，把客观世界中的实体抽象为问题域中的对象。

### 1.2.2 C#开发语言

C#是微软公司发布的一种面向对象的、运行于.NET Framework和.NET Core(完全开源，跨平台)之上的高级程序设计语言。C#是一种安全的、稳定的、面向对象的、面向组件的编程语言。它不仅去掉了 C++ 和 Java 语言中的一些复杂特性，还提供了可视化工具，能够高效地编写程序。C#提供了语言构造来直接支持这些概念，让C#成为一种非常自然的语言，可用于创建和使用软件组件。自诞生之日起，C# 就添加了支持新工作负载和新兴软件设计实践的功能。C#语言具备简单、安全、支持跨平台开发多种类型的程序等特点。与其他[面向对象语言](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E8%AF%AD%E8%A8%80/215742?fromModule=lemma_inlink)一样，C# 语言也具有面向对象语言的基本特征，即封装、继承、多态。

### 1.2.3 SQL Server数据库管理系统

SQL Server 是Microsoft 公司推出的关系型[数据库管理系统](https://baike.so.com/doc/2035924-2154168.html)。具有使用方便[可伸缩性](https://baike.baidu.com/item/%E5%8F%AF%E4%BC%B8%E7%BC%A9%E6%80%A7/1414004?fromModule=lemma_inlink)好与相关[软件集成](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E9%9B%86%E6%88%90/895406?fromModule=lemma_inlink)程度高等优点，可跨越从运行Microsoft Windows 98的膝上型电脑到运行[Microsoft Windows](https://baike.baidu.com/item/Microsoft%20Windows/3304184?fromModule=lemma_inlink) 2012 的大型[多处理器](https://baike.baidu.com/item/%E5%A4%9A%E5%A4%84%E7%90%86%E5%99%A8/53566627?fromModule=lemma_inlink)的服务器等多种平台使用。SQL (Structured Query Language) 是具有数据操纵和数据定义等多种功能的数据库语言，这种语言具有交互性特点，能为用户提供极大的便利，数据库管理系统应充分利用SQL语言提高计算机应用系统的工作质量与效率。SQL语言不仅能独立应用于终端，还可以作为子语言为其他程序设计提供有效助力。Microsoft按照[客户/服务器](https://baike.baidu.com/item/%E5%AE%A2%E6%88%B7%2F%E6%9C%8D%E5%8A%A1%E5%99%A8/3351471?fromModule=lemma_inlink)[体系结构](https://baike.baidu.com/item/%E4%BD%93%E7%B3%BB%E7%BB%93%E6%9E%84/8174145?fromModule=lemma_inlink)的分布进行操作。这种方法产生不必要的代价和复杂性。

### 1.2.4 Client/Server开发模式

Client/Server结构(C/S结构)是大家熟知的客户机和服务器结构。C/S结构在技能上非常成熟，它的重要特征就是交互性强、拥有安全的存取形式、网络通信数量低、响应速度快、利于处置大量数据。安全性能能够非常容易确保，C/S通常面向相比固定的用户群，程序越发注重过程，它能够对权限实行多层次校验，提供了更安全的存取形式，对信息安全的控制才能非常强。通常高度机密的信息系统选用C/S结构适宜。

## 1.3系统基本功能

在以上环境下对研究生信息管理系统进行开发与设计。主要实现系统登陆、用户管理、专业信息管理、课程信息管理、研究生信息管理、成绩信息管理、导师信息管理、研究生实习情况管理、研究生论文情况管理、研究生党员信息管理等功能。

# 第2章 系统设计

## 2.1 系统功能模块划分

研究生信息管理系统是研究生信息查询与修改流程中不可缺少的一个系统，它涉及到用户、课程、成绩、专业信息、实习情况、导师信息、论文发表情况等信息。研究生信息管理系统包括以下十二个模块：

1、登录模块

登录模块提供用户登录界面。在用户登录界面中，用户输入正确的用户名和密码后，即可进入系统主窗口（即导航页面），从而可以选择进入相应的子系统。用户还可以自主选择不同的系统主题。

2、注册模块

注册模块提供用户注册界面，当注册时，用户输入一个全新的用户名、密码以及正确的图形验证码，即可注册成功。

3、修改密码模块

在修改密码界面中，用户需提供正确的用户名和原密码，并确保原密码和新密码不一致，即可修改成功。

4、研究生信息管理模块

研究生信息管理模块主要管理研究生的基本信息，包括学号、姓名、性别、年龄、专业及所属院系，能将输入的研究生信息添加到数据库的研究生信息表中，对选中项研究生信息对应数据库中的研究生信息进行删除和修改等操作，对数据库对应研究生信息表导出为excel表格形式。

5、用户信息管理模块

用户信息管理模块主要管理用户的基本信息，包括姓名、性别、年龄及所属院系，能将输入的用户信息添加到数据库的用户信息表中，对选中项用户信息对应数据库中的用户信息进行删除和修改等操作，对数据库对应用户信息表导出为excel表格形式。

6、课程信息管理模块

课程信息管理模块主要管理课程的基本信息，包括课程编号、课程名称、授课教师、学分、学时及开设学期，能将输入的课程信息添加到数据库的课程信息表中，对选中项课程信息对应数据库中的课程信息进行删除和修改等操作，对数据库对应课程信息表导出为excel表格形式。

7、成绩信息管理模块

成绩信息管理模块主要管理成绩的基本信息，包括学生学号、姓名、课程编号、课程名称及成绩，能将输入的对应学生相应课程成绩添加到数据库的成绩信息表中，并可根据学生学号或课程的编号两种不同的方式对成绩完成查询。

8、专业信息管理模块

专业信息管理模块主要管理专业的基本信息，包括专业以及所属院系，能将输入的专业信息添加到数据库的专业信息表中，对选中项专业信息对应数据库中的专业信息进行删除和修改等操作，对数据库对应专业信息表导出为excel表格形式。

9、导师信息管理模块

导师信息管理模块主要管理导师的基本信息，包括姓名、性别、职称、研究方向、联系方式以及出生日期，能将输入的导师信息添加到数据库的导师信息表中，对选中项导师信息对应数据库中的导师信息进行删除和修改等操作，对数据库对应导师信息表导出为excel表格形式。

10、论文发表情况管理模块

论文发表情况管理模块主要管理论文发表的基本信息，包括学生学号、姓名、指导教师、论文题目以及发表时间，能将输入的论文发表情况信息添加到数据库的论文发表情况信息表中，并可根据学生姓名或指导教师两种不同方式对论文发表情况进行查询。

11、实习情况管理模块

实习情况管理模块主要管理学生实习的基本信息，包括姓名、学号、实习城市、岗位、薪资、实习起始日期、终止日期以及实习时长，能将输入的学生实习信息添加到数据库的实习情况信息表中，对选中项实习情况信息对应数据库中的实习情况信息进行删除和修改等操作，对数据库对应实习情况信息表导出为excel表格形式。

12、党员情况管理系统

党员信息管理模块主要管理学生党员的基本信息，包括姓名、学号、入党时间、民族、性别、出生日期、联系方式、专业以及所属院系，能将输入的学生党员信息添加到数据库的党员情况信息表中，对选中项党员信息对应数据库中的党员信息进行删除和修改等操作，对数据库对应党员信息表导出为excel表格形式。

研究生信息管理系统功能模块图如图2-1所示。

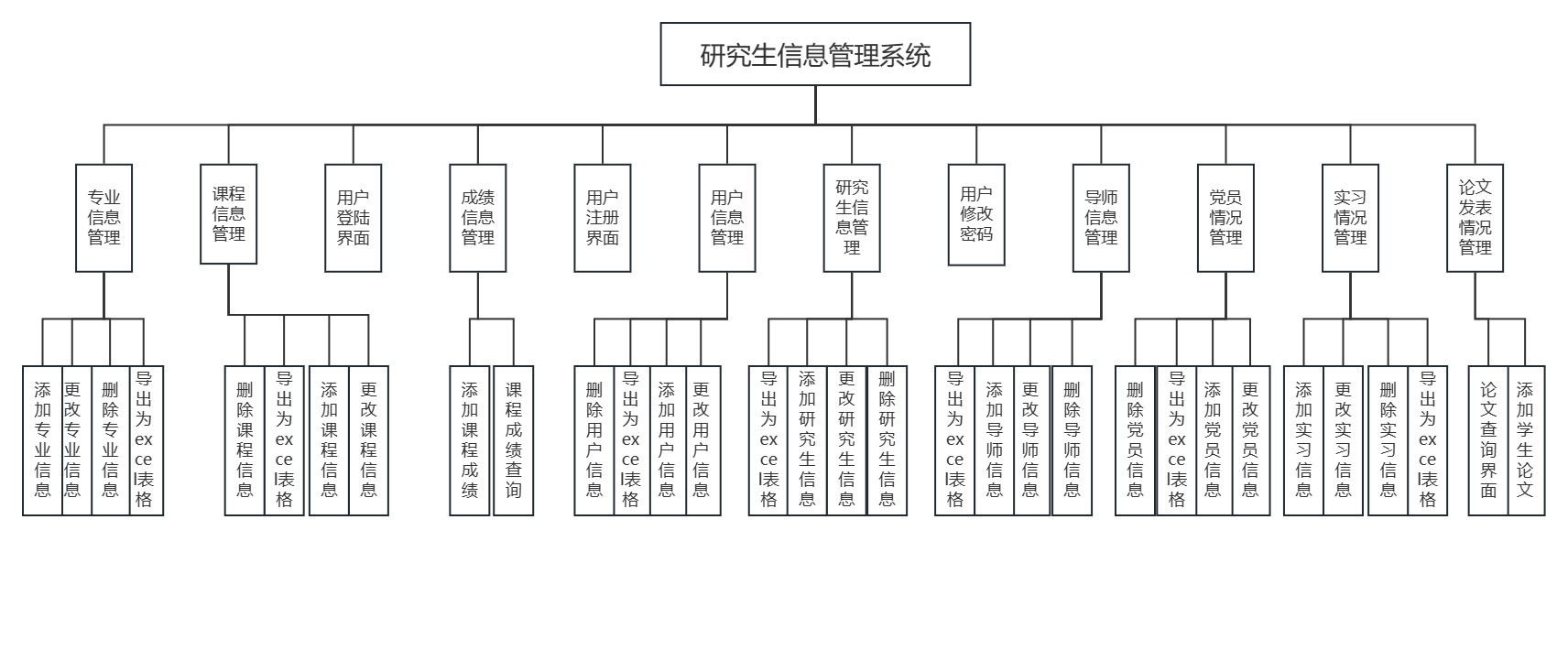


图2-1 系统功能模块图

## 2.2 数据库设计

根据前面的分析，研究生信息管理系统数据库（Postgraduate）中包括系统用户信息“tbl\_login”、用户信息“tbl\_user”、专业信息“tbl\_major”、课程信息“tbl\_Course”、成绩信息“tbl\_grade”、研究生信息“tbl\_student”、论文发表情况 “tbl\_paper”、党员信息“tbl\_party”、实习情况“tbl\_exercitation”和导师信息“tbl\_professor”十个数据表。表的结构、表字段的数据类型及相关说明如下：

1、系统用户表

系统用户表“tbl\_login”用于存放系统用户的相关数据，其结构如表2-1所示。

表2-1 系统用户表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Username | 用户名 | 字符串，长度为16 | 主键 |
| Userpassword | 用户密码 | 字符串，长度为16 | 非空 |
| Userpurview | 用户权限 | 字符串，长度为16 | 取值为“教师登录”或“学生登录” |

2、用户信息表

用户信息表“tbl\_user”用于存放用户基本信息，包括姓名、年龄、性别、所属院系等，其结构如表2-2所示。

表2-2 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Sname | 姓名 | 字符串，长度为16 | 主键 |
| Sdept | 所属院系 | 字符串，长度为16 | 非空 |
| Sage | 年龄 | 整数 | 非空 |
| Ssex | 性别 | 字符串，长度为2 | 取值为“男”或“女” |

3、专业信息表

专业信息表“tbl\_major”用于存放专业基本信息，包括专业和所属院系，其结构如表2-3所示。

表2-3 专业信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Subject | 专业 | 字符串，长度为16 | 主键 |
| Sdept | 所属院系 | 字符串，长度为16 | 非空 |

4、研究生信息表

专业信息表“tbl\_student”用于存放研究生基本信息，包括姓名、学号、年龄、性别、专业及所属院系，其结构如表2-4所示。

表2-4 研究生信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Sno | 学号 | 字符串，长度为10 | 主键 |
| Sname | 姓名 | 字符串，长度为50 | 非空 |
| Sdepartment | 所属院系 | 字符串，长度为50 | **—** |
| Sage | 年龄 | 整数 | **—** |
| Ssex | 性别 | 字符串，长度为2 | 取值为“男”或“女” |
| Smajor | 专业 | 字符串，长度为16 | **—** |

5、课程信息表

专业信息表“tbl\_Course”用于存放课程基本信息，包括课程编号、课程名称、学时、学分、授课教师及开设学期，其结构如表2-5所示。

表2-5 课程信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Cno | 课程编号 | 字符串，长度为10 | 主键 |
| Cname | 课程名称 | 字符串，长度为50 | 非空 |
| Cperiod | 学时 | 字符串，长度为10 | — |

续表2-5

|  |  |  |  |
| --- | --- | --- | --- |
| Csemester | 开设学期 | 字符串，长度为10 | — |
| Cprofessor | 授课教师 | 字符串，长度为50 | — |
| Ccredit | 学分 | 字符串，长度为10 | — |

6、成绩信息表

成绩信息表“tbl\_grade”用于存放成绩基本信息，包括学生学号、课程编号及成绩，其结构如表2-6所示。

表2-6 成绩信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| Sno | 学生学号 | 字符串，长度为10 | 主键，引用table\_student的外码 |
| Cno | 课程编号 | 字符串，长度为10 | 主键，引用table\_course的外码 |
| grade | 成绩 | 整数 | 取值0~100 |

7、导师情况信息表

导师情况信息表“tbl\_professor”用于存放导师基本信息，包括姓名、联系方式、职称、性别、出生日期及研究方向，其结构如表2-7所示。

表2-7 导师情况信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| T\_name | 姓名 | 字符串，长度为50 | 主键 |
| T\_telephone | 联系方式 | 字符串，长度为20 | 位数为11位 |
| T\_title | 职称 | 字符串，长度为50 | 非空 |
| T\_sex | 性别 | 字符串，长度为2 | 取值为“男”或“女” |
| T\_birthtime | 出生日期 | 时间日期类型 | — |
| T\_major | 研究方向 | 字符串，长度为50 | — |

8、论文发表情况信息表

论文发表情况信息表“tbl\_paper”用于存放论文基本信息，包括学生姓名、学号、指导教师、论文题目及发表时间，其结构如表2-8所示。

表2-8 论文发表情况信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| sno | 学号 | 字符串，长度为10 | 非空 |
| sname | 姓名 | 字符串，长度为50 | 主键 |
| title | 职称 | 字符串，长度为50 | 非空 |
| instructor | 指导教师 | 字符串，长度为50 | 主键 |
| paper\_time | 发表时间 | 时间日期类型 | 非空 |

9、党员情况信息表

党员情况信息表“tbl\_party”用于存放党员基本信息，包括学生姓名、学号、入党时间、出生日期、专业、所在院系、民族、性别及联系方式，其结构如表2-9所示。

表2-9 党员情况信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| sno | 学号 | 字符串，长度为10 | 主键 |
| sname | 姓名 | 字符串，长度为50 | 非空 |
| subject | 专业 | 字符串，长度为50 | — |
| sdept | 所在院系 | 字符串，长度为50 | — |
| party\_time | 入党时间 | 时间日期类型 | — |
| birthtime | 出生日期 | 时间日期类型 | — |
| sex | 性别 | 字符串，长度为2 | 取值为“男”或“女” |
| telephone | 联系方式 | 字符串，长度为50 | 位数为11位 |
| nation | 民族 | 字符串，长度为20 | — |

10、实习情况信息表

实习情况信息表“tbl\_exercitation”用于存放实习情况基本信息，包括学生姓名、学号、实习起始时间、出生日期、实习结束时间、实习时长、实习岗位、城市及薪资，其结构如表2-10所示。

表2-10 实习情况信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 约束 |
| sno | 学号 | 字符串，长度为10 | 主键 |
| sname | 姓名 | 字符串，长度为50 | 非空 |
| city | 城市 | 字符串，长度为50 | — |
| position | 实习岗位 | 字符串，长度为50 | — |
| payment | 薪资 | 字符串，长度为50 | — |
| exercutaition\_starttime | 实习起始时间 | 时间日期类型 | — |
| exercitation\_endtime | 实习结束时间 | 时间日期类型 | — |
| birthtime | 出生日期 | 时间日期类型 | — |
| timespan | 实习时长 | 字符串，长度为10 | — |

# 第3章 系统的详细设计及实现

## 3.1 公共类

考虑到系统的各个模块都需要访问数据库，本系统在开发时编写了一些访问数据库的方法，如返回数据集的公共查询方法，执行数据操作的公共方法，并把它们放在一个公共的类（DataBase）中，然后在各模块中调用这些方法来实现对数据库的访问。

另一方面，根据登录用户角色的不同，本系统提供了不同的访问权限。在每张表中都可能会涉及到不同的访问权限。本系统在DataBase的基础上编写了另一个公共的类（ClassShared），在其中建立了一个静态公共数组userinfo，预先将用户名和访问权限存到数组中，通过调用数组元素获知访问权限从而对每张表进行相应的操作。

### 3.1.1 编写DataBase公共类及公共方法

为系统添加一个名为“DataBase”的公共类，用于存放访问数据库的公共方法，为“DataBase”类声明几个变量以及数据库连接字符串，代码如下：

SqlConnection myconn;

SqlCommand mycmd;

SqlDataAdapter mydtadp;

DataSet ds;

string strconn = "data source =(local);uid=sa;pwd=1;database=Postgraduate";

1、公共查询方法GetDataFromDB

GetDataFromDB是一个返回数据集的公共查询方法，如果正常访问则返回查询结果；否则返回null。代码如下：

public DataSet GetDataFromDB(string mysql)

{

try

{

myconn = new SqlConnection(strconn);

myconn.Open();

mydtadp = new SqlDataAdapter(mysql, strconn);

ds = new DataSet();

mydtadp.Fill(ds);

myconn.Close();

}

catch (Exception exp)

{

MessageBox.Show(exp.Message);

myconn.Close();

}

if (ds.Tables[0].Rows.Count != 0)

{

return ds;

}

else

{

return null;

}

}

2、公共数据操作方法UpDataDB

公共数据操作方法UpDataDB用于对数据进行添加、修改和删除操作，若操作成功则返回true；否则返回false。代码如下：

public bool UpDataDB(string mysql)

{

try

{

myconn = new SqlConnection(strconn);

myconn.Open();

mycmd = new SqlCommand(mysql, myconn);

mycmd.ExecuteNonQuery();

myconn.Close();

return true;

}

catch (Exception exp)

{

MessageBox.Show(exp.Message);

myconn.Close();

return false;

}

}

### 3.1.2 添加ClassShared公共类

为项目添加一个名为“ClassShared”的公共类，为“ClassShared”类声明一个string类型数组，代码如下：

public class ClassShared

{

public static string[] userinfo = new string[2];

}

## 3.2 用户注册与修改密码

初始页面有一个菜单栏，包含用户登录、注册、修改密码以及改变系统主题四个菜单项。单击不同的菜单项以进入不同的功能界面。用户注册和修改密码界面都是一个功能完善的系统必不可缺的部分。

### 3.2.1 用户注册界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button等控件设计用户注册界面如图3-1所示。

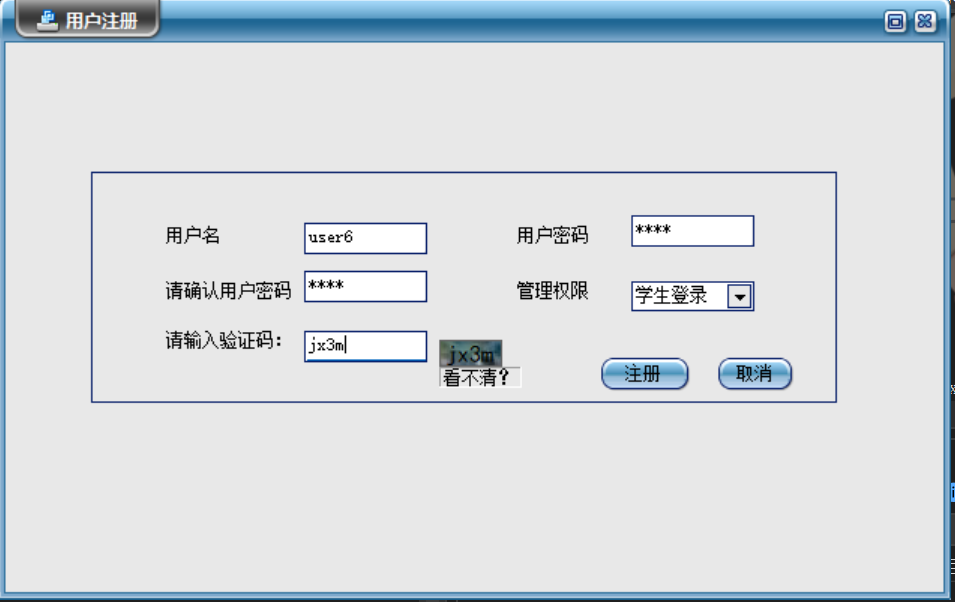


图3-1 用户注册界面

### 3.2.2 用户注册代码

在如图3-1所示的用户注册界面中，【注册】按钮用于验证输入的用户名是否未注册、两次输入的密码是否一致以及图形验证码是否正确，若都符合则注册成功；否则弹出相应错误提示，清空textbox控件的内容，并等待用户的重新输入。

设置一个公共静态方法CreatRandomCode，参数为length，定义string类型的变量list，令其值等于26个小写字母加上0-9十个数字，接着定义了Random随机数对象以及一个string类型的空变量code。代码如下：

string list = "abcdefghijklmnopqrstuvwxyz1234567890";

Random random = new Random();

string code = "";

然后对传递过来的参数length的值进行遍历，并利用Random生成随机数添加到变量code中，最后返回code的值。代码如下：

for (int i = 0; i < length; i++)

{

code += list[random.Next(0, list.Length - 1)];

}

return code;

在“FrmRegister”窗体中，双击【注册】按钮，编写其单击事件的代码，首先对所填信息进行判断，用户填写的用户名、密码不能为空，代码如下：

if (txtusername.Text.Trim() == "" || txtuserpassword.Text.Trim() == "")

{

MessageBox.Show("用户名或密码不得为空！");

}

然后利用异常处理结构对用户注册的操作进行捕获，防止系统不正常终止。定义DataBase的对象，以调用UpadataDB的方法，并写出sql语句mysql。代码如下：

DataBase DB = new DataBase();

string mysql = "insert into tbl\_login values('" + txtusername.Text.Trim() + "','" + txtuserpassword.Text.Trim() + "','" + cmbpurview.Text.Trim() + "')";

bool i = DB.UpDataDB(mysql);

接着判断两次输入的密码是否一致，如果不一致，弹出错误对话框并清空相应textbox内容。代码如下：

MessageBox.Show("两次输入密码不一致，请重输！");

txtuserpassword.Text = "";

txtpasswordok.Text = "";

对用户输入验证码进行判断，如果不正确，重新生成四位图形验证码，弹出错误对话框并清空相应textbox内容，代码如下：

if (label2.Text.Trim() != textBox1.Text.Trim())

{

MessageBox.Show("验证码不正确！请重新输入");

label2.Text = CreatRandomCode(4);

textBox1.Text = "";

}

如果用户看不清四位图形验证码，可以点击图形验证码下方的“看不清？”，以更新四位图形验证码。代码如下：

label2.Text = CreatRandomCode(4);

定义一个布尔类型的数据，使其等于公共方法UpDataDB的结果，如果为true，则弹出注册成功的对话框，清空文本框及组合框的内容，重新生成四位全新的图形验证码；如果为false，则修改失败，弹出错误对话框。代码如下：

MessageBox.Show("注册成功！");

txtuserpassword.Text = "";

txtusername.Text = "";

txtpasswordok.Text = "";

cmbpurview.Text = "";

label2.Text = CreatRandomCode(4);

textBox1.Text = "";

系统运行时，单击用户注册界面中的【取消】按钮，则关闭注册界面，其单击事件主要代码如下：

this.Close();

### 3.2.3 修改密码界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button等控件设计用户注册界面如图3-2所示。

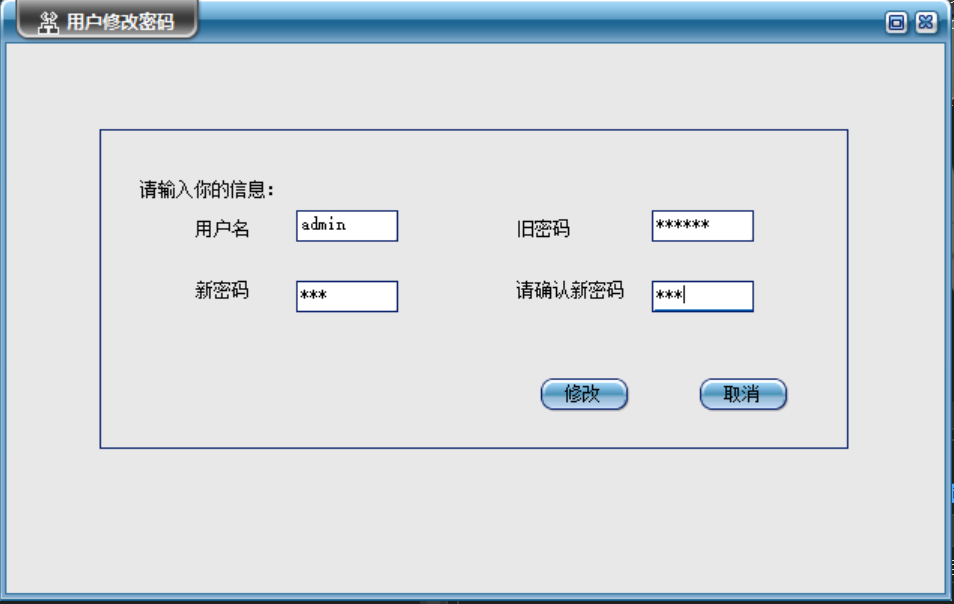


图3-2 用户修改密码界面

### 3.2.4 修改密码代码

在如图3-2所示的用户修改密码界面中，【修改】按钮用于验证用户名是否已经存在，原密码是否与用户名匹配，旧密码是否与新密码一致以及两次输入的新密码是否一致。若都符合则修改成功，清空textbox控件的内容；否则弹出相应错误提示，清空textbox控件的内容，并等待用户的重新输入。

首先判断输入的数据是否为空，如果有这样的“空数据”，弹出错误对话框。代码如下：

if (txtuser.Text == "" || txtpwd.Text == "" || txtnewpwd.Text == "" || txtnewpwdok.Text != "")

{

MessageBox.Show("请填入全部信息！");

return;

}

然后利用异常处理结构对用户修改密码的操作进行捕获，防止系统不正常终止。定义DataBase的对象，以调用GetDataFromDB的方法，写出sql语句mysql，并作为参数传递给公共方法GetDataFromDB。接着定义一个DataSet的对象，以承载查询的结果。核心代码如下：

string mysql = "select Username,Userpassword from tbl\_login where Username='" + txtuser.Text.Trim() + "' and Userpassword='" + txtpwd.Text.Trim() + "'";

myds = DB.GetDataFromDB(mysql);

最后判断这张数据表是否为空（即判断输入的用户名是否存在），如果为空，弹出错误信息。代码如下：

if (myds.Tables[0].Rows.Count <= 0)

{

MessageBox.Show("用户名不存在或输入错误！");

}

接着判断两次输入的密码是否一致，如果不一致，弹出错误对话框并清空textbox文本框内容。代码如下：

MessageBox.Show("两次输入密码不一致！请重输");

txtnewpwd.Text = "";

txtnewpwdok.Text = "";

然后判断新密码和旧密码是否一致，如果一致，弹出错误对话框。代码如下：

MessageBox.Show("新密码不能和旧密码一致！", "提示");

接着弹出一个确认窗口，询问是否确定修改，当取消修改时，textbox文本框内容清空，退出修改状态。代码如下：

txtnewpwd.Text = "";

txtnewpwdok.Text = "";

txtuser.Text = "";

txtpwd.Text = "";

编写sql语句，作为参数传递给公共方法UpDataDB。核心代码如下：

string updatesql = "update tbl\_login set Userpassword ='" + txtnewpwdok.Text.Trim() + "' where Username='" + txtuser.Text.Trim() + "'";

定义一个布尔类型的数据，使其等于公共方法UpDataDB的结果，如果为true，则弹出修改成功的对话框，清空文本框及组合框的内容。代码如下：

MessageBox.Show("修改成功！");

txtnewpwd.Text = "";

txtnewpwdok.Text = "";

txtuser.Text = "";

txtpwd.Text = "";

如果公共方法UpDataDB的结果为false，弹出错误对话框。

系统运行时，单击修改密码界面中的【取消】按钮，则关闭修改密码界面。

## 3.3 系统登录与主窗体

登录是每一个成功项目中不可缺少的模块，好的登录模块可以保证系统的可靠性和安全性。首先为“研究生信息管理系统”制作了一个简单的登录模块，登录成功后，应当进入系统的主窗体（主界面）。

### 3.3.1 登录界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button等控件设计登录界面如图3-3所示。



图3-3 登录界面

### 3.3.2 登录代码

在如图3-1所示的登录界面中，【登录】按钮用于验证输入的用户名和用户密码是否正确，若正确则进入系统主界面；否则弹出错误提示，清空textbox控件的内容，并等待用户的重新输入。

在“FrmLogin”窗体中，双击【登录】按钮，编写其单击事件的代码，首先对所填信息进行判断，用户填写的用户名、密码不能为空，代码如下：

if (textBox1.Text == "" || textBox2.Text == "")

{

MessageBox.Show("请输入用户名和密码！");

return;

}

然后利用异常处理结构对用户登录的操作进行捕获，防止系统不正常终止。定义DataBase的对象，以调用GetDataFromDB的方法，写出查询语句mysql，作为参数传递给公共方法GetDataFromDB，定义一个DataSet的对象，以承载查询的结果。核心代码如下：

string mysql = "select Userpassword,Userpurview from tbl\_login where Username ='" + textBox1.Text.Trim() + "' ";

myds = DB.GetDataFromDB(mysql);

对用户信息进行判断，数据库中的身份信息与用户所填写的信息是否一致。如果一致，利用公共类ClassShared定义的静态公共数组存储用户名和管理权限这一组信息，以便在各窗体中使用。代码如下：

if (myds.Tables[0].Rows[0].ItemArray[0].ToString().Trim()==textBox2.Text.Trim())

{

FrmMain main = new FrmMain();

ClassShared.userinfo[0]=textBox1.Text.Trim();

ClassShared.userinfo[1]=myds.Tables[0].Rows[0].ItemArray[1].ToString();

main.Show();

this.Hide();

}

如果不一致，此时弹出消息框，提示用户填写信息不正确并且将用户名和密码清空。

如果用户填写信息无误，则进入系统主界面，系统登录界面隐藏。

### 3.3.3 主窗体设计

在系统登录界面中，单击【登录】按钮验证用户名、用户密码及其身份，若正确则进入系统主界面；否则弹出错误提示，并等待用户的重新输入。单击【退出】按钮则关闭登录界面，退出系统。

因为登录代码中包含了显示系统主界面的代码，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmMain”的主窗体。如图3-4所示。

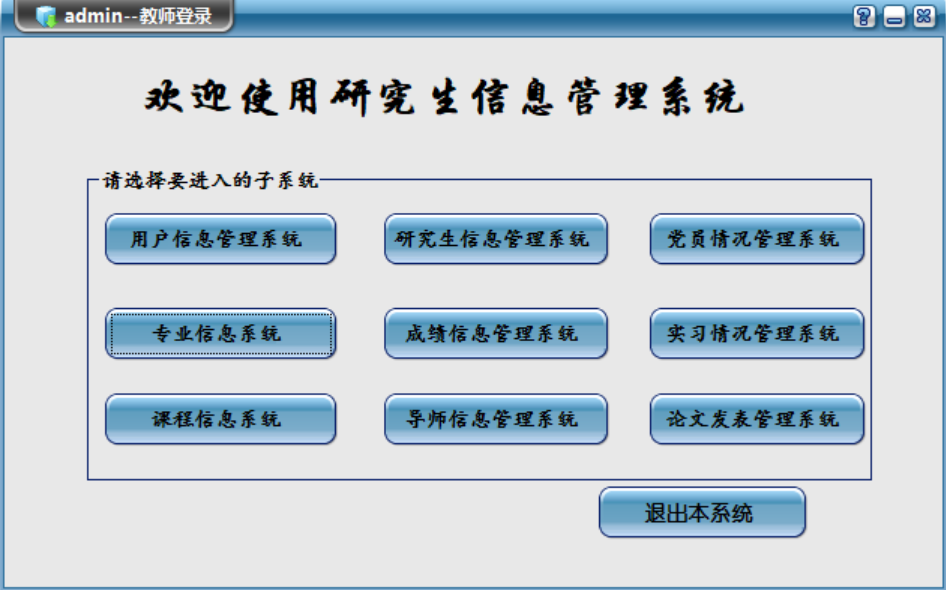


图3-4 主窗体界面

### 3.3.4 主窗体代码

主窗体即主界面，是用户进入各子系统（用户信息管理、研究生信息管理、党员信息管理、专业信息管理、成绩信息管理、实习情况管理、课程信息管理、导师信息管理和论文发表信息管理）的入口。

选择不同的功能即进入不同的子窗体，对应的子窗体显示以选择研究生信息管理为例，当单击【研究生信息管理系统】按钮时，主要代码如下：

FrmStudent student = new FrmStudent();

student.Show();

单击主界面中的【退出本系统】按钮，则关闭主界面，退出系统，其单击事件的主要代码如下：

Application.Exit();

主窗体载入时，标题自动根据输入的用户名发生改变，主要代码如下：

this.Text = ClassShared.userinfo[0] + "--" + ClassShared.userinfo[1];

## 3.4 用户信息管理

当点击主窗体的【用户信息管理系统】按钮时，需要弹出用户信息管理窗口以实现对用户信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmUser”的窗体。

### 3.4.1 用户信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、ComboBox、DataDridView等控件设计用户信息管理界面如图3-5所示。



图3-5 用户信息管理界面

### 3.4.2 用户信息管理模块编码实现

由于在各功能的实现过程中需进行多种频繁操作，为避免代码的繁琐，在进行各功能代码的编写之前，先将频繁操作编写成方法，以供调用。

使所有文本框及复选框皆不可用的方法DataClose，主要代码如下：

void DataClose()

{

textBox1.Enabled = false;

textBox2.Enabled = false;

comboBox1.Enabled = false;

comboBox2.Enabled = false;

}

使所有文本框及复选框皆可用的方法DataOpen，主要代码如下：

void DataOpen()

{

textBox1.Enabled = true;

textBox2.Enabled = true;

comboBox1.Enabled = true;

comboBox2.Enabled = true;

}

使所有文本框及复选框的内容都清空的方法DataClear，主要代码如下：

void DataClear()

{

textBox1.Text = "";

textBox2.Text = "";

comboBox1.Text = "";

comboBox2.Text = "";

}

令datagridview控件的表头显示出一个更加通俗易懂名字的方法SetHeaderText，其主要代码如下：

dgvuser.Columns[0].HeaderText = "姓名";

dgvuser.Columns[1].HeaderText = "性别";

dgvuser.Columns[2].HeaderText = "所在学院";

dgvuser.Columns[3].HeaderText = "年龄";

使数据表中的数据刷新并显示在datagridview控件上的方法refreshdata。先定义DataBase的对象，以调用GetDataFromDB的方法，写出查询语句mysql，作为参数传递给公共方法GetDataFromDB，定义一个DataSet的对象，以承载查询的结果。核心代码如下：

string mysql = "select \* from tbl\_user";

myds = DB.GetDataFromDB(mysql);

设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上，代码如下：

int i = dgvuser.CurrentCell.RowIndex;

textBox1.Text = dgvuser[0, i].Value.ToString();

textBox2.Text = dgvuser[3, i].Value.ToString();

comboBox1.Text = dgvuser[1, i].Value.ToString();

comboBox2.Text = dgvuser[2, i].Value.ToString();

当用户信息管理窗体载入时，自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及复选框不可用，代码如下：

refreshdata();

DataClose();

在如图3-5所示的用户信息管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmUser”窗体中，双击【添加】按钮，编写其单击事件的代码，首先对按钮的文本信息进行判断；如果是“添加”，则datagridview控件不可用，其余按钮都置灰，文本框及组合框都可用。此时按钮的文本信息更改为“确定”，并等待填入用户信息，代码如下：

btnadd.Text = "确定";

DataClear();

DataOpen();

dgvuser.Enabled = false;

btndelete.Enabled = false;

btnexit.Enabled = false;

btnupdate.Enabled = false;

如果按钮的文本信息是“确定”，则用户已完成添加操作。此时判断输入的数据是否为空，假如存在这样的“空数据”，则弹出错误提示并保持添加状态，直到消除所有的空数据。代码如下：

if (textBox1.Text == "" || textBox2.Text == "" || comboBox1.Text == "" || comboBox2.Text == "")

{

MessageBox.Show("输入的信息不完整，请继续输入！");

}

当消除所有空数据后，datadridview控件和其余按钮重新可用，此时文本框及组合框置灰，按钮的文本信息更改为“添加”。

接着利用异常处理结构对添加的操作进行捕获，防止系统不正常终止。然后定义DataBase的对象，以调用UpDataDB的方法，写出查询语句mysql，作为参数传递给公共方法UpDataDB，定义一个布尔类型的数据，使其等于公共方法UpDataDB的结果。核心代码如下：

DataBase DB = new DataBase();

string mysql = "insert into tbl\_user values('" + textBox1.Text.ToString().Trim() + "','" + comboBox1.Text.ToString().Trim() + "','" + comboBox2.Text.ToString().Trim() + "','" + textBox2.Text.ToString().Trim() + "')";

bool i = DB.UpDataDB(mysql);

如果公共方法UpDataDB的结果为true，则询问用户是否还要继续添加数据，若选择继续添加，则清空文本框及组合框数据，将新的数据表展现在datagridview控件上。此时按钮的文本信息变为“添加”并演示添加按钮的事件。代码如下：

if (DialogResult.Yes == MessageBox.Show("添加成功，您是否还要继续添加记录？", "提示", MessageBoxButtons.YesNo, MessageBoxIcon.Question))

{

DataClear();

refreshdata();

btnadd.Text = "添加";

btnadd.PerformClick();

}

如果公共方法UpDataDB的结果为false，弹出错误信息，清空文本框及组合框内容，并退出添加状态。

如果不继续添加，同样清空文本框及组合框内容，将新的数据表展现在datagridview控件上，并利用return语句退出添加状态。

在“FrmUser”窗体中，双击【修改】按钮，编写其单击事件的代码，首先判断是否选中了某一条数据，如果未选中，弹出错误信息并保持修改状态；如果已选中，则继续完成修改操作。代码如下：

if (textBox1.Text == "" || textBox2.Text == "" || comboBox1.Text == "" || comboBox2.Text == "")

{

MessageBox.Show("数据不得为空，请选中或继续输入！");

return;

}

接着对按钮的文本信息进行判断，如果是“修改”，则datagridview控件不可用，其余按钮按钮都置灰，文本框及组合框都可用。此时，按钮的文本信息更改为“确定”，等待修改用户信息。

如果按钮的文本信息是“确定”，则已完成修改操作。此时先判断是否有数据被赋予了空值，假如存在 “空数据”，则弹出错误提示并保持修改状态，直到消除所有的空数据。

当消除所有空数据后，datadridview控件和其余按钮重新可用，此时文本框及组合框置灰，按钮的文本信息更改为“修改”。

接着利用异常处理结构对修改的操作进行捕获，防止系统不正常终止。然后定义DataBase的对象，以调用UpDataDB的方法，写出查询语句mysql，作为参数传递给公共方法UpDataDB，定义一个布尔类型的数据，使其等于公共方法UpDataDB的结果，如果为true，则弹出修改成功的对话框并更新数据表中的数据，使其重新显示在datagridview控件上；如果为false，则修改失败，清空文本框及复合框内容，并退出修改状态。核心代码如下：

string mysql = "update tbl\_user set Sage='" + textBox2.Text.ToString().Trim() + "',Sdept='" + comboBox2.Text.ToString().Trim() + "',Ssex='" + comboBox1.Text.ToString().Trim() + "' where Sname='" + textBox1.Text.ToString().Trim() + "'";

在“FrmUser”窗体中，双击【删除】按钮，编写其单击事件的代码，首先判断是否选中了某一条数据，如果未选中，弹出错误信息并保持修改状态；如果已选中，则继续完成删除操作。

接着将“删除”按钮的文本信息更改为“确认”，并定义DataBase的对象，以调用UpDataDB的方法，接着写出查询语句mysql，作为参数传递给公共方法UpDataDB。核心代码如下：

string mysql = "delete from tbl\_user where Sname='" + textBox1.Text.ToString().Trim() + "'";

然后会弹出一个询问窗口，询问是否确定删除，当不确定删除时，按钮的text被更改回“删除”并退出删除状态。

当确定删除时，定义一个布尔类型的数据，使其等于公共方法UpDataDB的结果，如果为true，则弹出删除成功的对话框并更新数据表中的数据，使其重新显示在datagridview控件上，此时按钮的文本信息被更改为“删除”并退出删除状态。

在“FrmUser”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，首先定义默认文件名和文件保存名，并实例化文件对象，代码如下：

string fileName = "IC00";

string saveFileName = "";

SaveFileDialog saveDialog = new SaveFileDialog();

然后设置文件默认扩展名和当前文件名筛选器字符串，接着令对话框中保存的文件名为默认文件名。弹出保存窗口让用户选择路径和设置文件名，并令文件保存名为用户自设置文件名。代码如下：

saveDialog.DefaultExt = "xlsx";

saveDialog.Filter = "Excel文件|\*.xlsx";

saveDialog.FileName = fileName;

saveDialog.ShowDialog();

saveFileName = saveDialog.FileName;

接着判断用户是否点击了“取消”，如果是，退出当前对话框。代码如下：

if (saveFileName.IndexOf(":") < 0)

{

return;

}

然后实例化excel对象，判断用户电脑是否安装了excel软件。如果excel对象为空，弹出错误窗口，退出保存状态。代码如下：

Microsoft.Office.Interop.Excel.Application xlApp = new Microsoft.Office.Interop.Excel.Application();

if (xlApp == null)

{

MessageBox.Show("无法创建Excel对象，您的电脑可能未安装Excel");

return;

}

接着定义一个Microsoft Excel 工作簿并添加新工作表，取得worksheets[1]。代码如下：

Microsoft.Office.Interop.Excel.Workbooks workbooks = xlApp.Workbooks;

Microsoft.Office.Interop.Excel.Workbook workbook=workbooks.Add(Micr osoft.Office.Interop.Excel.XlWBATemplate.xlWBATWorksheet);

Microsoft.Office.Interop.Excel.Worksheet worksheet =(Microsoft.Off ice.Interop.Excel.Worksheet)workbook.Worksheets[1];

然后遍历循环获取DataGridView标题。代码如下：

for (int i = 0; i < dgv\_major.ColumnCount; i++)

{

worksheet.Cells[1, i + 1] = dgv\_major.Columns[i].HeaderText;

}

最后编写一个两层循环，在worksheets[1]中填入相应的数值并实时更新表格。代码如下：

for (int r = 0; r < dgv\_major.Rows.Count; r++)

{

for (int i = 0; i < dgv\_major.ColumnCount; i++)

{

worksheet.Cells[r + 2, i + 1] = dgv\_major.Rows[r].Cells[i].Value;

}

System.Windows.Forms.Application.DoEvents();

}

列宽自适应并弹出保存成功窗口，代码如下：worksheet.Columns.EntireColumn.AutoFit();

MessageBox.Show(fileName + "资料保存成功", "提示", MessageBoxButtons.OK);

单击用户信息管理界面中的【退出】按钮，则关闭用户信息管理界面，隐藏FrmUser窗体。

## 3.5 成绩信息管理

当点击主窗体的【成绩信息管理系统】按钮时，需要弹出成绩信息管理窗口以实现对成绩信息的录入、查询、查询全部、退出等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmGrade”的窗体。

### 3.5.1 成绩信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、ComboBox、DataDridView等控件设计研究生信息管理界面如图3-6所示。



图3-6 成绩信息管理界面

### 3.5.2 成绩信息管理模块编码实现

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字，和用户信息管理系统实现方法类似，在此不再赘述。当成绩信息管理窗体载入时，判断ClassShared类的userinfo[1]的值，如果是教师登录，则所有功能模块都能实现；如果是学生登录，导入成绩功能不可用。其主要代码如下：

if (ClassShared.userinfo[1] == "学生登录")

{

btninsert.Enabled = false;

}

else if (ClassShared.userinfo[1] == "教师登录")

{

btninsert.Enabled = true;

}

从tbl\_student中查询学生学号以及姓名，首先定义DataBase的对象，以调用GetDataFromDB的方法，查询tbl\_student表中的Sno、Sname，写出查询语句mysql，作为参数传递给公共方法GetDataFromDB，定义一个DataSet的对象，以承载查询的结果。核心代码如下：

string mysql = "select Sno,Sname from tbl\_student";

student = DBStudent.GetDataFromDB(mysql);

最后判断这张数据表是否为空，如果为空，弹出错误信息；不为空，作一循环，以向组合框cmbstudent中添加查询到的数据。代码如下：

if (student.Tables[0].Rows.Count > 0)

{

for (int i = 0; i < student.Tables[0].Rows.Count; i++)

{

cmbstudent.Items.Add(student.Tables[0].Rows[i].ItemArray[0].ToString().Trim() + "--" +student.Tables[0].Rows[i].ItemArray[1].ToString().Trim());

}

}

接着查询tbl\_Course表中的Cno、Cname，写出查询语句mysql，作为参数传递给公共方法GetDataFromDB，实现方法与上一查询步骤类似，在此不再赘述。

在如图3-6所示的成绩信息管理界面中，有导入成绩、查询、查询全部功能按钮。在“FrmGrade”窗体中，双击【导入】按钮，编写其单击事件的代码，首先判断输入数据是否为空，如果有空数据，弹出错误信息。然后定义DataBase的对象，以调用UpDataDB的方法，编写sql语句，将学号，课程号以及成绩添加到tbl\_grade数据表中，sql语句作为参数传递给公共方法UpDataDB，定义一个布尔类型的数据，等于公共方法UpDataDB的结果。核心代码如下：

string mysql = "insert into tbl\_grade values('" + cmbstudent.Text.ToString().Substring(0, 4) + "','" + cmbcourse.Text.ToString().Substring(0, 4) + "','" + txtgrade.Text + "')";

如果为true，则弹出修改成功的对话框；如果为false，则导入失败，弹出错误对话框。最后将所有文本框及组合框数据清空。

在“FrmGrade”窗体中，双击【查询】按钮，编写其单击事件的代码，首先判断cmbkind和cmbcontent的索引是否为-1（即没选中）。代码如下：

if (cmbcontent.SelectedIndex == -1 || cmbkind.SelectedIndex == -1)

{

MessageBox.Show("选择不能为空！");

}

定义DataBase的对象，以调用GetDataFromDB的方法，写出查询语句mysql，作为参数传递给公共方法GetDataFromDB，定义一个DataSet的对象，以承载查询的结果，实现方法在此不再赘述。核心代码如下：

string mysql = "select tbl\_grade.Sno,tbl\_student.Sname,tbl\_grade.Cno,tbl\_Course.Cname,tbl\_grade.grade from tbl\_student,tbl\_Course,tbl\_grade where tbl\_Course.Cno=tbl\_grade.Cno and tbl\_student.Sno=tbl\_grade.Sno and " + selectcondition + "='" + cmbcontent.Text.Trim() + "'";

最后判断这张数据表是否为空，如果为空，弹出错误信息；不为空，将查询到的数据表显示在datagridview控件上，并调用SetHeaderText方法。

在“FrmGrade”窗体中，双击【查询全部】按钮，编写其单击事件的代码，其实现方法和【查询】类似，在此不再赘述。核心代码如下：

string mysql="select tbl\_grade.Sno,tbl\_student.Sname,tbl\_grade.Cno,tbl\_Course.Cname,tbl\_grade.grade from tbl\_student,tbl\_Course,tbl\_grade where tbl\_Course.Cno=tbl\_grade.Cno and tbl\_student.Sno=tbl\_grade.Sno";

单击成绩信息管理界面中的【退出】按钮，则关闭成绩信息管理界面，隐藏FrmGrade窗体。

## 3.6 论文发表情况管理

当点击主窗体的【论文发表情况管理系统】按钮时，需要弹出论文发表情况管理窗口以实现对论文发表情况的上传及查询功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmPaper”的窗体。

### 3.6.1 论文发表情况管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、ComboBox、DataDridView等控件设计论文发表情况管理界面如图3-7所示。



图3-7 论文发表情况管理界面

### 3.6.2 论文发表情况管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

在如图3-17所示的论文发表情况管理界面中，有上传论文及查询功能按钮。

在“FrmPaper”窗体中，双击【上传】按钮，编写其单击事件的代码，其实现方法和成绩信息管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_paper values('" + txtinstructor.Text.Trim() + "','" + dtpublish.Value + "','" + txtno.Text.Trim() + "','" + txtname.Text.Trim() + "','" + txttitle.Text.Trim() + "')";

在“FrmPaper”窗体中，双击【查询】按钮，编写其单击事件的代码。

首先判断一下cmbcontent及cmbkind的索引值是否为空，即判断是否选中了某一项来进行查询。

利用异常处理结构对查询的操作进行捕获，防止系统不正常终止。定义DataBase的对象，以调用GetDataFromDB的方法，写出查询语句mysql，作为参数传递给公共方法GetDataFromDB，定义一个DataSet的对象，以承载查询的结果。核心代码如下：

string mysql = "select instructor ,paper\_time,sno,sname,title from tbl\_paper where instructor='" + cmbcontent.Text.Trim() + "' or sname='" + cmbcontent.Text.Trim() + "'";

然后判断一下这张数据表是否为空，为空时弹出错误信息；当不为空时，若cmbkind的索引值是0，先清空组合框cmbcontent数据，再向组合框cmbcontent中添加表中数据。代码如下：

cmbcontent.Items.Clear();

cmbcontent.Items.Add(myds.Tables[0].Rows[0].ItemArray[0]);

接着遍历一个两层循环，并在循环内部定义一个布尔型变量flag，用来判断添加的表中数据是否重复。如果重复，令flag=false，并终止循环；否则令flag=true，继续循环。代码如下：

if (myds.Tables[0].Rows[i].ItemArray[0].ToString().Trim() != myds.Tables[0].Rows[n].ItemArray[0].ToString().Trim())

{

flag = true;

}

在外层循环结束后，立即判断flag的真假，如果为真，才能添加表中数据。代码如下：

if (flag == true)

{

cmbcontent.Items.Add(myds.Tables[0].Rows[i].ItemArray[0]);

}

当cmbkind的索引值是1时，与索引值为0时实现的查询方法类似，在此不再赘述。

在“FrmPaper”窗体中，双击【查询全部】按钮，编写其单击事件的代码，其实现方法和查询类似，在此不再赘述。核心代码如下：

string mysql = "select instructor ,paper\_time,sno,sname,title from tbl\_paper ";

单击论文发表情况管理界面中的【退出】按钮，则关闭论文发表情况管理界面，隐藏FrmPaper窗体。

## 3.7 实习情况管理

当点击主窗体的【实习情况管理系统】按钮时，需要弹出实习情况管理窗口以实现对实习信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmExercitation”的窗体。

### 3.7.1 实习情况管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、ComboBox、DataDridView等控件设计实习情况管理界面如图3-8所示。



图3-8 实习情况管理界面

### 3.7.2 实习情况管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字。设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上。

当实习情况管理窗体载入时，自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及复选框不可用。

设置dtstarttime和dtendtime的ValueChanged事件，以自动计算实习时长并显示在txtspan控件上，代码如下：

DateTime dt1 = dtstarttime.Value;

DateTime dt2 = dtendtime.Value;

TimeSpan timeSpan = dt2 - dt1;

txtspan.Text = timeSpan.Days.ToString();

在如图3-8所示的实习情况管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmExercitation”窗体中，双击【添加】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_exercitation values('" + txtno.Text.Trim() + "','" + txtcity.Text.Trim() + "','" + txtposition.Text.Trim() + "','" + txtpayment.Text.Trim() + "','" + txtname.Text.Trim() + "','" + dtbirthtime.Value + "','" + dtstarttime.Value + "','" + dtendtime.Value + "','" + txtspan.Text.Trim() + "')";

在“FrmExercitation”窗体中，双击【修改】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "update tbl\_exercitation set exercitation\_starttime='" + dtstarttime.Value + "',city='" + txtcity.Text.Trim() + "',sname='" + txtname.Text.Trim() + "',payment='" + txtpayment.Text.Trim() + "',position='" + txtposition.Text.Trim() + "', exercitation\_endtime='" + dtendtime.Value + "',birthtime='" + dtbirthtime.Value + "',timespan='"+txtspan.Text.Trim()+"' where sno='"+ txtno.Text.Trim() + "'";

在“FrmExercitation”窗体中，双击【删除】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "delete from tbl\_exercitation where sno='" + txtno.Text.Trim() + "'";

在“FrmExercitation”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。

单击实习信息管理界面中的【退出】按钮，则关闭实习信息管理界面，隐藏FrmExercitation窗体。

## 3.8 专业信息管理

当点击主窗体的【专业信息管理】按钮时，需要弹出专业信息管理窗口以实现对专业信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmMajor”的窗体。

### 3.8.1 专业信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Combox、Label、TextBox、Button、DataDridView控件等设计专业信息管理界面如图3-9所示。



图3-9 专业信息管理界面

### 3.8.2 专业信息管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字。设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上。

当专业信息管理窗体载入时，判断ClassShared类的userinfo[1]的值，如果是教师登录，则所有功能模块都能实现；如果是学生登录，只有导出为excel功能可用。然后自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及组合框不可用。与用户信息管理系统实现方法类似，在此不再赘述。

在如图3-9所示的专业信息管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmMajor”窗体中，双击【添加】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_major (Subject,Sdept) values ('" + textBox1.Text.Trim() + "','" + textBox2.Text.Trim() + "')";

在“FrmMajor”窗体中，双击【删除】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "delete from tbl\_major where Subject='" + textBox1.Text + "'";

在“FrmMajor”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。

单击专业信息管理界面中的【退出】按钮，则关闭专业信息管理界面，隐藏FrmMajor窗体。

## 3.9 导师信息管理

当点击主窗体的【导师信息管理系统】按钮时，需要弹出导师信息管理窗口以实现对导师信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmProfessor”的窗体。

### 3.9.1 导师信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、ComboBox、DataDridView等控件设计导师信息管理界面如图3-10所示。



图3-10 导师信息管理界面

### 3.9.2 导师信息管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字。设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上。

当导师信息管理窗体载入时，判断ClassShared类的userinfo[1]的值，如果是教师登录，则所有功能模块都能实现；如果是学生登录，只有导出为excel功能可用。

自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及复选框不可用。

在如图3-16所示的导师信息管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmProfessor”窗体中，双击【添加】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_professor values('" + txtphone.Text.Trim() + "','" + txtname.Text.Trim() + "','" + txttitle.Text.Trim() + "','" + cmbsex.Text.Trim() + "','" + txtmajor.Text.Trim() + "','" + dtbirthtime.Value + "')";

bool i = DB.UpDataDB(mysql);

在“FrmProfessor”窗体中，双击【修改】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "update tbl\_professor set T\_telephone='" + txtphone.Text.Trim() + "',T\_birthtime='" + dtbirthtime.Value + "',T\_title='" + txttitle.Text.Trim() + "',T\_sex='" + cmbsex.Text.Trim() + "',T\_major='" + txtmajor.Text.Trim() + "'where T\_name='" + txtname.Text.Trim() + "'";

在“FrmProfessor”窗体中，双击【删除】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "delete from tbl\_professor where T\_name='" + txtname.Text.Trim() + "'";

在“FrmProfessor”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。

单击导师信息管理界面中的【退出】按钮，则关闭导师信息管理界面，隐藏FrmProfessor窗体。

## 3.10 研究生信息管理

当点击主窗体的【研究生信息管理系统】按钮时，需要弹出研究生信息管理窗口以实现对研究生信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmStudent”的窗体。

### 3.10.1 研究生信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、CheckBox、DataDridView等控件设计研究生信息管理界面如图3-11所示。



图3-11 研究生信息管理界面

### 3.10.2 研究生信息管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字。设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上。

当研究生信息管理窗体载入时，自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及复选框不可用。

在如图3-5所示的研究生信息管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmStudent”窗体中，双击【添加】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_student values('" + txtno.Text.Trim() + "','" + txtname.Text.Trim() + "','" + cmbdept.Text.Trim() + "','" + cmbsex.Text.Trim() + "','" + txtage.Text.Trim() + "','" + cmbmajor.Text.Trim() + "')";

在“FrmStudent”窗体中，双击【修改】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "update tbl\_student set Sname='" + txtname.Text.Trim() + "',Sdepartment='" + cmbdept.Text.Trim() + "',Ssex='" + cmbsex.Text.Trim() + "',Sage='" + txtage.Text.Trim() + "',Smajor='" + cmbmajor.Text.Trim() + "'where Sno='" + txtno.Text.Trim() + "'";

在“FrmStudent”窗体中，双击【删除】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "delete from tbl\_student where Sno='" + txtno.Text.Trim() + "'";

bool i = DB.UpDataDB(mysql);

在“FrmStudent”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。

单击课程信息管理界面中的【退出】按钮，则关闭研究生信息管理界面，隐藏FrmStudent窗体。

## 3.11 课程信息管理

当点击主窗体的【课程信息管理系统】按钮时，需要弹出课程信息管理窗口以实现对课程信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmCourse”的窗体。

### 3.11.1 课程信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、CheckBox、DataDridView等控件设计课程信息管理界面如图3-12所示。



图3-12 课程信息管理界面

### 3.11.2 课程信息管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字。设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上。

当课程信息管理窗体载入时，自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及复选框不可用。

在如图3-6所示的课程信息管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmCourse”窗体中，双击【添加】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_course values('" + txtcno.Text.Trim() + "','" + txtcredit.Text.Trim() + "','" + txtteacher.Text.Trim() + "','" + txtcname.Text.Trim() + "','" + cmbsemester.Text.Trim() + "','" + txtperiod.Text.Trim() + "')";

在“FrmCourse”窗体中，双击【修改】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "update tbl\_course set Ccredit='" + txtcredit.Text.Trim() + "',Cprofessor='" + txtteacher.Text.Trim() + "',Cname='" + txtcname.Text.Trim() + "',Csemester='" + cmbsemester.Text.Trim() + "',Cperiod='" + txtperiod.Text.Trim() + "'where Cno='" + txtcno.Text.Trim() + "'";

在“FrmCourse”窗体中，双击【删除】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "delete from tbl\_Course where Cno='" + txtcno.Text.Trim() + "'";

在“FrmCourse”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。

单击课程信息管理界面中的【退出】按钮，则关闭课程信息管理界面，隐藏FrmCourse窗体。

## 3.12 党员信息管理

当点击主窗体的【研究生信息管理系统】按钮时，需要弹出研究生信息管理窗口以实现对研究生信息的添加、删除、修改、导出为excel等功能，所以在编写代码之前需要为应用程序添加并设计一个名为“FrmParty”的窗体。

### 3.12.1 党员信息管理模块界面设计

新建一个Windows应用程序，使用Panel、Label、TextBox、Button、CheckBox、DataDridView等控件设计党员情况管理界面如图3-13所示。



图3-13 党员情况管理界面

### 3.12.2 党员信息管理模块编码实现

在代码中要用到的频繁操作同样要编写成方法，由于和用户信息管理系统实现方法类似，在此不再赘述。

编写方法SetHeaderText令datagridview控件的表头显示出一个更加通俗易懂名字。设置控件datagridview的RowHeaderMouseClick事件，使其实现这样的功能：每次单击datagridview控件的行首，就能将数据显示在相应文本框或复选框上。

当党员情况管理窗体载入时，自动将数据表里的数据呈现在datagridview控件上，并设初始情况下所有的文本框及复选框不可用。

在如图3-15所示的党员情况管理界面中，有添加、删除、修改、导出为excel等功能按钮。

在“FrmParty”窗体中，双击【添加】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "insert into tbl\_party values('" + dtparty.Value + "','" + txtno.Text.Trim() + "','" + txtname.Text.Trim() + "','" + cmbmajor.Text.Trim() + "','" + cmbdept.Text.Trim() + "','" + txtnation.Text.Trim() + "','" + dtbirth.Value + "','" + txtphone.Text.Trim() + "','" + cmbsex.Text.Trim() + "')";

bool i = DB.UpDataDB(mysql);

在“FrmParty”窗体中，双击【修改】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "update tbl\_party set sname='" + txtname.Text.Trim() + "',sex='" + cmbsex.Text.Trim() + "',nation='" + txtnation.Text.Trim() + "',subject='" + cmbmajor.Text.Trim() + "',sdept='" + cmbdept.Text.Trim() + "',telephone='" + txtphone.Text.Trim() + "',birthtime='" + dtparty.Value.ToShortDateString().Trim() + "',party\_time='" + dtparty.Value.ToShortDateString().Trim() + "' where sno='" + txtno.Text.Trim() + "'";

在“FrmParty”窗体中，双击【删除】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。核心代码如下：

string mysql = "delete from tbl\_party where sno='" + txtno.Text.Trim() + "'";

bool i = DB.UpDataDB(mysql);

在“FrmParty”窗体中，双击【导出为excel】按钮，编写其单击事件的代码，其实现方法和用户管理系统类似，在此不再赘述。

单击党员信息管理界面中的【退出】按钮，则关闭党员信息管理界面，隐藏FrmParty窗体。

# 第4章 系统测试

## 4.1 系统测试的原则

1、尽早地和不断地进行软件测试

应当把软件测试贯穿到整个软件开发的过程中，而不应该把软件测试看作是其过程中的一个独立阶段。因为在软件开发的每一环节都有可能产生意想不到的问题，其影响因素有很多，比如软件本身的抽象性和复杂性、软件所涉及问题的复杂性、软件开发各个阶段工作的多样性，以及各层次工作人员的配合关系等。所以要坚持软件开发各阶段的技术评审，把错误克服在早期，从而减少成本，提高软件质量。在本系统的开发编码阶段，每实现一个模块，就测试一下，看看与其他模块是否能连接上，提高系统的可靠性。

2、对测试用例要有正确的态度

第一，测试用例应当由测试输入数据和预期输出结果这两部分组成；第二，在设计测试用例时，不仅要考虑合理的输入条件，更要注意不合理的输入条件。因为软件投入实际运行中，往往不遵守正常的使用方法，却进行了一些甚至大量的意外输入导致软件一时半时不能做出适当的反应，就很容易产生一系列的问题，轻则输出错误的结果，重则瘫痪失效，因此常用一些不合理的输入条件来发现更多的鲜为人知的软件缺陷。

3、严格执行测试计划，排除测试的随意性，以避免发生疏漏或者重复无效工作。

4、应当对每一个测试结果进行全面检查。一定要全面地、仔细地检查测试结果，但常常被人们忽略，导致许多错误被遗漏。

5、妥善保存测试用例、测试计划、测试报告和最终分析报告，以备回归测试及维护之用。

在遵守以上原则的基础上进行软件测试，可以以最少的时间和人力找出软件中的各种缺陷，从而达到保证软件质量的目的。

## 4.2 测试结果

主要对系统功能模块进行测试，得到的测试结果如表4-1所示。

表4-1系统测试结果表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试项目 | 验证过程 | 预期结果 | 实际结果 | 结论说明 |

续表4-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户登录 | 输入用户名：admin密码：123456 | 进入主系统 | 进入主系统 | | 测试成功通过 |
| 用户注册 | 输入用户名：user12 密码：a123 | 注册成功 | 注册成功 | | 测试成功通过 |
| 用户修改密码 | 输入用户名：user1 原密码：a123  新密码：123 | 修改成功 | 修改成功 | | 测试成功通过 |
| 管理用户信息 | 进入用户信息管理窗口，点击添加，添加相关信息并提交；点击修改，选中某用户信息，修改相关信息并提交；点击删除，选中某用户信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | | 测试成功通过 |
| 管理专业信息 | 进入专业信息管理窗口，点击添加，添加相关信息并提交；点击修改，选中客房信息，修改相关信息并提交；点击删除，选中客房信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | | 测试成功通过 |
| 管理研究生信息 | 进入研究生信息管理窗口，点击添加，添加相关信息并提交；点击修改，选中车库信息，修改相关信息并提交；点击删除，选中车库信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | | 测试成功通过 |
| 管理课程信息 | 进入课程信息管理窗口，点击添加，添加相关信息并提交；点击修改，选中客户信息，修改相关信息并提交；点击删除，选中客户信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 测试成功通过 |
| 管理成绩信息 | 进入成绩信息管理窗口，点击查询，输入相关信息并提交；点击添加，添加相关信息并提交 | 提交后，DataGridView控件内仅出现所要查询的数据；提交后增加新数据信息 | | 提交后，DataGridView控件内仅出现所要查询的数据；提交后增加新数据信息 | 测试成功通过 |
| 管理党员信息 | 进入党员信息管理窗口，点击添加，添加相关信息并提交；击修改，选中客户信息，修改相关信息并提交；点击删除，选中客户信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 测试成功通过 |
| 管理论文发表情况 | 进入论文发表情况管理窗口，点击查询，输入相关信息并提交；点击添加，添加相关信息并提交 | 提交后，DataGridView控件内仅出现所要查询的数据；提交后增加新数据信息 | | 提交后，DataGridView控件内仅出现所要查询的数据；提交后增加新数据信息 | 测试成功通过 |
| 管理实习信息 | 进入实习情况管理窗口，点击添加，添加相关信息并提交；点击修改，选中客户信息，修改相关信息并提交；点击删除，选中客户信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 测试成功通过 |

续表4-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 管理导师信息 | 进入导师信息管理窗口，点击添加，添加相关信息并提交；点击修改，选中客户信息，修改相关信息并提交；点击删除，选中客户信息，提交并确认删除 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 提交后增加新数据信息；提交后数据信息发生改变；删除后列表记录数减少 | 测试成功通过 |

# 结 论

研究生信息管理系统基于C/S结构、C#开发语言、Visual Studio开发环境以及SQL Server数据库管理系统进行构建开发的。具体实现的功能如下：

1、用户登录、注册以及修改密码管理，在登陆界面中，用户通过输入正确的用户名和密码后，则可进入系统主窗口；用户注册和修改密码是一个功能完善的系统必不可缺的部分；只要符合一定的条件即可注册成功；

2、用户信息管理，完成管理用户信息的操作。在用户信息管理界面中，有添加、删除、修改、导出为excel等功能；

3、研究生信息管理，完成管理研究生信息的操作。在研究生信息管理界面中，有添加、删除、修改、导出为excel等功能；

4、课程信息管理，完成管理课程信息的操作。在课程信息管理界面中，有添加、删除、修改、导出为excel等功能；

5、成绩信息管理。在成绩信息管理界面中，可以完成成绩导入、查询等；

6、专业信息管理。在专业信息管理界面中，完成添加、导出为excel等功能；

7、党员信息管理，完成管理党员信息的操作。在党员信息管理界面中，有添加、删除、修改、导出为excel等功能；

8、实习信息管理，完成管理实习信息的操作。在实习信息管理界面中，有添加、删除、修改、导出为excel等功能；

9、导师信息管理，完成管理导师信息的操作。在导师信息管理界面中，有添加、删除、修改、导出为excel等功能；

10、论文发表情况管理，完成管理论文发表情况信息的操作。在论文发表情况管理界面中，有上传论文、查询、查询全部等功能。

在系统开发过程中使用和扩展的关键技术如下：

1、页面布局设计方面，使用干净简洁的皮肤，用户可以通过点击主界面的皮肤以自行更换系统主题；

2、为数据库中字段添加了check约束，提升了系统安全性。例如在数据表中将手机号位数设置为固定11位，性别只能填写“男”或“女”等；

3、增加了用户注册和用户修改密码界面，使系统功能更加完备；

4、用户注册界面增加了输入图形验证码，保证了系统的安全性；

5、增设了将数据导出为excel表格的功能，为用户提供更加便利的功能。

系统具有一定的局限性，具体体现在：

1、无法对修改前后数据未改变的情况进行验证；

2、未在用户信息表和系统用户表之间产生关联。

# 参考文献

[1]李春葆．C#程序设计教程[M]．北京：清华大学出版社，2019．

[2]吕凯．数据库原理及应用—SQL Server 2019 [M]．北京：清华大学出版社，2021．

[3]张才千，钱慎一．C#程序设计与开发经典课堂(高等院校课程设计案例精编)[M]．北京：清华大学出版社，2020．

[4]李运金．基于B/S模式的研究生管理信息系统设计与实现[J]．工业控制计算机，2021．

[5]占梅．基于C#面向对象程序设计的封装,继承和多态分析[J]．电子技术与软件工程，2021．

[6]赵健．基于数据挖掘的研究生信息管理系统设计[J]．信息记录材料，2023．

# 附录1团队开发会议记录1

研究生信息管理系统开发团队会议记录1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 开发小组 | | 研究生信息管理系统开发团队 | 指导教师 | 吴秀芹 |
| 团队组长 | | 陈勇 | 记录人员 | 陈勇 |
| 会议主题 | | 项目开发背景调研及基本功能模块设计 | 会议地点 | 1C-105 |
| 会议时间 | | 2023-07-03 | 记录时间 | 2023-07-03 |
| 团队组员 | | 冷娟、赵锐琪 | 记录形式 | 文档 |
| 会议内容概要 | | | | |
| 1 | 项目开发背景调研 | | | |
| 2 | 系统基本功能模块设计 | | | |
| 3 | 系统开发进度安排 | | | |
| 会议详情 | | | | |
| 1 | 项目开发背景调研：  通过文献检索、网络查询等方式，对项目开发背景进行调研。通过本系统的开发能改善研究生信息管理与查询，降低管理成本，应对庞大的信息量，能提高学生管理工作的效率。研究生管理系统将使研究生信息管理与查询工作规范化、系统化、程序化，避免研究生信息管理的随意性，提高信息处理的速度、准确性和保密性，能够准确、及时、有效的查询、更新和维护研究生的信息情况。 | | | |
| 2 | 系统基本功能模块设计：  （1）系统登录模块：完成用户的登录操作。  （2）用户管理模块：完成用户的信息管理。  （3）专业信息管理模块：完成专业的信息管理。  （4）课程信息管理模块：完成课程的信息管理。  （5）研究生信息管理模块：完成研究生的信息管理。  （6）成绩信息管理模块：完成成绩的信息管理。  （7）导师信息管理模块：完成导师的信息管理。 | | | |
| 3 | 系统开发进度安排：  （1）系统需求分析与功能模块设计：8学时  （2）数据库表结构设计及开发：4学时  （3）架构设计及公共类开发：8学时  （4）系统编码实现：28学时  （5）系统测试：8学时  （6）文档撰写：24学时 | | | |
| 小组沟通情况总结 | | | | |
| 1 | 项目成员沟通顺畅，充分了解彼此想法，有利于该项目的设计开发 | | | |
| 2 | 个人项目成员对讨论内容仍有疑问，经过详细沟通后，已明确思路，完成设计 | | | |
| 纪律情况总结 | | | | |
| 1 | 团队无人迟到早退，按时参加会议 | | | |
| 2 | 团队成员服从领导和安排 | | | |

# 附录2团队开发会议记录2

研究生信息管理系统开发团队会议记录2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 开发小组 | | 研究生信息管理系统开发团队 | 指导教师 | 吴秀芹 |
| 团队组长 | | 陈勇 | 记录人员 | 赵锐琪 |
| 会议主题 | | 项目开发关键技术交流及功能代码测试 | 会议地点 | 1C-105 |
| 会议时间 | | 2023-07-11 | 记录时间 | 2023-07-11 |
| 团队组员 | | 冷娟、赵锐琪 | 记录形式 | 文档 |
| 会议内容概要 | | | | |
| 1 | 项目开发关键技术交流 | | | |
| 2 | 系统功能模块测试 | | | |
| 会议详情 | | | | |
| 1 | 项目开发关键技术交流：  对系统开发过程中遇到了关键编码问题进行交流：（1）页面显示与布局方面；（2）系统用户注册及修改密码；（3）系统安全性：分析问题的可出现点及具体解决方案，使系统尽可能的容错；（4）系统可用性：符合普遍的研究生信息管理流程，并尽可能的容差，以求最大化的融入市场；（5）系统易用性：操作方式简而易懂，易被使用者接纳。 | | | |
| 2 | 系统功能代码测试：  （1）陈勇：学生管理模块添加时数据插入失败，需要调整  （2）冷娟：课程管理模块更改时密码更改失败，需要调整 （3）赵锐琪：成绩模块结算时数据计算错误，需要调整  （4）赵锐琪：成绩修改模块结算时数据计算错误，需要调整 （5）冷娟：党员管理模块更改信息时更改失败，需要调整  （6）陈勇：论文模块上传时失败，需要调整 | | | |
| 小组沟通情况总结 | | | | |
| 1 | 项目成员沟通顺畅，对于关键技术实现方法的交流让大家拓宽了自己程序开发的思路 | | | |
| 2 | 个人项目成员对讨论内容仍有疑问，经过详细沟通后，能够正视自己项目开发的问题，并积极进行改正 | | | |
| 纪律情况总结 | | | | |
| 1 | 团队无人迟到早退，按时参加会议 | | | |
| 2 | 团队成员服从领导和安排 | | | |