

数据库课程设计

**专业班级：计算机1302**

**题目：教室管理系统**

**姓名：曾彪**

**学号：3130602050**

**指导老师：朱玉全**

2015年 7 月 9 日

**目录**

[1.任务要求 1](#_Toc424300951)

[2 . 需求分析 1](#_Toc424300952)

[2.1 项目背景 1](#_Toc424300953)

[2.2学校的教室使用情况 1](#_Toc424300954)

[2.3用户要求 2](#_Toc424300955)

[2.4开发工具: 3](#_Toc424300956)

[3.概念结构设计 4](#_Toc424300957)

[3.1 E-R图 4](#_Toc424300958)

[3.2 数据字典 6](#_Toc424300959)

[4. SQL数据库的建立 7](#_Toc424300960)

[4.1 建立教室管理系统的库 7](#_Toc424300961)

[4.2 建立基本表 7](#_Toc424300962)

[4.3 建立触发器 8](#_Toc424300963)

[4.3.1插入教室申请记录： 8](#_Toc424300964)

[4.3.2更新教室申请的审核状态： 9](#_Toc424300965)

[4.3.3删除教室申请记录： 11](#_Toc424300966)

[4.3.4插入教室记录： 12](#_Toc424300967)

[4.3.5更新教室记录： 12](#_Toc424300968)

[5.用户操作界面设计 13](#_Toc424300969)

[5.1 注册界面 14](#_Toc424300970)

[5.2登录界面 16](#_Toc424300971)

[5.3 管理员操作界面 20](#_Toc424300972)

[5.4 学生操作界面 28](#_Toc424300973)

[5.5 负责人(后勤人员)操作界面 32](#_Toc424300974)

[6.课程设计总结 34](#_Toc424300975)

# 1.任务要求

《数据库系统原理》课程具有实践性强的特点，学生必须在一个实际的RDBMS软件平台上创建数据库及其应用系统，给出数据库设计各个阶段的详细设计报告，写出系统的主要功能和使用说明书，提交运行的系统。课程设计可以进一步深化学生对书本知识的理解和认识，使学生掌握数据库系统设计技能，培养自主学习能力、动手能力和团队合作精神。本次个人的设计任务是设计一个教室管理系统，以实现不同用户对教室的管理和使用目的。

# 2 . 需求分析

## 2.1 项目背景

本系统的用户面向在校学生、教师、管理人员、后勤人员（负责人），我根据从学校方面取得的对教室的使用情况，存在的问题等信息，根据我们日常生活中的生活经验，根据们所做的对在校师生的询问和调查，得出教室的下列实际情况：

## 2.2学校的教室使用情况

与教室管理相关的学校的成分有：学生、教师和教室。学校的所有日常工作都是围绕着这三大部分进行的。

每个教室在不同的时段可以有不同的使用情况，可以被不同的用户使用，教室主要有上课和自习两种用途，同一个时段只能有一个用途，如果已被申请上课，则不可以再自习，如果被申请为自习，则不可以在这个时段上课，但是只要余量不为0，就还可以接受自习申请。

## 2.3用户要求

(1)基本信息要求**：**

1. 教师信息

教师的基本信息(教师的教师编号，教师姓名，性别，职称,密码，电话)；

B、学生信息

学生的基本信息基本信息(学生的编号，姓名，性别，职称,密码，电话)；

1. 管理员信息

管理员基本信息(编号，姓名，性别，职称,密码，电话)；

D、负责人信息

负责人基本信息(编号，姓名，性别，职称,密码，电话)；

E、教室信息

教室基本信息(教室编号，楼栋，房间号，容量,损坏状况，负责人编号)；

F、教室申请情况

申请情况信息（教室编号，周次，星期，时段，申请人编号，用途，申请状态）

G、教室已使用情况

使用情况信息（教室编号，周次，星期，时段，用途，余量）；

(2)处理信息要求：

为了能够妥善管理教室信息，使得教室的利用效率提高，完善目前学校教室管理统的不足，充分满足教师、学生的要求，必须对教室的借用规则、使用情况做出明确的规定与清晰地显示。

1. 教师登录：

通过这个功能，可以修改相关教师的个人信息以及查看可以使用的教室，申请教室，查看已申请教室和申请通过情况，以便能更好地使用教室。

1. 学生登录：

通过这个功能，可以修改相关学生的个人信息以及查看任意时段可以使用的教室，申请教室，查看已申请教室和申请通过情况，以便能更好地使用教室。

1. 管理员登录

通过这个功能，可以管理所有的用户信息，增加、删除、修改等，可以管理所有的教室信息，增加、删除、修改教室信息，可以管理已使用的教室，可以管理教室的申请，通过或拒绝教室、学生提交的教室申请等

1. 负责人登录

通过这个功能，可以查询到自己负责的教室信息，查看哪些教室损坏了，当教室修复之后可以更改该教室的损坏情况。

(3)安全性要求：

a、访问用户的标识唯一以鉴别是否是合法用户，并要求合法用户设置其密码，保证用户身份不被盗用；

b、不同的数据设置不同的访问级别，限制访问用户可查询和处理数据的类别和内容；

c、不同用户设置不同的权限，区分不同的用户，如学生、教师和教务处管理人员。学生（只能查询教室、教师和课程信息以及提出借用教室申请），教师（只能查询教室、教师和课程信息，以及提出借用教室申请），管理人员（可进行日常事务的处理，如增加、删除、更新课程、教室、教师信息，并批准或拒绝借用教室的申请）。

(4)完整性要求

a、各种信息记录的完整性，信息记录内容根据实际情况判断是否能为空；

b、各种数据间相互的联系的正确性(外键、主键等的正确设置)；

c、相同的数据在不同记录中的一致性（数据连接过程中合理使用表间的联系元素）。

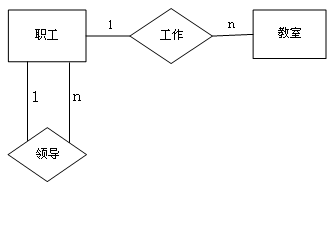
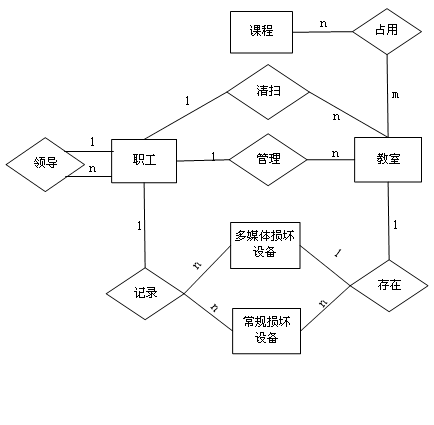
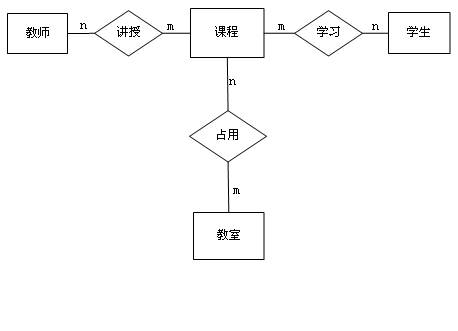
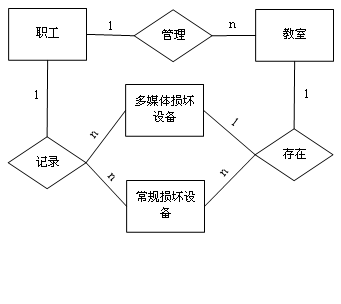
## 2.4开发工具:

Microsoft Visual Studio 2013，

Microsoft SQL Server 2008 SQL (Server Management Studio Express)

# 3.概念结构设计

## 3.1 E-R图



## 3.2 数据字典

**用户数据字典(people)：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 取值范围 | 备注 |
| 编号 | no | varchar | 10 |  | 主码 |
| 姓名 | name | varchar | 10 |  | not null |
| 性别 | sex | varchar | 5 | 男，女 | not null |
| 职务 | job | varchar | 10 | teacher,student，manage，admini | not null |
| 密码 | keyy | varchar | 15 |  | not null |
| 电话 | tel | varchar | 12 |  |  |

**教室数据字典(cr)：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 取值范围 | 备注 |
| 教室编号 | no | varchar | 10 |  | 主码 |
| 楼栋 | building | varchar | 10 |  | unique,  not null |
| 房间号 | room | varchar | 5 |  |
| 容量 | capacity | int |  | >0 and <200 | not null |
| 损坏状况 | damage | varchar | 10 | 损坏 or NULL |  |
| 负责人编号 | manageno | varchar | 10 |  | 外码（manage） |

**教室已使用情况数据字典(cruse)：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 取值范围 | 备注 |
| 教室编号 | no | varchar | 10 | foreign key | 主码 |
| 周次 | week | smallint |  | 1~20 |
| 星期 | weekday | smallint |  | 1~7 |
| 时段 | period | varchar | 10 | 上午,下午,晚上 |
| 用途 | usee | varchar | 10 | 上课，自习 | not null |
| 余量 | rest | int |  | 0~200 | not null |

**教室申请情况数据字典(crapply)：**

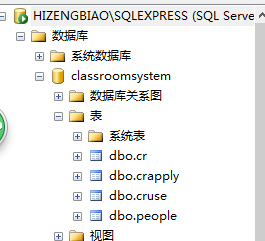
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 取值范围 | 备注 |
| 教室编号 | no | varchar | 10 | foreign key | 主码 |
| 周次 | week | smallint |  | 1~20 |
| 星期 | weekday | smallint |  | 1~7 |
| 时段 | period | varchar | 10 | 上午,下午,晚上 |
| 申请人编号 | applyno | varchar | 10 | check in(teacher or student) |
| 用途 | usee | varchar | 10 | 上课，自习 | not null |
| 申请状态 | condition | varchar | 10 | 通过，等待，拒绝 | not null |

# 4. SQL数据库的建立

## 4.1 建立教室管理系统的库

create database classroomsystem;

## 4.2 建立基本表



用户表：

create table people

(no varchar(10) primary key,

name varchar(10) not null,

sex varchar(5) not null check(sex in('男','女')),

job varchar(10) check(job in('teacher','student','manage','admini')),

keyy varchar(15) not null,

tel varchar(12) );

教师信息表：

create table cr

(no varchar(10) primary key,

building varchar(10) not null,

room varchar(5) not null,

capacity int not null check(capacity>0 and capacity<=200),

damage varchar(10) check (damage in('损坏','未损坏')),

manageno varchar(10),

--foreign key(manageno) references manage(no),

foreign key(manageno) references people(no),

unique(building,room));

教室已使用信息：

create table cruse

(no varchar(10),

week smallint check (week >= 1 and week<=20),

weekday smallint check(weekday between 1 and 7),

period varchar(10) check(period in('上午','下午','晚上')),

usee varchar(10) check(usee in('上课','自习')),

rest int not null check(rest between 0 and 200),

primary key(no,week,weekday,period),

foreign key(no) references cr(no));

教室申请信息：

create table crapply

(no varchar(10),

week smallint check (week >= 1 and week<=20),

weekday smallint check(weekday between 1 and 7),

period varchar(10) check(period in('上午','下午','晚上')),

applyno varchar(10),

usee varchar(10) check(usee in('上课','自习')),

condition varchar(10) check(condition in('通过','等待','拒绝')),

primary key(no,week,weekday,period,applyno),

foreign key(applyno) references people(no),

foreign key(no) references cr(no));

## 4.3 建立触发器

### 4.3.1插入教室申请记录：

create trigger apply\_insert

on crapply

instead of insert

as

declare @no varchar(10),@week smallint,@weekday smallint,@period varchar(10),@applyno varchar(10),@usee varchar(10),@condition varchar(10),@ifdamage varchar(10);

select@no=no,@week=week,@weekday=weekday,@period= period ,@applyno=applyno,@usee=usee from inserted;

select @ifdamage=damage from cr where cr.no=@no;if (exists (select \* from cruse where cruse.no=@no and cruse.week=@week and cruse.weekday=@weekday and cruse.period=@period and cruse.rest= 0) or (@usee='上课' and exists (select \* from cruse where cruse.no=@no and cruse.week=@week and cruse.weekday=@weekday and cruse.period=@period)))

begin

print '该教室余量不足';

end;

else if(@ifdamage='损坏')

begin

print '该教室已损坏，不能申请！';

end;

else

insert into crapply

values(@no,@week,@weekday,@period,@applyno,@usee,'等待');

### 4.3.2更新教室申请的审核状态：

create trigger apply\_pass

on crapply

instead of update

as

if update(condition)

begin

declare @no varchar(10),@week smallint,@weekday smallint,@period varchar(10),@applyno varchar(10),@usee varchar(10),@condition varchar(10);

select @no=no,@week=week,@weekday=weekday,@period=period,@applyno=applyno,@usee=usee,@condition=condition from inserted;

if(@condition='通过')

begin

--insert into crapply

--values(@no,@week,@weekday,@period,@applyno,@usee,'通过');

update crapply

set condition='通过'

where @no=no and @week=week and @weekday=weekday and @period=period and @applyno=applyno;

if(@usee='上课')

begin

insert into cruse

values(@no,@week,@weekday,@period,@usee,0);

end;

else

begin

if(exists (select \* from cruse where cruse.no=@no and cruse.week=@week and cruse.weekday=@weekday and cruse.period=@period))

begin

update cruse

set rest=rest-1

where @no=no and @week=week and @weekday=weekday and @period=period;

end

else

begin

declare @nn int;

select @nn=capacity from cr where @no=cr.no;

insert into cruse

values(@no,@week,@weekday,@period,@usee,@nn-1);

end;

end;

end;

if(@condition='拒绝')

begin

update crapply

set condition='拒绝'

where @no=no and @week=week and @weekday=weekday and @period=period and @applyno=applyno;

if(exists(select \* from cruse where cruse.no=@no and cruse.week=@week and cruse.weekday=@weekday and cruse.period=@period))

begin

if(@usee='上课')

begin

delete from cruse where @no=no and @week=week and @weekday=weekday and @period=period;

end;

else

begin

declare @nn2 int,@nn3 int;

select @nn2=capacity from cr where @no=cr.no;

select @nn3=rest from cruse where @no=no and @week=week and @weekday=weekday and @period=period;

set @nn3+=1;

if(@nn3=@nn2)

begin

delete from cruse where @no=no and @week=week and @weekday=weekday and @period=period;

end;

else

begin

update cruse

set rest=@nn3

where @no=no and @week=week and @weekday=weekday and @period=period;

end;

end;

end;

end;

end;

### 4.3.3删除教室申请记录：

create trigger apply\_delete

on crapply

for delete

as

begin

declare @no varchar(10),@week smallint,@weekday smallint,@period varchar(10);

select @no=no,@week=week,@weekday=weekday,@period=period from deleted;

begin

if(exists(select \* from cruse where cruse.no=@no and cruse.week=@week and cruse.weekday=@weekday and cruse.period=@period))

begin

declare @usee varchar(10);

select @usee=usee from cruse where cruse.no=@no and cruse.week=@week and cruse.weekday=@weekday and cruse.period=@period;

if(@usee='上课')

begin

delete from cruse where @no=no and @week=week and @weekday=weekday and @period=period;

end;

else

begin

declare @nn2 int,@nn3 int;

select @nn2=capacity from cr where @no=cr.no;

select @nn3=rest from cruse where @no=no and @week=week and @weekday=weekday and @period=period;

set @nn3+=1;

if(@nn3=@nn2)

begin

delete from cruse where @no=no and @week=week and @weekday=weekday and @period=period;

end;

else

begin

update cruse

set rest=@nn3

where @no=no and @week=week and @weekday=weekday and @period=period;

end;

end;

end;

end;

end

### 4.3.4插入教室记录：

create trigger cr\_insert

on cr

instead of insert

as

declare @no varchar(10),@building varchar(10),@room varchar(5),@capacity int,@damage varchar(10),@manageno varchar(10),@job varchar(10);

select @no=no,@building=building,@room=room,@capacity=capacity,@damage=damage,@manageno=manageno from inserted;

select @job=job from people where people.no=@manageno

if (@job='manage')

begin

insert into cr

values(@no,@building,@room,@capacity,@damage,@manageno);

end;

### 4.3.5更新教室记录：

create trigger cr\_update

on cr

instead of update

as

declare @no varchar(10),@building varchar(10),@room varchar(5),@capacity int,@damage varchar(10),@manageno varchar(10),@job varchar(10);

select @no=no from deleted;

select @building=building,@room=room,@capacity=capacity,@damage=damage,@manageno=manageno from deleted;

select @job=job from people where people.no=@manageno

if (@job='manage')

begin

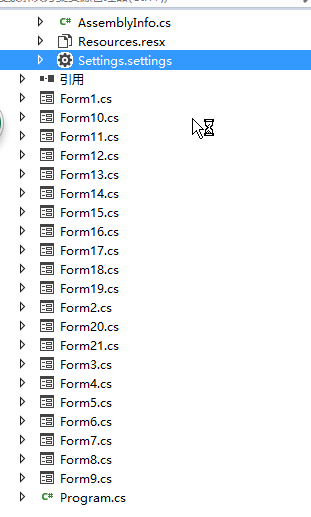
update cr

set building=@building,room=@room,capacity=@capacity,damage=@damage,manageno=@manageno

where @no=no;

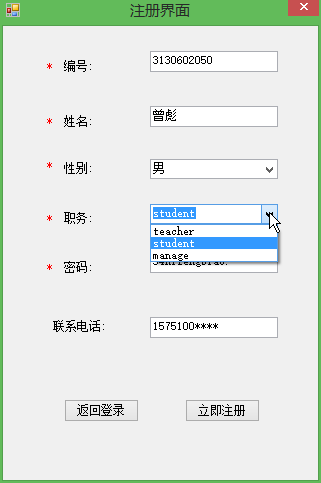
end;

# 5.用户操作界面设计



限于篇幅，以下仅展示部分有代表性的界面和代码

## 5.1 注册界面



using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace ClassRoomSystem

{

publicpartialclass注册界面 : Form

{

public 注册界面()

{

InitializeComponent();

dbConnection = newSqlConnection(ConnectString);

lo\_cmd = newSqlCommand(); //创建命令对象

lo\_cmd.Connection = dbConnection;

}

string ConnectString = "Data Source=HIZENGBIAO\\SQLEXPRESS;Initial Catalog=classroomsystem;User ID=sa;password=54hizengbiao";

SqlConnection dbConnection;

SqlCommand lo\_cmd;

privatevoid button1\_Click(object sender, EventArgs e)

{

try

{

dbConnection.Open();

string no = Convert.ToString(textBox1.Text);

string name = Convert.ToString(textBox2.Text);

string sex = Convert.ToString(comboBox1.Text);

string job = Convert.ToString(comboBox2.Text);

string keyy = Convert.ToString(textBox3.Text);

string tel = Convert.ToString(textBox4.Text);

string sql = String.Format("insert into [people](no,name,sex,job,keyy,tel)" + "values('{0}','{1}','{2}','{3}','{4}','{5}')", no, name, sex, job, keyy, tel);

lo\_cmd.CommandText = sql;

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0 )

{

MessageBox.Show("恭喜您注册成功！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("对不起，注册失败！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

}

privatevoid comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox1.Text = comboBox1.SelectedItem.ToString();

}

privatevoid comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

comboBox2.Text = comboBox2.SelectedItem.ToString();

}

privatevoid button2\_Click(object sender, EventArgs e)

{

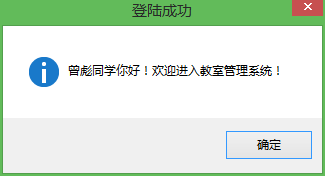
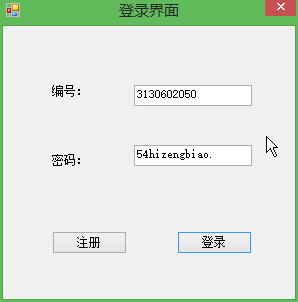
this.Close();

}

}

}

## 5.2登录界面



namespace ClassRoomSystem

{

publicpartialclass登录界面 : Form

{

public 登录界面()

{

InitializeComponent();

dbConnection = newSqlConnection(ConnectString);

//dbConnection = new System.Data.SqlClient.SqlConnection(ConnectString);

lo\_cmd = newSqlCommand(); //创建命令对象

lo\_cmd2 = newSqlCommand(); //创建命令对象

lo\_cmd3 = newSqlCommand(); //创建命令对象

lo\_cmd.Connection = dbConnection;

lo\_cmd2.Connection = dbConnection;

lo\_cmd3.Connection = dbConnection;

}

privatevoid 注册\_Click(object sender, EventArgs e)

{

// this.Close();

注册界面 mainForm = new注册界面();

mainForm.Show();

}

string ConnectString = "Data Source=HIZENGBIAO\\SQLEXPRESS;Initial Catalog=classroomsystem;User ID=sa;password=54hizengbiao";

SqlConnection dbConnection;

SqlCommand lo\_cmd;

SqlCommand lo\_cmd2;

SqlCommand lo\_cmd3;

privatevoid 登录\_Click(object sender, EventArgs e)

{

try

{

dbConnection.Open();

string UserName = textBox1.Text;

string PassWord = textBox2.Text;

string keyyy = null;

string jobb = null;

string namee = null;

lo\_cmd.CommandText = string.Format("select keyy from people where no={0};", UserName);

SqlDataAdapter dbAdapter = newSqlDataAdapter(lo\_cmd);

DataSet ds = newDataSet(); //创建数据集对象

dbAdapter.Fill(ds);

keyyy = ds.Tables[0].Rows[0][0].ToString();

lo\_cmd2.CommandText = string.Format("select job from people where no={0};", UserName);

SqlDataAdapter dbAdapter2 = newSqlDataAdapter(lo\_cmd2);

DataSet ds2 = newDataSet(); //创建数据集对象

dbAdapter2.Fill(ds2);

jobb = ds2.Tables[0].Rows[0][0].ToString();

lo\_cmd3.CommandText = string.Format("select name from people where no={0};", UserName);

SqlDataAdapter dbAdapter3 = newSqlDataAdapter(lo\_cmd3);

DataSet ds3 = newDataSet(); //创建数据集对象

dbAdapter3.Fill(ds3);

namee = ds3.Tables[0].Rows[0][0].ToString();

if (PassWord == keyyy)

{

if (/\*jobb[0]=='s' \*/jobb=="student")

{

string message;

message = string.Format("{0}同学你好！欢迎进入教室管理系统！", namee);

MessageBox.Show(message, "登陆成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

Form5 mainForm = newForm5(UserName,namee);

mainForm.Show();

// this.Hide();

}

elseif (jobb[0] =='t')

{

string message;

message = string.Format("{0}老师你好！欢迎进入教室管理系统！", namee);

MessageBox.Show(message, "登陆成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

Form16 mainForm = newForm16(UserName, namee);

mainForm.Show();

// this.Hide();

}

elseif (jobb[0] == 'm')

{

string message;

message = string.Format("{0}你好！欢迎进入教室管理系统！", namee);

MessageBox.Show(message, "登陆成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

Form20 mainForm = newForm20(UserName, namee);

mainForm.Show();

// this.Hide();

}

else

//if(jobb=="admini ")

{

/

string message;

message = string.Format("尊敬的管理员{0}欢迎进入教室管理系统！", namee);

MessageBox.Show(message, "登陆成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

Form3 mainForm = newForm3();

mainForm.Show();

// this.Hide();

}

}

else

{

MessageBox.Show("您输入的用户名或密码错误！", "登录失败", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

}

catch (Exception i)

{

MessageBox.Show("您输入的用户名或密码错误！", "登录失败", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

MessageBox.Show(i.Message);

}

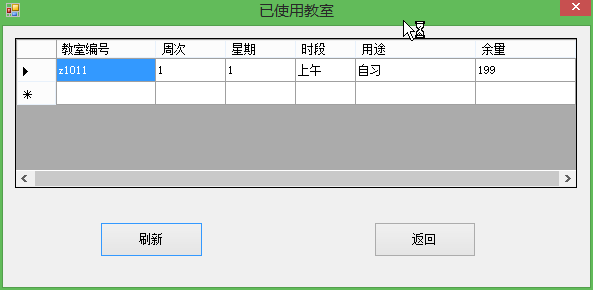
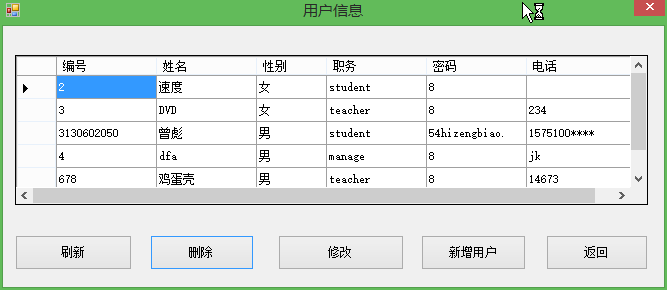
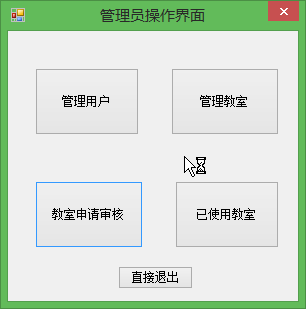
dbConnection.Close();

}

}

}

## 5.3 管理员操作界面



publicpartialclassForm4 : Form

{

public Form4()

{

InitializeComponent();

dbConnection = newSqlConnection(ConnectString);

lo\_cmd = newSqlCommand(); //创建命令对象

lo\_cmd.Connection = dbConnection;

ShowDeptResult();

}

string ConnectString = "Data Source=HIZENGBIAO\\SQLEXPRESS;Initial Catalog=classroomsystem;User ID=sa;password=54hizengbiao";

SqlConnection dbConnection;

SqlCommand lo\_cmd;

publicvoid ShowDeptResult()

{

dbConnection.Open(); //打开数据库连接

string sql = "select \* from people where job not like 'admin%';";

//返回查询结果

DataSet ds = newDataSet();

SqlDataAdapter dataAdpt = newSqlDataAdapter(sql,dbConnection); //执行查询

dataAdpt.Fill(ds);

dataGridView1.Rows.Clear(); //清空原有datagridview列表

//将SQL的查询结果显示在datagridview中

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

{

dataGridView1.Rows.Add(/\*i + 1, \*/ds.Tables[0].Rows[i][0], ds.Tables[0].Rows[i][1], ds.Tables[0].Rows[i][2], ds.Tables[0].Rows[i][3], ds.Tables[0].Rows[i][4], ds.Tables[0].Rows[i][5]);

}

dbConnection.Close();

}

string value3 = "";

string value1 = "";

string value2 = "";

string value4 = "";

string value5 = "";

string value6 = "";

privatevoid dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

//获得当前选中的行

int rowindex = e.RowIndex;

try

{

//获得当前行的第一列的值

value1 = dataGridView1.Rows[rowindex].Cells[0].Value.ToString().Trim();

//获得当前行的第二列的值

value2 = dataGridView1.Rows[rowindex].Cells[1].Value.ToString().Trim();

//获得当前行的第3列的值

value3 = dataGridView1.Rows[rowindex].Cells[2].Value.ToString().Trim();

//获得当前行的第4列的值

value4 = dataGridView1.Rows[rowindex].Cells[3].Value.ToString().Trim();

//获得当前行的第5列的值

value5 = dataGridView1.Rows[rowindex].Cells[4].Value.ToString().Trim();

//获得当前行的第6列的值

value6 = dataGridView1.Rows[rowindex].Cells[5].Value.ToString().Trim();

}

catch/\* (Exception exc)\*/ { /\*MessageBox.Show(exc.Message); \*/}

}

privatevoid button1\_Click(object sender, EventArgs e)

{

ShowDeptResult();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

Form9 mainForm = newForm9(value1,value2,value3,value4,value5,value6);

mainForm.Show();

// ShowDeptResult();

}

privatevoid button2\_Click(object sender, EventArgs e)

{

try

{

dbConnection.Open();

lo\_cmd.CommandText = string.Format("delete from people where no='{0}';", value1);

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("删除信息成功！", "删除成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("删除信息失败！", "删除失败", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

ShowDeptResult();

}

privatevoid button4\_Click(object sender, EventArgs e)

{

Form8 mainForm = newForm8();

mainForm.Show();

}

privatevoid button5\_Click(object sender, EventArgs e)

{

Form3 mainForm = newForm3();

mainForm.Show();

this.Close();

}

}

privatevoid button2\_Click(object sender, EventArgs e)

{

try

{

dbConnection.Open();

lo\_cmd.CommandText = string.Format("delete from crapply where no='{0}' and week={1} and weekday={2} and period='{3}' and applyno='{4}';", value1,value22,value33,value4,value5);

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("删除信息成功！", "删除成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("删除信息失败！", "删除失败", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

ShowDeptResult();

}

privatevoid button3\_Click(object sender, EventArgs e)

{

string sql;

if (value7 == "等待" || value7 == "拒绝")

{

sql = string.Format("update crapply set condition='通过' where no='{0}' and week={1} and weekday={2} and period='{3}' and applyno='{4}';", value1, value22, value33, value4, value5);

try

{

dbConnection.Open();

lo\_cmd.CommandText = sql;

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("审核通过！", "审核通过", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("审核失败！", "审核失败", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

ShowDeptResult();

}

elseif (value7 == "通过")

{ MessageBox.Show("该条记录已经审核通过，无需更改！"); }

else

{

MessageBox.Show("请选中一条申请记录后再进行操作！"); }

}

privatevoid button4\_Click(object sender, EventArgs e)

{

string sql;

if (value7 == "等待" || value7 == "通过")

{

sql = string.Format("update crapply set condition='拒绝' where no='{0}' and week={1} and weekday={2} and period='{3}' and applyno='{4}';", value1, value22, value33, value4, value5);

try

{

dbConnection.Open();

lo\_cmd.CommandText = sql;

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("拒绝成功！", "审核通过", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("拒绝失败！", "审核失败", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

ShowDeptResult();

}

elseif (value7 == "拒绝")

{ MessageBox.Show("该条记录已经拒绝，无需更改！"); }

else

{ MessageBox.Show("请选中一条申请记录后再进行操作！"); }

}

publicvoid ShowDeptResult()

{

dbConnection.Open(); //打开数据库连接

string sql = "select \* from cruse;";

//返回查询结果

DataSet ds = newDataSet();

SqlDataAdapter dataAdpt = newSqlDataAdapter(sql, dbConnection); //执行查询

dataAdpt.Fill(ds);

dataGridView1.Rows.Clear(); //清空原有datagridview列表

//将SQL的查询结果显示在datagridview中

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

{

dataGridView1.Rows.Add(/\*i + 1, \*/ds.Tables[0].Rows[i][0], ds.Tables[0].Rows[i][1], ds.Tables[0].Rows[i][2], ds.Tables[0].Rows[i][3], ds.Tables[0].Rows[i][4], ds.Tables[0].Rows[i][5]);

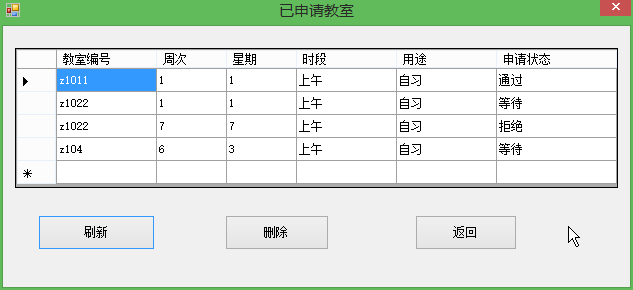
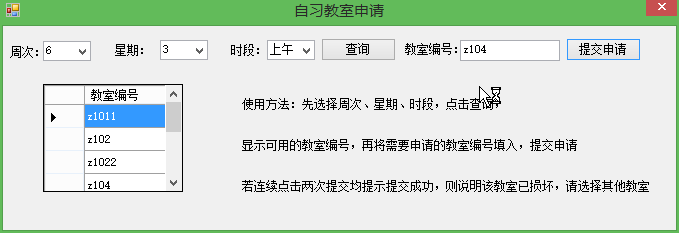
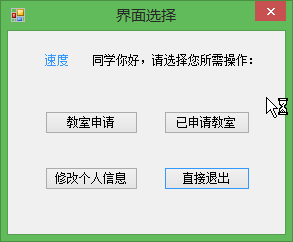
}

dbConnection.Close();

}

5.4 学生操作界面

（教室界面和学生的类似，不再列举）



privatevoid Form13\_Load(object sender, EventArgs e)

{

textBox1.Text = value1;

textBox2.Text = value2;

comboBox1.Text = value3;

textBox5.Text = value5;

textBox6.Text = value6;

}

privatevoid button1\_Click(object sender, EventArgs e)

{

try

{

string sql1;

dbConnection.Open();

sql1 = string.Format("update people set no='{0}',name='{1}',sex='{2}',job='{3}',keyy='{4}',tel='{5}' where no='{6}';", textBox1.Text, textBox2.Text, comboBox1.Text, value4, textBox5.Text, textBox6.Text, no);

lo\_cmd1.CommandText = sql1;

int count = lo\_cmd1.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("修改信息成功！", "修改成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

if (textBox1.Text != value1 || textBox5.Text != value5)

{

this.Close();

MessageBox.Show("您已修改了用户名或密码，请重新登录！");

//登录界面 mainForm = new 登录界面();

//mainForm.Show();

}

else

{

Form5 mainForm2 = newForm5(textBox1.Text, textBox2.Text);

this.Close();

mainForm2.Show();

}

}

else

{

MessageBox.Show("修改信息失败！", "修改失败", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

}

publicvoid ShowDeptResult()

{

int a, b;

a=Convert.ToInt32(comboBox1.Text);

b=Convert.ToInt32(comboBox2.Text);

dbConnection.Open(); //打开数据库连接

string sql = String.Format("select no from cr where no not in (select no from cruse where cruse.rest=0 and cruse.week={0} and cruse.weekday={1} and cruse.period ='{2}') and no not in(select no from crapply where crapply.week={0} and crapply.weekday={1} and crapply.period ='{2}' and crapply.applyno='{3}');", a, b, comboBox3.Text,no);

// string sql2 = "select no from cr where no not like 'admin%';";

//返回查询结果

DataSet ds = newDataSet();

SqlDataAdapter dataAdpt = newSqlDataAdapter(sql, dbConnection); //执行查询

dataAdpt.Fill(ds);

dataGridView1.Rows.Clear(); //清空原有datagridview列表

//将SQL的查询结果显示在datagridview中

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

{

dataGridView1.Rows.Add(/\*i + 1, \*/ds.Tables[0].Rows[i][0]);

}

dbConnection.Close();

}

privatevoid button2\_Click(object sender, EventArgs e)

{

ShowDeptResult();

}

privatevoid button1\_Click(object sender, EventArgs e)

{

try

{

dbConnection.Open();

string sql = String.Format("insert into [crapply](no,week,weekday,period,applyno,usee,condition)" + "values('{0}',{1},{2},'{3}','{4}','自习','等待')", textBox1.Text, comboBox1.Text, comboBox2.Text, comboBox3.Text, no);

lo\_cmd.CommandText = sql;

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("恭喜您提交成功！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("对不起，提交失败！", "提示", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

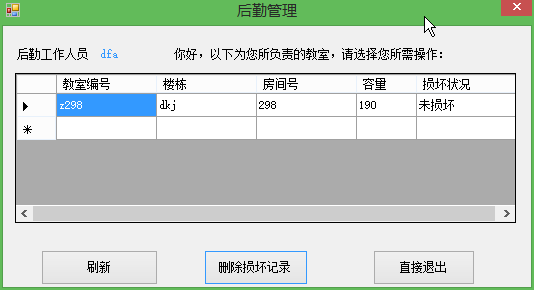
MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

}

## 5.5 负责人(后勤人员)操作界面



privatevoid button2\_Click(object sender, EventArgs e)

{

try

{

string sql1;

dbConnection.Open();

sql1 = string.Format("update cr set damage='未损坏' where no='{0}';", value1);

if (value5 == "未损坏")

{

MessageBox.Show("无需修改！");

}

else

{

lo\_cmd.CommandText = sql1;

int count = lo\_cmd.ExecuteNonQuery();

if (count > 0)

{

MessageBox.Show("修改信息成功！", "修改成功", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("修改信息失败！", "修改失败", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "操作数据库出错", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

}

dbConnection.Close();

}

# 6.课程设计总结

通过这次为期一周时间的课程设计让我对数据库这一学科的很多知识得到了一次更深的理解，当初原本以为自己已经基本掌握了数据库SQL的全部知识，但当我开始着手实际操作时才发现原来过去自己所掌握的知识是那么的贫乏。

这次我们所做的课程设计题目为教室信息管理系统，，开始做表时还算顺利，但当到了后来设计界面，很多东西都不懂，为此我们去上网、图书馆参阅了很多资料，也多次请教同学，但是因为我们的没有经验，所以设计出来的东西总是漏洞，不过在不断的调试与思考下，终于逐渐完善，界面设计也越来越娴熟，在运行出现问题的时候也能够马上推测出发生问题的大概原因。

总的来说这课程设计真的不容易，不仅需要丰富的数据库知识，严谨的思维，还需要耐心，而且之前我只学过C++，却不知道要怎么写软件的界面，从没有接触过C#，于是又特地花了几天时间学习C# ，时间确实有点不够了，不过好在C#和C++有很多相似之处，学起来并不费力，而且每当遇到不懂的地方，网上都能够找到很完美的解释，于是在很短的时间里就学会了使用C#，经过此次的课程设计不仅让我学到了很多过去漏掉的知识，而且也让我学到了很多新的知识并培养了我解决问题的能力。