疯狂的方言

组员：

林帆 计算机一班 20111003444

梁意 计算机一班 20111003441

胡建萍 计算机一班 20111003450

目录

[1 系统综述 4](#_Toc374652944)

[1.1 系统使用说明 4](#_Toc374652945)

[1.1.1 系统界面 4](#_Toc374652946)

[1.2 软件功能 11](#_Toc374652947)

[1.2.1 管理员功能 11](#_Toc374652948)

[1.2.2 用户功能 12](#_Toc374652949)

[1.2.3 功能结构图 12](#_Toc374652950)

[2 数据库设计说明 13](#_Toc374652951)

[2.1 数据字典 13](#_Toc374652952)

[3 概念模型（E-R图） 15](#_Toc374652953)

[4 关键代码 15](#_Toc374652954)

[4.1 多线程，播放背景音乐 16](#_Toc374652955)

[4.2 登录代码：判断是否登录，若登录，则判断为管理员或用户，显示首页，若未登录，则显示登录页面。 16](#_Toc374652956)

[4.3 上传头像，并显示图片 17](#_Toc374652957)

[4.4 清理面板控件，动态生成控件，动态绑定事件，使用匿名函数向事件函数传值 19](#_Toc374652958)

[4.5 在子窗口关闭的同时，刷新父窗口 21](#_Toc374652959)

[4.6 判断用户是否做过该试卷，若做过，则以红色绿色显示该题用户答案正确与否，若无做过，则实现点击单选按钮，改变题号颜色，标志该题已做过，同时记录用户答案。 22](#_Toc374652960)

[4.7 获取下拉框中选项的selectIndex并且对应到相应的数据库中的字段 26](#_Toc374652961)

[4.8 导入excel 27](#_Toc374652962)

[4.9 实现获取上级下拉框中的选项，加载相应的选项并显示在下级下拉框中 27](#_Toc374652963)

[4.10 修改密码，实现原密码验证，两次输入密码相同，并不可为空 30](#_Toc374652964)

[5 系统测试报告 31](#_Toc374652965)

[5.1 系统编写过程存在的问题： 31](#_Toc374652966)

[多线程循环播放背景音乐 31](#_Toc374652967)

[动态生成控件 31](#_Toc374652968)

[事件绑定 31](#_Toc374652969)

[5.2 如何解决： 32](#_Toc374652970)

[通过查阅资料 32](#_Toc374652971)

[单步运行调试 32](#_Toc374652972)

[5.3 最后的系统存在哪些不足： 32](#_Toc374652973)

[限制性：题库的导入文件格式限制为 xls，显得有些局限 32](#_Toc374652974)

[缺陷： 32](#_Toc374652975)

[（1）登录界面没有“记住密码”这项功能 32](#_Toc374652976)

[（2）没办法显示用户做完一道题的分数以及测试的计时 32](#_Toc374652977)

[建议： 32](#_Toc374652978)

[（1）再添加一个控件,能够使得用户可以设置为每做完一道题自动跳到下一道题，方便用户做题 32](#_Toc374652979)

[（2）按顺序摆放控件，以使可以通过TAB键跳到下一空格，方便用户填写资料 32](#_Toc374652980)

[（3）按下“退出登录”控件时，添加提示“确认要退出吗”窗口，以避免用户由于不小心的原因按了退出所造成的麻烦 32](#_Toc374652981)

[（4）登录之后可以直接跳转到主页面 32](#_Toc374652982)

[6 项目心得 32](#_Toc374652983)

[7 分工与合作说明 33](#_Toc374652984)

[8 分数评定 33](#_Toc374652985)

# 系统综述

## 系统使用说明

疯狂的方言系统包括：两个对象：管理员、用户。

用户功能：注册、登陆、修改密码、编制资料、选择方言类型、选择已加入方言类型试卷、做试卷、查看试卷分数、退出八个模块。

管理员功能：登陆、修改密码、编制资料、创建，编辑，修改方言类别、创建，删除试卷、查看用户资料、退出十个模块。

### 系统界面

**注册：**

****

**登录：**



**开始考试：进入选择方言类型页面。**





**点击查看考卷：**





**查看用户已做过试卷状态：**



**开始进入考试界面：**



**修改用户资料：**



**管理员首页：**



**查看方言类别：**



**查看所有试卷：**



**查看用户：**



**创建试卷：**



## 软件功能

### 管理员功能

**查看**

点击左上角导航“查看”，即可进入查看试题、用户资料，查看用户分数界面，对用户数据进行查看和修改。

**修改试题**

点击左上角导航“查看”，在下拉框选择查看试题，进入试题，对需要修改的试题进行修改并保存。

**创建试卷**

点击左上角导航“创建试卷”，即可进入创建试卷界面（见图4 创建试卷）点击“导入试卷”，选择需要导入的试卷excel表，下拉框“所属方言类别”内选择对应方言，点击“提交”，完成试卷的创建。

**修改资料**

点击左上角导航“修改资料”，即可进入资料编辑界面。

**创建方言类别**

点击左上角导航“查看”，在下拉框选择查看“方言类别”，进入方言类别界面，点击“新增”按钮，输入需要新增的类别名称，点击“确定”，完成方言类别的创建。

**修改密码**

点击左上角导航“修改资料”，进入资料编辑界面，点击“修改密码”，即可进入修改密码界面，进行密码的修改，按“确认”保存新密码。

**退出登陆**

点击左上角导航“退出登陆”，即可退出系统。

### 用户功能

选择方言语种

点击左上角导航“查看”，在下拉框选择查看方言语种，即可进入选择方言语种。

选择试卷

点击对应方言类型的查看试卷，即可选择进入试题。

做试卷

进入相应类别的试卷界面后，点击一套试卷按钮，即可进入考试阶段。

修改用户资料

点击左上角导航“修改资料”，即可进入资料编辑界面。

修改密码

点击左上角导航“修改资料”，进入资料编辑界面，点击“修改密码”，即可进入修改密码界面，进行密码的修改，按“确认”保存新密码。

退出登陆

点击左上角导航“退出登陆”，即可退出系统。

### 功能结构图



# 数据库设计说明

## 数据字典

#### 用户数据字典（user）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 用户编号 | user\_id | int |  | 系统用户编号 |
| 用户昵称 | username | string | 50 | 系统用户真实姓名 |
| 用户昵称 | userNickName | string | 50 | 系统用户昵称 |
| 用户密码 | password | string | 50 | 系统用户密码 |
| 用户身份 | identities | string | 2 | 系统用户身份 |
| 用户性别 | sex | int |  | 系统用户性别 |
| 用户生日 | birthday | timestamp |  | 系统用户生日 |
| 用户qq | qq | string | 20 | 系统用户qq |
| 用户邮箱 | email | string | 50 | 系统用户邮箱 |
| 用户MSM | MSM | string | 50 | 系统用户MSM |
| 用户相片 | picture | image |  | 系统用户相片 |

#### 方言类型数据字典（class）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 方言类型编号 | class\_id | int |  | 方言类型编号 |
| 方言类型名称 | class | string | 50 | 方言类型名称 |
| 管理员编号 | teacher\_id | int |  | 管理员编号 |
| 创建时间 | regist\_date | timestamp |  | 班级创建时间 |

#### 用户选择加入方言类型数据字典（stu\_class）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 班级编号 | class\_id | int |  | 班级编号 |
| 用户编号 | stu\_id | int |  | 用户编号 |
| 加入时间 | join\_date | timestamp |  | 用户加入该方言类型的时间 |

#### 试卷数据字典（exam）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 试卷编号 | exam\_id | int |  | 试卷编号 |
| 试卷名称 | exam | string | 50 | 试卷名称 |
| 管理员编号 | teacher\_id | int |  | 管理员编号 |
| 创建时间 | create\_date | timestamp |  | 创建试卷时间 |
| 方言类型编号 | class\_id | int |  | 试卷所属方言类型编号 |

#### 题目数据字典（question）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 题目编号 | question\_id | int |  | 题目编号 |
| 题目 | question | text |  | 题目内容 |
| 试卷编号 | exam\_id | int |  | 所属试卷编号 |
| 答案 | answer | text |  | 题目的答案 |
| 选项 | option | text |  | 题目提供的选项 |

#### 成绩数据字典

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 存储代码 | 类型 | 长度 | 备注 |
| 用户编号 | student\_id | int |  | 用户编号 |
| 试卷编号 | eam\_id | int |  | 试卷编号 |
| 成绩 | score | float |  | 用户成绩 |
| 用户答案 | answers | text |  | 用户答案 |
| 完成试卷日期 | answer\_date | timestamp |  | 完成试卷日期 |

# 概念模型（E-R图）



# 关键代码

## 多线程，播放背景音乐

private void axWindowsMediaPlayer1\_PlayStateChange(object sender,

AxWMPLib.\_WMPOCXEvents\_PlayStateChangeEvent e)

{

if (axWindowsMediaPlayer1.playState ==

WMPLib.WMPPlayState.wmppsMediaEnded)

{

Thread thread = new Thread(new ThreadStart(PlayThread));

thread.Start();

}

}

private void PlayThread()

{

axWindowsMediaPlayer1.URL = "song.mp3";

axWindowsMediaPlayer1.Ctlcontrols.play();

}

## 登录代码：判断是否登录，若登录，则判断为管理员或用户，显示首页，若未登录，则显示登录页面。

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

{

string username = txtUserName.Text.Trim();

string password = txtPassword.Text.Trim();

bool identity = (radioStu.Checked || radioTea.Checked);

string table = radioStu.Checked ? "0" : "1";

if (username == "" || password == "" || !identity)

{

MessageBox.Show("用户名,密码以及身份不能为空！");

txtUserName.Focus();

return;

}

else

{

BLL.User user = new BLL.User();

DataTable dt = user.Login(username, password, table);

if (dt.Rows.Count > 0)

{

UserHelper.userID = int.Parse(dt.Rows[0]["user\_id"].ToString());

UserHelper.userNickName = (dt.Rows[0]["userNickName"].ToString());

UserHelper.password = password;

UserHelper.identity = table;

MessageBox.Show("登陆成功！");

showIndex();

}

else

{

MessageBox.Show("用户名或密码错误，请重新输入！", "错误");

txtUserName.Text = "";

txtPassword.Text = "";

txtUserName.Focus();

}

}

}

## 上传头像，并显示图片

public void ShowPhoto(DataRow info)

{

if ((picStu.Image != null))

{

picStu.Image.Dispose();

picStu.Image = null;

}

if (!(info["picture"] == System.DBNull.Value))

{

byte[] BBytes = null;

BBytes = (byte[])info["picture"];

if (BBytes.LongLength > 0)

{

System.IO.MemoryStream Mystrm = new

System.IO.MemoryStream(BBytes);

Bitmap Bmp = new Bitmap(Mystrm);

picStu.Image = Bmp;

}

}

}

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

{

OpenFileDialog fd = new OpenFileDialog();

if (fd.ShowDialog() == DialogResult.OK)

{

byte[] B = user.uploadPic(UserHelper.userID, fd.FileName);

this.userInfo["picture"] = B;

ShowPhoto(this.userInfo);

MessageBox.Show("图片写入成功");

}

}

## 清理面板控件，动态生成控件，动态绑定事件，使用匿名函数向事件函数传值

//清理控件

for (int i = flowLayoutPanel2.Controls.Count - 1; i >= 0; i--)

{

Control con = flowLayoutPanel2.Controls[i];

if (con is Label || con is LinkLabel)

{

flowLayoutPanel2.Controls.RemoveAt(i);

}

}

for (int i = flowLayoutPanel3.Controls.Count - 1; i >= 0; i--)

{

Control con = flowLayoutPanel3.Controls[i];

if (con is Label || con is LinkLabel)

{

flowLayoutPanel3.Controls.RemoveAt(i);

}

}

//清理控件结束

DataTable dt = classes.getOwnClass(UserHelper.userID);

DataTable dtall = classes.getAllClass(dt);

for (int i = 0; i < dt.Rows.Count; i++)

{

Label label1 = new Label();

label1.Location = new Point(3, 0);

label1.Size = new Size(200, 20);

label1.Text = dt.Rows[i]["class"].ToString();

flowLayoutPanel3.Controls.Add(label1);

LinkLabel linklabel1 = new LinkLabel();

linklabel1.AutoSize = true;

linklabel1.Location = new Point(172, 0);

linklabel1.Size = new Size(170, 20);

linklabel1.Name = dt.Rows[i]["class\_id"].ToString();

//绑定事件

linklabel1.Click += new EventHandler(linklabelCancel\_Click);

linklabel1.Text = "退出";

//将控件动态加载到面板中

flowLayoutPanel3.Controls.Add(linklabel1);

LinkLabel linklabel2 = new LinkLabel();

linklabel2.AutoSize = true;

linklabel2.Location = new Point(272, 0);

linklabel2.Size = new Size(170, 20);

linklabel2.Name = dt.Rows[i]["class\_id"].ToString();

//使用匿名函数向事件函数传值

linklabel2.Click += delegate(Object o, EventArgs ee) { doSomething(linklabel2); };

linklabel2.Text = "查看考卷";

flowLayoutPanel3.Controls.Add(linklabel2);

}

for (int i = 0; i < dtall.Rows.Count; i++)

{

Label label2 = new Label();

label2.Location = new Point(3, 0);

label2.Size = new Size(250, 20);

label2.Text = dtall.Rows[i]["class"].ToString();

flowLayoutPanel2.Controls.Add(label2);

LinkLabel linklabel2 = new LinkLabel();

linklabel2.AutoSize = true;

linklabel2.Location = new Point(175, 0);

linklabel2.Size = new Size(77, 20);

linklabel2.Text = "加入";

linklabel2.Name = dtall.Rows[i]["class\_id"].ToString();

linklabel2.Click += new EventHandler(linklabelEnter\_Click);

flowLayoutPanel2.Controls.Add(linklabel2);

}

## 在子窗口关闭的同时，刷新父窗口

**用户完成试卷窗口代码：**

//刷新父级窗口

public delegate void updateParentData(object sender);

public event updateParentData updateIt;

//完成试卷提交按钮事件

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

{

if (exam.calculateScore(dt, this.\_exam\_id, UserHelper.userID))

{

MessageBox.Show("提交考卷成功");

this.Close();

if (this.updateIt != null)

{

updateIt(null);

}

}

else

{

MessageBox.Show("提交考卷失败");

}

}

**用户显示全部试卷窗口代码：**

public void label1\_Click(object sender, EventArgs e)

{

Label clickedLabel = sender as Label;

int exam\_id = int.Parse(clickedLabel.Name);

CDDoExam doexam = new CDDoExam(exam\_id, this);

doexam.updateIt += new CDDoExam.updateParentData(doSomething);

doexam.Show();

}

public void labelButton\_Click(object sender, EventArgs e)

{

Button clickedLabel = sender as Button;

int exam\_id = int.Parse(clickedLabel.Name);

CDDoExam doexam = new CDDoExam(exam\_id, this);

//给用户完成试卷窗口绑定事件，若完成则调用父窗口dosomething函数

doexam.updateIt += new CDDoExam.updateParentData(doSomething);

doexam.Show();

}

## 判断用户是否做过该试卷，若做过，则以红色绿色显示该题用户答案正确与否，若无做过，则实现点击单选按钮，改变题号颜色，标志该题已做过，同时记录用户答案。

public CDDoExam(int exam\_id, Form index, bool status = true)

{

InitializeComponent();

this.status = status;

this.\_exam\_id = exam\_id;

this.\_index = index;

//将四个单选按钮放入数组中，方便操作

radioButtons[1] = radioButton1;

radioButtons[2] = radioButton2;

radioButtons[3] = radioButton3;

radioButtons[4] = radioButton4;

//获取该试卷所有的试题，放入datatable dt中

dt = exam.getQuestion(exam\_id);

//如果该试卷已做 则获取学生答案

if (this.status == false)

{

studentAnswer = exam.getUserAnswers(UserHelper.userID, this.\_exam\_id);

}

//动态创建题目标号 label

for (int i = 0; i < dt.Rows.Count; i++)

{

Label label = new Label();

label.BackColor = SystemColors.ActiveCaption;

label.BorderStyle = BorderStyle.Fixed3D;

label.Location = new Point(30 \* i + 12, 325);

label.Size = new Size(30, 30);

label.TabIndex = 4;

label.Text = (i + 1).ToString();

label.TextAlign = ContentAlignment.MiddleCenter;

label.Click += new EventHandler(label\_Click);

if (this.status == false)

{

if (dt.Rows[i]["answer"].ToString() == studentAnswer[i])

{

label.BackColor = Color.Green;

}

else

{

label.BackColor = Color.Red;

}

}

flowLayoutPanel1.Controls.Add(label);

labels[i] = label;

this.ResumeLayout(false);

}

}

//放置当前题目的题号

private int num;

//form load完之后需要加载当前题目的选项

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

{

textBox1.DataBindings.Add(new Binding("Text", dt, "question"));

string[] optionArray = Regex.Split(dt.Rows[0]["options"].ToString(), "::",

RegexOptions.IgnoreCase);

radioButton1.Text = "A:" + optionArray[0];

radioButton2.Text = "B:" + optionArray[1];

radioButton3.Text = "C:" + optionArray[2];

radioButton4.Text = "D:" + optionArray[3];

this.BindingContext[dt].Position = 0;

num = this.BindingContext[dt].Position;

if (this.status == false)

{

for (int i = 1; i < radioButtons.Count(); i++)

{

radioButtons[i].Enabled = false;

}

btnSubmit.Hide();

}

StudentOption();

}

//题号click操作的公共函数

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

{

Label clickedLabel = sender as Label;

this.BindingContext[dt].Position = int.Parse(clickedLabel.Text)-1;

num = this.BindingContext[dt].Position;

string[] optionArray = Regex.Split(dt.Rows[num]["options"].ToString(), "::", RegexOptions.IgnoreCase);

radioButton1.Text = "A:"+optionArray[0];

radioButton2.Text = "B:" + optionArray[1];

radioButton3.Text = "C:" + optionArray[2];

radioButton4.Text = "D:" + optionArray[3];

StudentOption();

}

//radio单选click之后的公共函数

private void radio\_click(object sender, EventArgs e)

{

RadioButton clickRadio = sender as RadioButton;

labels[num].BackColor = Color.FromArgb(((int)(((byte)(255)))), ((int)(((byte)(255)))),

((int)(((byte)(128)))));

string option = clickRadio.Text.Substring(0, 1);

exam.setStudentOption(option, num);

}

private void StudentOption()

{

string[] studentOptions;

if(this.status == false)studentOptions = this.studentAnswer;

else studentOptions = exam.getStudentOption();

if (studentOptions[num] == null||(studentOptions[num] == ""))

{

for (int i = 1; i < radioButtons.Count(); i++)

{

radioButtons[i].Checked = false;

}

}

else

{

string option = studentOptions[num];

int intoption =

option=="A"?1:(option=="B"?2:(option=="C"?3:(option=="D"?4:0)));

radioButtons[intoption].Checked = true;

}

}

## 获取下拉框中选项的selectIndex并且对应到相应的数据库中的字段

private void cbClass\_SelectedIndexChanged(object sender, EventArgs e)

{

this.selectedClassID = int.Parse(teacherClass.Rows[Convert.ToInt32(this.cbClass.SelectedIndex)]["class\_id"].ToString());

}

## 导入excel

private void bind(string fileName)

{

string strConn = "Provider=Microsoft.Jet.OLEDB.4.0;" + "Data Source=" + fileName + ";" + "Extended Properties='Excel 8.0;HDR=Yes; IMEX=1'";

OleDbDataAdapter da = new OleDbDataAdapter("Select \* From [Sheet1$]", strConn);

DataSet ds = new DataSet();

try

{

da.Fill(ds, "exam");

dt = ds.Tables["exam"];

dataGridView1.DataSource = dt;

}

catch (Exception err)

{

MessageBox.Show("操作失败！" + err.ToString());

}

}

## 实现获取上级下拉框中的选项，加载相应的选项并显示在下级下拉框中

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

{

panelExam.Hide();

panelExam.Hide();

panelIndex.Hide();

panelManageIndex.Hide();

panelLogin.Hide();

panelUserInfo.Hide();

panelStuClass.Hide();

panelCheckUser.Show();

panelCheckUser.Controls.Add(menuStrip1);

showMenu();

for (int i = cbClass.Items.Count - 1; i >= 0; i--)

{

cbClass.Items.RemoveAt(i);

}

DataTable dt = classes.getClass(UserHelper.userID);

cbClassDataTabel = dt;

cbMouseLeave(cbClass, dt, Convert.ToInt32(cbClass.SelectedValue));

for (int i = 0; i < dt.Rows.Count; i++)

{

cbClass.Items.Add(dt.Rows[i]["class"].ToString());

}

cbClass.TextChanged += delegate(Object o, EventArgs ee) { cbMouseLeave(cbClass, dt, Convert.ToInt32(cbClass.SelectedValue)); };

cbClass.SelectedIndexChanged += delegate(Object o, EventArgs ee) { cbMouseLeave(cbClass, dt, Convert.ToInt32(cbClass.SelectedValue)); };

btnSelectByClass.Click += delegate(Object o, EventArgs ee) { btnSelectByClass\_Click(cbClass, Convert.ToInt32(cbClass.SelectedValue), Convert.ToInt32(cbTest.SelectedValue)); };

}

public void cbMouseLeave(object sender, DataTable cbDT, int selectNum)

{

ComboBox cb = sender as ComboBox;

DataTable dt;

for (int i = cbTest.Items.Count - 1; i >= 0; i--)

{

cbTest.Items.RemoveAt(i);

}

if (cb.Text == "")

{

dt = bllExam.getTeacherExam(UserHelper.userID);

cbExamDataTabel = dt;

for (int i = 0; i < dt.Rows.Count; i++)

{

cbTest.Items.Add(dt.Rows[i]["exam"].ToString());

}

}

else

{

int index = cbClass.FindString(cbClass.Text);

dt = bllExam.getExamByClassID(int.Parse(cbDT.Rows[index]["class\_id"].ToString()));

cbExamDataTabel = dt;

for (int i = 0; i < dt.Rows.Count; i++)

{

cbTest.Items.Add(dt.Rows[i]["exam"].ToString());

}

}

}

private void btnSelectByClass\_Click(object sender, int selectClass, int selectExam)

{

bool selectClassBool = false, selectExamBool = false;

if (cbClass.Text != "") { selectClassBool = true; }

if (cbTest.Text != "") { selectExamBool = true; }

bool selectGirl = radioGirl.Checked;

bool selectBoy = radioBoy.Checked;

bool selectAll = radioAll.Checked;

int selectClassInt = 0, selectExamInt = 0;

if (selectClassBool)

{

selectClassInt = int.Parse(cbClassDataTabel.Rows[selectClass]["class\_id"].ToString());

}

if (selectExamBool)

{

selectExamInt = int.Parse(cbExamDataTabel.Rows[selectExam]["exam\_id"].ToString());

}

DataTable dt = user.selectByClass(selectClassInt, selectExamInt, selectGirl, selectBoy, selectAll, selectClassBool, selectExamBool);

dataGridUser.DataSource = dt;

}

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

{

string username = txtUserName1.Text;

dataGridUser.DataSource = user.selectByUserName(username);

}

## 修改密码，实现原密码验证，两次输入密码相同，并不可为空

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

{

if (txtOldPass.Text == "" || txtReNewPass.Text == "" || txtNewPass.Text == "")

{

label5.Text = "以上不可为空";

}

else if (txtReNewPass.Text != txtNewPass.Text)

{

label5.Text = "两次输入新密码不同";

}

else if (!user.checkPass(UserHelper.userID, txtOldPass.Text.Trim()))

{

label5.Text = "原密码不正确";

}

else

{

user.changePass(UserHelper.userID, txtOldPass.Text.Trim(), txtReNewPass.Text.Trim());

MessageBox.Show("修改密码成功");

this.Hide();

}

}

# 系统测试报告

## 系统编写过程存在的问题：

## 多线程循环播放背景音乐

## 动态生成控件

## 事件绑定

## 如何解决：

## 通过查阅资料

## 单步运行调试

## 最后的系统存在哪些不足：

## 限制性：题库的导入文件格式限制为 xls，显得有些局限

## 缺陷：

## （1）登录界面没有“记住密码”这项功能

## （2）没办法显示用户做完一道题的分数以及测试的计时

## 建议：

## （1）再添加一个控件,能够使得用户可以设置为每做完一道题自动跳到下一道题，方便用户做题

## （2）按顺序摆放控件，以使可以通过TAB键跳到下一空格，方便用户填写资料

## （3）按下“退出登录”控件时，添加提示“确认要退出吗”窗口，以避免用户由于不小心的原因按了退出所造成的麻烦

## （4）登录之后可以直接跳转到主页面

# 项目心得

经过本次的数据库设计以及三层架构的运用，深深体会到要做好一个数据库系统的管理是那么的不容易。既要对需求进行详细的分析，又要对每一个table的属性进行严谨的思考，还要细致地把E-R图画出来，本数据库系统由于功能的需要，还需要利用HTML、PHP、JS以及SQL方面的知识来实现试题的上传、显示，界面的美化以及自动评分等功能。设计过程中，细心是最重要的，对属性的命名，修改的时候要特别注意把其他表格里也包含该属性名也一起修改，否则就会产生错误。

# 分工与合作说明

主题：由小组成员一起讨论取定；

组员林帆：系统代码的编写，实现所需功能，负责数据库的设计，搭建概念模型，绘制E-R图，并编写相关部分的技术文档。

组员梁意：负责收集相关素材，界面的设计及实现，测试系统并编写系统测试报告。

组员胡建萍：负责界面的初步设计，测试使用系统，编写使用说明、软件功能部分的文档并整合。

# 分数评定

|  |  |  |  |
| --- | --- | --- | --- |
| **组员**  **评分人员** | **林帆** | **梁意** | **胡建萍** |
| **林帆** | 90 | 85 | 80 |
| **梁意** | 94 | 80 | 80 |
| **胡建萍** | 92 | 83 | 80 |
| **平均得分** | **92** | **83** | **80** |