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.IO;

namespace CSLG.CTalk.UI

{

public partial class FrmTalk : Form

{

private Model.userInfo \_me;

private Model.userInfo \_friend;

private string \_nickName;

private FrmMain \_frm;

private List<RelFriendToFrmTalk> \_ft;

private UCMessageHistory currentRight = new UCMessageHistory();

public FrmTalk(List<RelFriendToFrmTalk> ft, FrmMain frm, Model.userInfo me, Model.userInfo friend, string nickName)

{

InitializeComponent();

this.\_me = me;

this.\_friend = friend;

this.\_nickName = nickName;

this.\_frm = frm;

this.\_ft = ft;

}

Image \_fimage;

Image \_iimage;

int \_fid;

int \_iid;

string \_fname;

bool \_fisopen;

string \_iname;

public FrmTalk(Image HeadImage,Image Iimage ,int UserID,int id,string UserName,bool frmtalkIsOpen,string iname )

{

InitializeComponent();

this.\_fid = UserID;

this.\_fimage = HeadImage;

this.\_iimage = Iimage;

this.\_fisopen = frmtalkIsOpen;

this.\_fname = UserName;

this.\_iid = id;

this.\_iname = iname;

}

public FrmTalk()

{

InitializeComponent();

}

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

{

this.Close();

}

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

{

if (this.\_me != null)

{

this.Text = "与" + this.\_nickName + "聊天中...";

this.picUser.Image = this.\_frm.imageList1.Images[Convert.ToInt32(this.\_me.userHeadImage)];

this.picFriend.Image = this.\_frm.imageList1.Images[Convert.ToInt32(this.\_friend.userHeadImage)];

}

else

{

this.Text = "与" + this.\_fname + "聊天中...";

this.picUser.Image = this.\_iimage;

this.picFriend.Image = this.\_fimage;

}

}

private void FrmTalk\_FormClosed(object sender, FormClosedEventArgs e)

{

if (this.\_me != null)

{

for (int i = 0; i < this.\_ft.Count; i++)

{

if (this.\_ft[i].FriendId == this.\_friend.userID)

{

this.\_ft[i].IsOpenTalkForm = false;

break;

}

}

}

}

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

{

if (this.\_me != null)

{

SendMsg();

}

else

{

SendMsg2();

}

}

private void SendMsg()

{

string comment = this.txtmessge.Text;

if (comment == "")

{

return;

}

BLL.messsageInfo BllMsgInfo = new BLL.messsageInfo();

Model.messsageInfo msg = new Model.messsageInfo();

msg.messageContent = comment;

msg.senderID = this.\_me.userID;

msg.receiverID = this.\_friend.userID;

msg.messageState = 1;

msg.sendTime = BLL.NormalOperation.getServerTime();

int res = BllMsgInfo.Add(msg);

if (res <= 0)

{

MessageBox.Show("消息发送失败或检查网络连接！！");

}

this.txtmessge.Text = "";

ShowMsg(msg, false);

bool isExist = false;

FileStream fst = new FileStream(Application.StartupPath + "\\" + "friend.txt", FileMode.Open, FileAccess.Read);

StreamReader srt = new StreamReader(fst);

while (srt.Peek() > -1)

{

string qq = srt.ReadLine();

string[] re = qq.Split(' ');

if (re[1] == \_friend.userID.ToString() && re[0] == this.\_me.userID.ToString())

{

isExist = true;

}

}

srt.Close();

fst.Close();

if (isExist == false)

{

using (StreamWriter sw = File.AppendText(Application.StartupPath + "\\" + "friend.txt"))

{

sw.WriteLine(msg.senderID + " " + msg.receiverID);

sw.Close();

}

}

}

private void SendMsg2()

{

string comment = this.txtmessge.Text;

if (comment == "")

{

return;

}

BLL.messsageInfo BllMsgInfo = new BLL.messsageInfo();

Model.messsageInfo msg = new Model.messsageInfo();

msg.messageContent = comment;

msg.senderID = this.\_iid ;

msg.receiverID = this.\_fid ;

msg.messageState = 20;

msg.sendTime = BLL.NormalOperation.getServerTime();

int res = BllMsgInfo.Add(msg);

if (res <= 0)

{

MessageBox.Show("消息发送失败或检查网络连接！！");

}

this.txtmessge.Text = "";

ShowMsg2(msg, false);

}

private void ShowMsg2(Model.messsageInfo msg, bool isSendFromME)

{

int startPos = this.txtinfo.Text.Length;

string title = "";

Color fColor = Color.Black;

if (isSendFromME)

{

title = this.\_iname + msg.sendTime.ToLongTimeString() + "\r\n\t";

fColor = Color.Red;

}

else

{

title = this.\_fname + msg.sendTime.ToLongTimeString() + "\r\n\t";

fColor = Color.Blue; ;

}

int titlelen = title.Length;

this.txtinfo.AppendText(title);

this.txtinfo.SelectionStart = startPos;

this.txtinfo.SelectionLength = titlelen;

this.txtinfo.SelectionColor = fColor;

startPos = this.txtinfo.Text.Length;

string comment = msg.messageContent + "\r\n";

int commentlen = comment.Length;

this.txtinfo.AppendText(comment);

this.txtinfo.SelectionStart = startPos;

this.txtinfo.SelectionLength = commentlen;

this.txtinfo.SelectionColor = Color.Black;

}

private void txtmessge\_KeyPress(object sender, KeyPressEventArgs e)

{

if (e.KeyChar == (char)Keys.Enter)

{

if (this.\_me!=null )

{

SendMsg();

}

else

{

SendMsg2();

}

}

}

private void TmPicker\_Tick(object sender, EventArgs e)

{

if (this.\_me != null)

{

BLL.messsageInfo bllmsg = new BLL.messsageInfo();

List<Model.messsageInfo> msgInfo = bllmsg.GetModelList("senderID= '" + this.\_friend.userID + "'and receiverID= '"+this.\_me.userID +"' and messageState=1 or messageState=20 ");

foreach (Model.messsageInfo m in msgInfo)

{

ShowMsg(m, true);

BLL.messsageInfo BllMsgInfo = new BLL.messsageInfo();

Model.messsageInfo msg = new Model.messsageInfo();

msg.messageContent = m.messageContent;

msg.senderID = m.senderID;

msg.receiverID = m.receiverID;

msg.messageState = 2;

msg.sendTime = m.sendTime;

msg.messageID = m.messageID;

BllMsgInfo.Update(msg);

}

}

else

{

BLL.messsageInfo bllmsg = new BLL.messsageInfo();

List<Model.messsageInfo> msgInfo = bllmsg.GetModelList("senderID= '" + this.\_fid + "'and messageState=20 or messageState=1 ");

foreach (Model.messsageInfo m in msgInfo)

{

ShowMsg2(m, true);

BLL.messsageInfo BllMsgInfo = new BLL.messsageInfo();

Model.messsageInfo msg = new Model.messsageInfo();

msg.messageContent = m.messageContent;

msg.senderID = m.senderID;

msg.receiverID = m.receiverID;

msg.messageState = 21;

msg.sendTime = m.sendTime;

msg.messageID = m.messageID;

BllMsgInfo.Update(msg);

}

}

}

public void ShowMsg(Model.messsageInfo msg, bool isSendFromME)

{

int startPos = this.txtinfo.Text.Length;

string title = "";

Color fColor = Color.Black ;

if (!isSendFromME)

{

title = this.\_me.userName + msg.sendTime.ToLongTimeString() + "\r\n\t";

fColor = Color.Red;

}

else

{

title = this.\_nickName + msg.sendTime.ToLongTimeString() + "\r\n\t";

fColor = Color.Blue; ;

}

int titlelen = title.Length;

this.txtinfo.AppendText(title);

this.txtinfo.SelectionStart = startPos;

this.txtinfo.SelectionLength = titlelen;

this.txtinfo.SelectionColor = fColor;

startPos = this.txtinfo.Text.Length;

string comment = msg.messageContent + "\r\n";

int commentlen = comment.Length;

this.txtinfo.AppendText(comment);

this.txtinfo.SelectionStart = startPos;

this.txtinfo.SelectionLength = commentlen ;

this.txtinfo.SelectionColor = Color.Black ;

}

List<UCMessageHistory> listmessage = new List<UCMessageHistory>();

private void button1\_Click(object sender, EventArgs e)

{

getmessage();

}

private void getmessage()

{

this.Width = 640;

this.pnHistory.Controls.Clear();

this.pnHistory.Visible = true;

BLL.messsageInfo bllmess = new BLL.messsageInfo();

List<Model.messsageInfo> lmess = bllmess.GetModelList("receiverID='" + this.\_friend.userID + "'or senderID='" + this.\_friend.userID + "'");

int index = 0;

foreach (Model.messsageInfo mess in lmess)

{

BLL.userInfo blluser = new BLL.userInfo();

Model.userInfo username = blluser.GetModel(mess.senderID);

UCMessageHistory uc = new UCMessageHistory();

if (mess.receiverID == this.\_friend.userID)

{

uc.Ucid = mess.messageID;

uc.Top = uc.Height \* index;

uc.Username = username.userName;

uc.Time = mess.sendTime;

uc.Message = mess.messageContent;

uc.BackColor = Color.Beige;

uc.Me = false;

uc.MouseClick += new MouseEventHandler(uc\_MouseClick);

listmessage.Add(uc);

uc.MouseMove += new MouseEventHandler(uc\_MouseMove);

pnHistory.Controls.Add(uc);

index++;

}

else

{

uc.Ucid = mess.messageID;

uc.Top = uc.Height \* index;

uc.Username = username.userName;

uc.Time = mess.sendTime;

uc.Message = mess.messageContent;

uc.BackColor = Color.Beige;

uc.Me = true;

listmessage.Add(uc);

uc.MouseMove += new MouseEventHandler(uc\_MouseMove);

uc.MouseClick += new MouseEventHandler(uc\_MouseClick);

pnHistory.Controls.Add(uc);

index++;

}

}

}

void uc\_MouseClick(object sender, MouseEventArgs e)

{

if (e.Button != System.Windows.Forms.MouseButtons.Right)

{

return;

}

currentRight = (UCMessageHistory )sender;

this.contextMenuStrip1 .Show(Control.MousePosition);

}

void uc\_MouseMove(object o, MouseEventArgs e)

{

UCMessageHistory uc = (UCMessageHistory)o ;

foreach (UCMessageHistory ucmh in listmessage)

{

ucmh.BackColor = Color.Beige;

}

uc.BackColor = Color.Bisque;

}

private void pictureBox1\_Click\_1(object sender, EventArgs e)

{

this.Width = 519;

this.pnHistory.Visible = false;

}

private void 删除此条信息ToolStripMenuItem\_Click(object sender, EventArgs e)

{

UCMessageHistory uc = (UCMessageHistory)currentRight ;

BLL.messsageInfo bll = new BLL.messsageInfo();

int mid = uc.Ucid ;

bll.Delete(mid );

getmessage();

}

private void 删除所有信息ToolStripMenuItem\_Click(object sender, EventArgs e)

{

}

}

}