

基于中央定位服务器的 P2P 网络聊天系统

Parlons 设计文档

自 54 田毅 2015011451

2017 年 12 月

目录

1	需求分析	3
1.1	必做内容	4
1.1.1	账号登录上线	4
1.1.2	查询好友是否在线	4
1.1.3	P2P 下的一对一通信	4
1.1.4	下线功能	5
1.1.5	文件传输	5
1.2	选做内容	5
1.2.1	具有友好的用户界面	5
1.2.2	具有群聊功能	5
1.2.3	个性化功能	5
1.2.4	使用手机编程实现 (Android 或者 iOS)	5
2	总体设计	6
2.1	Parlons 命名说明	6
2.2	系统架构	6
2.2.1	有限状态机	6
2.2.2	程序流程图	7
2.2.3	系统类图	8
2.3	回顾：本项目完成的选做内容	9

目录	2
3 详细设计	9
3.1 服务器连接模块	10
3.2 P2P 连接模块	12
3.3 登录模块	14
3.4 Parlons 模块	16
3.4.1 用户模块	21
3.4.2 窗口控件	22
4 结果分析	23
4.1 登录界面	23
4.2 Parlons 主界面	24
5 遇到的主要问题及解决	32
5.1 程序退出后仍有残存线程的问题	32
5.2 表情包大量读取后内存不足的问题	33
6 总结	33

1 需求分析

本项目的目标是实现一个网络聊天系统。

网络聊天系统是一种特殊目的的网络应用程序（Network Application）。网络应用程序架构（Architecture）分为三类 [1]：（1）客户端 -服务器架构（C/S 架构，Client-Server Architecture）；（2）点对点架构（P2P 架构，Peer-to-Peer Architecture）；（3）二者的混合。本项目要求实现的是第三种类型的架构，即基于中央定位服务器的 P2P 网络聊天系统，具体形式如下图。

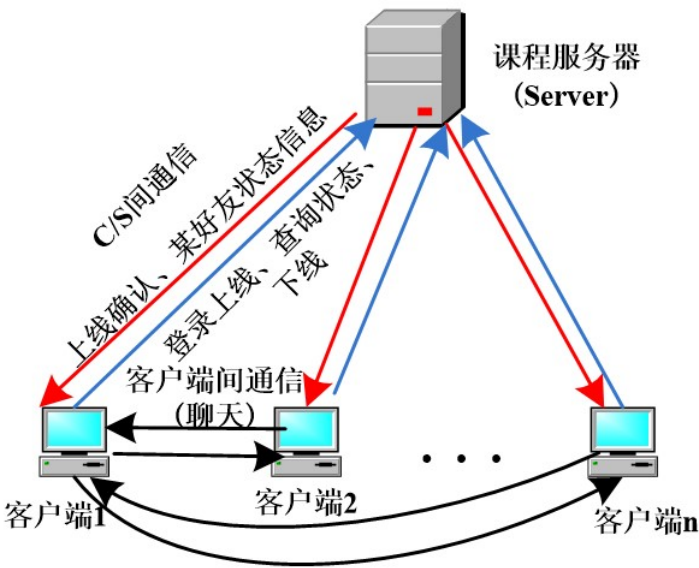


图 1: 基于中央定位服务器的 P2P 网络聊天系统

程序质量的衡量应从两个维度考虑：（1）**程序功能**：用户能用这个程序做什么？（2）**用户体验**：用户使用后的感受如何？前者决定有相关需求的用户是否选择尝试这个程序，后者决定用户在尝试后是否继续使用这个程序。

需求分析与**价值观**密切相关。我们分析依据的价值观包括：（1）**轻松性**：现代社会已带给人们太多的焦虑和压力，我们希望本产品最小化这些负面情绪，带给人们**轻松的感受**；（2）**可行性**：任何项目的实施都在一定的约束条件下进行，项目在谋划时对这些约束条件应有清晰的认知，本项目的约束条件主要是不可扩展的中央服务器，即**中央服务器不可临时储存并中转发送消息、不可注册新的账号、不可为每个用户储存好友列表等**，因此一些成熟聊天软件中的功能在本项目中无法实现。

下面从程序功能和用户体验两个维度出发，逐一分析必做和选做内容。

1.1 必做内容

1.1.1 账号登录上线

程序功能 与中央服务器通信，告知上线。

用户体验 良好的用户体验要求这一部分有单独的界面。如登录成功，则切换到聊天程序的主界面；否则应提供对应提示。

1.1.2 查询好友是否在线

程序功能 与服务器通信，服务器将返回好友状态。程序需将这一状态告知用户。本大作业的中央服务器

用户体验 好友列表是成功聊天软件的标准配置。但是否在好友列表中显示好友的在线状态是一个值得讨论的话题。QQ[2] 和微信 [3] 是两款具有代表性的聊天软件，其中 QQ 提供此功能，微信不提供此功能。我们认为，（1）显示在线无形中给人一种回复消息的压力；（2）在线状态属于个人隐私，非必要不应提示。因此，我们得到结论：依据轻松性原则，舒适的用户体验要求不必要时不显示在线状态；依据可行性原则，服务器不提供消息临时存储功能，发送消息或文件时必须查询好友是否在线。

1.1.3 P2P 下的一对一通信

程序功能 不通过服务器，客户端之间通过 socket 实现一对一通信，但通信前需要确认好友是否在线。

用户体验 消息发送后应自动清空输入框；消息发送和接收应提供日期、时间信息。针对于陌生人是否可以发送消息，QQ 和微信同样采取了不同的设计方式。我们认为，（1）中央服务器内存储的用户均为清华校内同学¹，清华作为一个以理工科见长的综合型、研究型大学应鼓励大家的相互认识和沟通，如果可能，通过学号联系到同学是亟待实现的功能，这对于提高校内脱单率等具有积极意义，而不应以陌生人间不可沟通的权限增强校内的紧张；（2）中央服务器不可存储好友列表、不可新增用户的性质，实际上等价于所有当前存储的用户彼此间均为好友。基于轻松性原则和可行性原则，我们的结论是：应该允许未在好友列表的用户与自己通信，且通信后应立即加入好友列表。

¹具体来说，可能是选修计算机网络课程的同学。

1.1.4 下线功能

程序功能 与服务器通信，告知下线。提供下线选项。

用户体验 下线应予以提示，可取消；应提供切换账号功能，此时应返回登录界面。

1.1.5 文件传输

程序功能 如果好友在线，能够向好友发送文件，不通过服务器。

用户体验 接收不需要的文件是一种负担，依据**轻松性原则**，应提供是否接收文件选项。

1.2 选做内容

1.2.1 具有友好的用户界面

本项目在时间允许的范围内尽可能使用扁平化设计（Flat Design[4]），力求去除冗余、厚重、繁杂的装饰效果，让“信息”本身重新作为核心被凸显出来。在设计元素上，强调抽象、极简、和符号化。

1.2.2 具有群聊功能

程序功能 同组内的好友可以实现群聊，不通过服务器，客户端之间通过 socket 实现一对多通信。这可被视作是在**应用层上实现的多播**。

用户体验 为便于检索，应有**群聊名**功能。

1.2.3 个性化功能

自由发挥，比如语音传输、动态表情、聊天记录查询等。

本项目实现的程序功能是**发送表情功能**。众所周知，现代社会的聊天大量依赖于表情，以含蓄、戏谑地表达自己的情绪。从**轻松性原则**出发，这是一个不能不实现的功能。

1.2.4 使用手机编程实现 (Android 或者 iOS)

多终端是当今世界发展的趋势。遗憾的是，本项目仅支持 Windows 操作系统。

2 总体设计

2.1 Parlons 命名说明

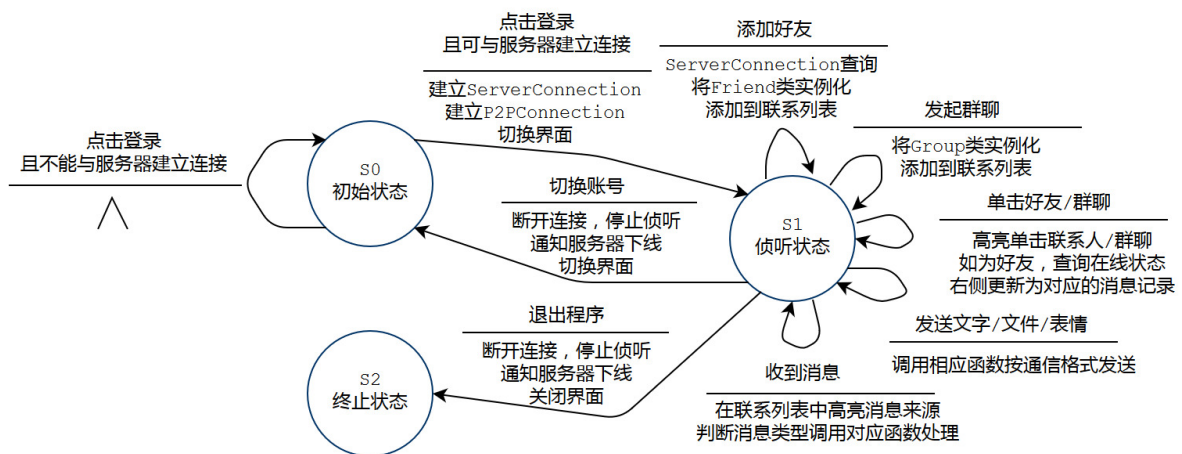
好的名字是成功的项目必不可少的组成部分。本聊天系统名称 Parlons 读作 [par'lõ]，源自法语。法语中，parler 是“谈论、聊天”的意思，例如 parler de chose et d'autres 即为“闲聊”的意思。parlons 是 parler 的第一人称复数变位²，如 nous parlons（我们谈话）。Parlons 本身可被视作是祈使句，意为：让我们聊！

因此，选择 Parlons 作为聊天系统名字意义包括：（1）具有“让我们聊”的意义，清楚地表明本软件的用途，与微信（Wechat）不谋而合；（2）以法语命名，体现对多元文化和多元价值观的尊重，非洲国家多以法语为官方语言，这种命名方式可令相关国家使用者感到亲切，符合“一带一路倡议”的方向；（3）Parlons 虽为法语，但可以顺畅地用英语读作 [pa:lɒns]，这种读法也被我们认可，因此可适应说英语者。

2.2 系统架构

我们从有限状态机（Finite State Machine, FSM）、程序流程图（Program Flowchart）、系统类图（Class Diagram）三方面阐述我们设计的系统架构。

2.2.1 有限状态机



针对上述需求分析，我们设计了如上图的有限状态机。

²变位是指动词形式随主语而变，一般发生在词尾，例如英语中动词在第三人称单数后的变化。

图中设计了三个状态；遵循一般有限状态机作图规范，横线以上表示触发事件，横线以下表示响应动作。**S0 状态**与登录界面对应，针对点击“登录”后的不同情况做出响应；**S1 状态**与主界面对应，可处理好友查询、群聊建立、三种类型的消息收发等各类事件；**S2 状态**是终止状态，程序完全退出。

2.2.2 程序流程图

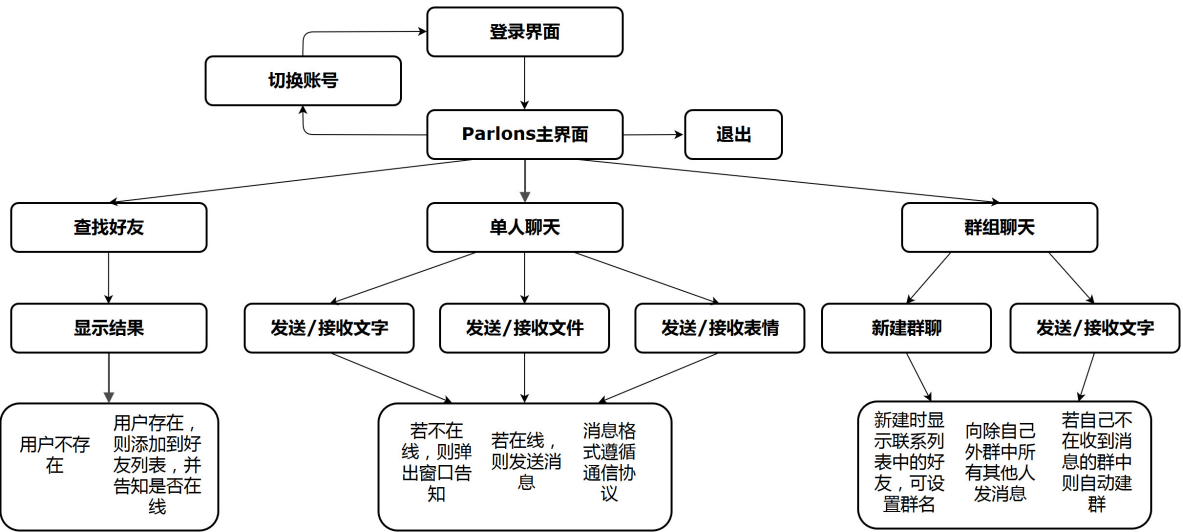


图 3: 系统流程图

系统流程图描述了我们提供给用户的接口。

程序共有登录界面与主界面两个，支持的主要功能从图中清晰地可见。各功能的具体实现细节，我们将在下面说明。

2.2.3 系统类图

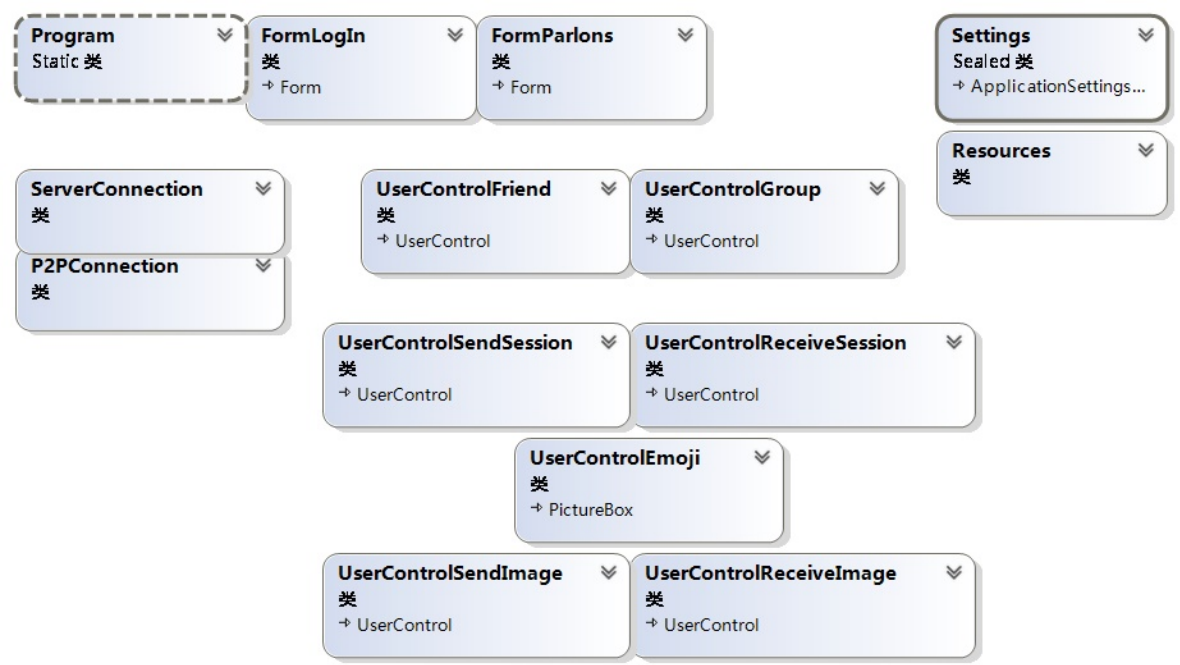


图 4: 系统类图

系统类图描述了程序的架构。除默认的 Settings、Resources、Program 类外，我们共定义、实现了 11 个类。这 14 个类在图中已按相互关系组织在了一起：

主界面与主逻辑 2 个。FormLogin 与 FormParlons 类。前者实现登录界面与逻辑，后者实现聊天主界面与主逻辑。

网络通信 2 个。ServerConnection 与 P2PConnection 类。前者实现与中央服务器的通信，后者实现 P2P 通信。

聊天用户 2 个。UserControlFriend 与 UserControlGroup 类。前者是每个添加的朋友维护一个对象，后者为添加的每一个群聊维护一个对象。

窗口控件 5 个。UserControlSendSession、UserControlReceiveSession、UserControlEmoji、UserControlSendImage、UserControlReceiveImage 类。分别为发送会话控件、接收会话控件、表情控件、发送图片控件、接收图片空间。

2.3 回顾：本项目完成的选做内容

Parlons 实现了全部必做内容。我们在此将 Parlons 实现的选做内容总结如下。

具有友好的用户界面 Parlons 设计了友好的用户界面。主界面参照微信设计，简洁大方。例如：（1）去除了 C# 默认的菜单栏，自己添加了最小化与关闭图标，并设计拖动效果；（2）界面文件发送、表情发送、新建群聊等功能以图标代替 Button；（3）允许通过回车一键发送消息；（4）点击的联系列表中的联系对象高亮显示；（5）新增消息后消息界面滚动条自动滚到最下方；（6）发送与接收的文件若为图片则在聊天窗口中生成预览；（7）简单的头像功能；等等。在界面的很多细节上花费了很多心思。

具有群聊功能 Parlons 实现了简单的群聊功能。具体包括：（1）发起群聊的界面及功能；（2）设置群名的功能，群名可自动同步到所有群中成员的程序中去；（3）群消息的发送与接收，建群后在发第一次消息时群成员才会知道建群，且允许群成员间相互不是好友。

个性化功能：表情包 Parlons 使用代码完成了表情包的构建与显示。表情包具体功能包括：（1）系统提供 846 个默认的表情包的预览和发送功能；（2）可通过在文件夹中加入图片而方便地扩展表情；（2）支持大表情的快速传送，因为原理上在传输字符串而非图片文件。

3 详细设计

以各类为线索，我们对我们的设计做出详细介绍。编程中有参考部分 C# 微软官方文档 [5] 及其他相关博客。

3.1 服务器连接模块



图 5: 服务器连接模块类图

服务器连接模块不依赖于界面，其作用是完成与中央服务器的通信。

ServerQuery 函数向中央服务器发送字符串，并返回中央服务器响应的字符串。这一功能是整个程序运行的基础，为确保不会出现问题，设置了多组 try 和 catch 机制。关键代码如下：

```
serverSocket = new Socket(AddressFamily.InterNetwork,
    SocketType.Stream, ProtocolType.Tcp);
// set up connection
try
{
    serverSocket.Connect(serverEndPoint);
}
catch (SocketException se)
{
    MessageBox.Show(se.Message, "温馨提示");
    return resultStr;
}

serverSocket.Send(queryByte);

// receive from server
```

```
try
{
    length = serverSocket.Receive(resultByte);
}
catch (SocketException se)
{
    MessageBox.Show(se.Message, "温馨提示");
    serverSocket.Close();
    return resultStr;
}
catch (Exception e)
{
    MessageBox.Show(e.Message, "温馨提示");
    serverSocket.Close();
    return resultStr;
}

resultStr = System.Text.Encoding.UTF8.GetString(
    resultByte, 0, length);
serverSocket.Close();
return resultStr;
```

3.2 P2P 连接模块

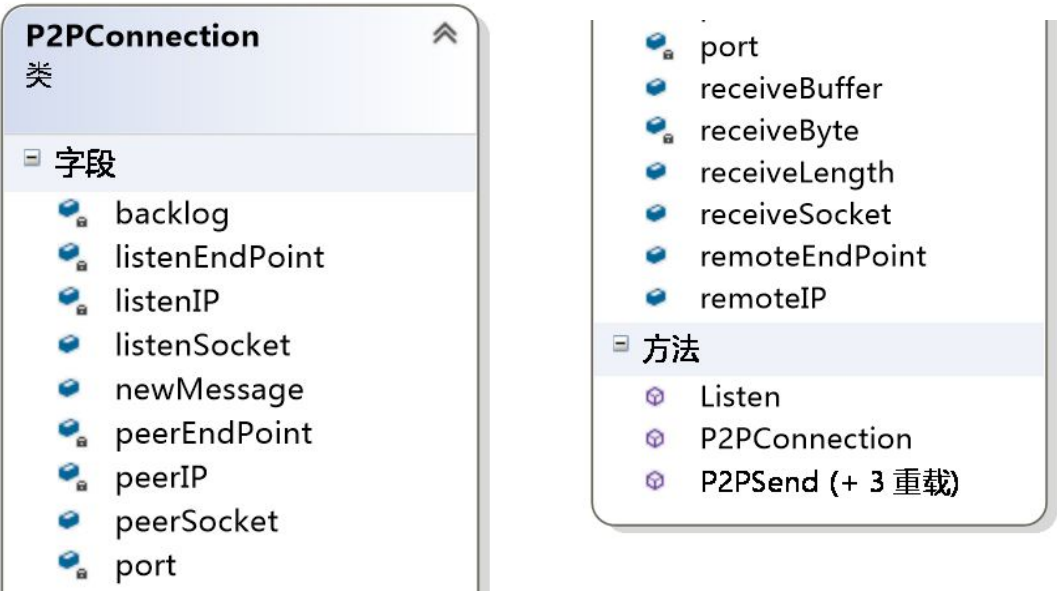


图 6: P2P 连接模块类图

P2P 模块功能有二：（1）作为客户端 P2P 发送消息；（2）作为服务器监听其他用户发来的消息。

功能（1）由函数 P2PSend 实现。这一函数共有 4 种重载形式，兼容 IP 输入为字符串/IPAddress 两种格式、发送内容为 Byte 数组/字符串两种格式。关键代码如下：

```
// send bytes to a peer
public void P2PSend(IPAddress peerIP , byte[] sendByte
)
{
    peerEndPoint = new IPEndPoint(peerIP , port);
    peerSocket = new Socket(AddressFamily.
        InterNetwork , SocketType.Stream , ProtocolType.
        Tcp);

    // set up connection
    try
```

```
{
    peerSocket.Connect(peerEndPoint);
}
catch (SocketException se)
{
    MessageBox.Show(se.Message, "温馨提示");
    return;
}

// send the bytes
peerSocket.Send(sendByte);
peerSocket.Close();
}
```

功能（2）由函数 P2PConnection 实现。通过循环使用 Socket 接收消息，接收到后再使用循环取出内容，直至收到约定的终止消息（1 个全 0 的 Byte）退出循环。实际调用时在线程中调用，因此受到终止消息后线程结束。关键代码如下：

```
while (true)
{
    int length;

    receiveSocket = listenSocket.Accept();
    length = receiveSocket.Receive(receiveByte);
    receiveLength = 0;
    Array.Copy(receiveByte, 0, receiveBuffer,
        receiveLength, length);
    receiveLength += length;

    remoteEndPoint = (IPEndPoint)receiveSocket.
        RemoteEndPoint;
    remoteIP = remoteEndPoint.Address;

    while (length > 0)
```

```

{
    length = receiveSocket.Receive(receiveByte);
    Array.Copy(receiveByte, 0, receiveBuffer,
        receiveLength, length);
    receiveLength += length;
}

// 1 byte of 0 is defined as the terminating
// signal
if (receiveLength == 1 && receiveBuffer[0] == 0)
    break;

receiveSocket.Close();
// set flag to remind the FormParlons
newMessage = true;
}

```

3.3 登录模块

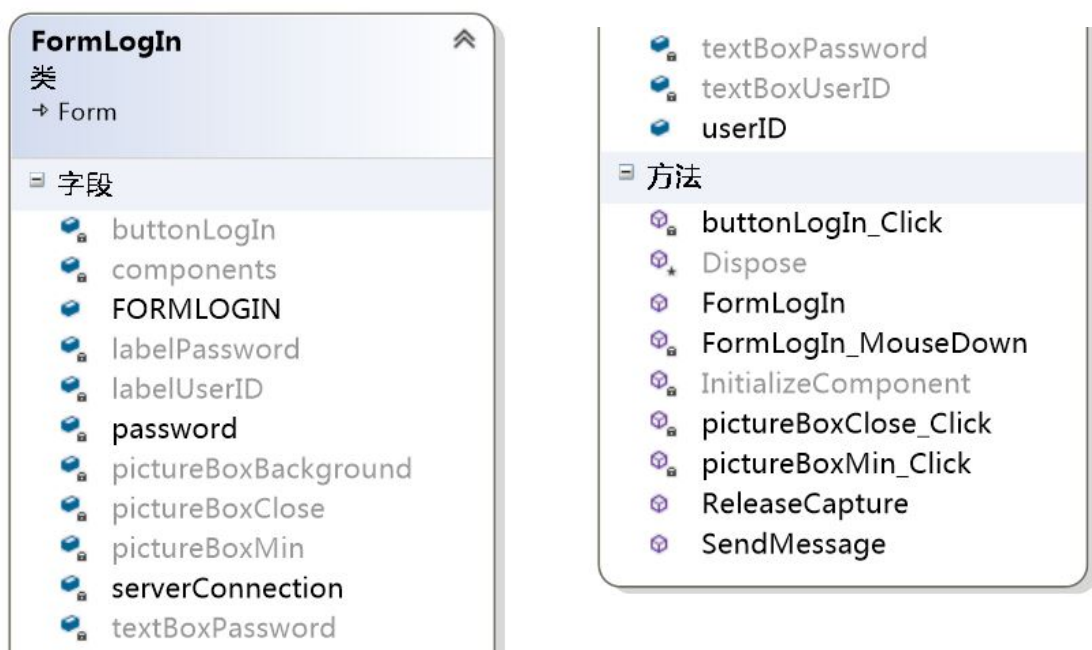


图 7: 登录模块类图

登录模块逻辑与界面一体。从类图可看到有许多界面触发事件，体现了界面的精心设计。例如，ReleaseCapture、SendMessage、FormLogin_MouseDown 三个函数实现了界面随鼠标拖动而移动；而字段 FORMLOGIN 则是指向 this 的静态引用，在从主界面切换回来时使用。

buttonLogIn_Click 是此模块主要逻辑。通过调用 ServerConnection 实例的方法建立与中央服务器的连接。关键代码为：

```
private void buttonLogIn_Click(object sender ,
    EventArgs e)
{
    // get the information in the text boxes
    userID = textBoxUserID.Text.ToString();
    password = textBoxPassword.Text.ToString();
    // form a log-in string
    string logInStr = userID + "_" + password;

    // log in
    if (serverConnection.ServerQuery(logInStr) == "
        lol")
    {
        this.Hide();
        FormParlons formParlons = new FormParlons(
            serverConnection);
        formParlons.Show();
    }
    else
    {
        MessageBox.Show("用户名或密码有误哦！", "温馨提
            示");
    }
}
```

3.4 Parlons 模块

FormParlons

类

→ Form

字段

buttonChangeUser

buttonQuery

buttonSend

checkedListBoxGroup

components

currGroupOrderID

currOrderID

flowLayoutPanelFriends

flowLayoutPanelSession

formatLength

friends

friendsDict

groups

groupsDict

isGroup

listening

listening

myIPStr

myUserID

p2pConnection

panelEmoji

PARLONS

pictureBoxClose

pictureBoxGroup

pictureBoxGroupCancel

pictureBoxGroupConfirm

pictureBoxMin

pictureBoxSendEmoji

pictureBoxSendFile

serverConnection

textBoxGroupName

textBoxQuery

textBoxSession

threadListening

threadMonitoring

方法

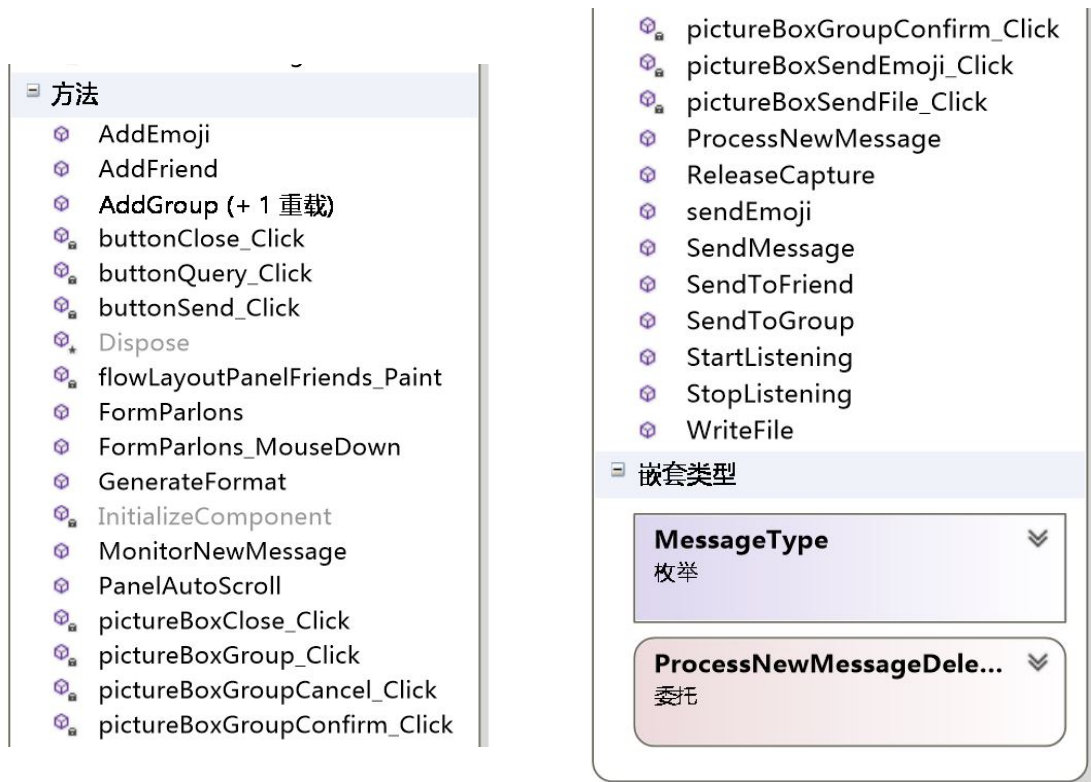


图 8: Parlons 模块类图

Parlons 模块是本项目的主界面与主逻辑。拥有大量字段和方法，其方法可归入以下几类：

构造函数 实现字段赋值、界面元素的隐藏与初始化的功能，开启两个线程分别监听和监控新消息，并自动添加自己为好友。

界面配置 包括设置界面随鼠标拖动而移动的三个函数、界面关闭与切换账号时触发的函数等。关键代码以关闭界面函数为例：

```
// stop listening
StopListening();
while (threadListening.IsAlive) ;
while (threadMonitoring.IsAlive) ;

// log out
string logOutStr = "logout" + myUserID;
```

```

serverConnection.ServerQuery(logOutStr);
MessageBox.Show("您已经下线了哦!", "温馨提示");

this.Close();
FormLogIn.FORMLOGIN.Close();
Application.Exit();

```

处理并发送消息 包括格式生成、增加表情、发送给好友、发送给群组、发送表情等函数。这些函数的核心是**通信协议**的定义。通信协议（B 表示 Byte 单位）为：

文字消息：10B[发送者 ID] + 1B[发送类型] + [文字消息]

文件消息：10B[发送者 ID] + 1B[发送类型] + 1B[文件名长度] + [文件名] + [文件]

表情消息：10B[发送者 ID] + 1B[发送类型] + 1B[0] + [表情名]

群组文字消息：10B[发送者 ID] + 1B[群聊人数 N] + (10N)B[群 ID] + 1B[群名长度] + [群名] + [文字消息]

其中发送者 ID 即 10 位学号，是每个好友的唯一标识码；群 ID 是建群时所有好友的 ID 的拼接，长度为 10 倍群聊人数，是每个群组的唯一标识码。

发送类型对于文字为 0，对于文件和表情为 1；限制群聊人数至少为 2 人，因此 2 以上表示为群聊。

文件名长度为 0 表示为表情，其后内容按字符串发送与接收，接收方从本地表情库中找到此表情显示在聊天窗口；否则为文件，按文件发送与接收。

群名任意设置、可以重复，不作为群的唯一标识码。对于二人群组而言，这一功能与备注功能相近。

通信协议可以定义的方式很多，在此定义的仅为一种可行方案。

关键代码以给群组发送文字消息的代码为例：

```

// generate sendByte
// generate format
byte[] formatByte = GenerateFormat(MessageType.TEXT);
    // 11 bytes of format
formatByte[formatLength - 1] = (byte)peerUserIDs.
    Count; // the last byte of format represents the
    number of people in this group

```

```
// generate groupID
byte [] groupIDByte = System.Text.Encoding.UTF8.
    GetBytes(groups[currGroupOrderID].groupID);
// generate groupName
byte [] groupNameByte = System.Text.Encoding.UTF8.
    GetBytes(groupName);

// generate session
string session = groups[currGroupOrderID].
    textBoxSession.Text.ToString();
byte [] sessionByte = System.Text.Encoding.UTF8.
    GetBytes(session);
byte [] sendByte = new byte[formatByte.Length +
    groupIDByte.Length + 1 + groupNameByte.Length +
    sessionByte.Length];

// combine the 5 parts: format + groupID +
    len_groupName + groupName + session to form the
    sendByte
formatByte.CopyTo(sendByte, 0);
groupIDByte.CopyTo(sendByte, formatLength);
sendByte[formatLength + groupIDByte.Length] = (byte)
    groupNameByte.Length;
groupNameByte.CopyTo(sendByte, formatLength +
    groupIDByte.Length + 1);
sessionByte.CopyTo(sendByte, formatByte.Length +
    groupIDByte.Length + 1 + groupNameByte.Length);
```

接收并处理消息 包括开始监听、终止监听、监控消息、处理消息等函数。监听部分调用 `P2PConnection` 类的实例。使用委托方式在线程中触发处理消息函数，以避免在子线程中不能改动控件的问题。

ProcessNewMessage是整个接受消息处理的核心。处理的依据是通信协议。关

键代码以 Emoji 传输消息解析为例：

```
case 1: // FILE
    if (p2pConnection.receiveBuffer[formatLength] ==
        0)
    {
        // an emoji
        string emojiName = System.Text.Encoding.UTF8.
            GetString(p2pConnection.receiveBuffer,
                formatLength + 1,
                p2pConnection.receiveLength -
                    formatLength - 1);

        UserControlReceiveImage
            userControlReceiveImage = new
            UserControlReceiveImage(Image.FromFile(
                emojiName));
        friends[currOrderID].flowLayoutPanelSession.
            Controls.Add(userControlReceiveImage);
        PanelAutoScroll();
    }
    else
    {
        // ...
    }
```

好友相关 包括添加好友、好友查询函数。好友查询调用了 `ServerConnection` 的实例。

群聊相关 包括添加群组、新建群组控件事件处理函数等。对新建群聊 `CheckedListBox` 的处理方式为隐藏与显示。

3.4.1 用户模块

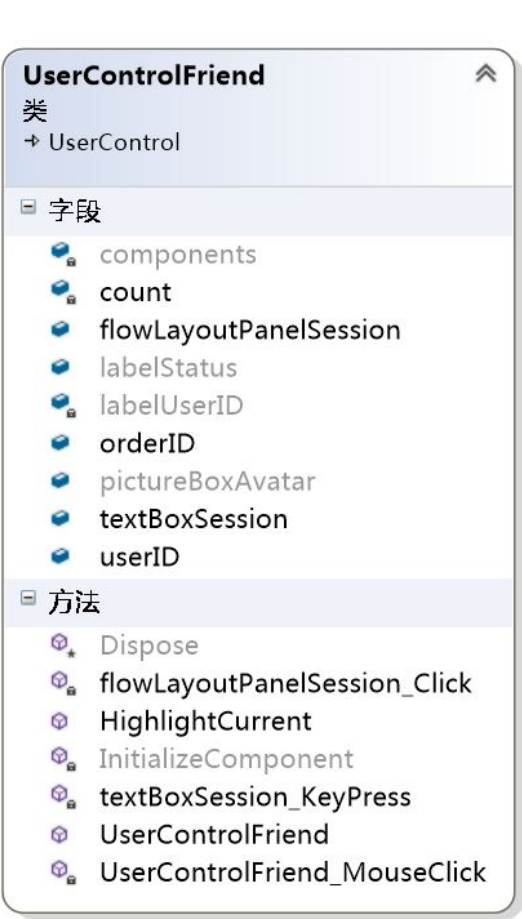


图 9: 好友控件类图

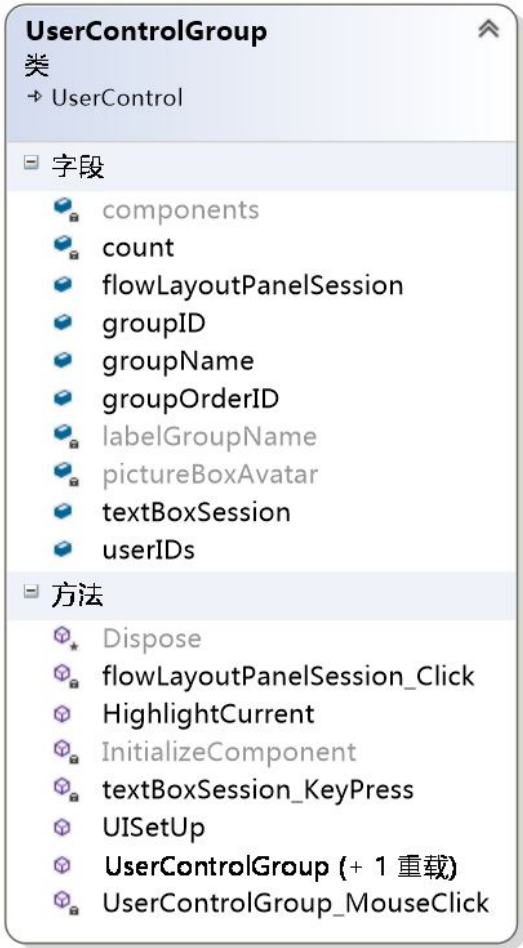


图 10: 群组控件类图

用户模块逻辑与界面位于同一个文件。

逻辑上，好友与群组是两个类，含有 ID 号、顺序 ID 号等字段。这里的顺序 ID 号体现了程序设计的思想。分配给每个好友一个在 List 中的顺序号，这样，通过建立“ID 号 - 顺序号”字典、好友 List，则可在很短时间内由 ID 号定位到好友实例。而通过记录当前顺序号则可以获知当前显示的是与哪一个好友聊天的窗口。每个好友实例都有自己聊天消息窗口与消息输入窗口。

群组与之类似，在 FormParlons 类中引入了 isGroup 的布尔变量以区分当前活动的聊天窗口是群组还是单个好友。为实现好友与群组在单击后的高亮效果，二者均编写了 Highligh 函数。二者均支持‘Enter’一键发送消息功能。

界面上，二者都是显示在左侧联系列表中的控件，可以被各种事件触发。

3.4.2 窗口控件



图 11: 表情控件类图

表情控件继承自 PictureBox 类。相比于 PictureBox 类，表情控件新增了表情名的字段，用于在单击事件触发后可以方便地获取表情名以按通信协议发送。表情包使用代码在 Panel 控件中排列，具体由代码进行设置。

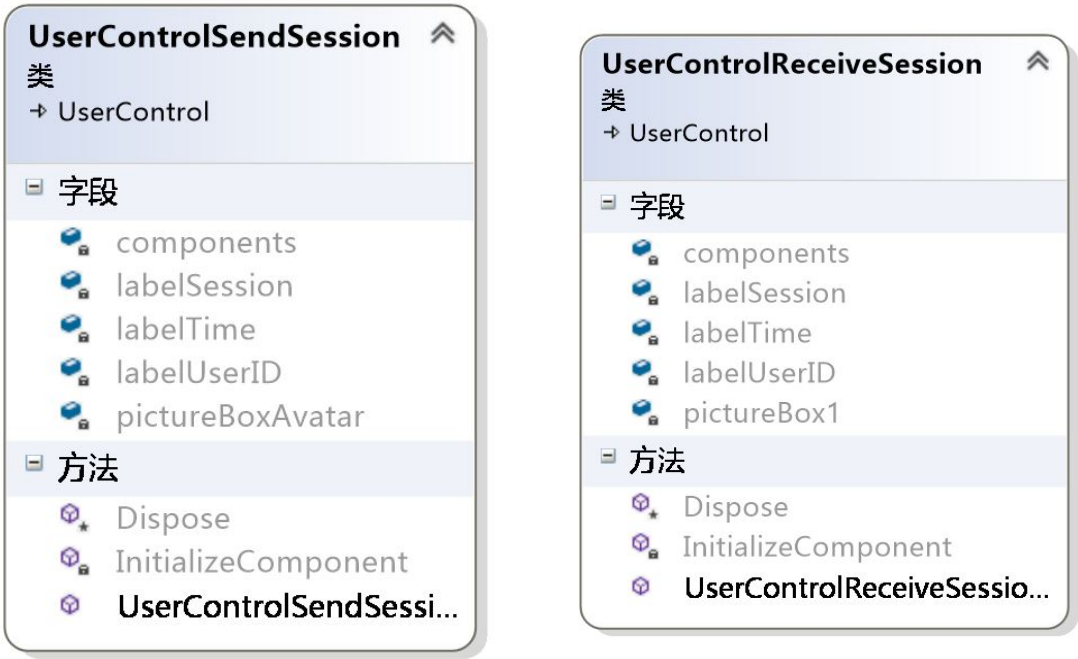


图 12: 发送会话控件类图

图 13: 接收会话控件类图

发送与接收会话控件二者是用于显示在右侧聊天消息窗口中控件，用于显示文字消息及文件消息。为每个会话特意设置了发送人、时间信息的提示。需要指出的是，对于群聊而言，发送人的提示是必不可少的。



图 14: 发送图片控件类图

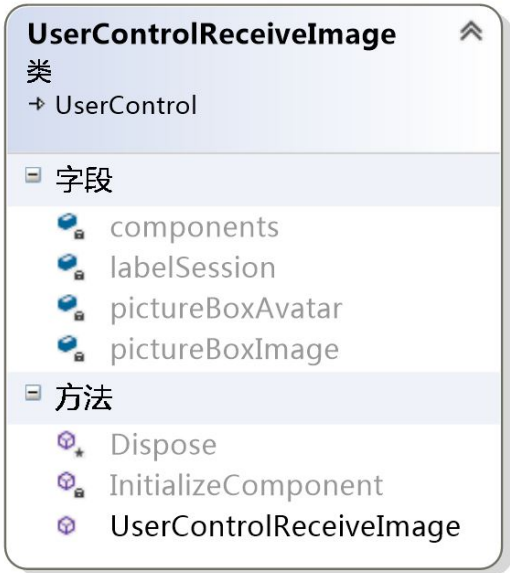


图 15: 接收图片控件类图

发送与接收图片控件为图片文件的发送与接收预览而设计。支持 JPG 与 PNG 格式图片的预览，显示在聊天消息窗口中，功能与“发送/接收会话控件”相同。

4 结果分析

运行程序，观察运行结果。

4.1 登录界面

打开程序，首先显示登录界面。

界面上有账号、密码选项，其中密码设置了暗文显示，默认为“net2017”。自制了“最小化”与“关闭”按钮。



图 16: 登录界面

4.2 Parlons 主界面

向中央服务器发起查询，用户名与密码无误后进入 Parlons 主界面。

添加好友 在左上角输入用户 ID 后，单击查找，与服务器通信，如用户存在，则添加到自己左侧的好友列表中；否则弹出提示信息。

显示在好友的列表的同时显示该好友是否在线。自己账号默认为第一个好友。添加好友且对方在线即可开始聊天。



图 17: 添加好友

发送消息 在右下方空白区域输入文字，“回车”或单击“发送”即可给在线好友或群组发送消息。如发送消息时需换行，则需使用“Ctrl”+“回车”。

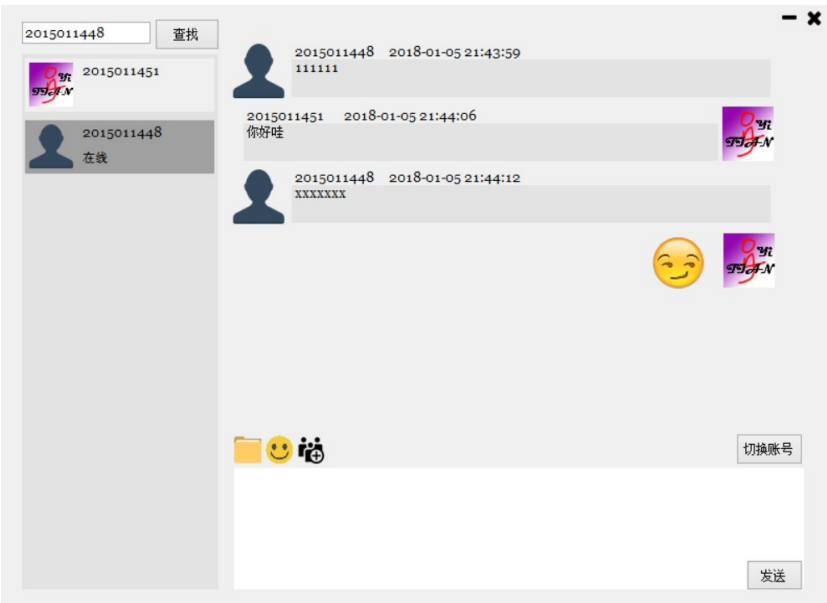


图 18: 发送消息

由于支持“回车”一键发送消息，且消息允许为空，所以可以通过一直敲击回车制造刷屏效果，简称“回车刷屏”。

从下图可见，消息窗口支持自动滚动条滚动到最下方的功能。

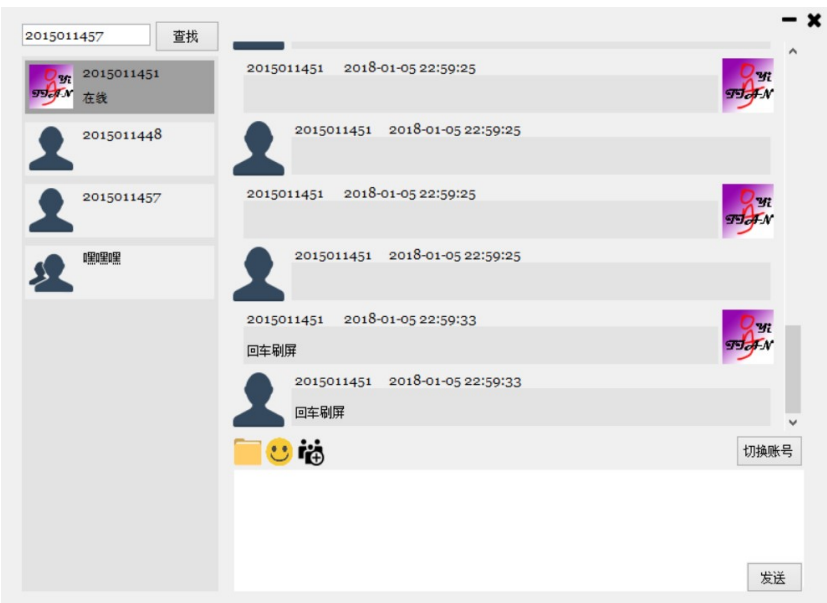


图 19: 回车刷屏

发送/接收文件 通过单击文件图标进行文件的发送与接收。二者都会弹出文件资源管理器窗口，从中选择要发送的文件或接收后存储的位置。

发送文件后会有相应提示：

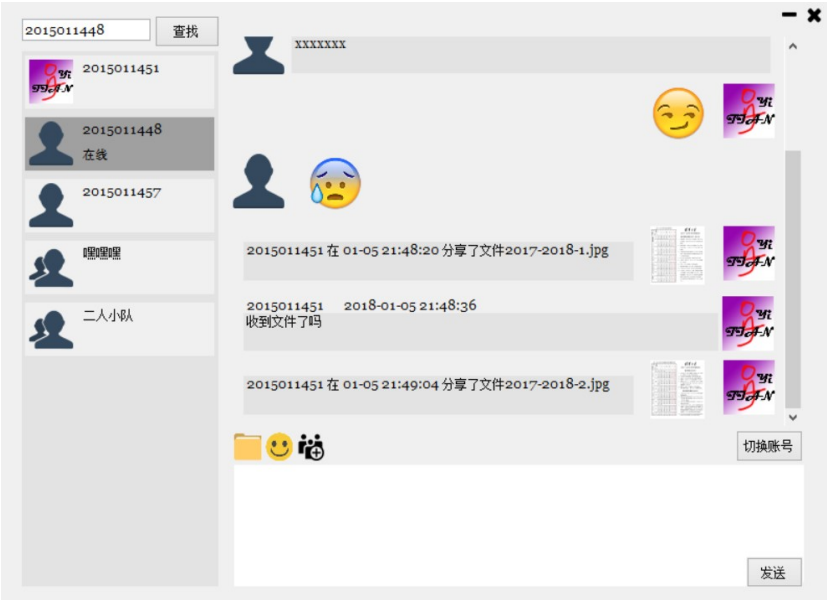


图 20: 发送文件

文件到来，会询问是否接受，如不接收，则可自动丢弃：

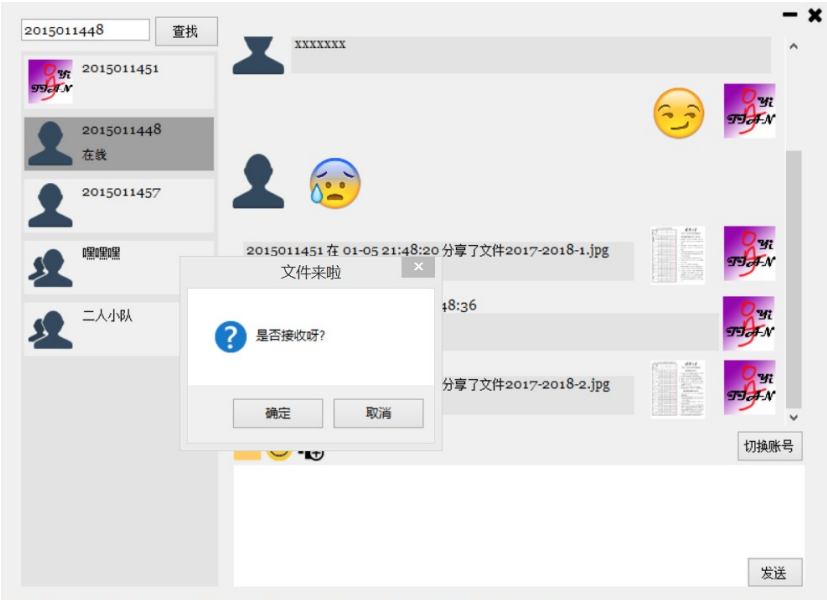


图 21: 接收文件

接收文件后会有相应提示：

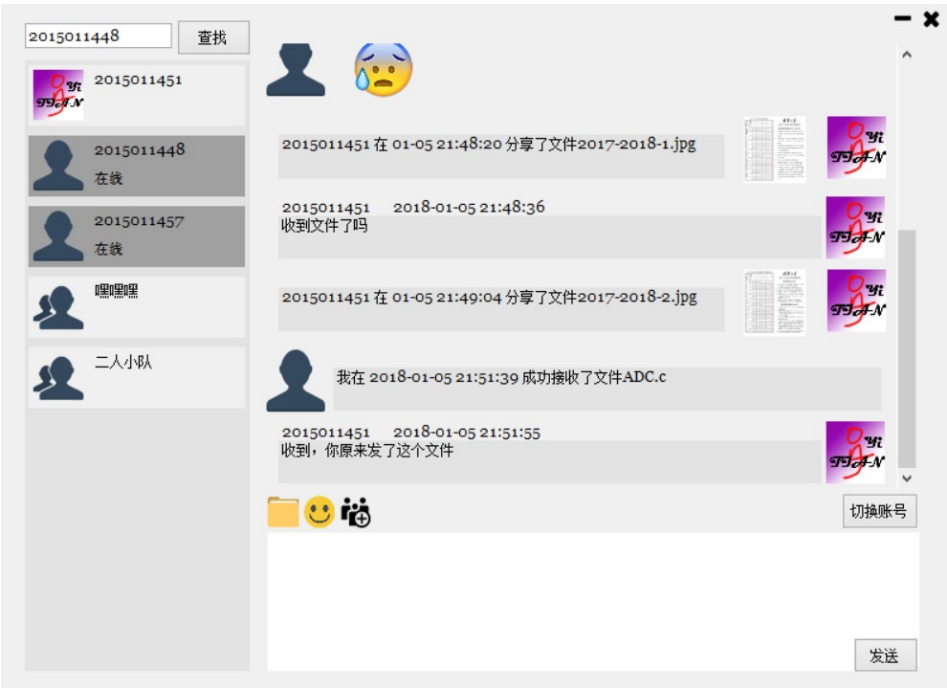


图 22: 接收成功

发送表情 Parlons 默认支持 846 个表情，单击表情图标，即在下方显示出表情，单击后直接发送。

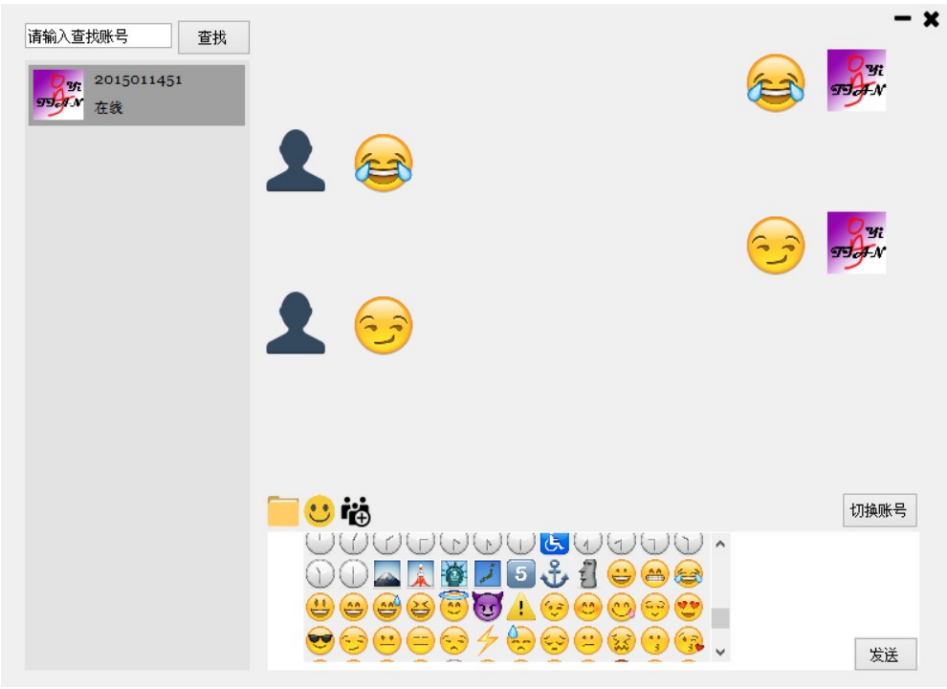


图 23: 发送表情

发起群聊 单击“群聊发起”图标，显示出已添加好友列表，可从中选择，可在输入群名位置设置群聊名。

新建群完毕后，单击绿色“对号”图标，则成功建群；如欲取消，则单击黑色“叉号”图标，可选建群的好友列表消失。

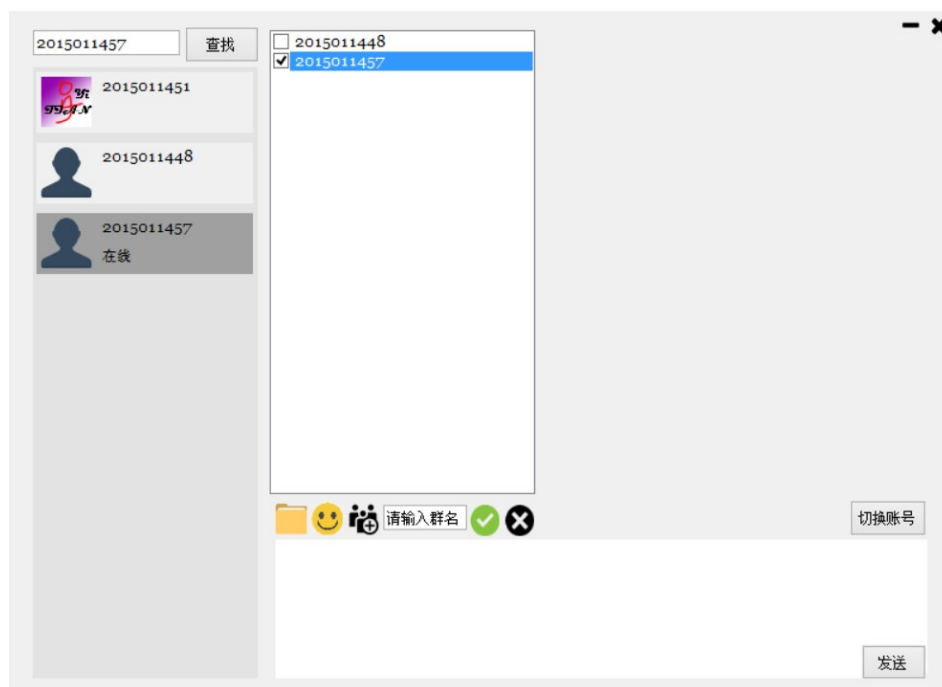


图 24: 发起群聊

发群消息 建群后，不会立刻通知群中联系人，而是在发消息后通知。收到消息的联系人的界面上自动出现以群主发起时设置的群名命名的群。群主以外的其他人可以相互不是好友。

在群中发消息时，消息发送者身份的提示就十分必要了。

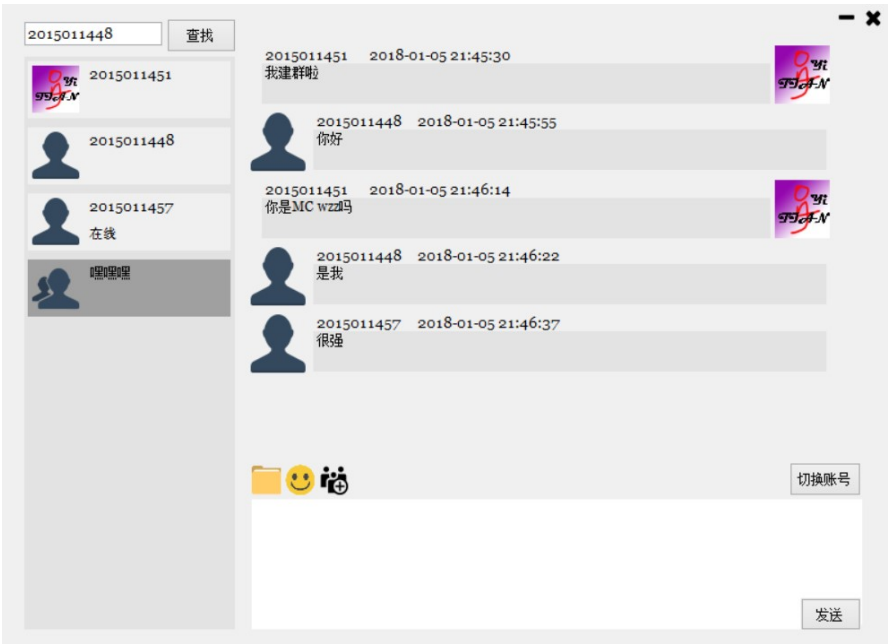


图 25: 发群消息

倘若有群成员不在线，不会像好友消息一样不能发送，而是发送后提示发送者相应不在线的成员，如图所示。这一功能既保证了群消息能顺畅地发送给在线的群成员，又可使发消息者知道不在线的成员。

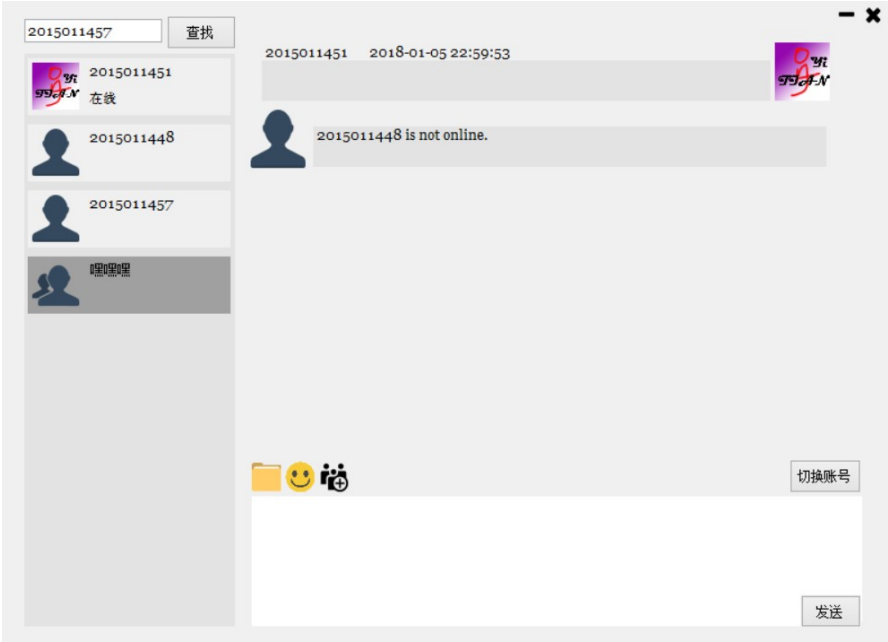


图 26: 群成员不在线

本程序支持二人群聊，即群中只有群发起者（群主）和一个好友。此时群名有“备注”功能。

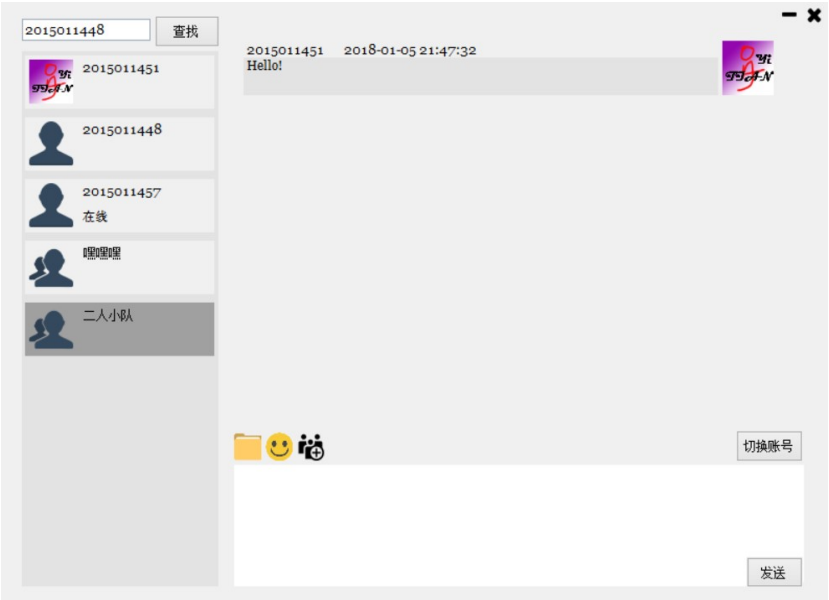


图 27: 二人群聊

切换账号 单击界面中“切换账号”按键，可以切换账号，返回登录界面。在返回前，会向中央服务器发出下线通知，并将下线消息告知用户。

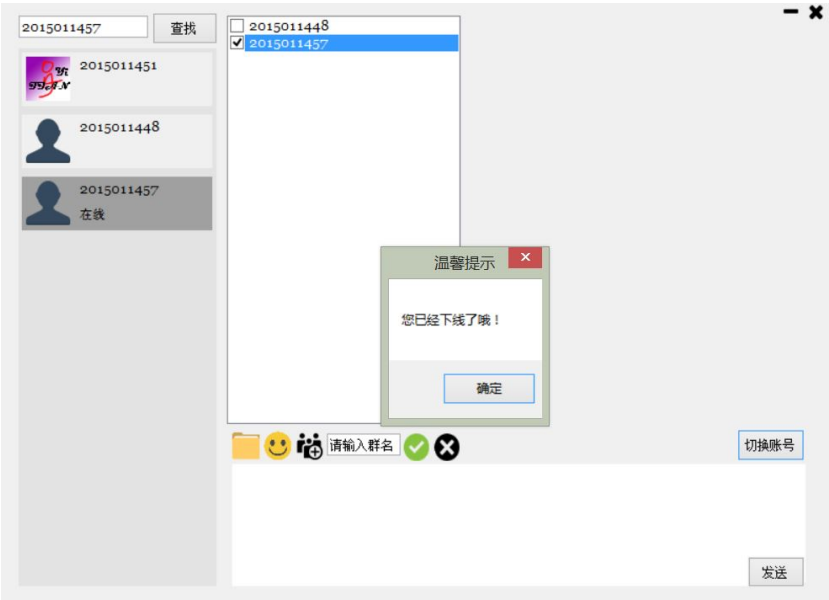


图 28: 切换账号

返回登录界面，可以不同身份登录。



图 29: 返回登录界面

退出 单击右上方“关闭”图标，将弹出询问是否退出的提示。可通过选择“取消”返回；或选择“确定”发送下线消息并提示，释放资源，程序完全退出。

当然，右上方的“最小化”图标也是有效的按钮。

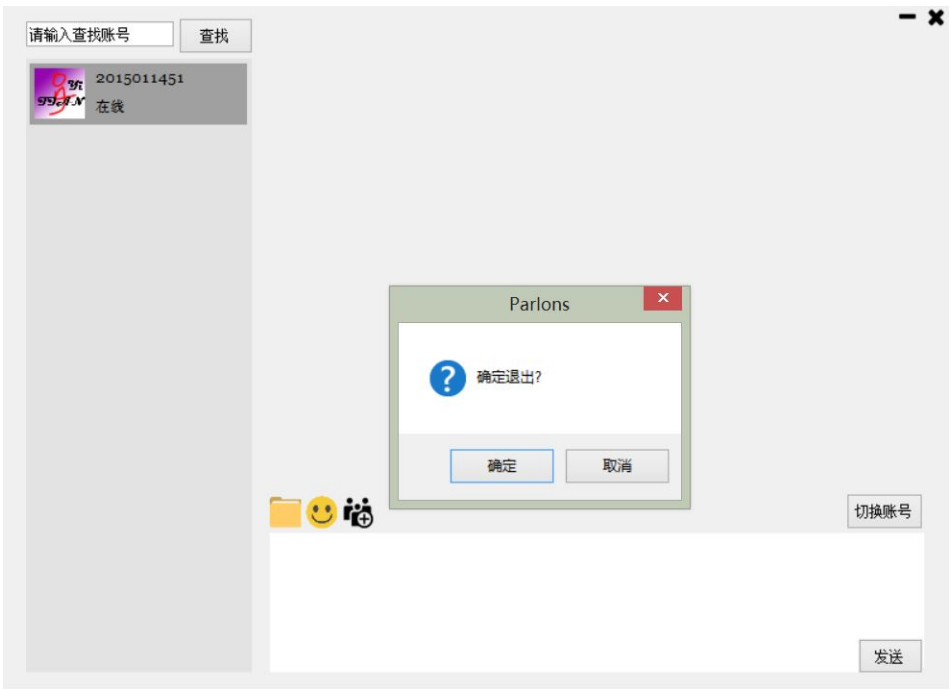


图 30: 退出询问

5 遇到的主要问题及解决

本项目完成过程中遇到过一些问题，多数经过分析与查阅资料都能够顺利解决。在此介绍将两个较为典型的问题。

5.1 程序退出后仍有残存线程的问题

程序最初编写完成后，点击关闭（关闭按钮出发 `Application.Exit` 函数）后仍不能完全退出，以至于再次运行会报端口号已被占用的错误。打开任务管理器，可看到 `Parlons` 仍在运行。经过调试分析，发现问题在于，**线程在程序结束后仍在运行**。

原本希望通过将布尔变量 `listening` 置为 `false` 来实现线程的结束。但发现，在监听时，如果不来新消息，根本不会运行到 `while` 循环条件判断。因此想到，如果在需要结束时，给自己发送一条 `1Byte 0` 的消息，即可奏效。改动后果然成功解决了这一问题。

5.2 表情包大量读取后内存不足的问题

在登录后，会在后台进行表情包读入的操作。共读入 **846 个表情**，使用 `Image.FromFile` 函数时报出内存不足的错误。

为此，与同学交流后，改用两步走的方式：（1）先将图片文件以流的形式读入；（2）再调用 `Image.FromStream`。修改后，这个问题不再出现，这体现了文件流是一种更好的封装。

6 总结

本次大作业累计投出时间在五个整天，全部三天元旦假期和此前的两天都投入到 `Parlons` 项目的完成中来。这五天里，专心做好了这一件事，回顾起来，收获可以概括为以下三方面。

知识增长 在本项目中，我第一次使用 `C#` 进行 `Socket` 编程、第一次使用 `C#` 进行多文件编程、第一次使用 `C#` 进行界面美化。这些增加了我的编程知识，令我印象深刻。而另一方面，在本项目中，我进一步体会了计算机网络及应用课程中学习的 `CS` 架构与 `P2P` 架构的含义；在一次使用 `NAT` 网络测试时更是切身体会到 `NAT` 给 `P2P` 程序带来的影响。知识上的增长是我在本项目上的直接收获。

能力提升 第一，在**编程能力**上，近 2000 行的代码量使得我的编程能力在这次大作业中得到巩固和提升。我遵循统一的命名规范，恰当撰写全英文注释，降低了本项目中代码调试的难度。第二，在本项目中我**快速学习的能力**得到锻炼。逻辑的核心代码在三天内写完，后期所做工作主要为界面调整与逻辑修改。这样的进度需要快速学习、特别是从代码中学习的能力，这种能力将使我以后受益良多。第三，在最初网络通信部分较多参考了现成的代码外，其他大部分内容均由我自己创造性地完成。**基于理解进行创造**，这是在本项目中锻炼的又一能力。

价值塑造 本项目的完成注定是我大学一段难忘的经历。在五天内，专心做好一件事，这种专注是现代生活中所少有的。而当全身心投出进去，有时会感到创作的思绪如泉涌，这种愉快的体验，让我感受到**专注的魅力**。当完成以后欣赏自己的程序，我相信，每个完成这次大作业的同学都会感到**巨大的成就感**，这有力地提升了我们对自己能力的认知，提升了我们解决问题的自信。最后，聊天系统是现代社会的必需品，在大学生活中微信的作用就好像润滑剂对于汽车的作用。亲自编写这样一个类似的聊天系统，打开了聊天软件这个黑盒子，**满足了我对此长久以来的好奇心**——实际上，这也正是一学期以来计算机网络及应用课程最吸引我的地方。

参考文献

- [1] James F Kurose and Keith W Ross. *Computer networking: a top-down approach*. Addison-Wesley Reading, 2010.
- [2] 百度百科. qq. <https://baike.baidu.com/item/QQ>.
- [3] 百度百科. 微信. <https://baike.baidu.com/item/%E5%BE%AE%E4%BF%A1/3905974>.
- [4] Wikipedia. Flat design. https://en.m.wikipedia.org/wiki/Flat_design.
- [5] MSDN. Socket class. <https://msdn.microsoft.com/en-us/library/system.net.sockets.socket.aspx>.