

基于中央定位服务器的 P2P 网络聊天系统设计

班级： 自 73

姓名： 陈昱宏

学号： 2017011507

一、项目介绍：

本次项目是基于中央定位服务器（由助教维护）的 P2P 网络聊天系统，透过 socket 编程实现基本的 TCP 传输，完成网络聊天。

本项目设计的功能如下：

- （一）用户登入和登出。
- （二）好友查询和通讯录维护。
- （三）P2P 模式，基于 TCP 的文字传输。
- （四）P2P 模式，基于 TCP 的文件传输（10M，单对单）。
- （五）创建群组 and 群组聊天。
- （六）语音通话。
- （七）友好的用户界面。

二、编程环境

使用 Win10 系统，Python3.7 进行编程，另外需要安装额外的模块，包括 PyQt5、PyAudio、json。

三、项目结构设计与实现

这一部分将会介绍主要功能实现的原理。

（一）数据报文结构：

数据报的格式我采用字典的形式来设计，分为以下几个字段，用来进行数据类别分类和其他控制操作：

Type 字段	Source 字段	Target 字段	Time 字段	可选字段	Data 字段
数据类型	源用户账户	目的用户账户	发送时间	其他控制	数据内容

其中可选字段中有以下可选内容：

1.FileSize 字段：传输文件模式下，表达文件大小的字段。

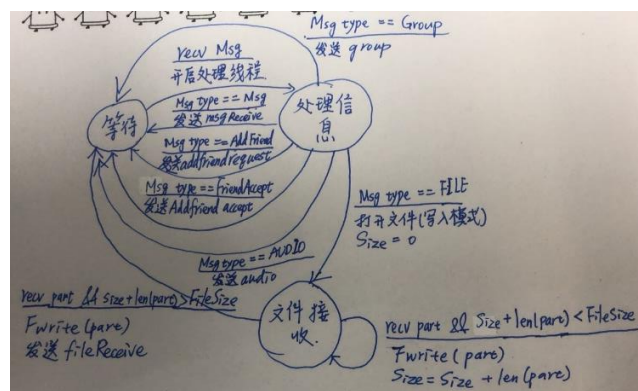
2.GROUPNAME 字段：群组模式下，标记群组名称的字段。

根据上述的报文结构，可以实现各个模式下数据的传输和读取；由于是使用字典的形式封装数据报，在发送前需要先用 json 模块转换成二进制流，接收时也要用 json 模块把二进制流转换成字典形式。

（二）P2P 服务器：

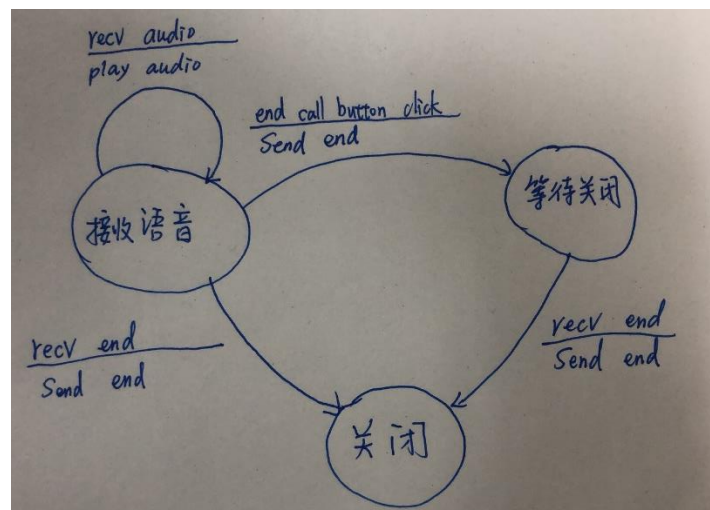
为了实现接收 P2P 模式下的信息，需要开设一个 P2P 服务器用来随时接收，此项目中设计了一个接收消息的服务器和语音通话的服务器。

接收消息的服务器可以同时接受 5 个 TCP 请求，每次收到消息后，会根据数据报的类型执行对应的操作，状态机如下（图中的发送指的是发送信号）：



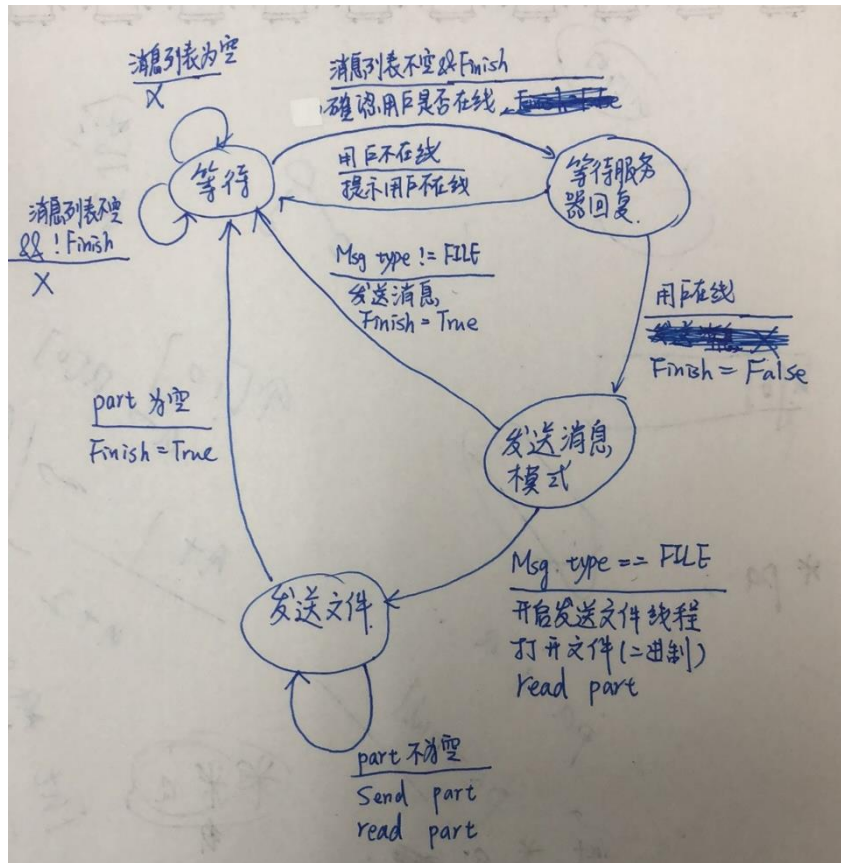
这里需要注意的是服务器的信号会与主界面连接，通知主界面收到什么类型的数据，其中 audio 信号会配置 voicecall 界面并显示。为了方便在同一台电脑调试，服务器端口的设计上，采用学号后五位来当作端口号。

语音通话的服务器主要运行于语音通话期间，主要接收语音信息和挂断控制信息，并且可以播放收到的语音信息，状态机如下：



(三) P2P 客户端：

为了避免发送消息时因为网络波动导致界面卡死，额外开了一个线程去发送消息，采用 P2P 客户端的模式来设计，其状态机如下：



(四) 功能实现介绍:

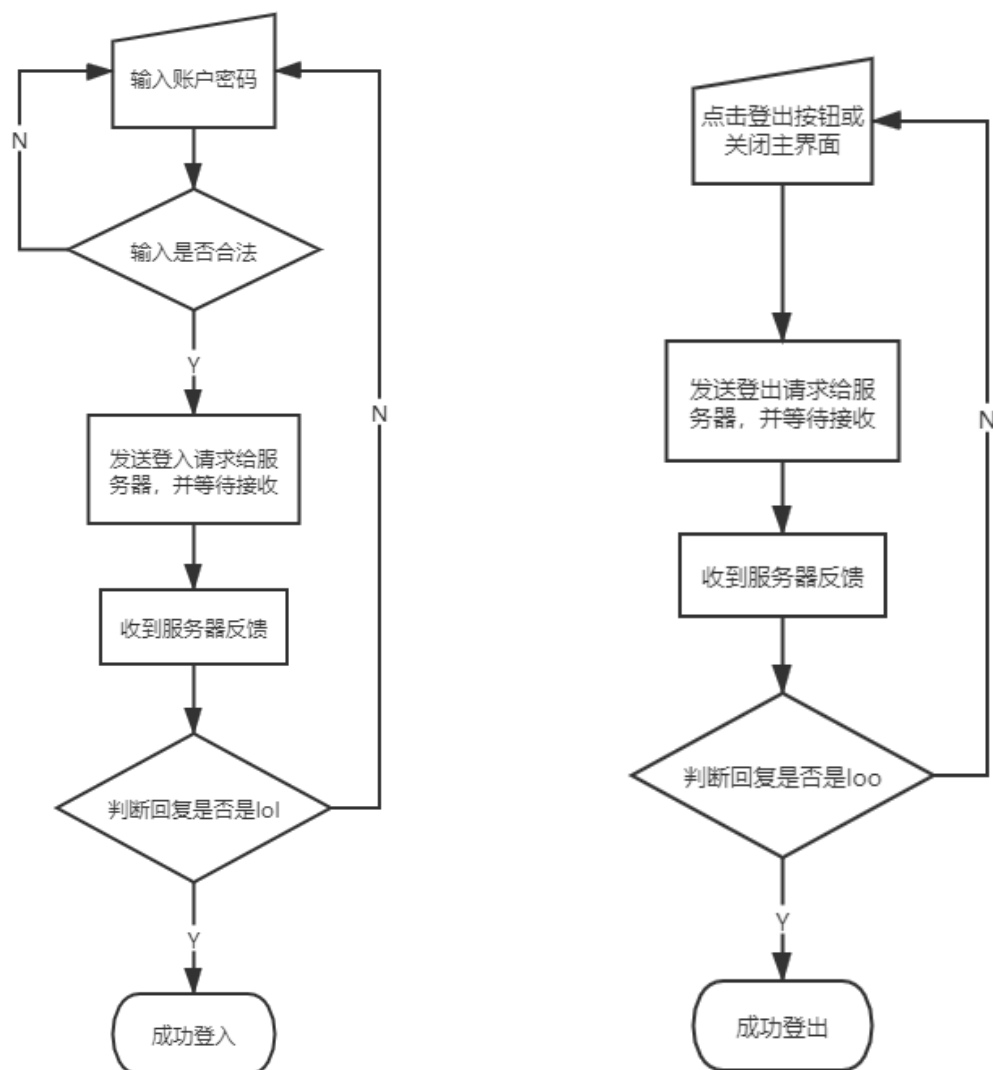
1.用户登入和下线:

此部分功能需要与中央定位服务器交互才能实现，主要是向服务器发送对应的指令，服务器相应指令如下:

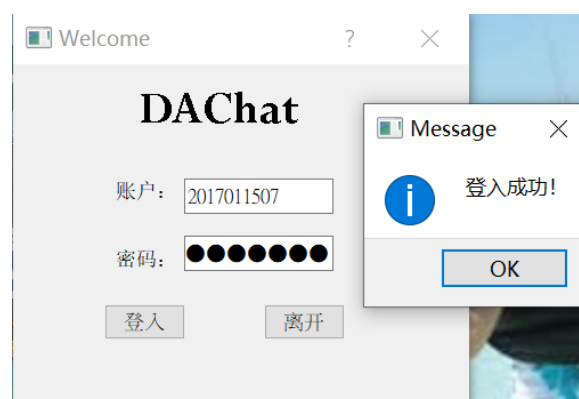
通信类型	客户端发送指令	服务器返还指令
登录	用户名: 本人学号, 密码: net2019 例: "2017011000_net2019"	"lol"
查询好友状态	"q+ 好友学号" 例: "q2019011001"	IP 地址 (在线) "n" (不在线)
下线	"logout+ 本人学号" 例: "logout2019011000"	"loo"

根据服务器返回的信息进行相应的操作，具体流程图如下（登

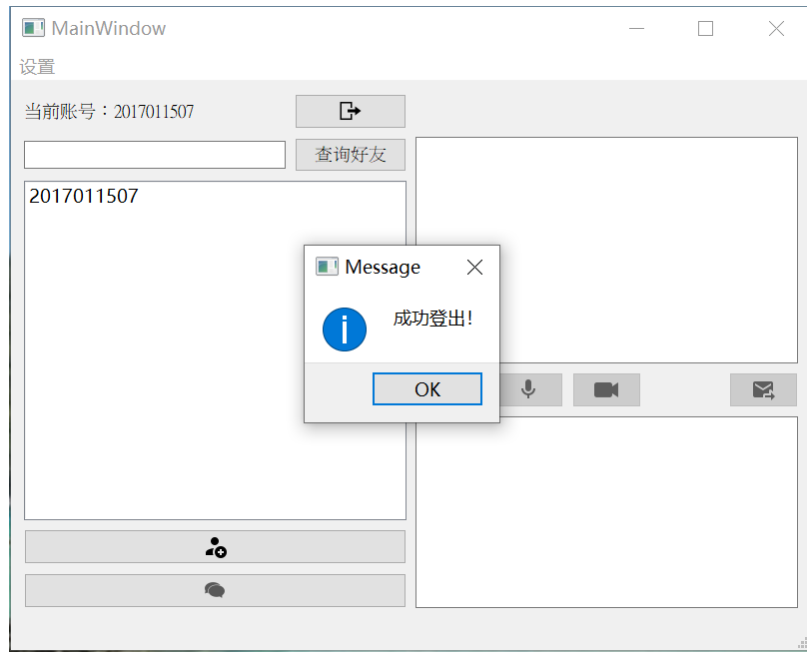
入为左图，登出为右图)：



登入画面如下：



登出画面如下：



2.好友查询与通信录维护：

查询好友是否在线需要中央定位服务器的协助，同样为向服务器发送特定指令，等待服务器的回复。

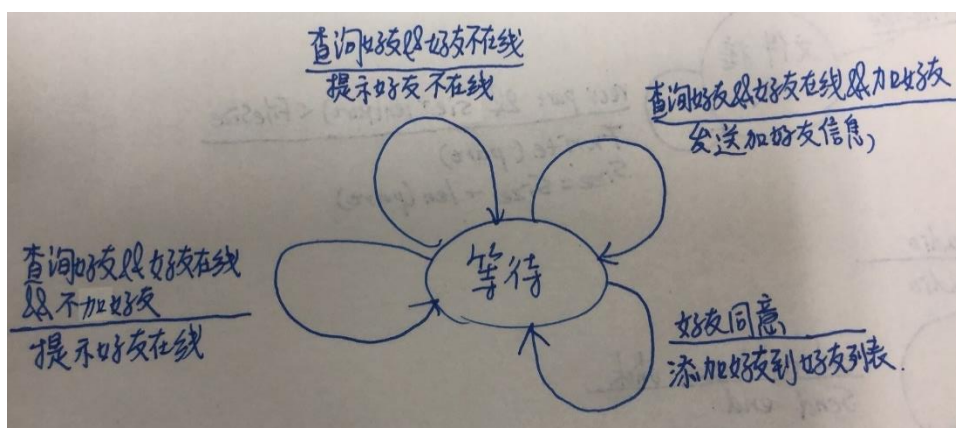
加好友的报文格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	Data 字段
AddFriendRequest	源用户账户	目的用户账户	发送时间	AddFriend

同意好友的报文格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	Data 字段
FriendAccept	源用户账户	目的用户账户	发送时间	Accept

查询好友并加好友的状态机如下：



除了发送好友申请外，也可以点击左下角的加好友图标按钮，可以回复加好友申请，如果同意，会发送同意消息并添加好友；如果拒绝，则直接将申请删除不发生消息。

3.TCP 文字传输：

用户选择想要聊天的好友后，直接在输入框中输入想要发送的消息，点击发送消息的按钮后，即可发送消息。

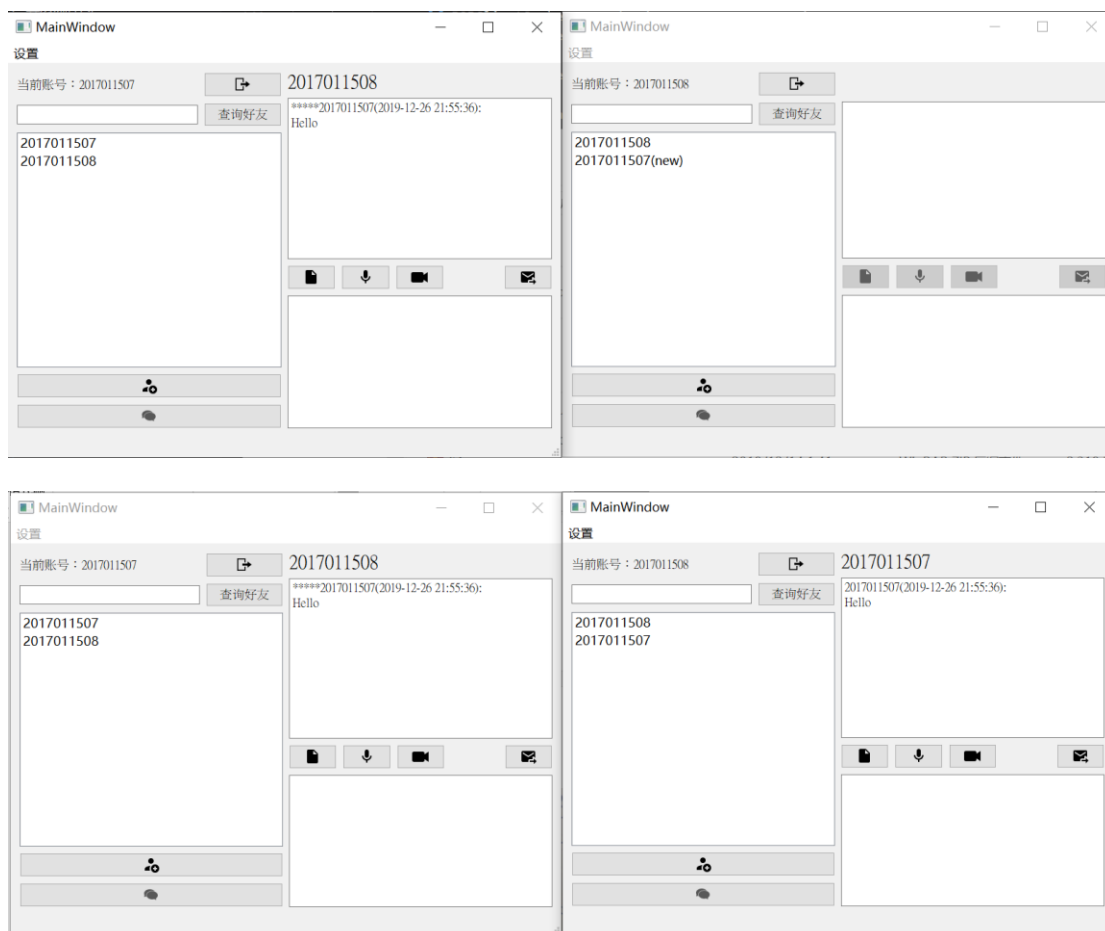
文字传输的报文格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	Data 字段
Msg	源用户账户	目的用户账户	发送时间	信息

在发送消息时，会先将消息交给 P2P 客户端，根据 P2P 客户端的状态机情况进行发送消息。

接受到消息时，根据 P2P 服务器状态机的情况，收到 msgReceive 信号，会将好友列表中的用户更改成“学号（new）”，代表该用户发送了新消息。

具体发送消息的画面如下：



如果是自己发送的消息，会在用户名前显示*****，以区分双方用户。

4.TCP 文件传输（10M，单对单）

用户选择想要发送文件的好友后，点击文件图标的按钮后，可以选择想要发送的文件（会开启文件资源管理器）。

传输文件的报文格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	FileSize 字段	Data 字段
FILE	源用户账户	目的用户账户	发送时间	文件大小	文件名

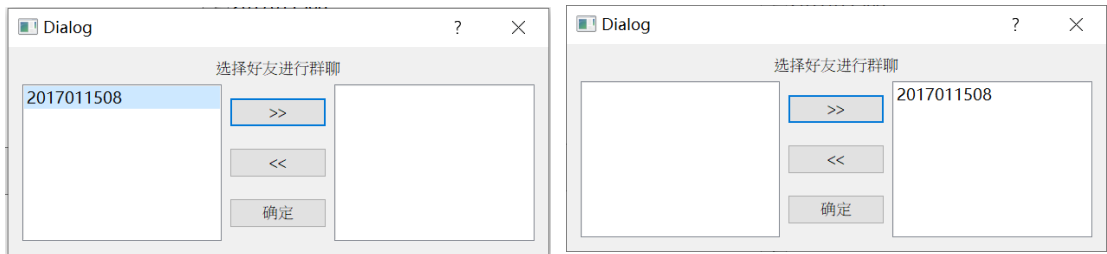
在传输文件时，会首先开启文件传输模式（详见 P2P 客户端的状态机），首先将上面提到的数据报传给对方，让对方知道文件名和

文件大小，随后打开文件以二进制读取，依次将读到的内容传输给对方，直到文件结束。

接收方的 P2P 服务器在收到第一个传输文件数据报之后，会先判断文件夹是否存在，若不存在则创建一个文件夹。之后开启接收文件的模式，依次接收对方发送的二进制流并写入文件中，根据是否接收到指定长度来停止接收。

5.群聊：

点击左下角的群聊图标按钮可以开启创建群聊的界面（如下），可以在左边选取想要邀请进群的好友，也可以在右边移除已选择的好友。



发起群聊的报文格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	GROUPNAME 字段	Data 字段
GROUP	源用户账户	目的用户账户	发送时间	群组名	Group Create

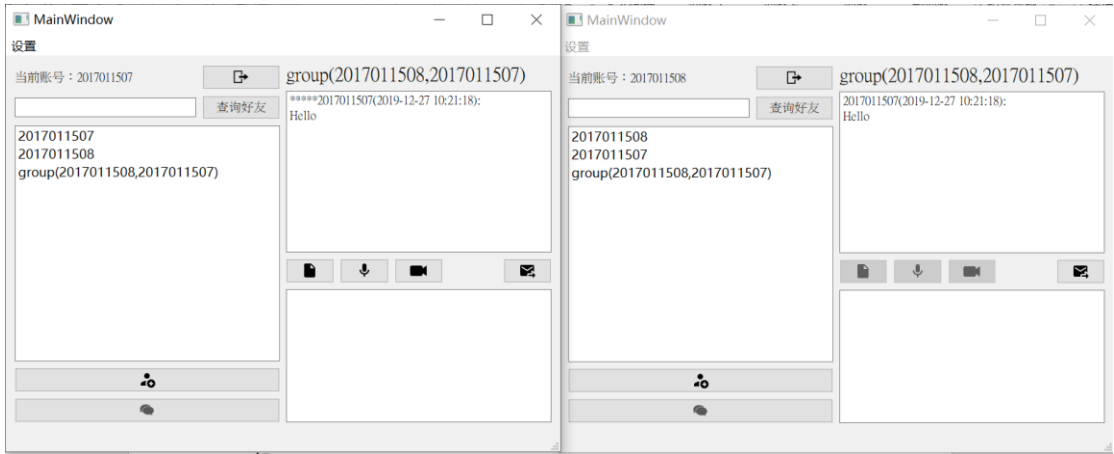
发送群组信息的报文格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	GROUPNAME 字段	Data 字段
GROUP	源用户账户	目的用户账户	发送时间	群组名	信息

在此部分功能需要注意的是，为了避免处理一些很细节的东西，此处不让用户修改群组名，且不能创建两个群组人员相同的群

组。

发送群组消息的画面如下：



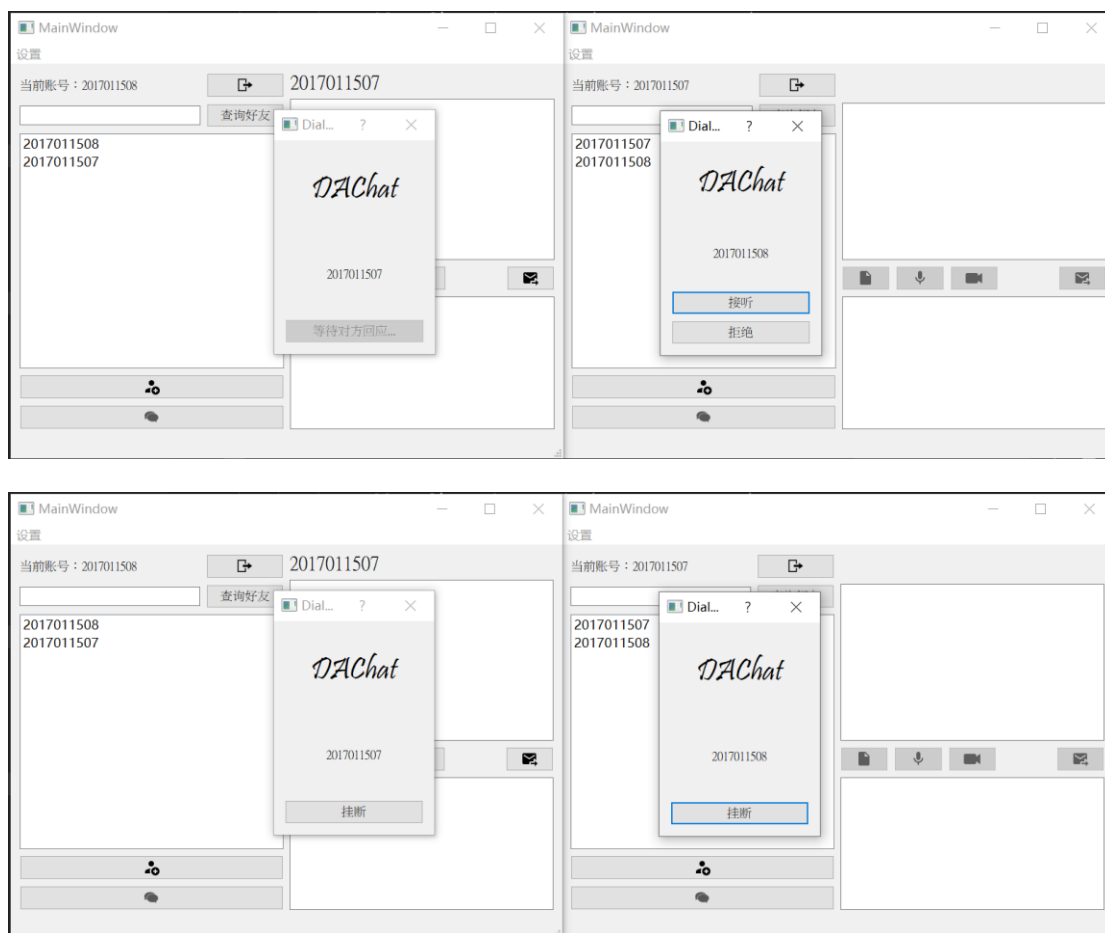
6.语音通话：

用户选择想要语音通话的好友后， 点击语音图标的按钮后， 可以发送语音聊天请求给对方， 等对方同意后， 双方便可以开始语音聊天。

发起语音聊天的数据报格式如下：

Type 字段	Source 字段	Target 字段	Time 字段	Data 字段
AUDIO	源用户账户	目的用户账户	发送时间	自己的 IP 地址

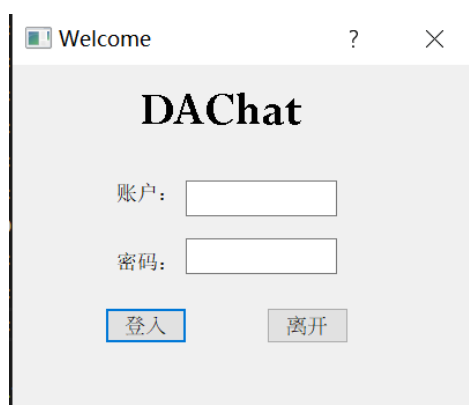
当用户挂断的时候， 会先发送 End 报文， 等待收到对方接收的 End 报文后， 自己再结束会话（详细见语音通话服务器的状态机）。



四、UI 界面设计

本网络聊天程序设计了 5 个界面，分为：登入界面、主界面、回复好友申请界面、发起群聊界面和语音通话界面，以下将会一一展开介绍：

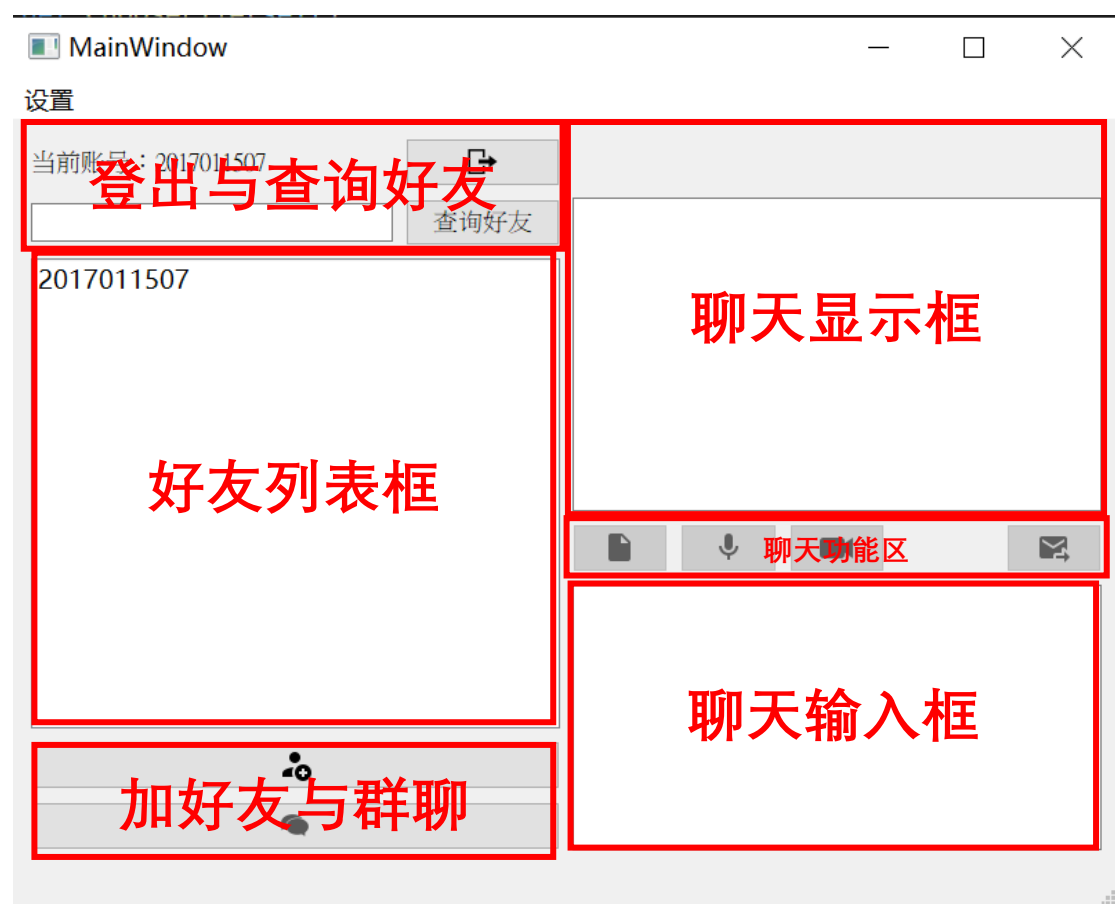
(一) 登入界面：



在账户密码的输入框中输入相应的账户信息，点击登入按钮，如果账户或密码错误会有提示，如果正确的话提示登入成功，点击OK后可以进入主界面。

点击离开按钮，则结束程序。

(二) 主界面：



上图标示了主界面的分区，登出与查询好友区可以进行用户登出和查询好友，在输入框输入好友的账户，点击查询好友即可查询好友是否在线与添加好友。

好友列表框显示你已添加的好友列表；加好友与群聊区中，点

击加好友图标按钮，会开启回复好友申请界面，可以回复其他人发出的好友申请，点击群聊图标按钮，会打开发起群聊界面，可以选择想要群聊的好友。

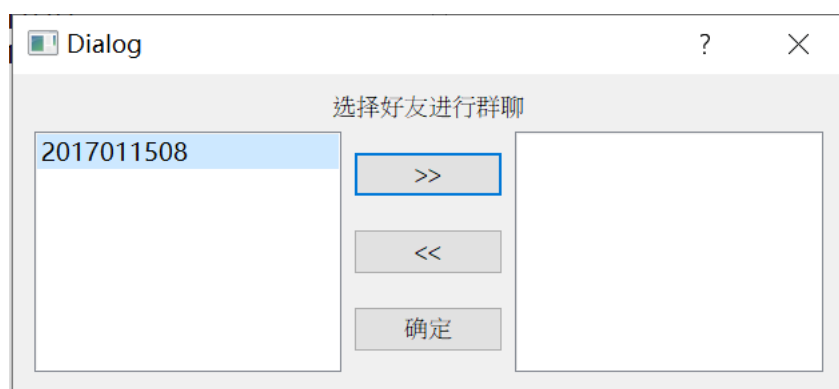
聊天显示框上方空白当选中聊天的好友时，会出现好友的账户名（或群组名），下方的显示框会显示与好友的聊天记录；聊天功能区由四个按钮组成，从左至右分别为：发送文件、语音聊天、视频聊天（此功能未实现）和发送消息的按钮；聊天输入框支持用户输入消息。

（三）回复好友申请界面：



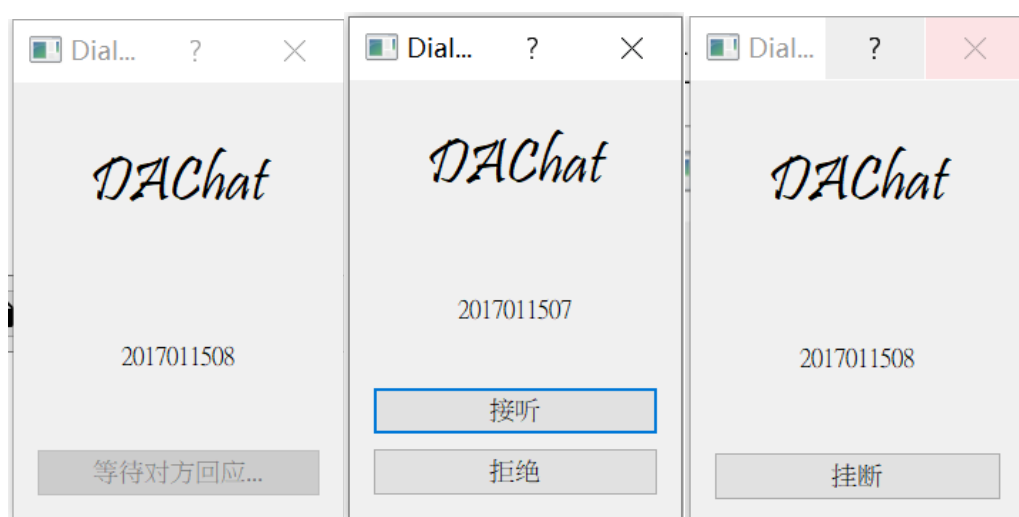
当用户在左边的框选中后，右侧的两个按钮将会变成可以按的模式，点击相应的按钮，可以实现同意申请或拒绝申请。

（四）发起群聊界面：



中间三个按钮从上到下功能分别为：添加好友进入群聊（从左框移到右框）、将好友移除群聊（从右框移到左框）和确定群聊人员。用户选择完群组人员后，点击确定按钮可以创建相应的群组。

（五）语音聊天界面：



上图从左到右分别为拨打电话中、接收方显示、通话中的三个不同情况下的界面，界面由上到下分别是：软件标签、通话的好友账户、功能按钮，可以根据按钮上的提示说明来执行相应的功能。

五、创新点和使用注意事项：

本项目采用多线程的技术来实现，除了与服务器交互的登入登出和查询好友之外，其余的网络功能采用多线程实现，避免界面卡

死（由于用户在使用与服务器交互的功能时，不太会出现同时做其他事情的情形，所以并不需要使用多线程）；用户在不关闭程序的前提下，所有的聊天记录都会保存下来，当用户更换聊天对象时，会自动显示与该聊天对象的聊天记录；此外，当用户登出账户时，会自动切换会登入界面，可以直接重新登入或是登入其他账号。

注意事项：

（一）请不要删除名为 data 和 resources 的文件夹，否则文件传输和界面图标会出现问题。

（二）P2P 服务器我只允许同时有 5 个连接，所以请不要同时使用大于 5 个用户对同一用户发起连接请求。

（三）由于端口号使用的是学号后五位，请不要在同一台电脑上登入两个后五位学号一致的用户，除此之外，由于语音通话会占用 8000 和 8003 两个端口，请注意不要使用到后五位为 08000 或 08003 的用户。

六、实验心得与体会：

本次实验项目是我第一次利用网络编程来实现一个项目，在过程中也遇到了一些关于 socket 的 bug，但经过上网查询和与同学讨论后，最后都成功解决了，这让我学习去利用手边的资源来完成以前没有接触过的任务。

这次的项目最理想的状态应该是要将前端和后端分离，但由于我在设计前并没有事先想好，导致我的前后端混在一起编程，也导致后续新功能增加和调试都变得较为困难，希望我下次能够在编写项目前，事先想好具体结构设计再进行编写，完善自己的代码质量。

七、参考文献：

- [1]使用 PyAudio 模块播放音频流之体,<https://www.jianshu.com/p/ba82e90ce706>
- [2] Python 音频操作工具 PyAudio 上手教程详解,<https://www.jb51.net/article/163992.htm>