

# 计算机网络及应用

## 大作业

**学号**      2017011589

**姓名**      吾尔开西

**班级**      自 76

## 目录

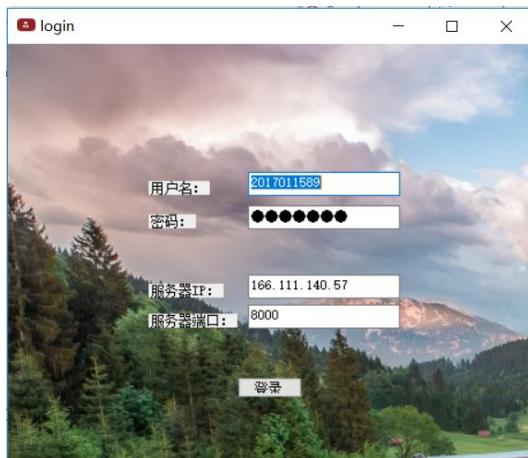
一、功能介绍.....	3
1、登录.....	3
2、查询好友是否在线，添加好友/接受好友请求.....	4
3、与好友文字聊天.....	6
4、文件传输.....	7
5、多人文字聊天（选做）.....	10
6、P2P 文件分发（选做）.....	10
7、UDP 协议通信（选做）.....	12
8、设计模拟服务器（选做）.....	13
二、通信设计.....	14
1、与服务器的通信.....	14
2、添加好友.....	14
3、单人聊天.....	15
4、发起群聊.....	16
5、群聊.....	17
6、UDP 文字聊天.....	18

7、模拟服务器 .....	20
三、通信协议 .....	20
1、添加好友 .....	20
2、单人聊天 .....	20
3、群聊 .....	21
四、界面设计 .....	23
1、客户端 .....	23
2、模拟服务器 .....	27
五、总结 .....	28

## 一、功能介绍

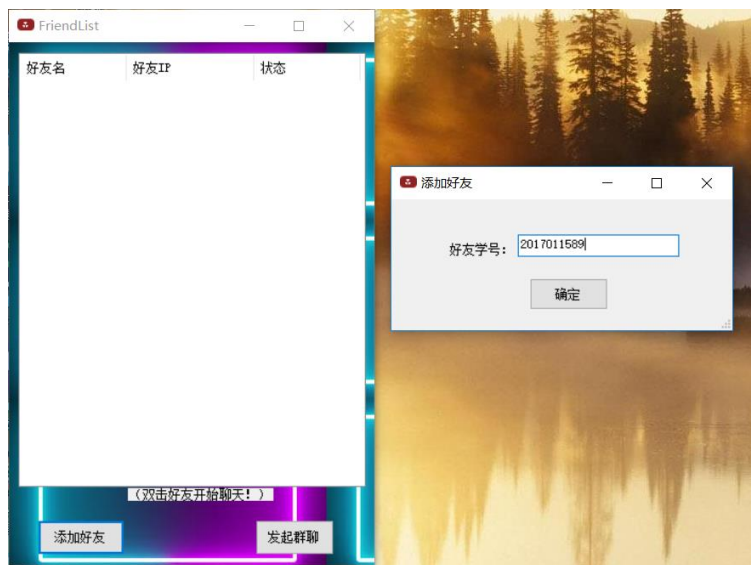
### 1、登录

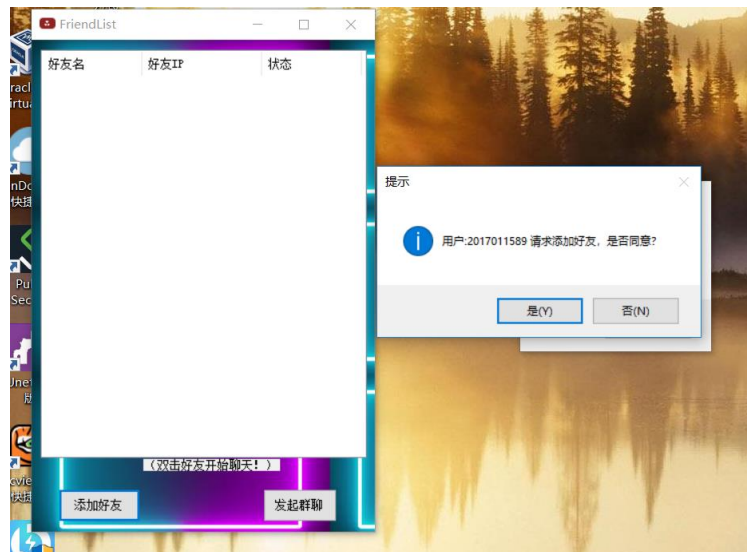
用户凭用户名和密码登录系统，可以更换服务器 IP 和端口，以防网络不畅通，连接服务器有 3 秒的超时时延。



## 2、查询好友是否在线，添加好友/接受好友请求

用户可以通过好友学号添加好友，如果好友在线，发出好友请求，对方可以选择接受或拒绝好友请求。





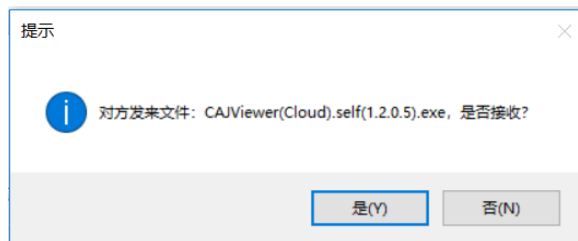
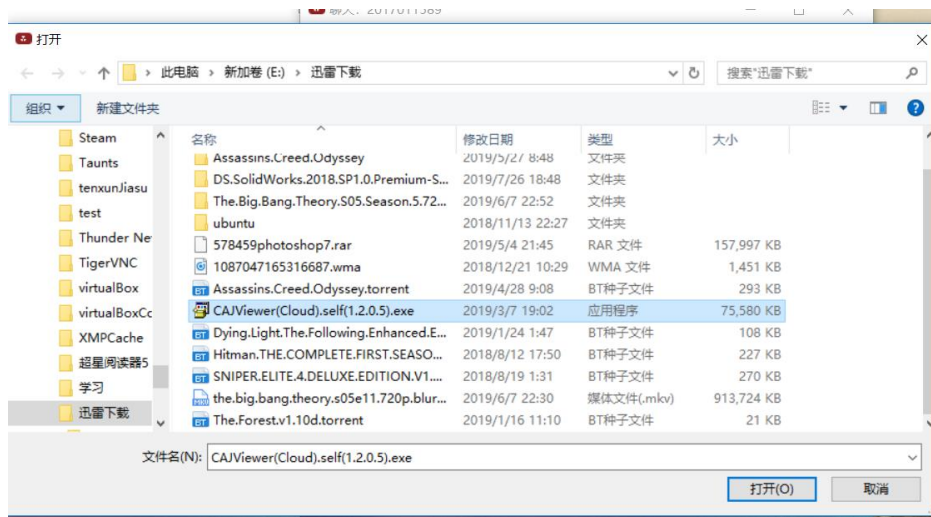
### 3、与好友文字聊天

添加好友后，可双击好友名打开聊天窗口，与好友进行文字聊天。

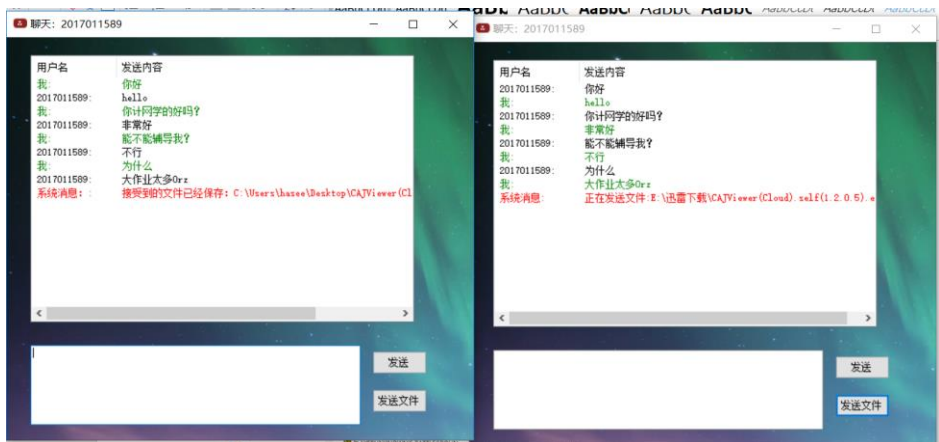
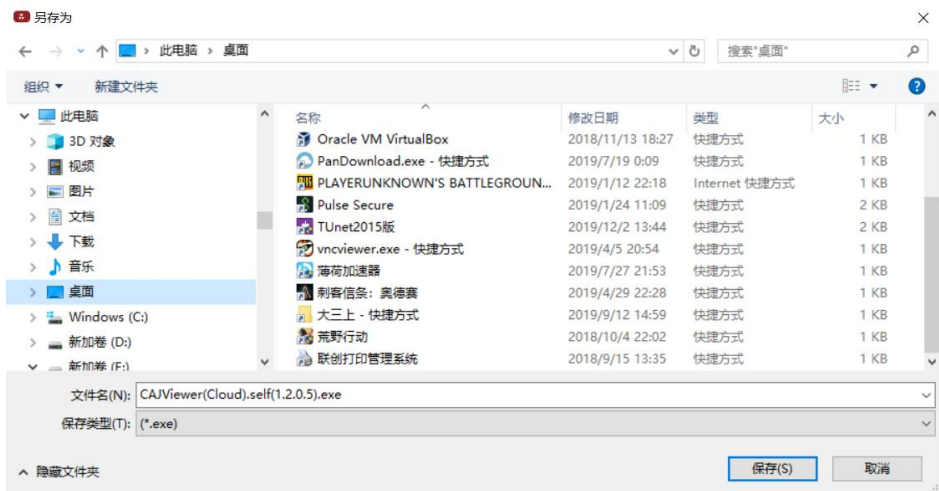


#### 4、文件传输

在聊天窗口，用户可以向好友发送文件，好友收到文件名和大小后决定是否接收。

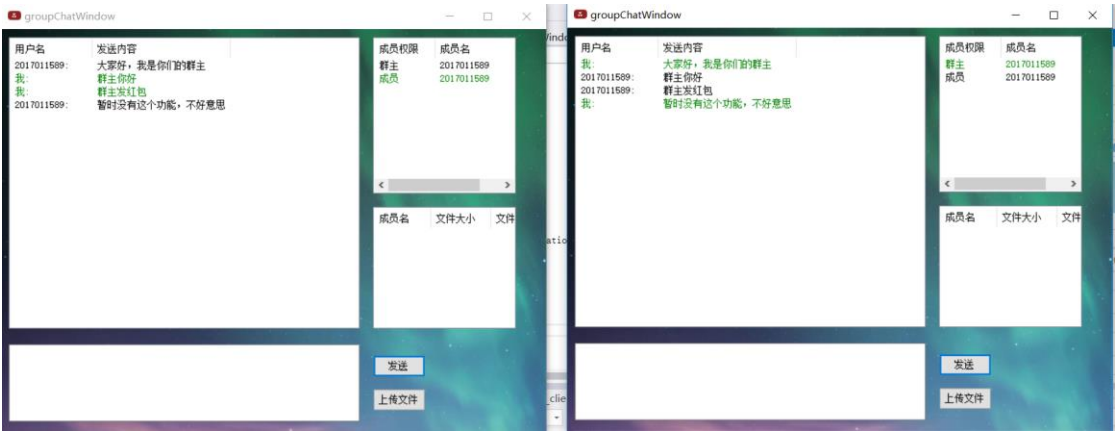






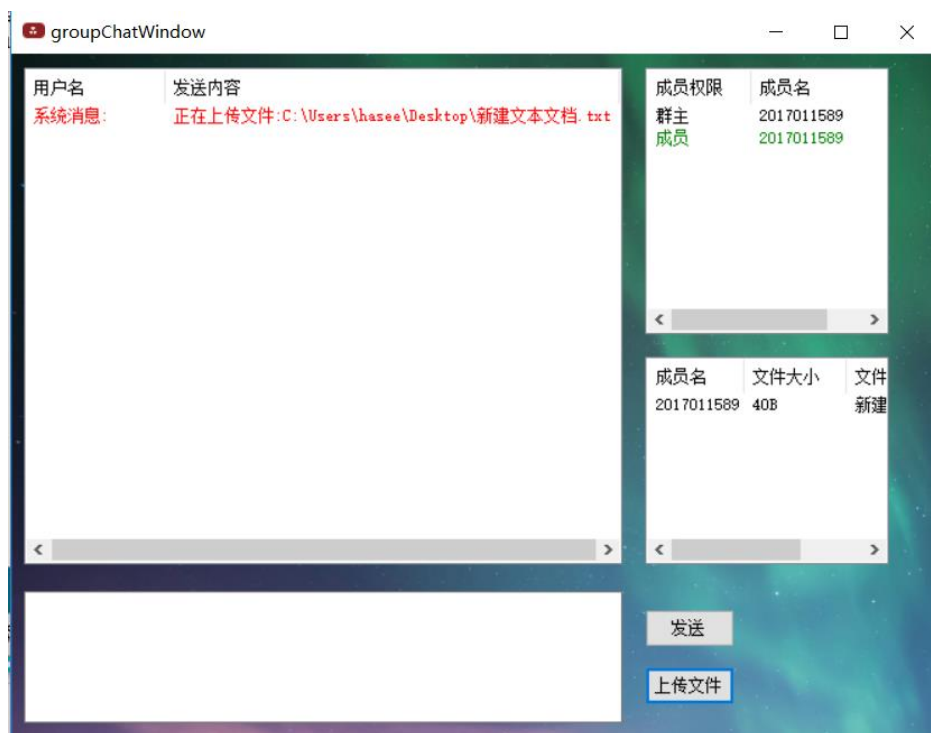
## 5、多人文字聊天（选做）

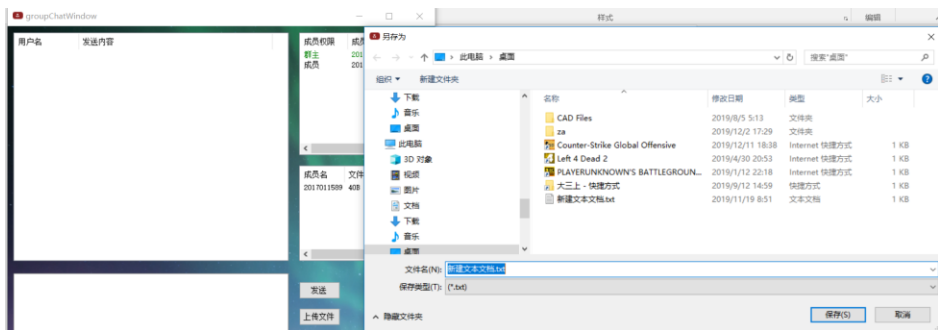
用户可以选择多个用户发起群聊，成为群主。群聊窗口打开后，群成员们可以多人聊天



## 6、P2P 文件分发（选做）

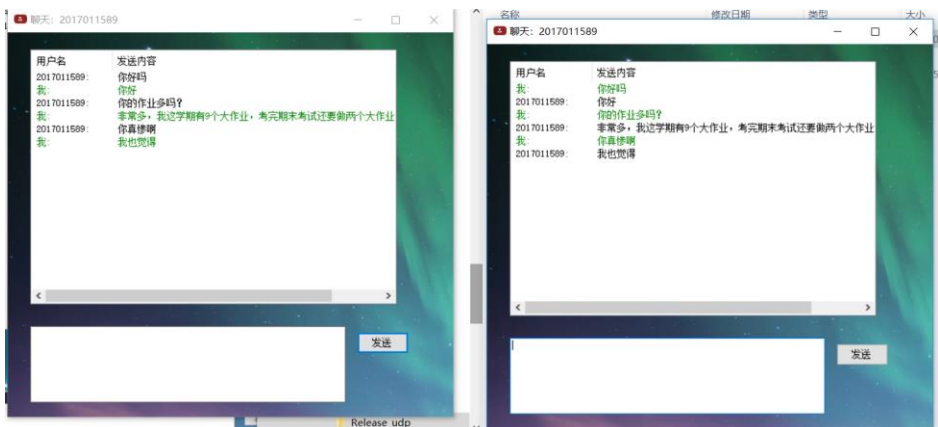
在群聊窗口中，每个群成员都可以上传文件，群成员上传的文件信息展示在群文件列表中。每个群成员都可以在群文件列表中双击下载文件。





## 7、UDP 协议通信（选做）

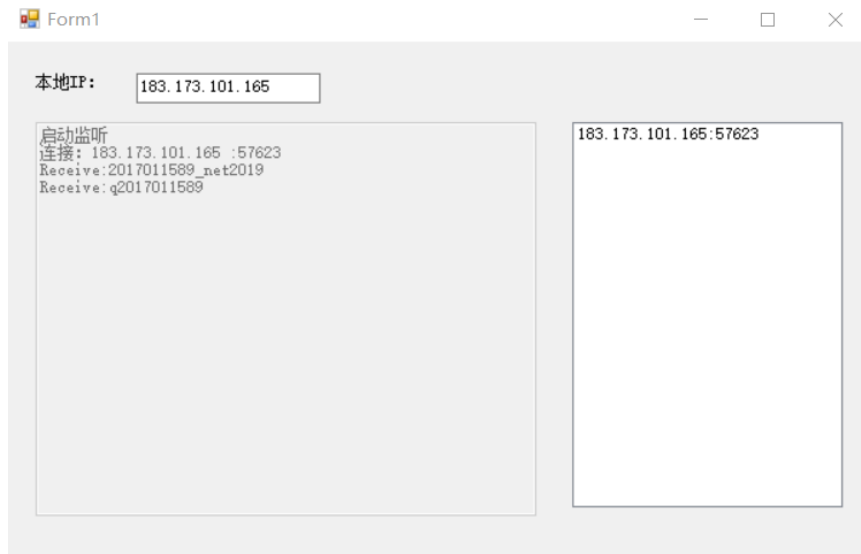
完成了使用 UDP 协议的文字聊天功能，这个功能的实现是在一个单独的工程中，因为如果两个客户端使用不同的协议，则二者不能通信，所以一个工程中最好使用统一的通信协议。



## 8、设计模拟服务器（选做）

为了调试的方便，我设计了模拟服务器（助教在课程群里说这可以作为选做之一）。

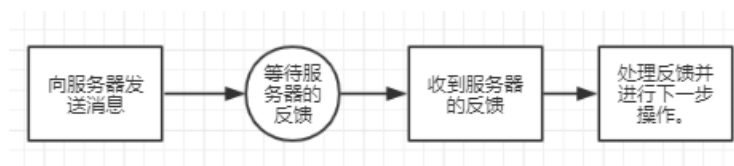
模拟服务器能完成助教提供服务器的所以功能，同时在 UI 界面显示收到的消息。



## 二、通信设计

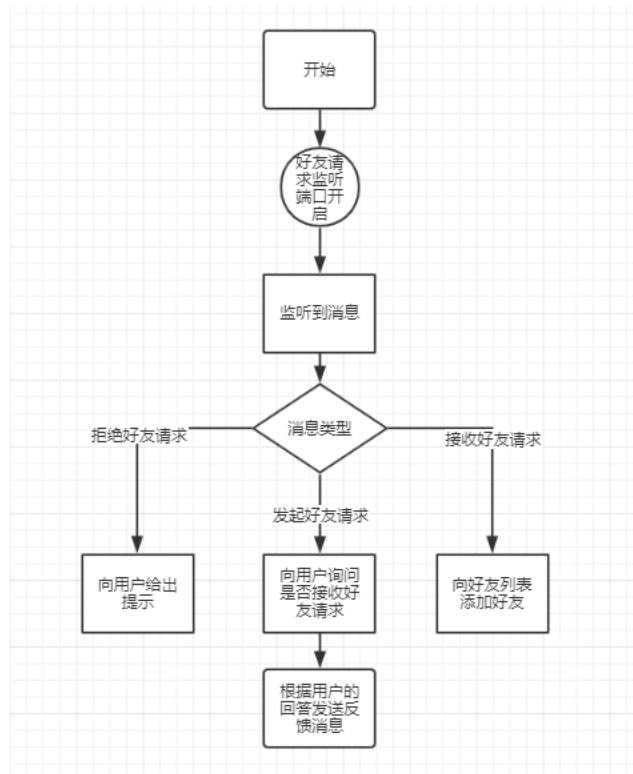
### 1、与服务器的通信

登录、查询好友状态等操作需要与服务器进行通信。由于这些操作一般不需要与其他操作同时进行（不需要并行），可以让用户等待服务器的回应，所以用简单的串行同步通信完成。



### 2、添加好友

每个客户端都需要设置一个好友请求的监听端口（异步监听与异步接收），当监听并收到到好友请求时，用户选择是否接收好友请求，根据用户的选择向对方的好友请求监听端口发送回应。



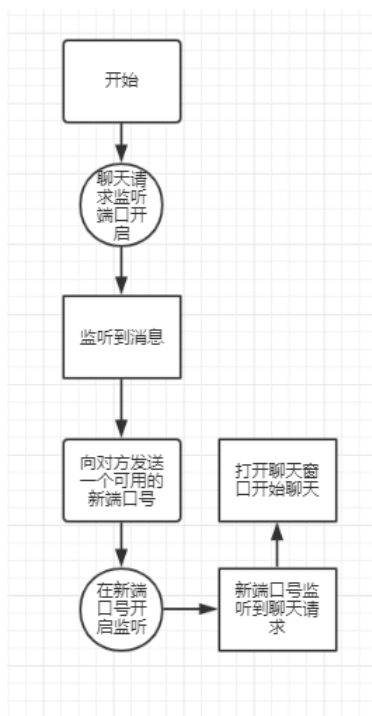
### 3、单人聊天

每个客户端都需要设置一个聊天请求的监听端口（异步监听与异步接收）。

在用户双击好友名时，会向对方的聊天请求监听端口发送聊天请求。

当客户端聊天请求的监听端口接收到连接时，马上向对方发送自己另一个可用的端口号，并在这个端口号开启监听，对方便会向这个端口号发起连接。

双方在新的端口连接成功后，创建聊天窗口，并传入这个 TCP 连接，双方之后的通信（异步接收与发送）都在这个 TCP 连接内完成。



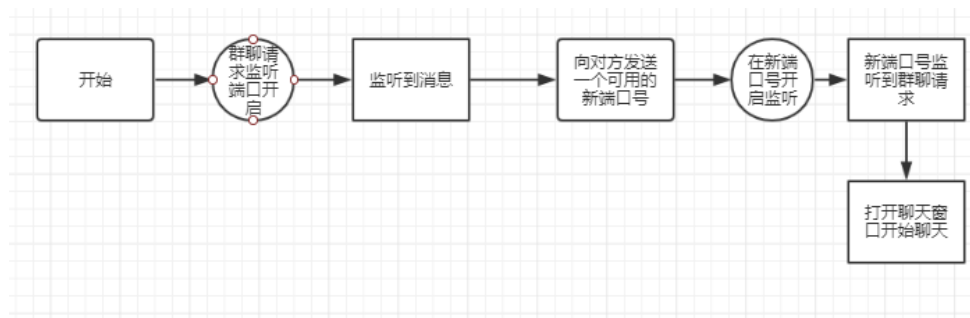
#### 4、发起群聊



每个客户端都需要设置一个群聊请求的监听端口（异步监听与异步接收）。

当用户选择多个好友并点击“发起群聊”按钮时，会向所选的每个好友的群聊监听端口发送群聊请求。

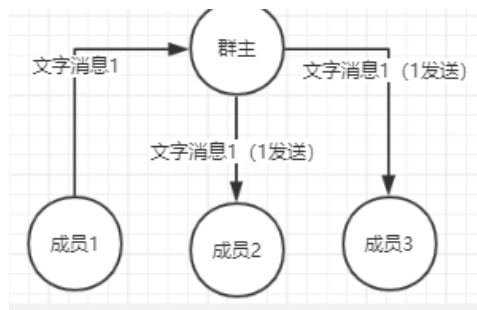
当客户端群聊请求的监听端口接收到连接时，马上向对方发送自己另一个可用的端口号，并在这个端口号开启监听，群聊发起方便会向这个端口号发起连接。



## 5、群聊

在群聊过程中，为了保证群主的权限，以及避免冗余的传输，群主将承担消息中转站的作用。每个人的消息将发送给群主，群主再将这些消息发送给除发送人以外的每个群成员。

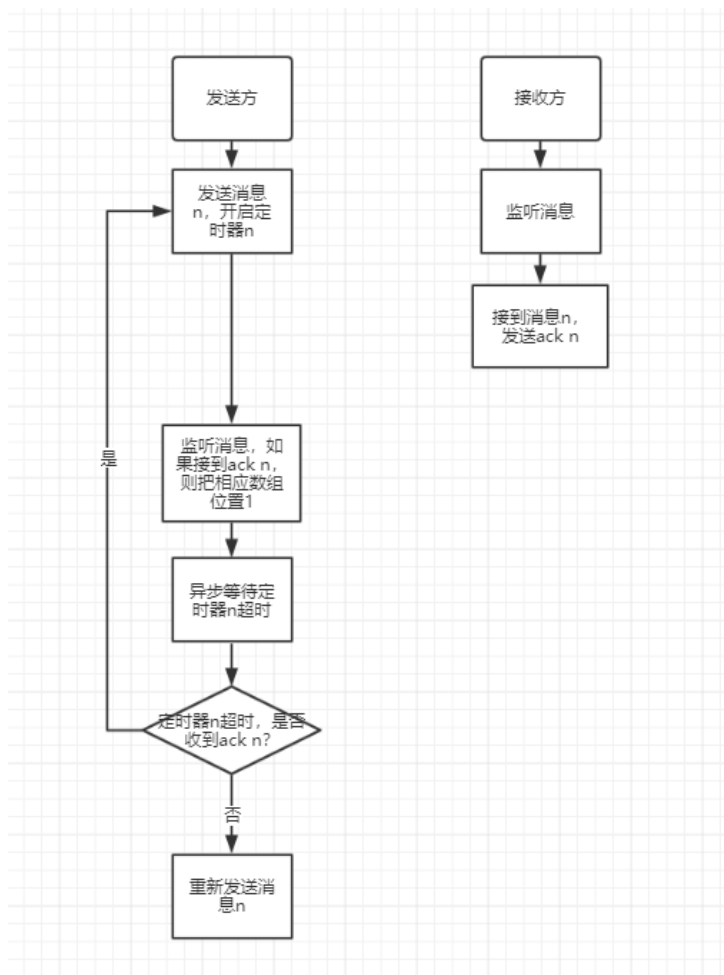
文件的信息（包括文件名和文件大小）会像文字消息一样分发，但文件内容会存储在群主本地，当有用户请求文件时向其发送。



## 6、UDP 文字聊天

UDP 文字聊天中，有单独的接收线程，如果接收到一条消息，则向对方发送 ack n；如果接收到 ack k，则记录下来。

当发送消息 n 时，开启 5 秒的定时器 n，如果定时器 n 当时，还没有收到 ack n，则向用户提示并重新发送。



## 7、模拟服务器

模拟服务器的设计采用了经典的服务器监听设计，服务器在端口 8000 进行监听，当有客户端发送消息时开辟一个新的线程对其内容进行处理并回复。

# 三、通信协议

## 1、添加好友

在添加好友的端口，由于接收方有拒绝接收好友请求的功能，所以需要 一个通信协议。双方消息类型有三类：添加好友请求、接受好友请求、拒绝好友请求。`message` 为消息内容。

消息类型	<code>message</code>
添加好友请求	<code>'_f'+对方用户名</code>
接受好友请求	<code>'_y'+对方用户名</code>
拒绝好友请求	<code>'_n'+对方用户名</code>

## 2、单人聊天

单人聊天时，双方之间发送的消息可能有三类：文字消息、文件内容、文件名。

将一方收到的字节流设为 `message`，采用下面的通信协议：

消息类型	<code>message[0]</code>	<code>message[1:end]</code>
文字消息	1	文字（string）
文件内容	2	文件内容（bytes）
文件名	3	文件名（string）

### 3、群聊

群聊的情况比较复杂，群聊中分两种用户：群主与群成员。

群主从群成员处收到的消息有 5 种，设群主收到的字节流为 `message`，采用下面的通信协议：

消息类型	<code>message[0]</code>	<code>message[1:end]</code>
文字消息	0	文字（string）
文件内容	2	文件内容（bytes）
文件长度和文件名	3	文件长度（string）+“_”+ 文件名（string）

用户退出消息	4	“q” (string)
请求文件	5	文件长度 (string) + “_” + 文件名 (string)

群主需要向群成员发送几种消息，包括聊天信息（某位成员发送的信息）、群成员名（群聊刚建立以及有成员退出群聊时，群主向每位成员发送全部成员名）、文件内容（当群成员向群主申请文件时，群主向该成员发送文件内容）、文件信息（当群里有其他成员上传文件时，群主向大家发送文件信息）、群组解散、没有找到文件（群主没有找到成员申请的文件）。

设群成员收到的字节流为 `message`，采用下面的通信协议：

消息类型	<code>message[0]</code>	<code>message[1:end]</code>
文字消息	0	作者名 (string) + “_” + 消息内容 (string)
成员名	1	所有成员名 (string) 拼接，群主名在第一个。
文件内容	2	文件内容 (bytes)

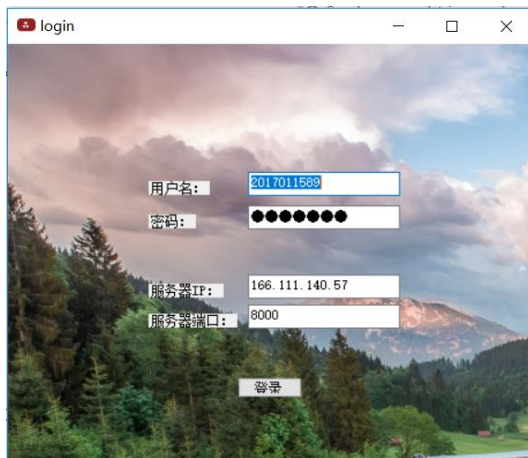
文件消息	3	文件长度 (string) +“_”+上传者名字 (string)+”_”+文件名 (string)
群组解散	4	“q”
没有找到文件	5	

## 四、界面设计

本项目的界面包括客户端和模拟服务器两大部分，客户端包括登录界面、好友列表界面、聊天窗口和群聊窗口。

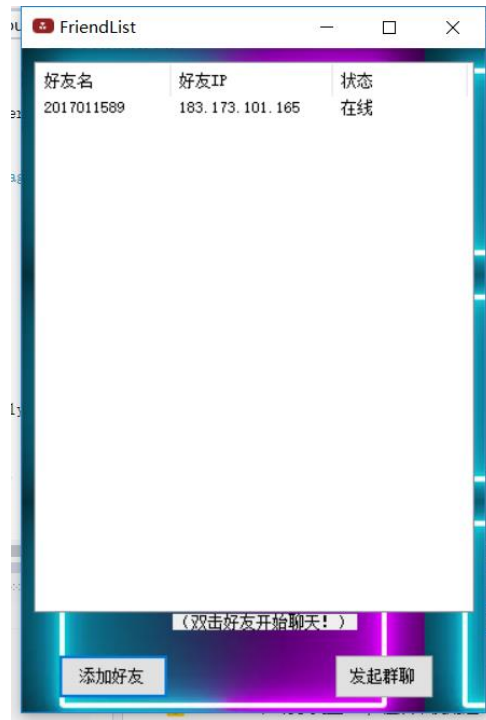
### 1、客户端

在登录界面、用户可输入用户名、密码、服务器 IP 和端口。这些都有默认值，用户只需进行修改。

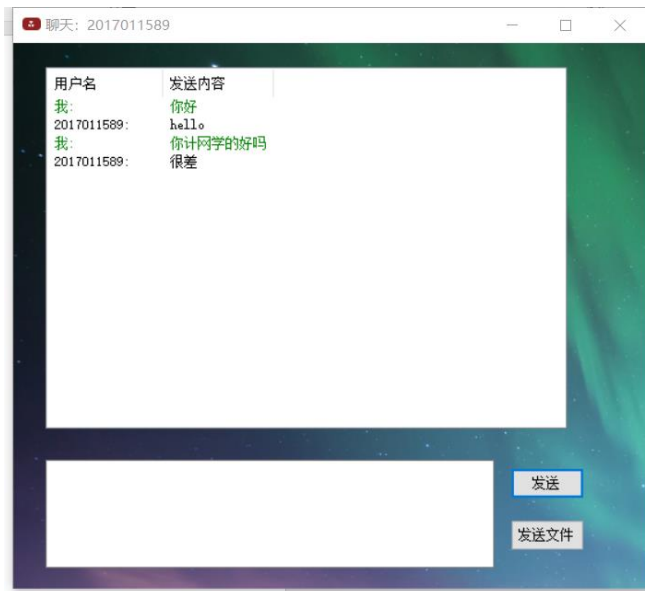


在好友列表界面，用户可实现添加好友、发起聊天、发起群聊功能。用户添加的好友会显示在好友列表，用户双击好友名发起聊天。用户在好友列表单击选中若干好友，点“发起群聊”按钮发起群聊。





在聊天窗口，用户可输入并发送文字信息，也可以向对方发送文件。

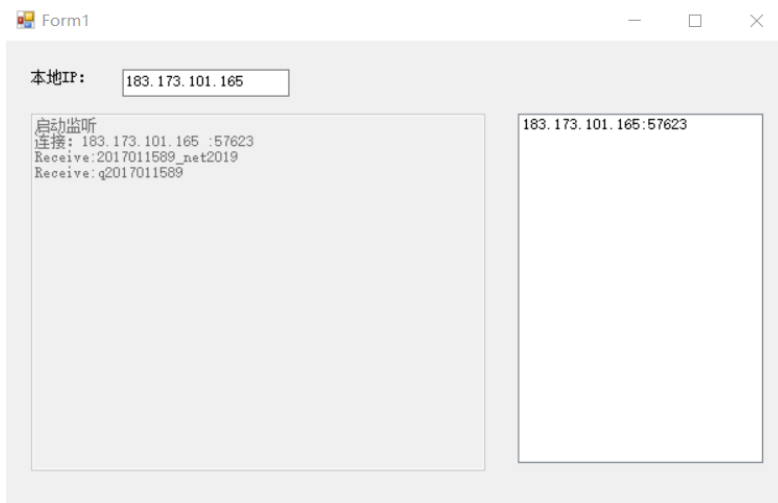


在群聊窗口，用户可以输入文字信息和上传文件。用户上传的文件显示在群里所有人的文件列表内，每个人都可以双击某文件来下载。



## 2、模拟服务器

模拟服务器能完成助教提供服务器的所以功能，同时在 UI 界面显示收到的消息。



## 五、总结

这次大作业完成了相对比较完整的聊天软件的设计，体验了自己设计软件通信、软件界面的感受，属实不易。在完成大作业的过程中遇到不少问题，比如两个客户端通信的时序问题、双方握手的协议问题等。

软件的一个界面内可能需要完成多个任务、需要监听多个端口，这就要求设计者有一个清晰的思路和鲁棒的通信方式，否则很容易陷入混乱。比如两个客户端在好友列表界面发起聊天，如果直接将二者发起聊

天的 socket 连接传入聊天窗口进行通信，会出现通信与监听混乱的问题。所以需要当二者第一次连接后，在新的端口重新进行连接。

总之，通过完成这次大作业，我对课本理论知识的理解得到了提高，动手能力和解决问题的能力得到锻炼。将理论知识应用到实际中，给我带来了很大的成就感。