

基于中央定位服务器的 P2P 网络聊天系统设计

林嘉成 2016011498
(自动化系 自 66)

目录

1	需求分析	2
2	总体设计	2
2.1	整体流程图	2
2.2	状态机	2
2.3	选做内容	3
2.4	开发环境	3
3	详细设计	3
3.1	界面层	3
3.2	逻辑层	4
3.2.1	账号登录上线及下线	4
3.2.2	通讯录 (查询好友是否在线)	4
3.2.3	P2P 通信	4
3.2.4	文件传输	4
3.2.5	聊天记录查询	4
3.2.6	动态表情发送	5
3.2.7	好友查询及跳转	5
3.2.8	新消息提醒	5
3.2.9	实时显示提示信息	6
3.3	数据层	6
4	结果分析	7
4.1	登录界面	7
4.2	聊天主界面	7
5	总结与反思	11

1 需求分析

即时通讯 (Instant Messaging) 是目前 Internet 上最为流行的通讯方式，各种各样的即时通讯软件也层出不穷；服务提供商也提供了越来越丰富的通讯服务功能。不容置疑，Internet 已经成为真正的信息高速公路。

微信 (Wechat) 是腾讯公司于 2011 年 1 月 21 日推出的一个为智能终端提供即时通讯服务的免费应用程序。截止到 2016 年第二季度，微信已覆盖中国 94% 以上的智能手机，并由其界面简练的特性使得多数人放弃使用 QQ 转而使用微信。

本项目要求实现基于中央定位服务器的 P2P 网络聊天系统设计。客户端与服务器之间使用 C-S 通信，客户端之间对话使用 P2P 进行通信，这两点即为本项目需要实现的核心。

2 总体设计

2.1 整体流程图

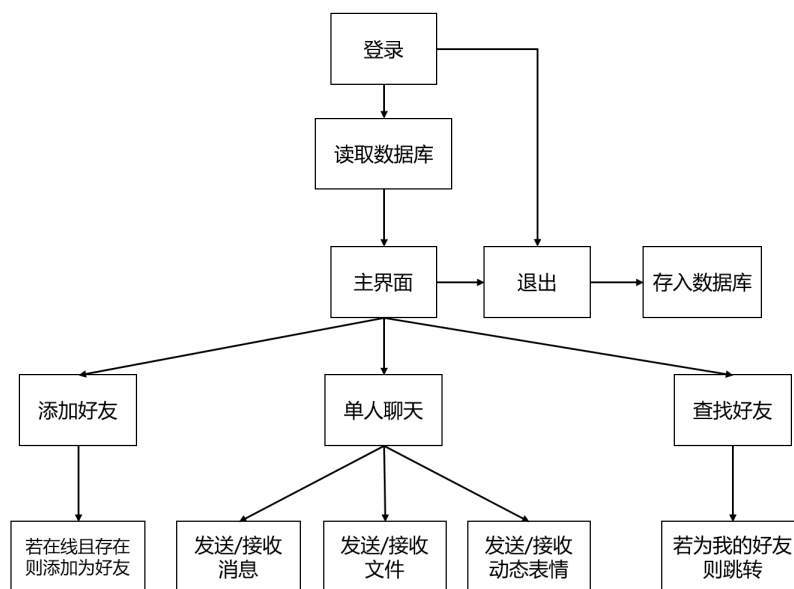


图 1: 整体框架图

2.2 状态机

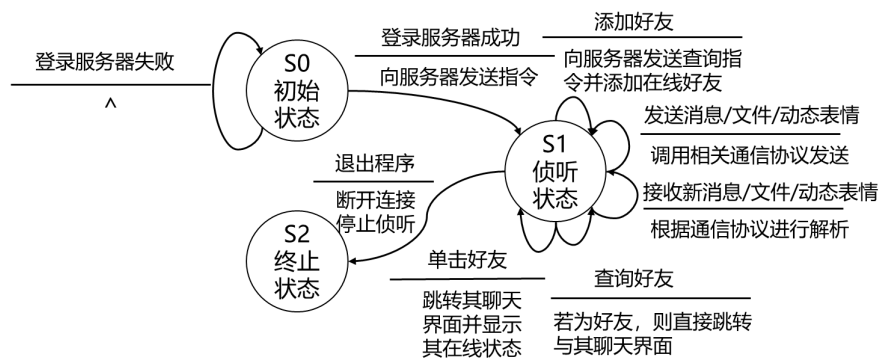


图 2: 整体状态机

2.3 选做内容

本项目完成了所有的必做任务，选做内容如下。

动态表情发送 本项目实现了对动态表情发送的功能，支持的文件格式为 gif。

聊天记录查询 使用 SQL 数据库对聊天记录进行存储。在每次登出时，将所有聊天记录存入数据库；在每次登录成功后，读取数据库中的聊天记录，以实现在登出后再次登录时，能够查询之前的聊天记录。

快速查找好友并发起对话 一般来讲，每个人会有数百个好友。如果想与某一位好友对话时，如果一点一点翻联系人列表会耗费大量时间，故设计快速查找好友功能以实现对话的快速发起。

新消息提醒功能 在收到消息时，在列表的正下方会进行消息提醒，即提醒发来新消息的人的学号。

2.4 开发环境

本项目基于 VS2017 及 Blend for VS2017，使用 C# 的 WPF 框架进行开发。其中逻辑编写使用 VS2017，界面的开发基于 Blend。

3 详细设计

该系统主要分为三个模块：逻辑层、界面层以及数据层。三个层次彼此分明：逻辑层和数据层通过初始化时载入数据库以及退出程序时数据保存至数据库的方式对接；逻辑层和界面层通过事件对接。

3.1 界面层

界面层通过对 .xaml 文件的编写进行设计。其中，界面风格采用了 material 风格。所设计的聊天界面如下。其中按钮分别为添加好友、添加群聊¹、好友查找跳转、动态表情发送、文件传输、语音发送²、消息列表³。上方学号旁的绿灯表示在线，若不在线则为灰色。

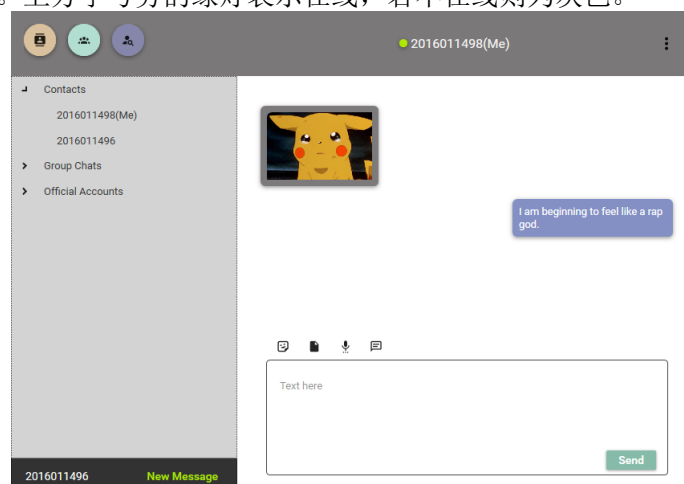


图 3: 聊天界面

¹由于期末时间关系，功能未实现

²功能未实现

³功能未实现

3.2 逻辑层

3.2.1 账号登录上线及下线

账号登录上线即通过对服务器发送指令，账号密码均正确后即登录成功。与服务器之间的交互使用了 C# 中的 System.Net 以及 System.Net.Socket 库⁴。下线操作即在程序完全退出前，对服务器发送相关指令，基本操作与登录上线相似。

与服务器的交互过程为，客户端向服务器发起连接，连接后再发送相关的指令。

3.2.2 通讯录 (查询好友是否在线)

查询好友是否在线即向服务器询问所要查询的好友的在线状态及其 IP 地址。若好友在线，则好友指示灯亮起；若好友不在线或者用户不存在以及输入格式出现错误，会通过 MessageBox 进行提醒。

3.2.3 P2P 通信

P2P 通信分为发送部分和接收部分。下面分别叙述本项目的传输协议、P2P 发送以及 P2P 侦听部分。

协议部分 本项目的协议共 14 位字节，例：20160114980000。其中，前 10 位为学号，第 11 位和第 12 位为类别，第 13 位和第 14 位为扩展位。

P2P 发送部分 P2P 发送部分与和服务器之间的交互的发送部分几乎相同，即先与 Peer 进行连接，再进行发送。发送前要对相关的信息转换为与协议相关的格式。

P2P 侦听部分 P2P 接收部分单独开了两个线程进行接收。在其中一个线程下不断循环进行侦听，另一个线程监测是否有新消息提醒，若有新消息提醒，则使用 Invoke 对界面进行更新，以防止程序出现崩溃。侦听在聊天界面的初始化时开始，在完全退出程序前停止侦听。停止侦听采用软停止，即对自己发送 1 个字节的 0，当侦听到时，退出循环即停止侦听。

3.2.4 文件传输

协议说明 文件传输的应用层协议为，在 14 位通信协议的基础上，先是用一个字节表示文件名字的长度，再用若干字节表示文件名字，剩下的为文件的具体内容。

文件传输 文件传输功能能够传输的最大文件限制在 1G 左右，这一点仿照微信对文件大小限制在 100M。除此之外，接收文件时需要手动保存的路径，若放弃选择路径则表示拒绝接收文件。

3.2.5 聊天记录查询

内存记录 为了在程序中记录消息记录，定义了两个聊天记录类，即全部聊天记录类和单条聊天记录类。具体如下。

```
1 //单条聊天记录纪录
2 public class chatHistorySingle
3 {
4     public string Speaker; //谁说的话
5     public int chatid;
6     public string Mess;
7     public string Date = "2019/1/1";
```

⁴程序书写参考了 MSDN

```

8      public int ModeFile = 0;
9      public chatHistorySingle(string speaker, int Chatid, string
      mess,int modefile,string date)
10     {
11         Speaker = speaker;
12         chatid = Chatid;
13         Mess = mess;
14         ModeFile = modefile;
15         Date = date;
16     }
17 }
18
19 //整个聊天记录
20 public class chatHistory
21 {
22     public string Monitor;      //主机（谁登录的）
23     public string Contact;      //给谁发
24     public List<chatHistorySingle> History = new List<
      chatHistorySingle>();
25     public int preLoadHistoryID = 1;
26     public chatHistory(string monitor, string contact)
27     {
28         Monitor = monitor;
29         Contact = contact;
30     }
31     public void AddHistory(chatHistorySingle chs)
32     {
33         History.Add(chs);
34     }
35 }

```

其中单条聊天记录类用于记录说话的人、聊天记录、记录 ID、时间、文件类型；整个聊天记录类用于记录登录 ID、联系人 ID 以及众多的单条聊天记录。

数据库记录 通过 SQL 接口，将聊天记录信息存入数据库。且在每次界面初始化时，将数据读入，以实现在登出后再次登录时，能够再次查询之前的聊天记录。

3.2.6 动态表情发送

动态表情发送与发送文件类似，将收到的动图自动存在当前文件夹中的 Image 文件夹中。通过使用 WPF 的 MediaElement 控件，以及在后端调整其帧使得其能够循环播放显示。

3.2.7 好友查询及跳转

通过使用 C# 的 Lambda 表达式，查询好友列表中是否存在好友且得到其 index，通过与界面层的结合实现跳转功能，便于在数百个好友中迅速跳转到想要对话的好友。

3.2.8 新消息提醒

为了达到即时通讯的目的，需要在接收到消息时了解到消息的发送者是谁，所以需要进行消息提醒操作。使用 SnackbarMessage 控件进行对新消息的提醒，便于操作。在点击 NewMessage 之后，提醒框会自动消失。



图 4: 新消息提示

3.2.9 实时显示提示信息

在查找好友和添加好友时，实现对提示信息的实时显示，使用户体验更好。

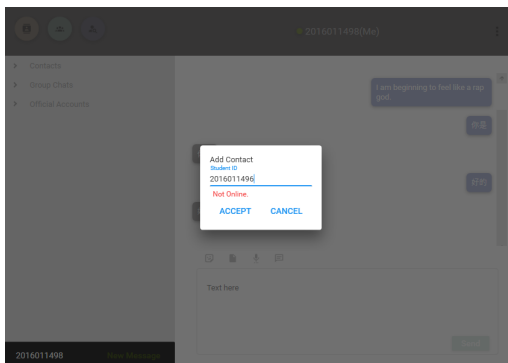


图 5: 实时显示提示信息 (添加好友时好友不在线)

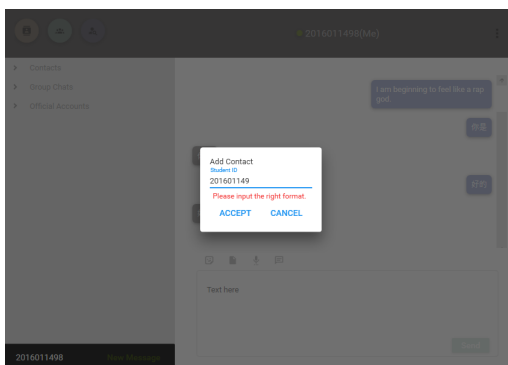


图 6: 实时显示提示信息 (添加好友时好友格式输入错误)

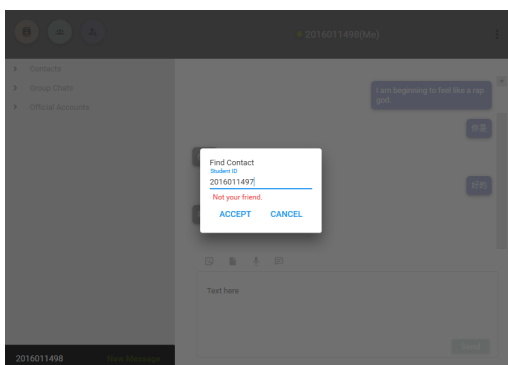


图 7: 实时显示提示信息 (查询好友时，该好友不为你为好友，无法跳转)

3.3 数据层

数据层主要调用了 SQLite 数据库。存储数据分别为登录用户学号、消息发送者学号、联系人学号、聊天消息记录、日期、消息类型、消息的 ID。

MONITOR	SPEAKER	PEER	CONTENT	DATE	KIND	ID
2016011498	2016011498	2016011498	111	2019/1/1	0	13
2016011498	2016011498	2016011498	dd.txt	2019/1/1	0	14
2016011498	2016011498	2016011498	111	2019/1/1	0	15
2016011498	2016011498	2016011498	111	2019/1/1	0	16
2016011498	2016011498	2016011498	1111	2019/1/1	0	17
2016011498	2016011498	2016011498	1111	2019/1/1	0	18
2016011498	2016011498	2016011498	2123231	2019/1/1	0	19
2016011498	2016011498	2016011498	2123231	2019/1/1	0	20
2016011498	2016011498	2016011498	111	2019/1/1	0	21
2016011498	2016011498	2016011498	111	2019/1/1	0	22
2016011498	2016011498	2016011498	2222	2019/1/1	0	23
2016011498	2016011498	2016011498	2222	2019/1/1	0	24
2016011498	2016011498	2016011498	新建文本...	2019/1/1	0	25
2016011498	2016011498	2016011498	111	2019/1/1	0	26
2016011498	2016011498	2016011498	111	2019/1/1	0	27
2016011498	2016011498	2016011498	dd (2).txt	2019/1/1	0	28
2016011498	2016011498	2016011498	lyrics.txt	2019/1/1	0	29
2016011498	2016011498	2016011498	11	2019/1/1	0	30
2016011498	2016011498	2016011498	11	2019/1/1	0	31
2016011498	2016011498	2016011498	22	2019/1/1	0	32
2016011498	2016011498	2016011498	22	2019/1/1	0	33
2016011498	2016011498	2016011498	dd (2).txt	2019/1/1	0	34
2016011498	2016011498	2016011498	111	2019/1/1	0	35
2016011498	2016011498	2016011498	111	2019/1/1	0	36
2016011498	2016011498	2016011498	222	2019/1/1	0	37
2016011498	2016011498	2016011498	222	2019/1/1	0	38

图 8: 数据库表格

4 结果分析

4.1 登录界面

登录界面使用了 Card、Textblock、Button、passwordBox 控件。账号密码默认为 2016011498 和 net2018。除此之外，隐藏了窗口边缘并自制了最小化和关闭按键，以保证界面的美观，同时拖动窗口的任意位置可以对窗口进行拖动。

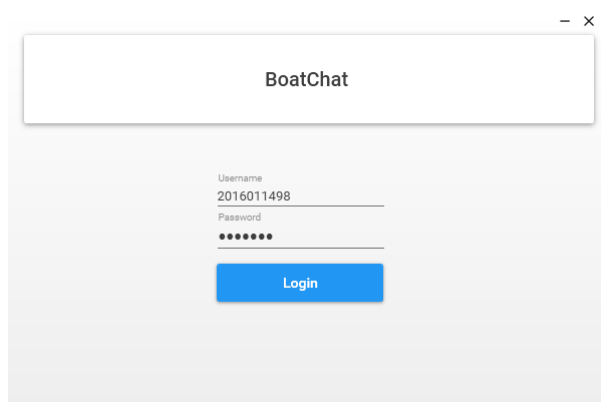


图 9: 登录界面

4.2 聊天主界面

向中央服务器发出查询，若无误则登录成功进入聊天主界面。

添加好友 点击左上第一个按键，进行好友添加。添加过程中会有实时提醒输入格式是否正确、好友是否在线、好友是否存在。若好友在线，则添加入通讯录中。

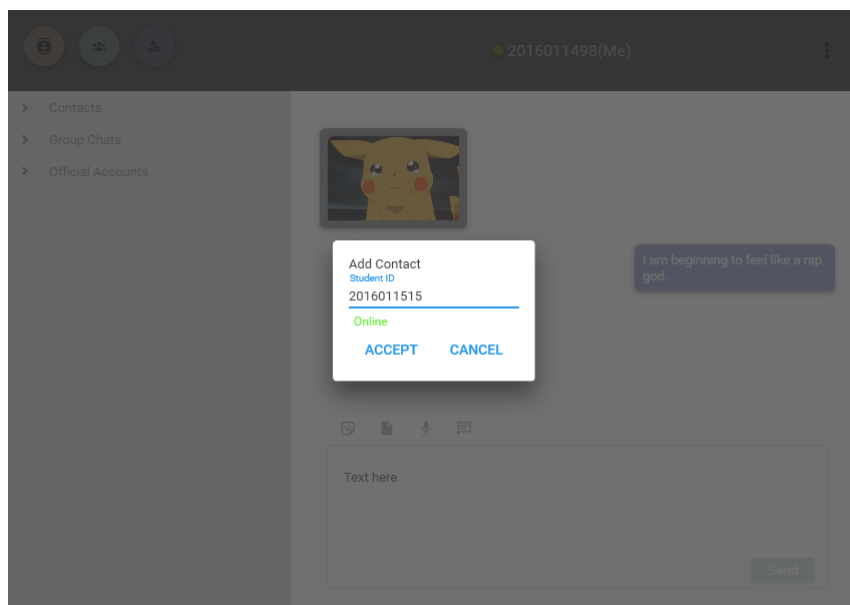


图 10: 添加好友界面

发送消息 点击左侧的通讯录即跳转至与其聊天的界面。在对话框中发送消息，以及发送动图和文件。相关信息会显示在对话框中，且设定滚动条自动追踪最后一条消息。其中文字消息的发送可以点击 Send 键或者直接回车发送。其中，右上的亮灯表示该好友在线。

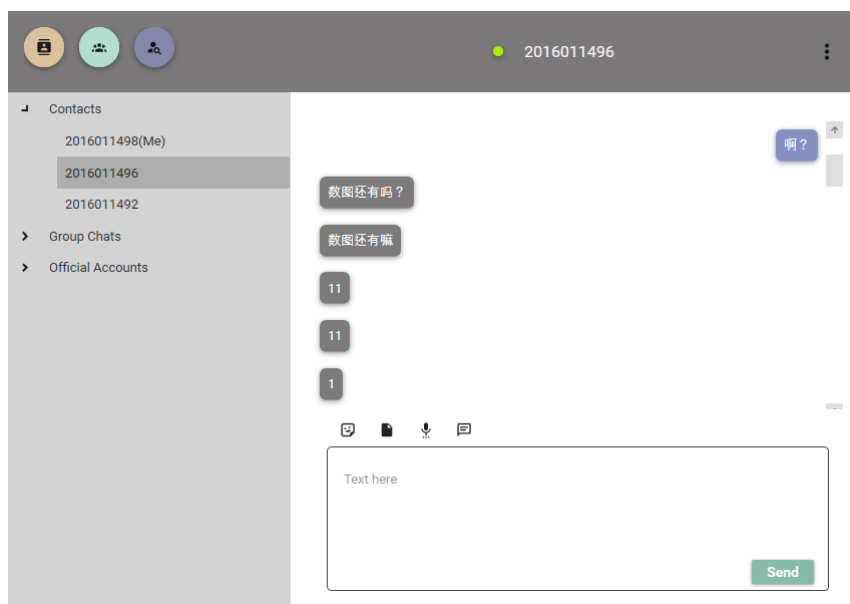


图 11: 发送消息窗口

查找好友并跳转 点击左上方第三个按钮，则弹出好友查询对话框，在输入学号的过程中同样也会做即时提醒。若为好友则直接跳转与其之间的聊天界面。

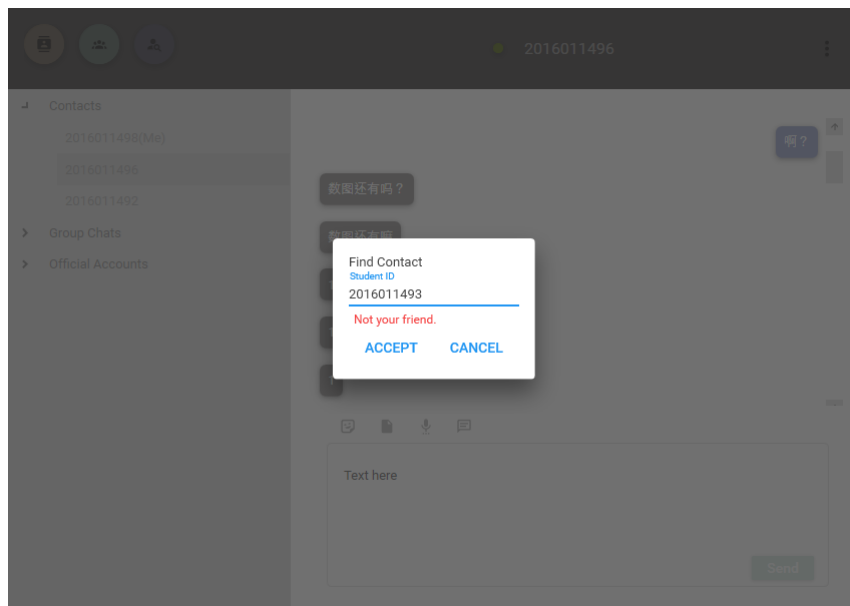


图 12: 查找好友并跳转

发送/接收动态表情 本项目只支持发送并显示动态表情，即电机 StickerButton，然后选择动图并进行发送。



图 13: 发送/接收动态表情

发送/接收文件 本项目限制最大文件大小为 1G，这一点模仿微信。发送文件即点击文件发送按钮，并选择发送的文件。

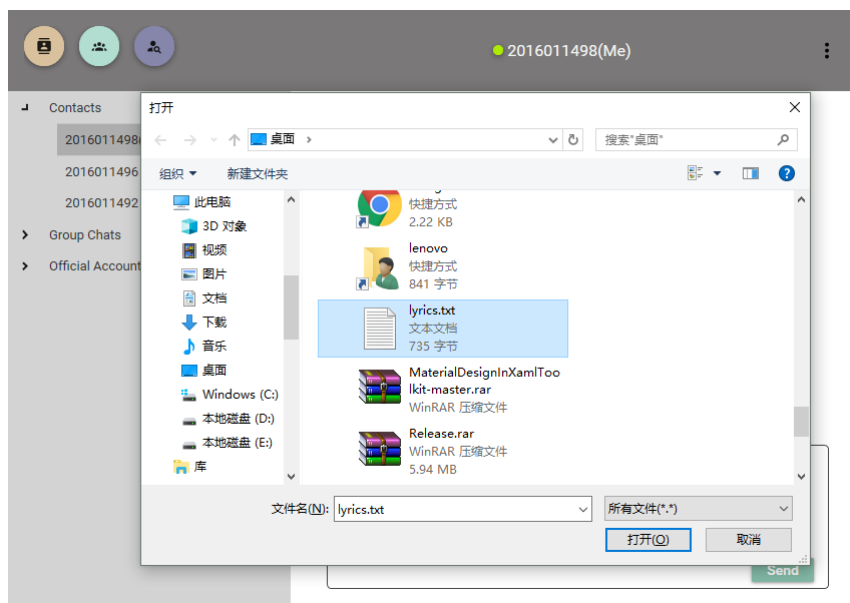


图 14: 发送文件

接收到文件时会自动弹出保存窗口，选择保存的路径，并且左小角会提醒文件的来源

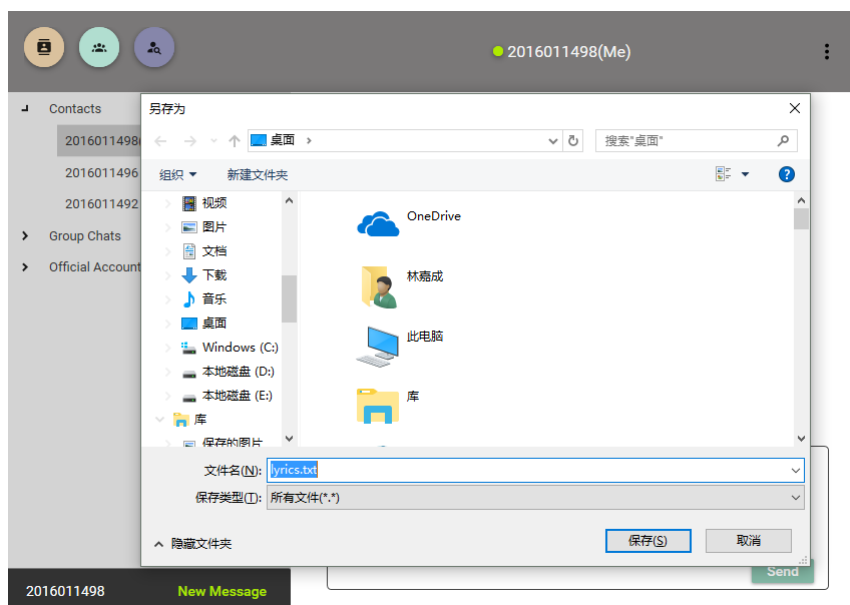


图 15: 发送文件

退出程序 若要退出程序，则点击右上角的按键，点击 Quit 再确认即退出。退出后所有数据将存入数据库。

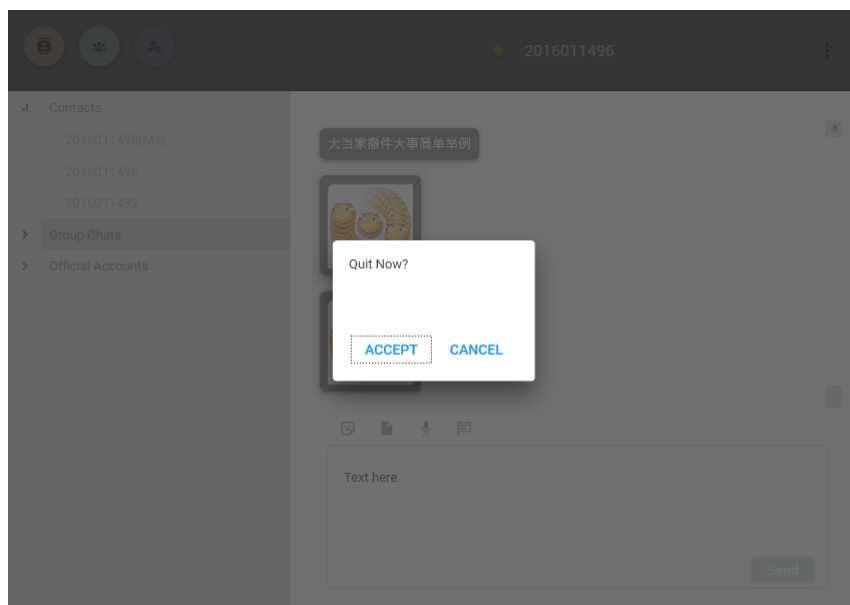


图 16: 退出程序

5 总结与反思

由于这学期其他课程众多的大作业以及期末复习的压力，本次计网大作业我在 2018 年底写了整整四天。四天里一直专心的写计网大作业，且这是我第一次使用 WPF 进行界面设计以及第一次使用 C# 进行网络通信编写。本次大作业极大的锻炼了我的快速学习能力以及泛化能力。

除了对计网课上所讲的相关内容在实践过后有了更近一步的了解，我逐渐感觉到 WPF 的厉害之处。这四天里，一天半界面设计，两天半逻辑编写，以及对 WPF 各种特性的测试，让我越发越感受到其远远超过 Winform 的地方，在编写的过程中让我回想起当初人智作业用 Winform 编写时各种不规范的编程习惯，比如 Winform 的界面更新建议另开线程 (Invoke)，否则可能在界面同时刷新时程序会出现崩溃。但是，我了解的 WPF 可能只是冰山一角，比如本次大作业的编写由于时间关系，并没有使用 WPF 的优点之一的 binding 功能。这可能是一种遗憾。

不过更遗憾的是，我没有去实现群聊和语音功能。一是期末复习压力，时间不够，二是由于群聊的编写可能要稍微修改一下我之前的代码架构，可能会耗费太多的时间。此外，再进行深入思考之后，我感觉或许微信的群聊功能是基于中央服务器的，即可能不是 P2P 的。如果可以的话，其实可以让我们也对服务器部分的代码进行编写，这样更能锻炼我们的应用能力，进而对相关知识有更进一步的了解。

参考文献

[1] <https://docs.microsoft.com/en-us/dotnet/index>(网络部分)

[2] <https://github.com/MaterialDesignInXAML/MaterialDesignInXamlToolkit>(界面部分)