四川大學

实时通讯工具
软件开发过程

学　　院　建筑与环境学院

专　　业　工程力学与软件交叉

学生姓名　徐添

学　　号　2019141470418　年级　2019

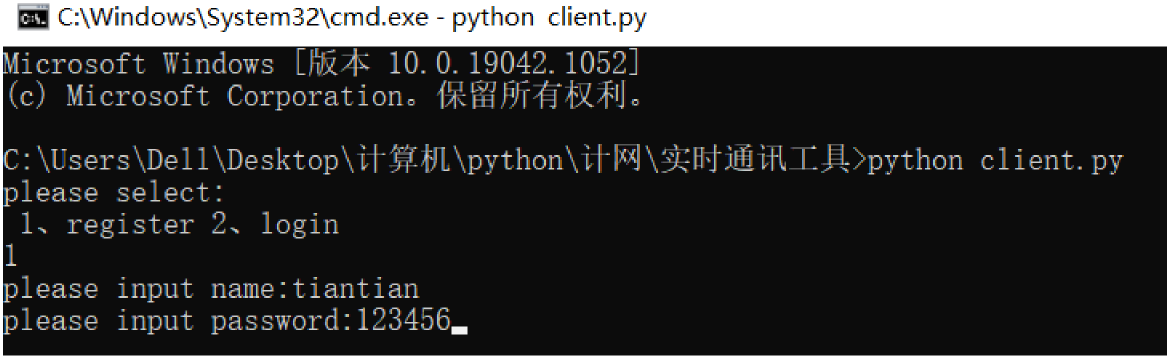
指导教师　宋万忠

二０二一年　6月　25日

一、 测试结果：

我选择的是实时通讯工具这一Socket编程项目。我所采用的协议是UDP关联。其能够实现：用户注册、用户登录、公聊、私聊、退出五个功能，测试如下：

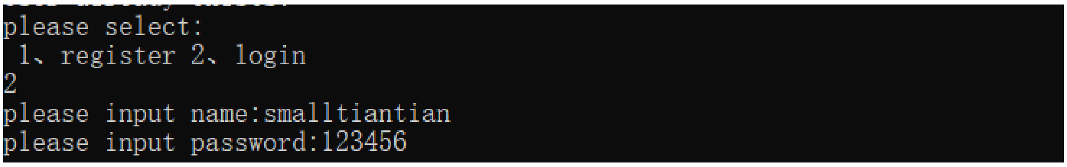
1、 用户注册



C:\Windows\System32\cmd.exe - python client.py

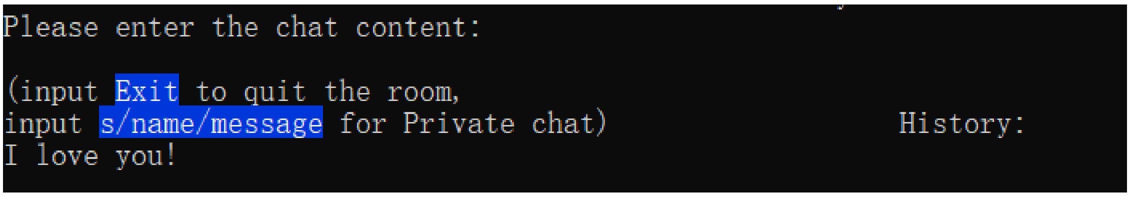
```
Microsoft Windows [版本 10.0.19042.1052]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Dell\Desktop\计算机\python\计网\实时通讯工具>python client.py
please select:
 1、register 2、login
1
please input name:tiantian
please input password:123456
```

2、 用户登录

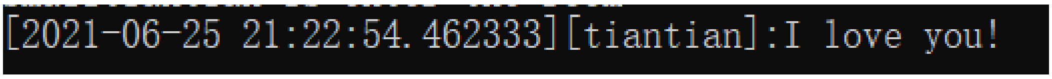


```
please select:
 1、register 2、login
2
please input name:smalltiantian
please input password:123456
```

3、 公聊

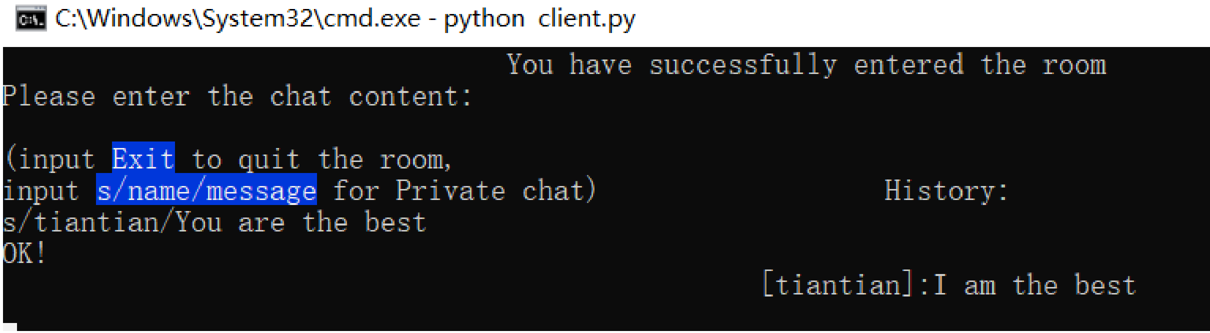


```
Please enter the chat content:

(input Exit to quit the room,
input s/name/message for Private chat)                    History:
I love you!
```

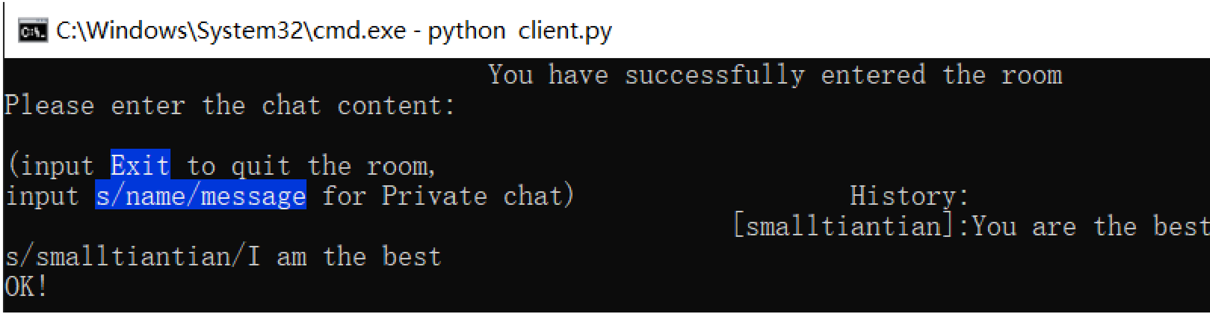


```
[2021-06-25 21:22:54.462333][tiantian]:I love you!
```

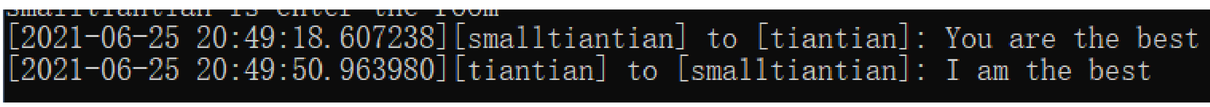
4、 私聊



```
                                You have successfully entered the room
Please enter the chat content:

(input Exit to quit the room,
input s/name/message for Private chat)                        History:
s/tiantian/You are the best
OK!
                                                 [tiantian]:I am the best
```

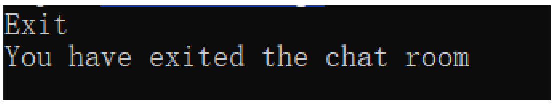


```
                               You have successfully entered the room
Please enter the chat content:

(input Exit to quit the room,
input s/name/message for Private chat)                      History:
                                               [smalltiantian]:You are the best
s/smalltiantian/I am the best
OK!
```



```
smalltiantian is enter the room
[2021-06-25 20:49:18.607238][smalltiantian] to [tiantian]: You are the best
[2021-06-25 20:49:50.963980][tiantian] to [smalltiantian]: I am the best
```

5、　退出



```
Exit
You have exited the chat room
```

综上所述，测试成功。

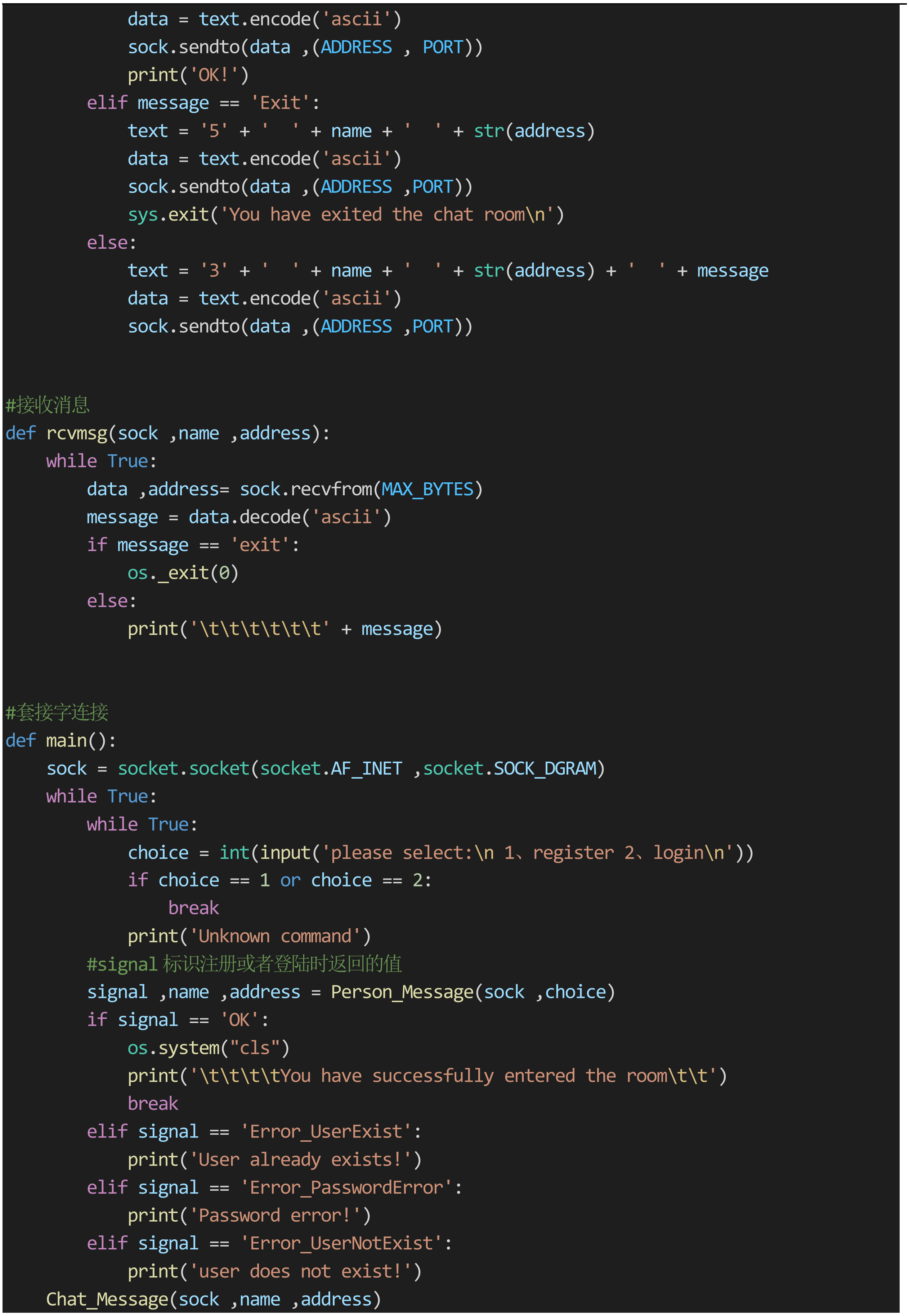
二、　发现问题及解决办法：

未能实现界面操作，我尝试学习通过 PyQt5 来编辑界面，但由于 python 语言编写能力较弱，无法实现界面配置后的代码实现，bug 较多。最后我选择通过 cmd 命令指示符对服务器和用户端进行界面显示，虽然有些简陋，但功能均可实现。

三、 代码附件：

client.py

```
import socket
import sys
from multiprocessing import Process
import os

MAX_BYTES = 65535
ADDRESS = '127.0.0.1'
PORT = 1600


#将注册或者登陆信息打包发送给服务器端
def Person_Message(sock ,choice):
    name = input('please input name:')
    password = input('please input password:')
    text = str(choice) + '  ' + name + '  ' + password
    data = text.encode('ascii')
    sock.sendto(data ,(ADDRESS ,PORT))
    data ,address= sock.recvfrom(MAX_BYTES)
    return data.decode('ascii') ,name ,address


#将用户聊天信息传送给公共频道
def Chat_Message(sock ,name ,address):
    print('Please enter the chat content:\n\n(input \033[1;44mExit\033[0m to quit the room,\n'
          'input \033[1;44ms/name/message\033[0m for Private chat)\t\t\tHistory:')
    #创建进程，父进程发送消息，子进程接受消息
    p = Process(target = rcvmsg ,args = (sock ,name ,address))
    p.start()
    sendmsg(sock ,name ,address)


#发送消息
def sendmsg(sock ,name ,address):
    while True:
        message = input()
        Words = message.split('/')
        if Words[0] == 's':
            Destination = Words[1]
            true_message = Words[2]
            text = '4' + '  ' + name + '  ' + str(address) + '  ' + true_message + '  ' + Destination
```

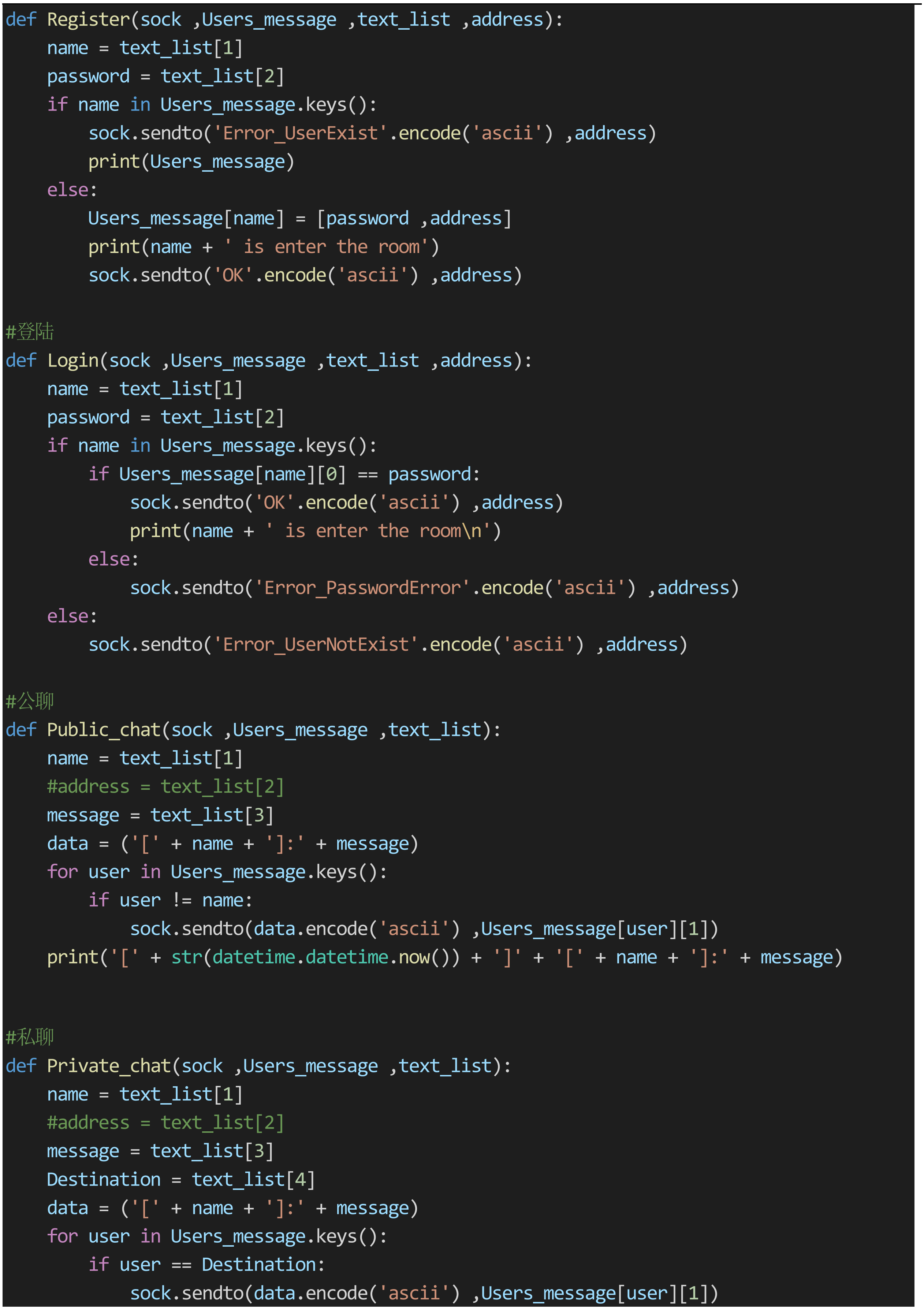


```
            data = text.encode('ascii')
            sock.sendto(data ,(ADDRESS , PORT))
            print('OK!')
        elif message == 'Exit':
            text = '5' + '  ' + name + '  ' + str(address)
            data = text.encode('ascii')
            sock.sendto(data ,(ADDRESS ,PORT))
            sys.exit('You have exited the chat room\n')
        else:
            text = '3' + '  ' + name + '  ' + str(address) + '  ' + message
            data = text.encode('ascii')
            sock.sendto(data ,(ADDRESS ,PORT))


#接收消息
def rcvmsg(sock ,name ,address):
    while True:
        data ,address= sock.recvfrom(MAX_BYTES)
        message = data.decode('ascii')
        if message == 'exit':
            os._exit(0)
        else:
            print('\t\t\t\t\t\t' + message)


#套接字连接
def main():
    sock = socket.socket(socket.AF_INET ,socket.SOCK_DGRAM)
    while True:
        while True:
            choice = int(input('please select:\n 1、register 2、login\n'))
            if choice == 1 or choice == 2:
                break
            print('Unknown command')
        #signal 标识注册或者登陆时返回的值
        signal ,name ,address = Person_Message(sock ,choice)
        if signal == 'OK':
            os.system("cls")
            print('\t\t\t\tYou have successfully entered the room\t\t')
            break
        elif signal == 'Error_UserExist':
            print('User already exists!')
        elif signal == 'Error_PasswordError':
            print('Password error!')
        elif signal == 'Error_UserNotExist':
            print('user does not exist!')
    Chat_Message(sock ,name ,address)
```

```
if __name__ == "__main__":
    main()
```

server.py

```
import socket
import datetime

MAX_BYTES = 65535
ADDRESS = '127.0.0.1'
PORT = 1600


#选项菜单
def menu(sock ,Users_message):
    while True:
        # data ,address = sock.recvfrom(MAX_BYTES)
        # text = data.decode('ascii')
        # print('message from {} is {}'.format(address ,text))
        # text = 'hello ,too'
        # data = text.encode('ascii')
        # sock.sendto(data ,address)
        data ,address = sock.recvfrom(MAX_BYTES)
        text_list = data.decode('ascii').split('  ')

        if int(text_list[0]) == 1:
            #注册
            Register(sock ,Users_message ,text_list ,address)
        if int(text_list[0]) == 2:
            #登陆
            Login(sock ,Users_message ,text_list ,address)
        if int(text_list[0]) == 3:
            #公聊
            Public_chat(sock ,Users_message ,text_list)
        if int(text_list[0]) == 4:
            #私聊
            Private_chat(sock ,Users_message ,text_list)
        if int(text_list[0]) == 5:
            #退出
            Exit(sock ,Users_message ,text_list)

#注册
```



```
def Register(sock ,Users_message ,text_list ,address):
    name = text_list[1]
    password = text_list[2]
    if name in Users_message.keys():
        sock.sendto('Error_UserExist'.encode('ascii') ,address)
        print(Users_message)
    else:
        Users_message[name] = [password ,address]
        print(name + ' is enter the room')
        sock.sendto('OK'.encode('ascii') ,address)

#登陆
def Login(sock ,Users_message ,text_list ,address):
    name = text_list[1]
    password = text_list[2]
    if name in Users_message.keys():
        if Users_message[name][0] == password:
            sock.sendto('OK'.encode('ascii') ,address)
            print(name + ' is enter the room\n')
        else:
            sock.sendto('Error_PasswordError'.encode('ascii') ,address)
    else:
        sock.sendto('Error_UserNotExist'.encode('ascii') ,address)

#公聊
def Public_chat(sock ,Users_message ,text_list):
    name = text_list[1]
    #address = text_list[2]
    message = text_list[3]
    data = ('[' + name + ']:' + message)
    for user in Users_message.keys():
        if user != name:
            sock.sendto(data.encode('ascii') ,Users_message[user][1])
    print('[' + str(datetime.datetime.now()) + ']' + '[' + name + ']:' + message)


#私聊
def Private_chat(sock ,Users_message ,text_list):
    name = text_list[1]
    #address = text_list[2]
    message = text_list[3]
    Destination = text_list[4]
    data = ('[' + name + ']:' + message)
    for user in Users_message.keys():
        if user == Destination:
            sock.sendto(data.encode('ascii') ,Users_message[user][1])
            print('[' + str(datetime.datetime.now()) + ']' + '[' + name + ']' + ' to [' + Destination + ']: ' + message)


#退出程序
def Exit(sock ,Users_message ,text_list):
    name = text_list[1]
    address = text_list[2]
    print(address)
    data = 'exit'
    for user in Users_message.keys():
        if name == user:
            sock.sendto(data.encode('ascii') ,Users_message[user][1])
            print(name + ' is quit the room\n')


#套接字连接
def main():
    #用户信息存在字典中，实现可持久化存储可将用户信息写入 txt 等文本内
    Users_message={}
    sock = socket.socket(socket.AF_INET ,socket.SOCK_DGRAM)
    sock.bind((ADDRESS ,PORT))
    print('listen to {}'.format(sock.getsockname()))
    menu(sock ,Users_message)


if __name__ == "__main__":
    main()
```