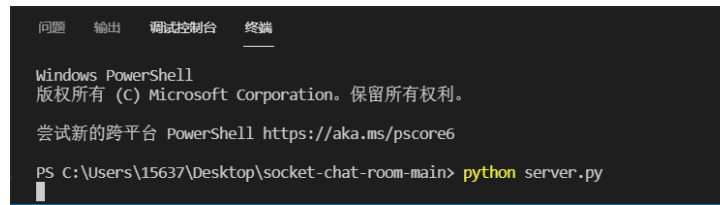


## 开发过程文档

测试过程：具体使用时的正常和异常路径流

- 1、在终端运行 server.py，即启动服务器



```
Windows PowerShell
版权所有 (c) Microsoft Corporation。保留所有权利。

尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\15637\Desktop\socket-chat-room-main> python server.py
```

- 2、在另一个终端运行 client.py，启动客户端



```
153 if __name__ == '__main__':
154     users = loadUser()
155     history = loadHistory()
156
157     #启动多线程tcpServer作为服务器
158     app = socketserver.ThreadingTCPServer(('', 8080), MyHandler)
159     app.serve_forever()
```

登录

账号

密码

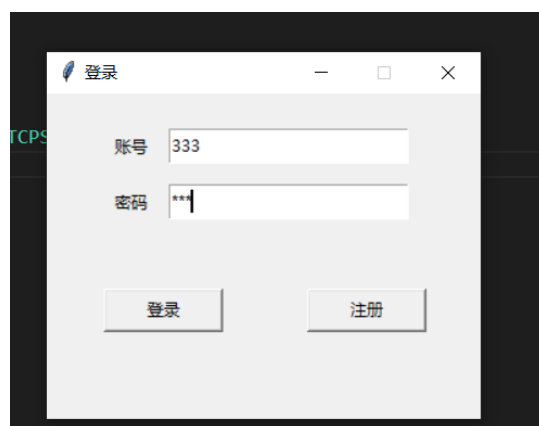
```
Windows PowerShell
版权所有 (c) Microsoft Corporation。保留所有权利。

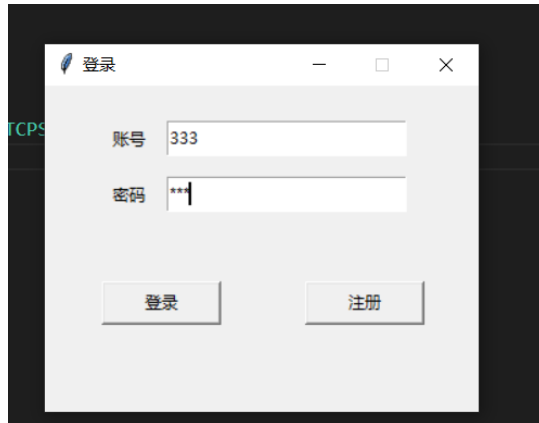
尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\15637\Desktop\socket-chat-room-main> python client.py
```

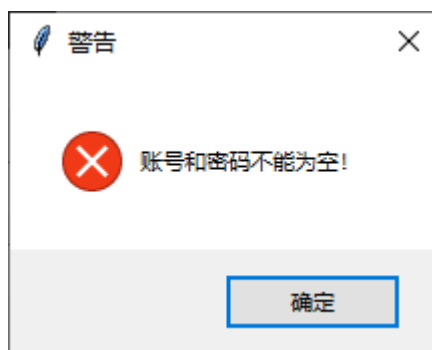
如图可见窗口程序已经显示

- 3、新用户输入账号和密码后，点击注册，显示注册成功

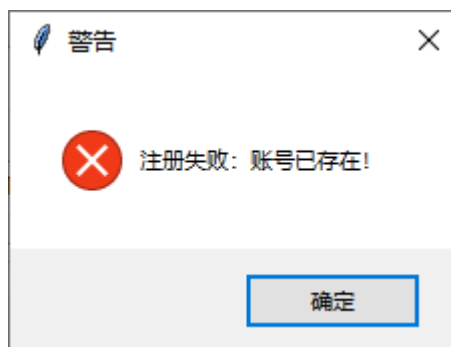




如果不输入账号密码，应显示提示输入



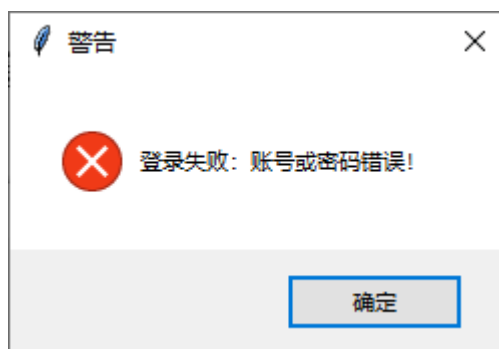
如果用户名已经存在，应显示用户名已占用



- 4、注册后再次点击登录，进入主窗口，左边显示聊天记录，右边显示在线用户，默认为公共聊天室



如果用户不存在应显示账号或密码错误

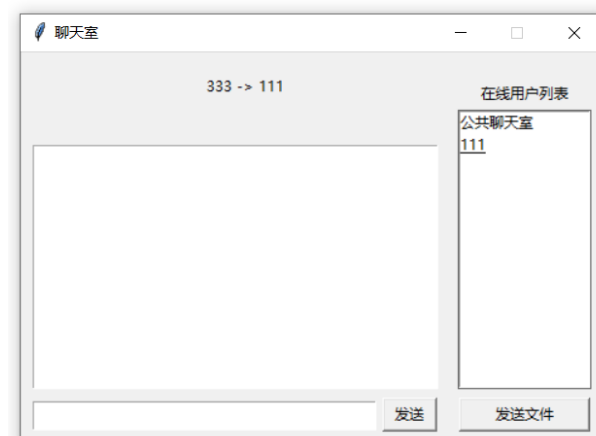


在公共聊天室发送消息，应正确显示在所有用户的公共聊天室窗口（或者右侧

公共聊天室一栏显示\*号）

公共聊天室不能发送文件，其按钮为灰色

5、 点击右侧在线用户（非公共聊天室），进入私聊窗口



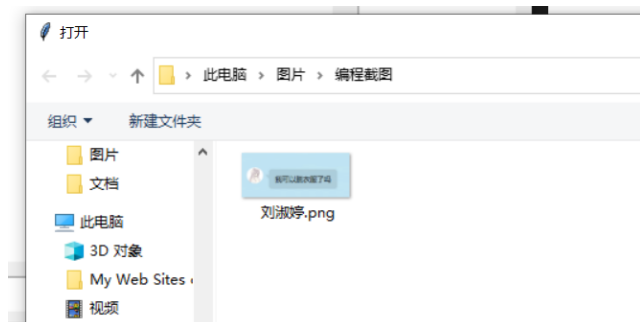
6、 在下方输入框中输入文字内容，点击发送，上方显示发送内容



- 7、 对方在接收到消息后 ,若聊天窗口并非对应的私聊窗口 ,其右侧对应列表中 333 的名称后面会显示\*表示有新消息未查看

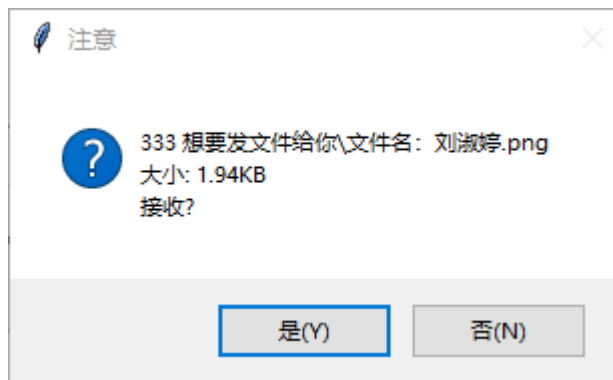


- 8、 点击发送文件，弹出文件管理器选择文件，选择完成后，显示对方正在对请求做出回应。相应的对方客户端显示是否接受发送请求

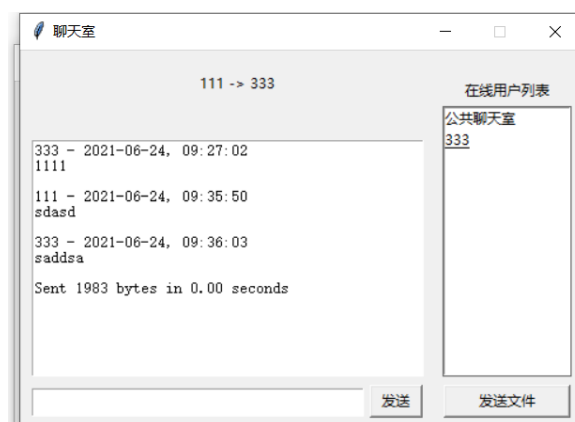
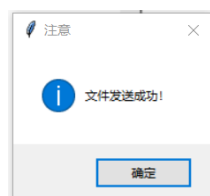




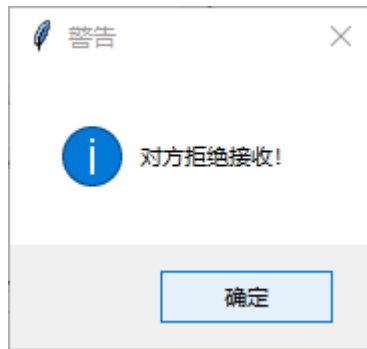
对方应弹出同意窗口



9、 对方同意后，文件开始传输，并弹出窗口显示发送成功



若对方不同意，应显示对方拒收



点击确定窗口消失

10、 点击右上角 X，窗口将关闭，同时程序终止

### 问题与解决

1 Linux 环境测试：在虚拟机环境 Ubuntu20.04 中测试，需要 sudo 安装 tk 包，在安装完成后，可正常运行。同时由于虚拟机网段与主机不同，需更改虚拟机网段才能链接服务器。

2 局域网连接服务器测试：由于 windows 的防火墙禁止了其他设备的链接，需要关闭 Windows 的防火墙（所有主机）才能够建立局域网下的链接（即 ping 命令可正常工作），通过在同一局域网下的不同电脑启动客户端，均可以连接到服务器进行消息的发送。

3 Python Socket 的阻塞问题，socket 默认为阻塞式的，当 recv()函数被调用时，如果没有明确的定义数据流在什么字符串被读取时停止传输，则 recv()会一直阻塞线程，以至于下面的代码无法执行，所以采用设置明确的 socket 的 timeout 来达成非阻塞，在消息发送完后再执行 settimeout(None)重新设置 socket 为阻塞

4 数据的传输以及处理问题，由于 socket TCP 传输字节流，需要使用较为麻烦的字符串大量处理数据，故而采用 struct 和 json 两个包对数据进行 json 文件格式化的写入以及读取。

5 能处理并发请求的服务端。服务器端能处理并发请求，每当有客户端请求连接时，服务端都会开启一个线程进行处理，因此当有多个客户端同时请求服务时不会造成阻塞。