1. 系统介绍
2. 概述

一个基于python的在线聊天室软件，结合tkinter图形库、threading多线程和socket套接字等工具，实现了用户即时登录登出、在线聊天消息的接收转发、表情包发送、文本文件上传、查看在线用户列表等功能。系统采用C/S架构，采用TCP协议进行网络数据传输，服务端负责用户地址、消息和信息的存储接收转发，客户端负责图形界面系统响应以及和服务端的通讯。

1. 功能
2. 用户登录

用户可以通过使用默认的服务器端口地址，或者自定义端口（在客户端修改端口绑定代码即可），输入随意昵称，点击登录按钮，登录聊天系统。

1. 发送消息

用户可以通过在文本框输入文字消息，点击发送按钮，将消息发送到所有人可见的群聊列表内。

1. 发送表情包

用户可以通过点击表情包按钮，选择默认的表情包进行发送（表情包只接受png格式，可以通过emoji文件增加图片，并在代码段进行添加）。

1. 上传文本文件

用户可以通过点击上传文件按钮，浏览选择电脑本地的文件上传（文件只接受txt文档），上传后的文本文件会自动将文字内容发送到群聊消息中。

1. 用户登出

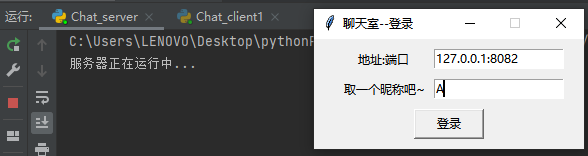
用户点击右上角取消按钮，即可退出聊天软件，并弹出已退出系统的界面提示；其他在线用户则可看到在线用户列表实时变化。

1. 多用户登录

一个client端代码代表一个用户，要想多用户同时登录并且彼此交互，可以依次运行各个client端代码进行登录，系统内的在线用户列表会随用户数量增加实时更新显示，但后登录用户无法查看历史消息。

1. 运行演示
2. 单人登录

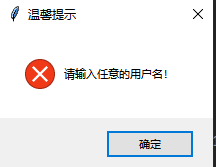
先启动Chat\_server，控制台显示“服务器正在运行中”，再启动Chat\_client1，随后跳出登录界面，如下图所示：



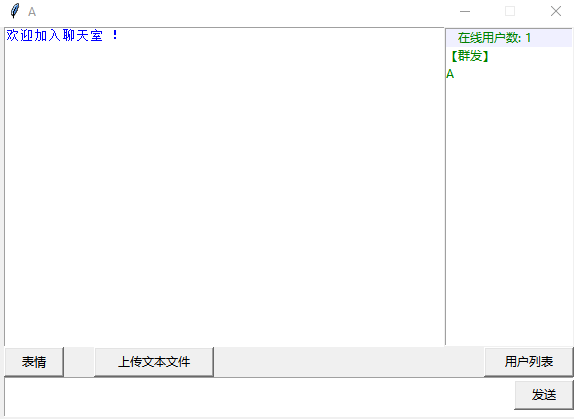
第一栏 地址：端口 是系统默认的地址

第二栏 昵称 可以随意输入字符数字字母

若什么都不填 会弹出错误窗口，如下图所示：



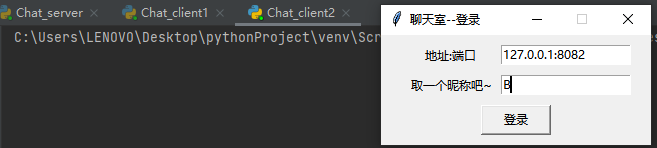
填好昵称后，点击登录按钮，成功进入聊天室，如下图所示：



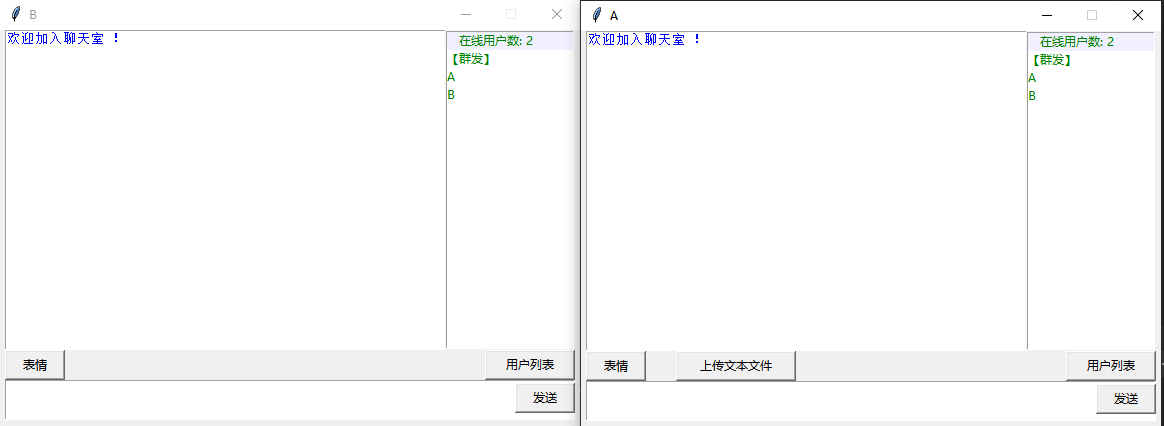
1. 多人登录

再运行一个Chat\_client2，依照1）的步骤登录，如下图所示：

（注：要想继续增加用户数量，再新建一个py文件Chat\_client3/4/……，将Chat\_client1的代码完全复制过去再运行即可）



登录成功后，两个客户端的在线用户列表都会同时刷新，如下图所示：



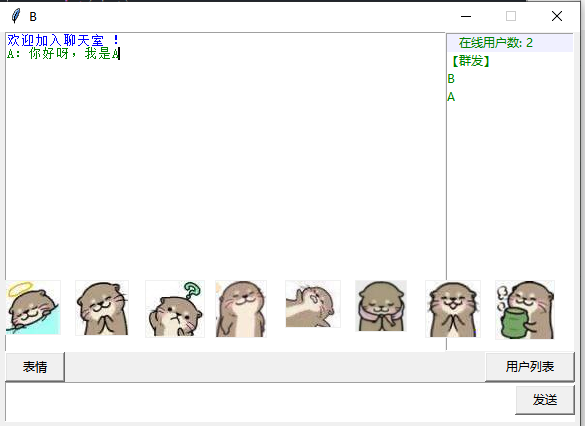
1. 发送消息

例如 A发送一个“你好呀，我是A”的消息，结果如下图所示：



1. 发送表情包

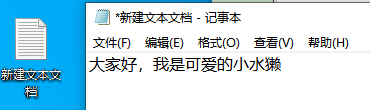
例如，B发送一个小水獭的表情包，结果如下图所示：



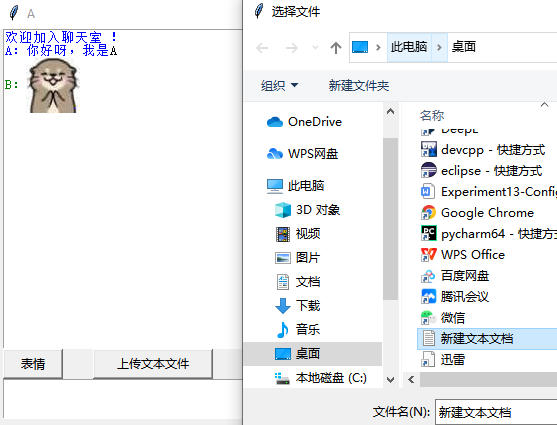


5）上传文本文件内容

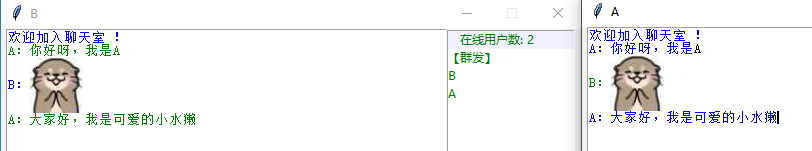
例如，在电脑桌面新建一个文本txt文件，内容为“大家好，我是可爱的小水獭”



然后点击上传文件按钮，选择这个文本文件，如图所示：



文本文件的内容直接被发送成消息形式，结果如下图所示：

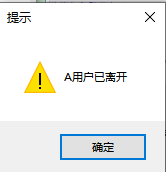


（下图为后来修改： 加上了文字标识）

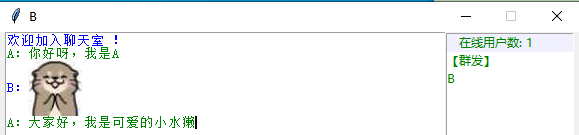


1. 退出

首先，在A的图形界面右上角点击×，A退出聊天系统，并提示下线



此时B的界面上，在线用户只剩下了B：



1. 项目总结

总的来说系统开发难度不大，因为网上有很多现成的教程和代码，个人在这个基础上对复用的框架做了修改、添加以及优化，但仍然存在一定缺陷，比如没有实现其他格式文件的读取，以及文件的下载功能，也没有实现消息的私聊功能，因此在这个项目课程设计收获很多的同时，也看到了很多值得继续钻研的方向，总的来说获益满满。