软件使用说明：

实现的功能：多用户即时在线聊天室

1）注册功能：使用 pickle 包在本地储存用户信息，可以持久保存用户名、登录密码。

2）登录功能：通过登录界面拦截用户，需进行身份验证。

3）聊天功能：聊天室可实现多人在线聊天。

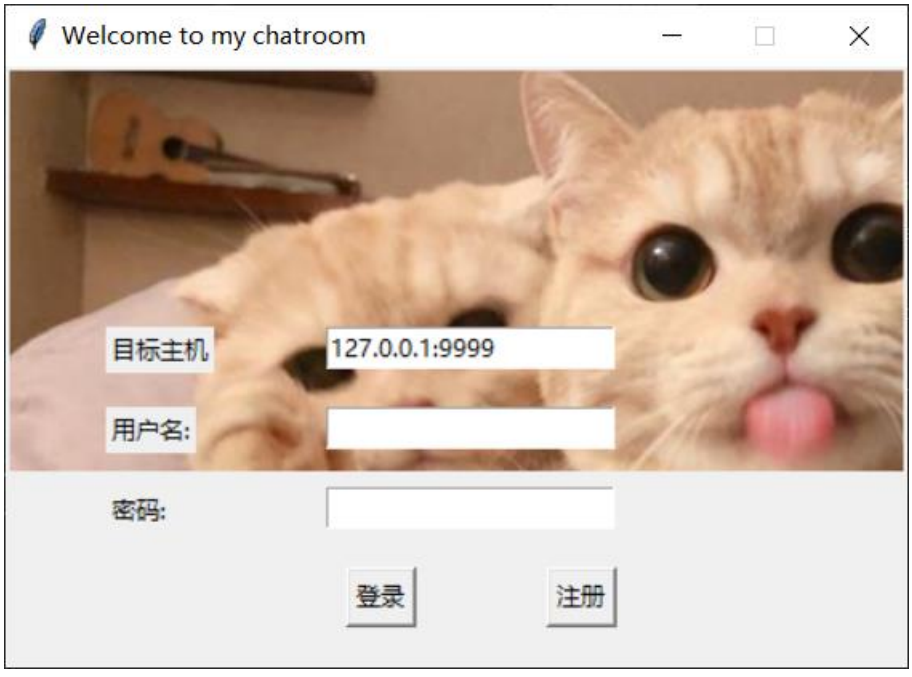
4）基本的防错：用户名不存在提示注册，密码错误提示错误，注册成功后会有提示

操作步骤：

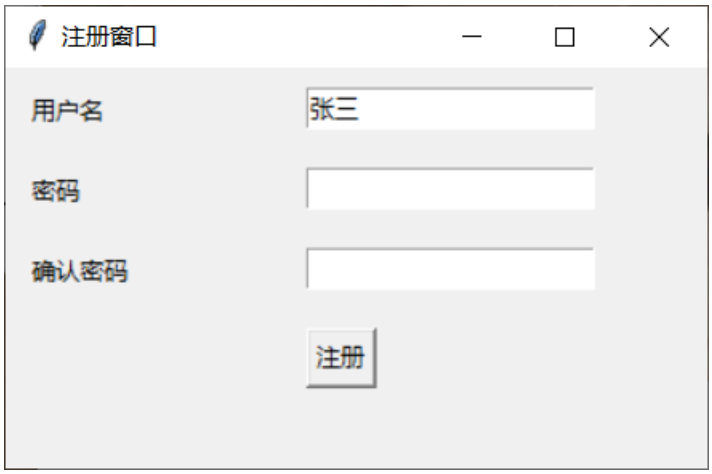
1，先运行 server.py,再运行 client1.py 和 client2.py 出现以下登录界面

登录主机默认本地，端口默认 9999（可在 new.ini 中修改）

如果想实现局域网下的互联，可以让一台 PC 作为服务器，其他两台 PC 作为客户端进行访问（注意：服务器端要打开网络共享才可实现连接）



2，点击注册按钮，输入注册信息，姓名默认为张三，可自行修改



3，注册成功以后返回登录界面，用注册好的用户名和密码登录，进入聊天室

4，在聊天室下方空白区域输入聊天内容，点击发送即可

开发过程：

测试结果：不仅实现了本地的多用户聊天，也成功在不同 PC 机，同一局域网下的互联

发现的问题：

1， 多线程问题：项目开始阶段不知道如何实现服务器端的多线程接受客户端响应，

在百度学习了 threading 包以后成功编写好了多线程代码：主类的 run 方法执行一个阻塞式循环，每接收到一个请求开启一个新的线程。建立好一个线程后，为该线程分配一部分内存（即创建一个 message 变量），发送函数不断检查 message 是否为空，若不为空，则表明当前用户将要发送数据，于是 message 发送，并清空

2，图形界面问题：起初的构思是想运用 web 前后端，但是考虑到可能会用到一些框架，背离使用基础的 socket 编码要求而放弃，最后选择的是 python 自带的 tkinter 作图包。（负面效果就是使用起来不够美观，且 tkinter 语法较为晦涩）。解决方法是仿照网上线程的一些界面，进行修改后使用。

3，服务器\客户端的交互：客户端需要进行登录操作，也就需要数据的持久化以验证用户身份，笔者使用了 pickle 包来生成服务器端的用户信息。用户身份验证还涉及注册问题、密码验证问题，防错提示等问题，在登录交互的逻辑设计上笔者花费了大量时间，最后采用 if else 式的分支判断逐步解决了问题