## 客户端：

#### 登录：

写： Login:username:password

阻塞

读： Failed

Successed

Refused

\*\*\*\*读数据：

#### 接收消息功能：

服务端推送消息 Server:内容:时间

接受聊天（单聊和群聊）内容: Chat:发送方：内容：时间 ChatAll:发送方：内容：时间

#### 获取在线用户功能：

新客户端： 获取用户列表并添加 UserList:用户1：用户2

旧客户端:

添加单个用户 AddUser:用户名

移除单个用户 RemoveUser:用户名

#### 关闭功能：

服务端关闭 ServerClosed 弹出提醒框

客户端关闭： Server\_Agree\_Exit

\*\*\*\*写数据

#### 发送功能：

发送群聊信息 ChatAll:内容

发送一对一单聊信息 Chat:接收方：内容

#### **关闭功能：**Client\_Request\_Exit

## 服务端：

\*\*\*\*读数据：

#### 关闭功能：

停止服务端的主线程：ServerStop

\*\*\*\*写数据：

推送消息功能：Server:内容:时间

告知关闭功能：

——告知客户端，我已关闭"ServerClosed"

—— 模拟一个客户端建立一个连接，发送"ServerStop"

获取在线用户功能： 遍历SocketThread集合

#### 连接用户功能：

为每个客户端都创建一个线程：SocketThread 【形参：当前连接，文本框引用，在线用户列表，服务端界面引用】

客户端的包装类：SocketThread

\*\*\*\*读数据：

#### 验证用户登录功能：

Login:username:password —— new LoginManager().isLogin(strs[1], strs[2])

#### 处理用户聊天功能：

Chat:发送方：内容：时间

ChatAll:发送方：内容：时间

#### 处理用户下线功能： Client\_Request\_Exit

\*\*\*\*写数据：

#### 验证用户登录功能：

用户登录成功:Successed

—— 向该客户端发送其登陆前已在线用户列表UserList:用户1：用户2

—— 向该客户端发送：将该客户端自己添加要在线用户列表的信息 AddUser:%s:%s

用户登录重复:Refused —— 遍历在线用户名，进行比对

用户登录失败:Failed

处理用户聊天功能： 遍历socketThreads集合，挨个发送

Chat:发送方：内容：时间

ChatAll:发送方：内容：时间

#### 处理用户下线功能：

——通知其他客户端，移除下线客户：RemoveUser:username:time

——通知要下线的客户端：Server\_Agree\_Exit

——将该用户从服务端用户列表中删除

——刷新服务端的用户列表

#### 关闭SocketThread

因服务端关闭而导致的客户端关闭，进而关闭SocketThread

服务端关闭后，通知到客户端，客户端通过ClientClosed指令，进而关闭服务端中的SocketThread：ClientClosed

关闭最后一个销毁服务端的socket: ServerStop