# SOCKET网络实验附件

## 系统概述

运行环境：windows 10

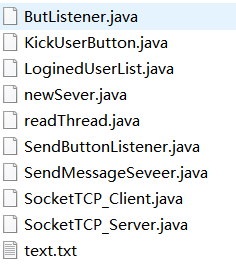
编译环境：JAVA 8

使用方法：将程序源文件解压，使用IDEA等JAVA运行软件打开对应的文件夹，先运行服务器端，再运行客户端。

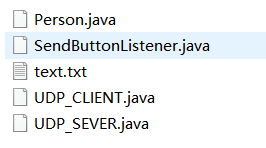
实现环境：IntelliJ IDEA 2022.2.1

程序文件列表：

**TCP:**



**UDP:**



## 主要数据结构

HashMap、ArrayList

## 主要算法描述

**TCP:**

1. 首先运行服务端，服务器创建seversocket监听9999端口等待连接，随后初始化图形化界面。然后从根目录下的text.txt文本中读取所有用户的账号和密码，将其放入到userLiat这一hashmap中，形成一一对应的映射关系。
2. 初始化连接人数，通过while循环为每一个连接的客户端创建一个单独的服务器线程。
3. 对于每一个启动的服务器，先对客户端传递的身份信息进行验证，若验证失败，则显示某用户连接失败并向客户端发送false信号，表示账号密码错误，若验证成功，则返回正确，同时将该用户添加进已连接用户的ArrayList数组中去，更新已连接人数。同时，在该用户分配一个强制下线按钮，展示在服务器端的指定位置。最后为输入输出流分别创建一个线程，用于向客户端接收和发送信息，在图形化页面中显示相关信息内容。
4. 当服务器点击某个用户的强制下线按钮时，会向客户端发送“#exit”关键字，当客户端接收到该信息时，便会自动关闭并用提示框提示用户“远程主机强制关闭了一个现有的连接”，实现强制下线的功能。同时，服务器端会把该客户端的身份验证等信息从ArrayList中剔除，根据登陆顺序将按钮重新调整位置，实现服务器端手动强制下线客户端某一用户的功能。
5. 当管理员在服务器端的发送框内以username:info的格式发送消息的时候，服务器端会先截取username，到已登录用户信息的数组中去检索，若该用户已登录，则会查询map，获取对应socket将消息发送给该用户。
6. 当服务器接收到一个格式为name：info的信息后，默认认为是A客户端发送消息给名称为name的客户端，服务器再处理后将信息发送给客户端B（用户名为name）。相当于一个转发的过程。
7. 对于客户端来说，首先创建和展示登录窗口，监听用户输入的账号和密码，当用户点击登录按钮时，会和服务器进行一次连接，并将输入的身份信息发给服务端，若身份错误，则接收到true信号，退出该次连接，释放资源。若身份正确，则弹窗登陆成功并展示新的界面，同时创建输入输出流的两个线程，监控服务器消息。
8. （若无格式发送信息）当客户端点击发送消息后，通过输出流将信息发送给服务器端，服务器端收到后显示在界面上。
9. （若发送的信息具有格式name：info），则认为是A客户端向B客户端发送消息，将该消息先发送到服务器，由服务器转发给客户端B。

**UDP：**

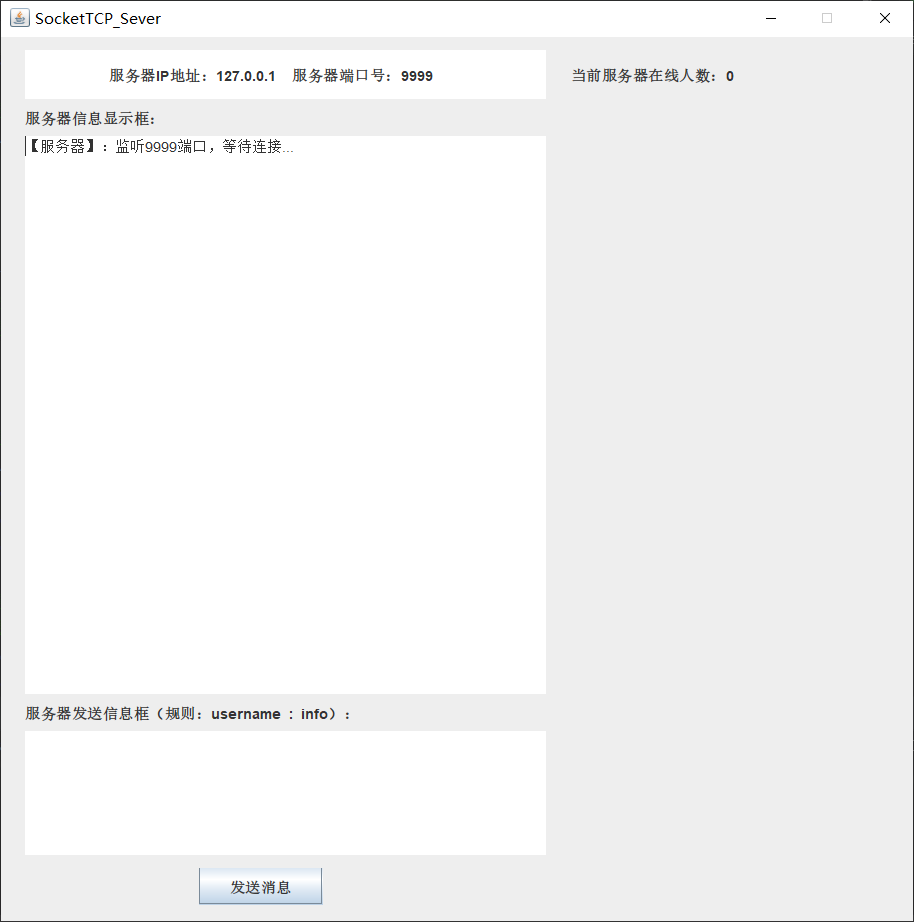
1. UDP相对而言较为简单，无需连接，只需启动服务器端监听发送到指定端口的消息，判断是否为固有用户，是，则接收并将消息转发给目标端口（规则和TCP相同），不判断目标用户是否接收，不是则拒收。
2. 客户端只需将消息以一定的格式发送给服务端即可。

其他：若客户端退出，或服务端退出，由于socket已关闭，资源已释放，另一方均可根据socket的异常抛出判断对方是否下线，从而进行其他操作。

## 用户使用手册

# （TCP）

1. 启动服务器，左上方显示服务器IP地址以及监听端口，下方为服务器日志显示框，右侧为当前服务器在线人数以及强制下线按钮。



**图1 服务器启动界面**

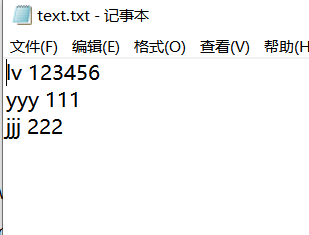
1. 启动客户端，展示客户端登录界面。





**图2 客户端登陆界面**

1. 输入用户名密码，服务器认可的用户名密码存放在根目录下的text.txt文件中。



**图3 服务器认可的账号密码**

1. 客户端登陆成功和失败界面以及服务端日志信息展示。



**图4 登陆失败界面**



**图5 登陆成功页面**

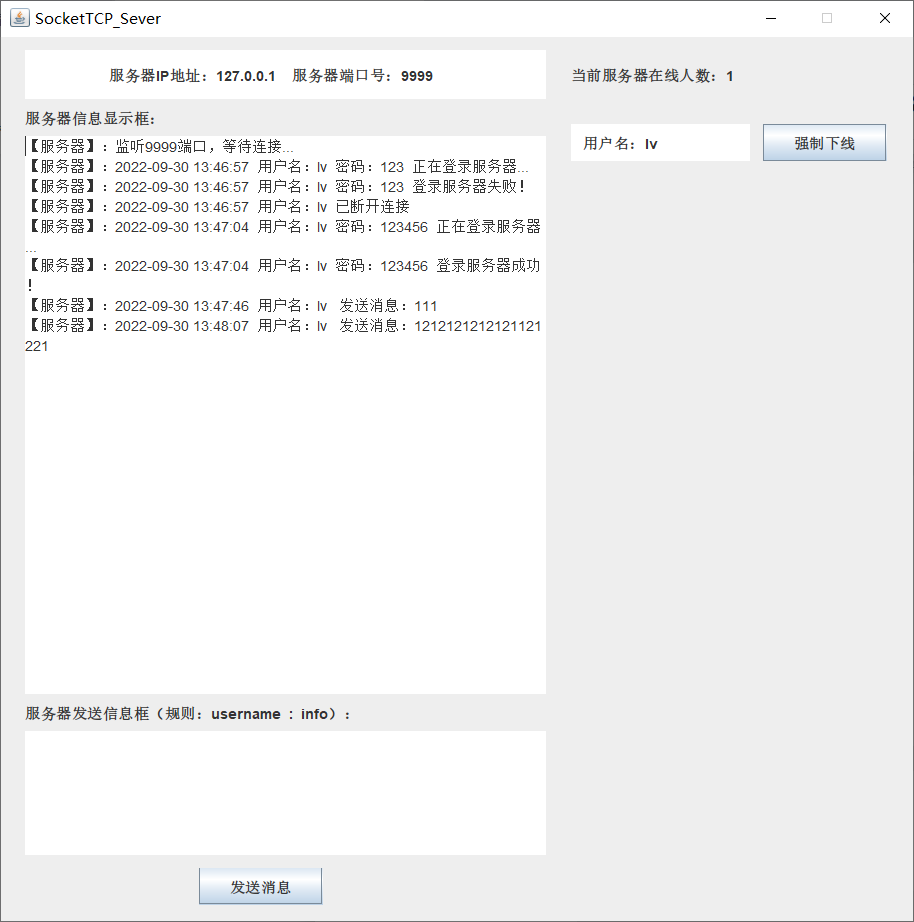


**图6对应服务器信息显示**

1. 客户端发送消息（无格式），服务端接收并显示在页面内

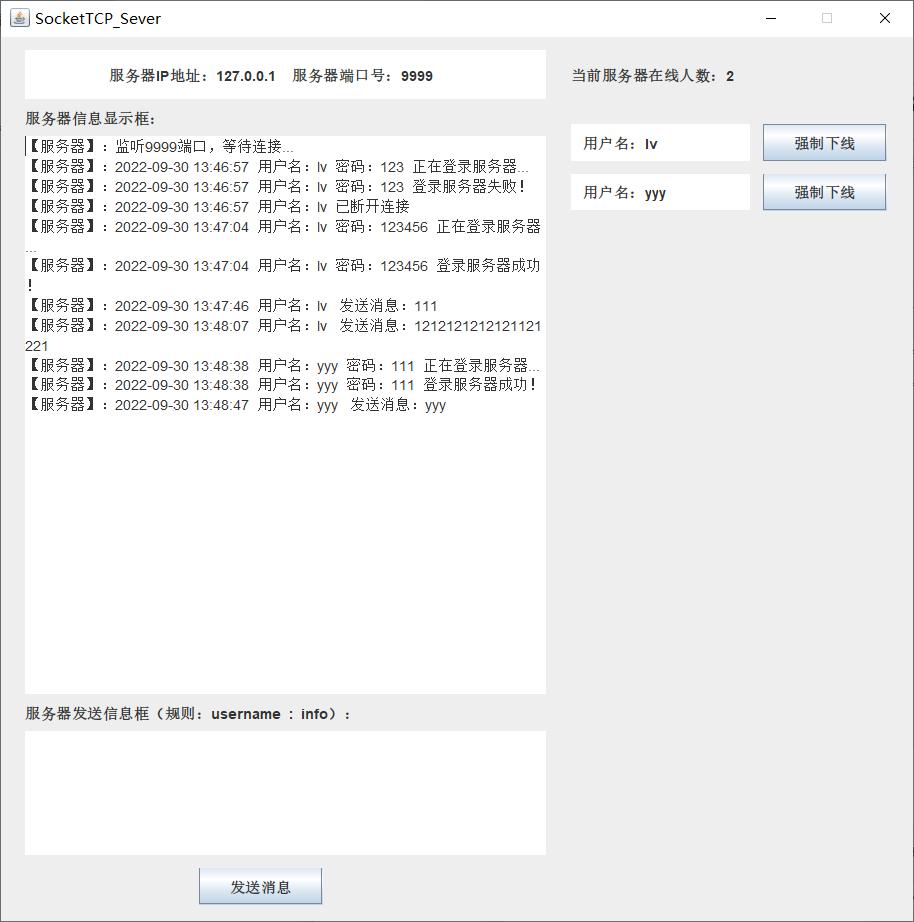


**图6客户端发送信息**



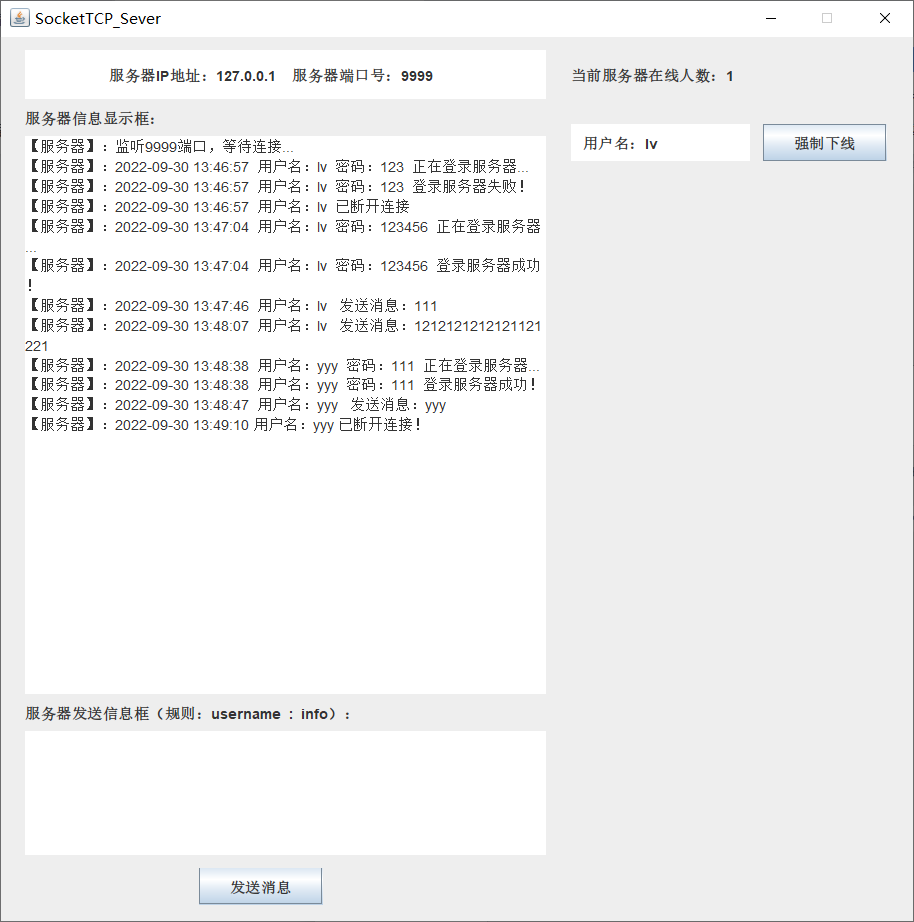
**图7 服务端接收信息**

7、多线程一对多，多个客户端均可同时通信



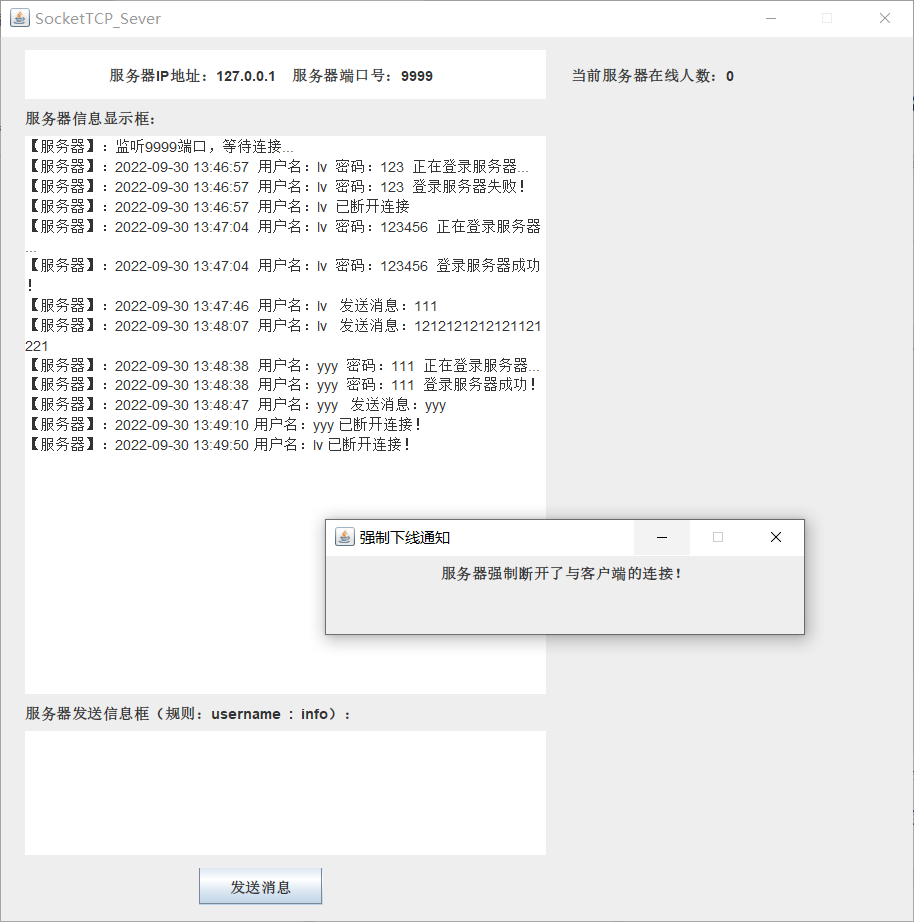
**图8 多线程通信**

8、关闭客户端后服务器端实时进行调整。



**图9 客户端退出，服务器端显示**

1. 服务器强制断开客户端的连接，客户端弹框：“服务器强制断开了与客户端的连接”

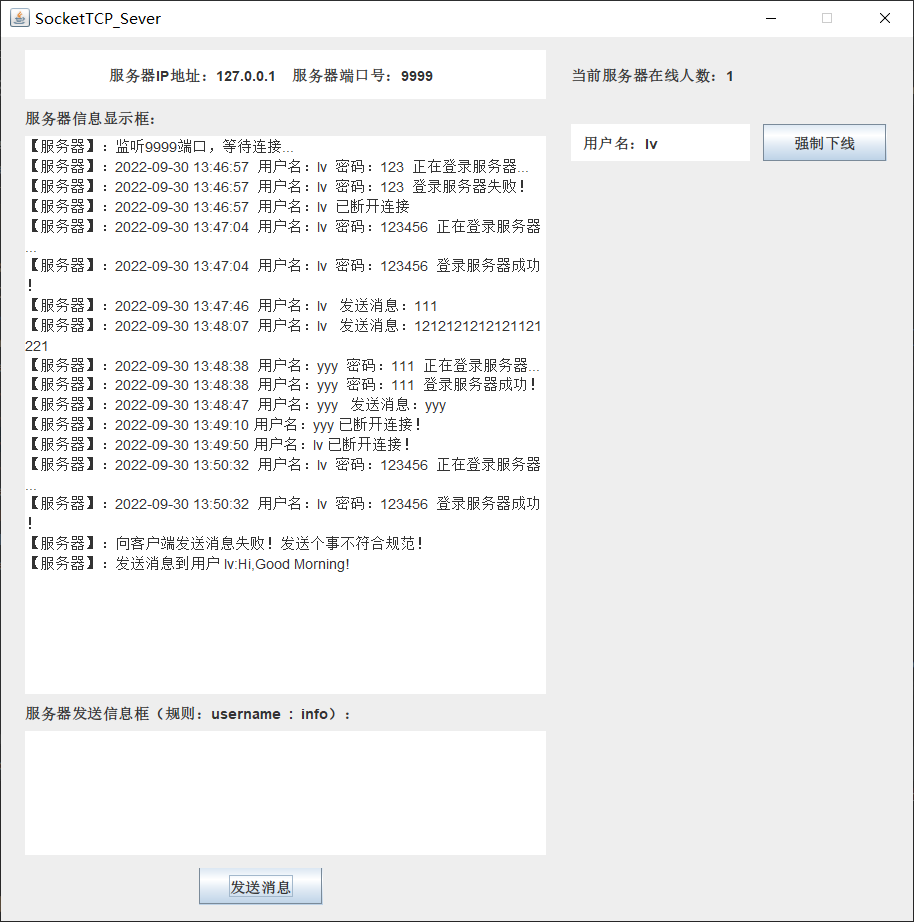


**图10 服务器端强制断开与指定客户端的连接**

10、服务器发送给客户端信息，实现服务器与客户端之间的全双工通信。



**图11 客户端接收服务器消息**



**图12 服务端向客户端发送消息**

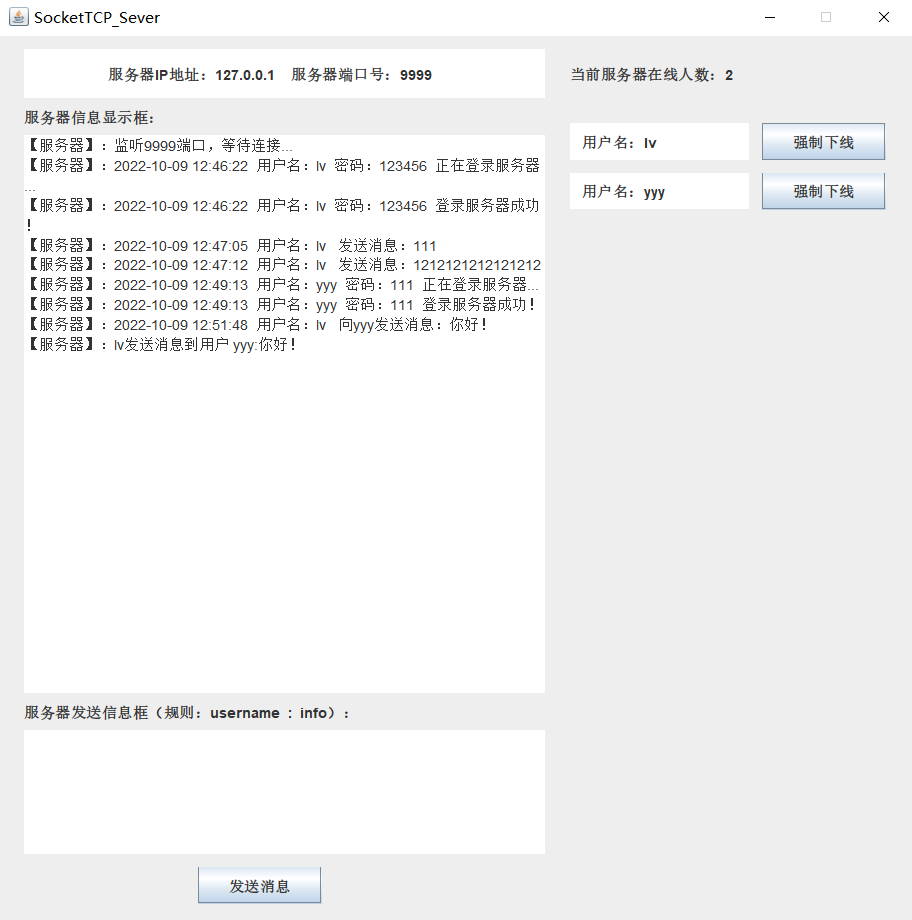
11、客户端间全双工通信，客户端A只需要直到客户端B的用户名并按照规则发送信息，即可将信息发送给客户端B。下图为用户lv发送消息给用户yyy的全过程。



**图13 用户lv发送信息给用户yyy**

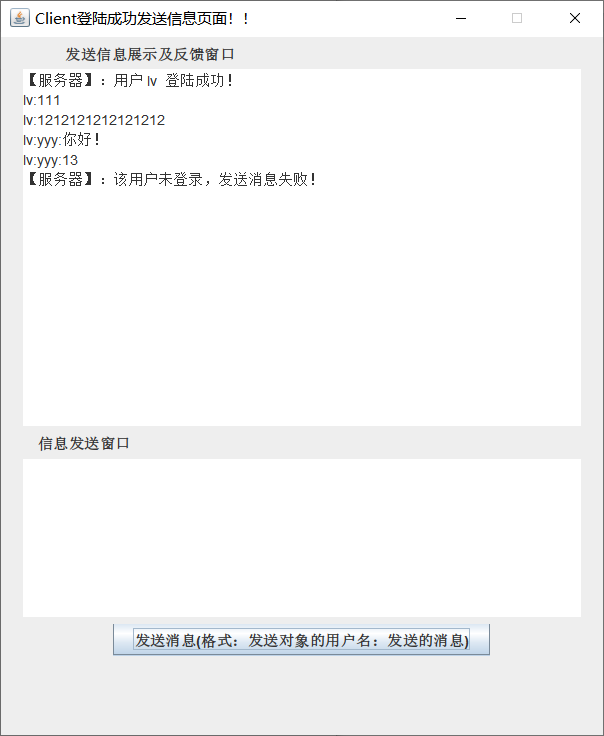


**图14 用户yyy接收到来自用户lv的消息**

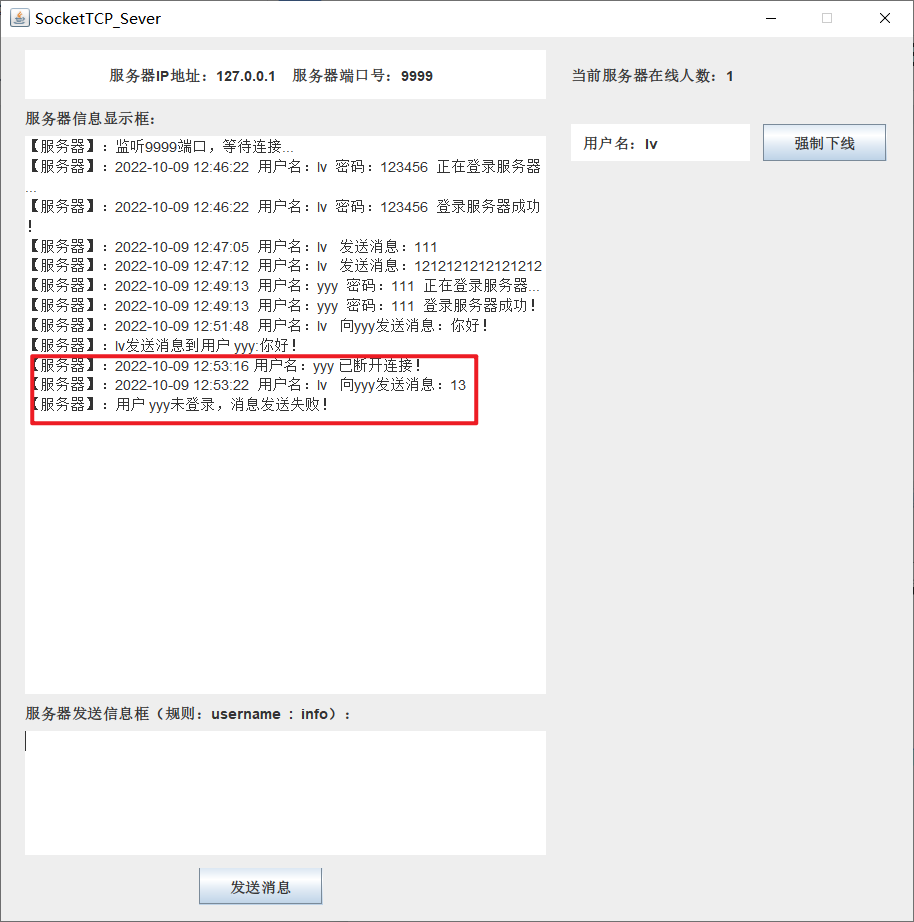


**图15 服务器转发用户lv发送给用户yyy的消息**

1. 当用户下线后，用户lv再发送消息，服务器会告知用户lv 用户yyy已下线，发送信息失败。

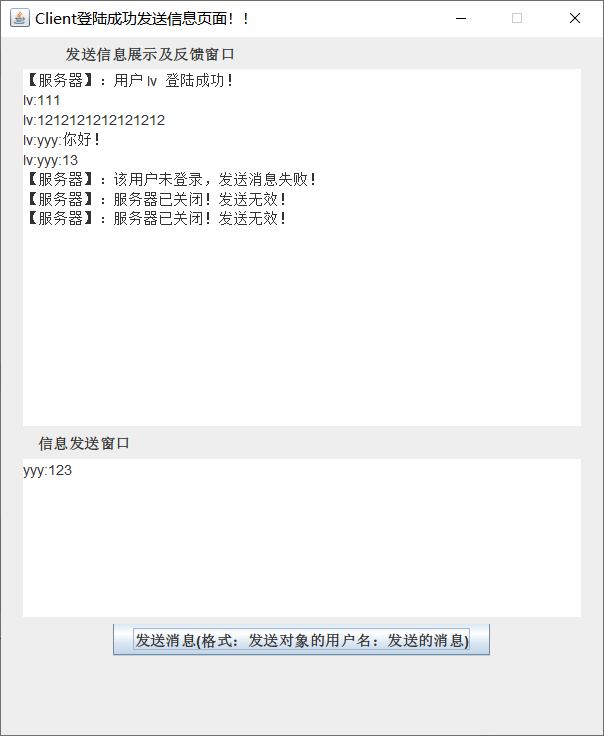


**图16 用户下线后发送消息**



**图17用户yyy下线后服务器显示**

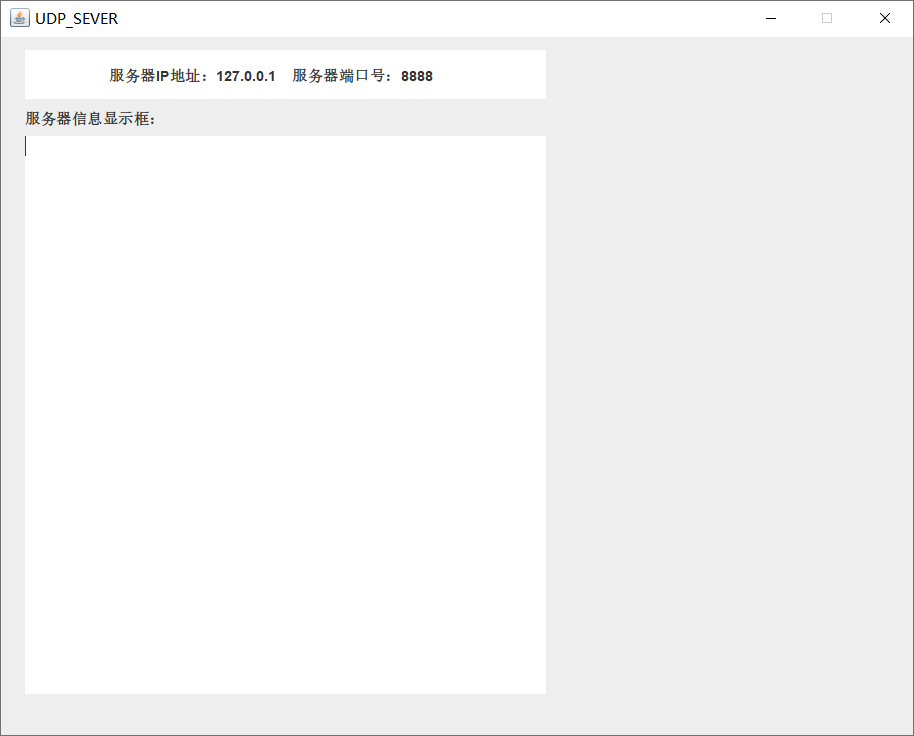
13、服务端已退出，客户端发送信息返回服务器已关闭的消息。



**图18 服务端退出，客户端反应**

# （UDP）

1. 服务器界面以及客户端界面与TCP，只是UDP的服务端少了在线人数，因为UDP是无连接的，所以不需要。



**图19 UDP服务器端**

1. 客户端端口号11111的用户发送消息给端口号为22222的用户信息，实现全双工通信，也由服务端进行转发。



**图20 客户端之间全双工通信，端口11111发送信息**



**图21 客户端之间全双工通信，端口22222接收信息**



**图22 客户端之间全双工通信服务器转发**