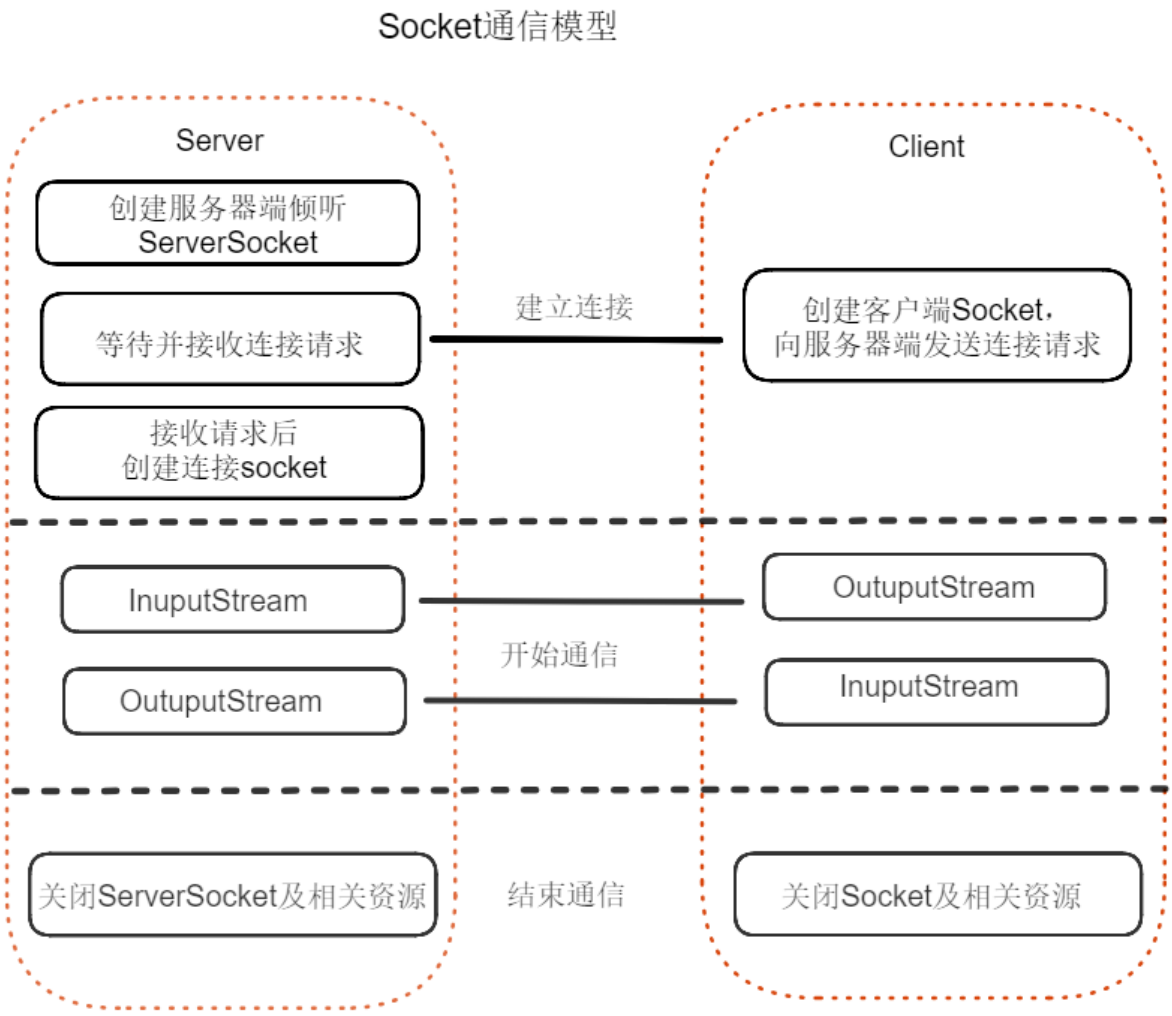
项目演示PPT提纲

项目所实现的主要功能

使用基于 TCP 协议的 Socket 类实现通信

Socket通信模型:



服务器与多客户端的通信:

1.服务器创建ServerSocket，循环调用 accept() 等待客户端连接

```
ServerSocket serverSocket = new ServerSocket(11111);
while (true) {
    Socket socket = serverSocket.accept();
    ...
}
```

2.客户端创建一个socket并请求和服务器端连接

```
socket = new Socket("127.0.0.1", 11111);
```

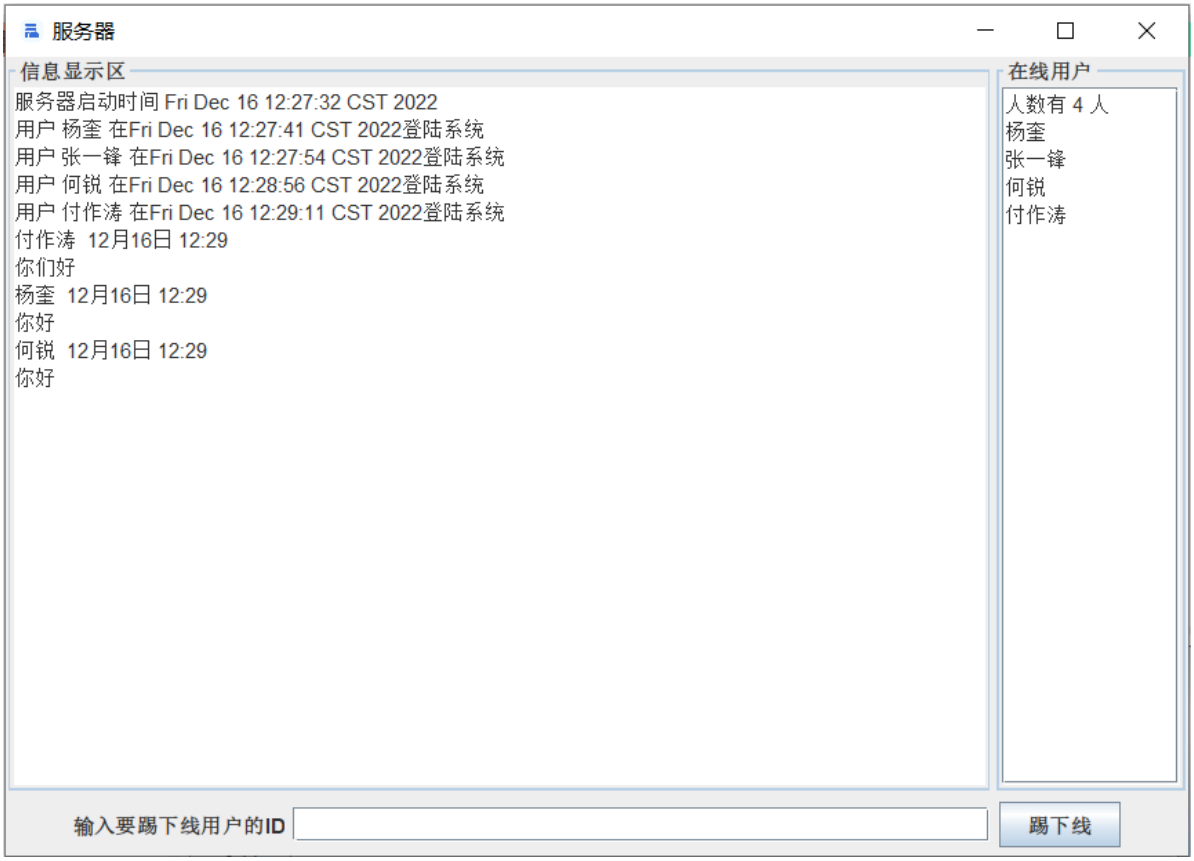
3.服务器端接受客户端请求，创建socket与该客户端建立专线连接

4.建立连接的两个socket在一个**单独的线程**上保持对话

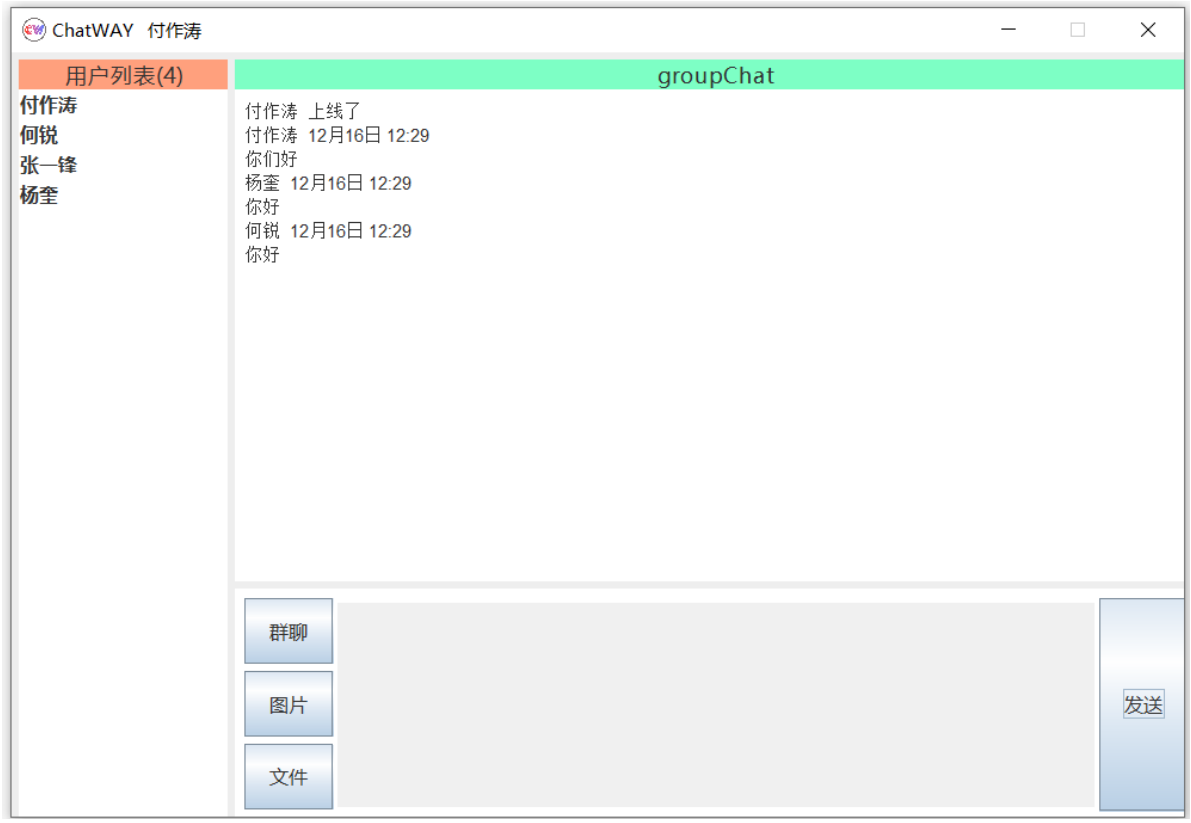
5.服务器端继续等待新的连接

服务器端和客户端图形用户界面（GUI）

服务器端:



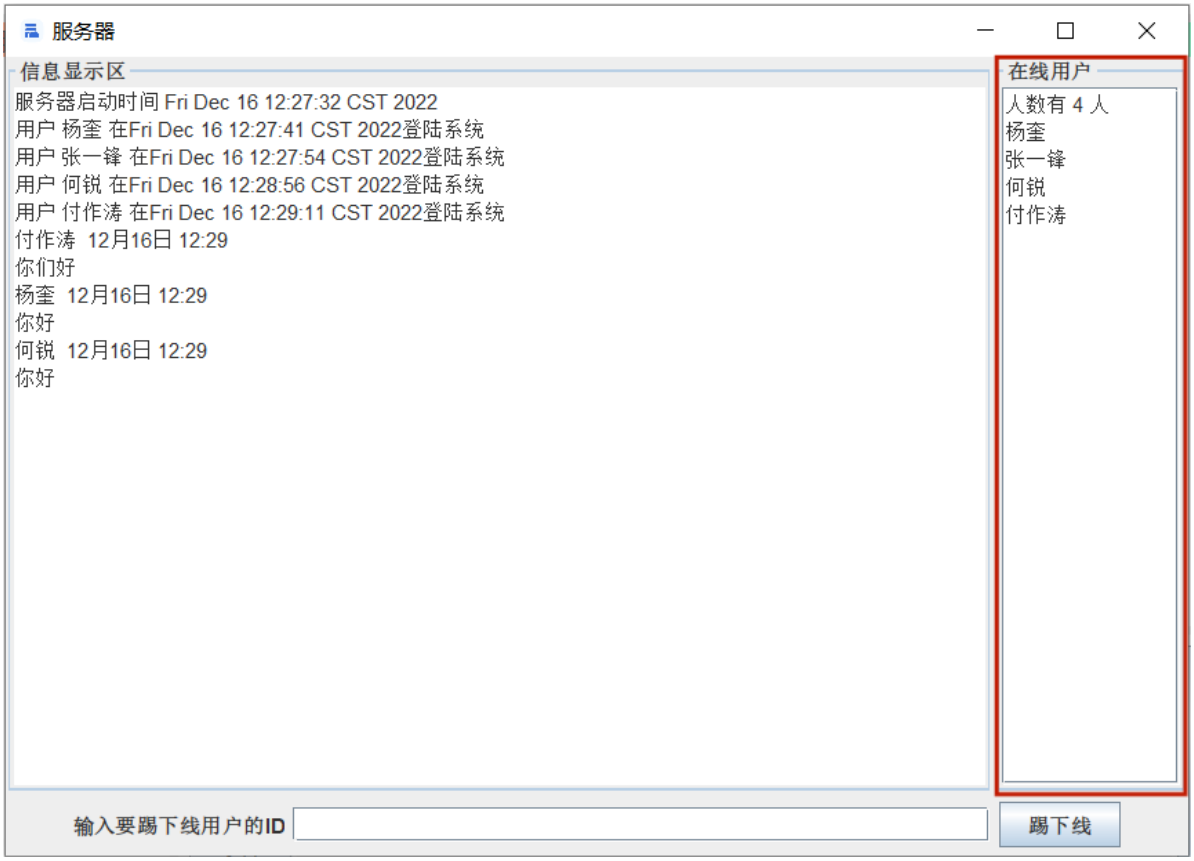
客户端:



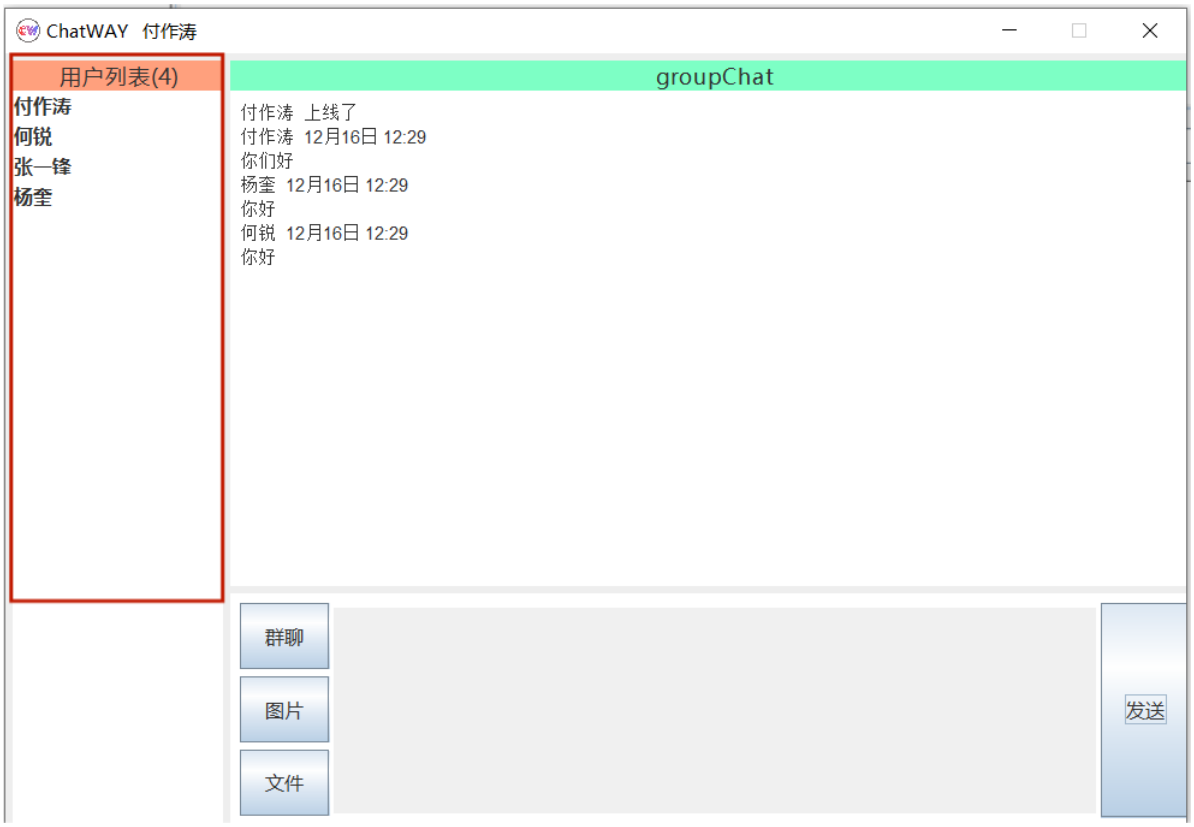
服务器端实时维护一个用户列表，客户端可以读取用户列表

```
//用户名列表，用于展示在线用户
ArrayList<String> username_list = new ArrayList<>();

//用户列表，存放处于连接中的用户信息
ArrayList<User> user_list = new ArrayList<>();

其中 User 类封装了用户名username、密码password、连接socket
```

服务器端用户列表：

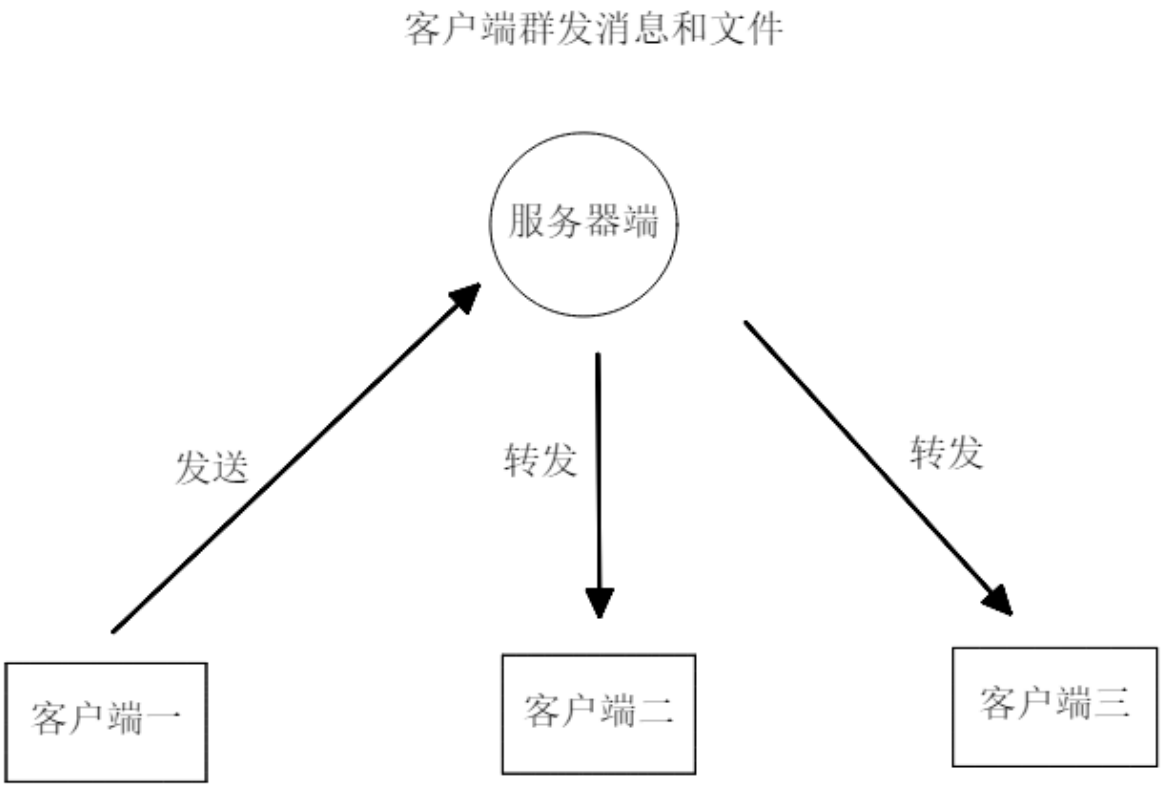

服务器
信息显示区
服务器启动时间 Fri Dec 16 12:27:32 CST 2022
用户 杨奎 在Fri Dec 16 12:27:41 CST 2022登陆系统
用户 张一锋 在Fri Dec 16 12:27:54 CST 2022登陆系统
用户 何锐 在Fri Dec 16 12:28:56 CST 2022登陆系统
用户 付作涛 在Fri Dec 16 12:29:11 CST 2022登陆系统
付作涛 12月16日 12:29
你们好
杨奎 12月16日 12:29
你好
何锐 12月16日 12:29
你好
在线用户
人数有 4 人
杨奎
张一锋
何锐
付作涛
输入要踢下线用户的ID
踢下线

客户端用户列表：

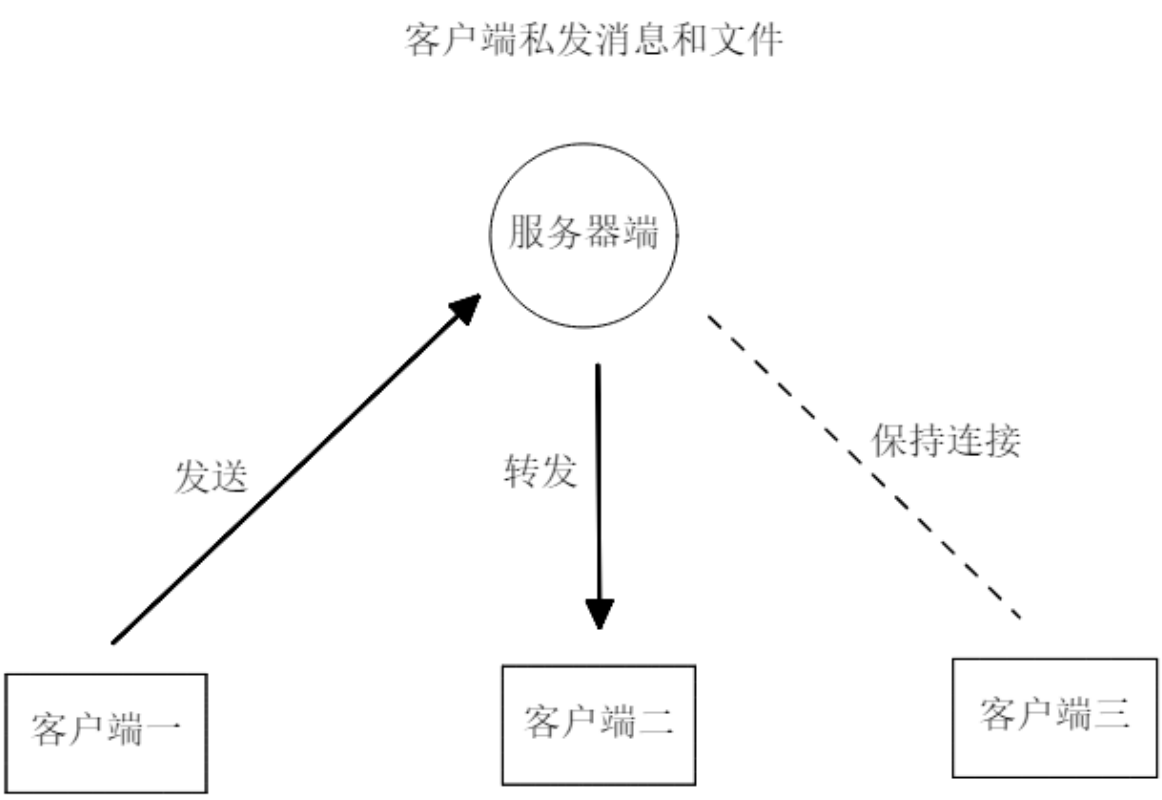

ChatWAY 付作涛
用户列表(4)
付作涛
何锐
张一锋
杨奎
groupChat
付作涛 上线了
付作涛 12月16日 12:29
你们好
杨奎 12月16日 12:29
你好
何锐 12月16日 12:29
你好
群聊
图片
文件
发送

服务器端负责多个客户端之间的数据交换

群发:

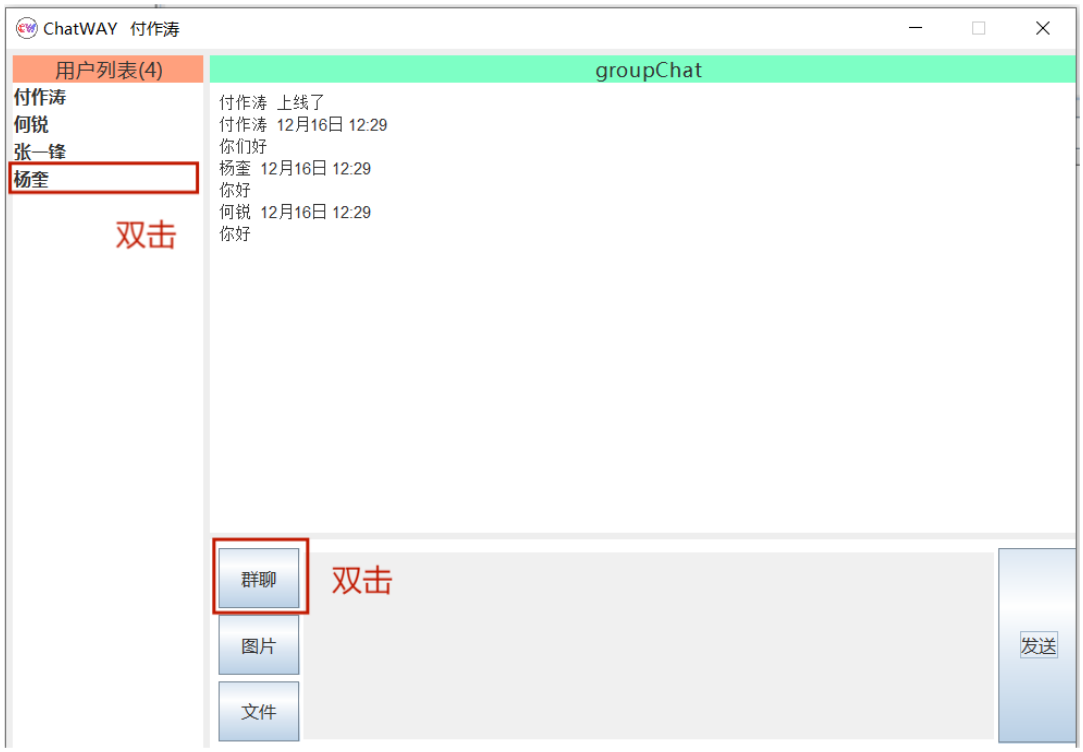


私发:



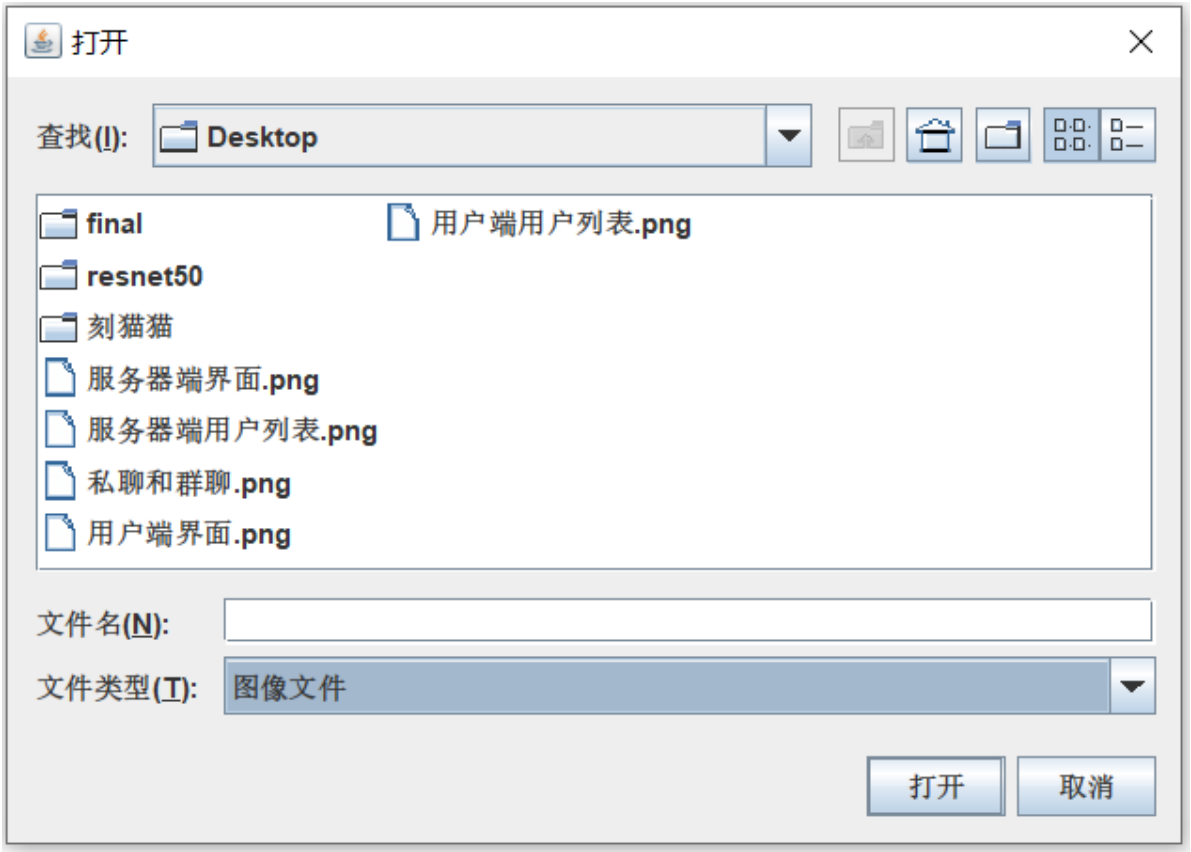
同时支持一对一通信和组通信

1. 用户可在用户列表任选一人双击切换到私聊界面，进行一对一通信，
2. 用户点击群聊按钮选择加入群聊，进行组通信



同时支持文件和图片的私发和群发

与发送消息类似，先切换到私聊或群聊界面，然后点击文件按钮选择文件进行发送



服务器端:

```
//创建文件传输线程
ServerFileThread serverFileThread = new ServerFileThread();
serverFileThread.start();
```

客户端:

```
ClientFileThread fileThread = new ClientFileThread(username);
fileThread.start();
```

使接发消息的线程与接发文件的线程隔离开

技术亮点

数据库管理账号

Duriud连接池

使用阿里巴巴 Druid 连接池接口，连接 MySql 数据库，存放账号信息

```
 InputStream is =
JDBCUtils.class.getClassLoader().getResourceAsStream("druid.properties"); //获取输入流
 pro.load(is);  //读取 Properties
 ds = DruidDataSourceFactory.createDataSource(pro);  //获取连接池对象
```

连接池负责创建和管理数据库连接

上层程序只负责取用与归还

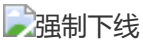
JDBC库

```
//创建JdbcTemplate对象,DataSource即连接池对象
private final JdbcTemplate template = new
JdbcTemplate(JDBCUtils.getDataSource());

//数据库查询
User user = template.queryForObject(sql, new BeanPropertyRowMapper<User>
(User.class),
    login_user.getUsername(), login_user.getPassword());

//数据库更新
int count =
template.update(sql,register_user.getUsername(),register_user.getPassword());
```

使用JDBC库接口方便地对数据库进行操作

服务器端统一管理

服务器端拥有强制下线功能

 强制下线

输入用户名，删除系统中维护的 user_list对应项，断开与客户端的连接并退出客户端进程

客户端之间的通信都要经过服务器端的转发

无论消息还是文件，群发还是私发，流数据都要经过服务器端的转发，保证了通信的有序

自由通信

用户可与任意其他用户进行通信

用户可以双击列表上任意一人与之通信，也可以切换回群聊

```
Map messageMap = new HashMap();
String currentChatObject = "groupChat";
```

保存界面：在每一个用户端进程中创建一个 Map 存储与其他用户、群聊的聊天界面及相关内容

取出界面：currentChatObject 用于从 Map 中取出相应内容。初始化为 "groupChat"，双击后切换为对应用户名

用户可传输任意文件

无论是文本文件、图片文件还是其他文件，全部以**二进制流**的形式进行传输

发送方（接收方与之类似）：

```
File file = new File(path);
fileReader = new DataInputStream(new FileInputStream(file));

fileOut.writeUTF(file.getName());  // 发送文件名字
fileOut.flush();

fileOut.writeUTF(currentChatObject);  // 发送私发对象
fileOut.flush();

fileOut.writeLong(file.length());  // 发送文件长度
fileOut.flush();
int length = -1;
byte[] buff = new byte[1024];
while ((length = fileReader.read(buff)) > 0) {  // 发送内容

fileOut.write(buff, 0, length);
fileOut.flush();
```

```
}
```

服务器:

```
 String textName = input.readUTF();      //文件名
 String people = input.readUTF();        //私发对象
 long textLength = input.readLong();     //文件长度
 if (currentChatObject.equals("groupChat")){    //进行群发
    ...
}else{      //进行私发
    ...
}
```

遵循发送方 -> 服务器 -> 接收方模式

自定义通信规则

传输

```
import net.sf.json.JSONObject;      //导入JSON包
JSONObject data = new JSONObject(); //实例化
```

发送时将JSON对象转换为JSON字符串:

```
String JSONStr = data.toString()
```

接收时转换为JSON对象:

```
JSONObject data = data.fromObject(JSONStr)
```

内容

```
"username":username     //username字段携带用户名
"msg":msg               //msg字段携带信息
"time":time             //time字段携带时间
"private":private       //private字段携带私聊对象
```

互不干扰

聊天线程与文件传输线程分离

服务器端:

聊天线程:ServerThread

文件转发线程:serverFileThread

客户端：

聊天线程：Read

文件接发线程：ClientFileThread

运行效果

放录制视频

自我总结（学到了哪些知识技术，存在的不足与进一步改进的方向）

1. 学习到如何利用 Swing 工具进行图形用户界面开发
2. 学习到如何使用 JAVA SOCKET API 实现网络通信
3. 学习到使用多线程、长连接技术保障通信的稳定