**Messenger项目设计文档**

**专 业：软件工程**

**班 级：2021级**

**姓 名：潘绪康  
 罗怿  
 彭帝霖**

**2024年6月**

**1.项目概述**

项目旨在开发一个类似于飞鸽传书的软件，采用 C++ 语言结合 C++ 标准库和 UNIX 网络编程的 Sockets API，基于 ipmsg 协议的messenger。主要要求实现以下功能：

(1)局域网即时通讯：软件能够在局域网内提供即时通讯功能，用户可以实时发送和接收消息。

(2)局域网文件传输：除了消息通讯以外，还支持在局域网内进行方便的文件传输，用户可以轻松发送和接收文件。

**2.需求分析**

这个项目的需求可以分为两部分：

(1)局域网即时通讯：

I.在线状态显示：显示用户当前的在线状态。

II.实时消息发送和接收：用户可以即时发送和接收消息。

III.消息记录查看：用户可以查看历史消息记录。

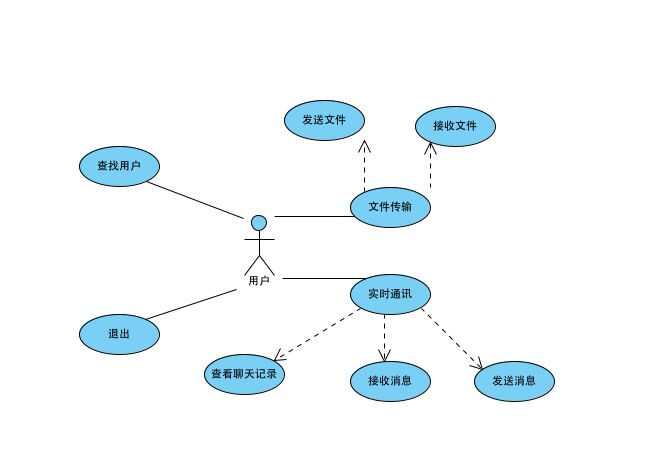
(2)局域网文件传输：

I.文件发送和接收：用户可以直接发送和接收文件。

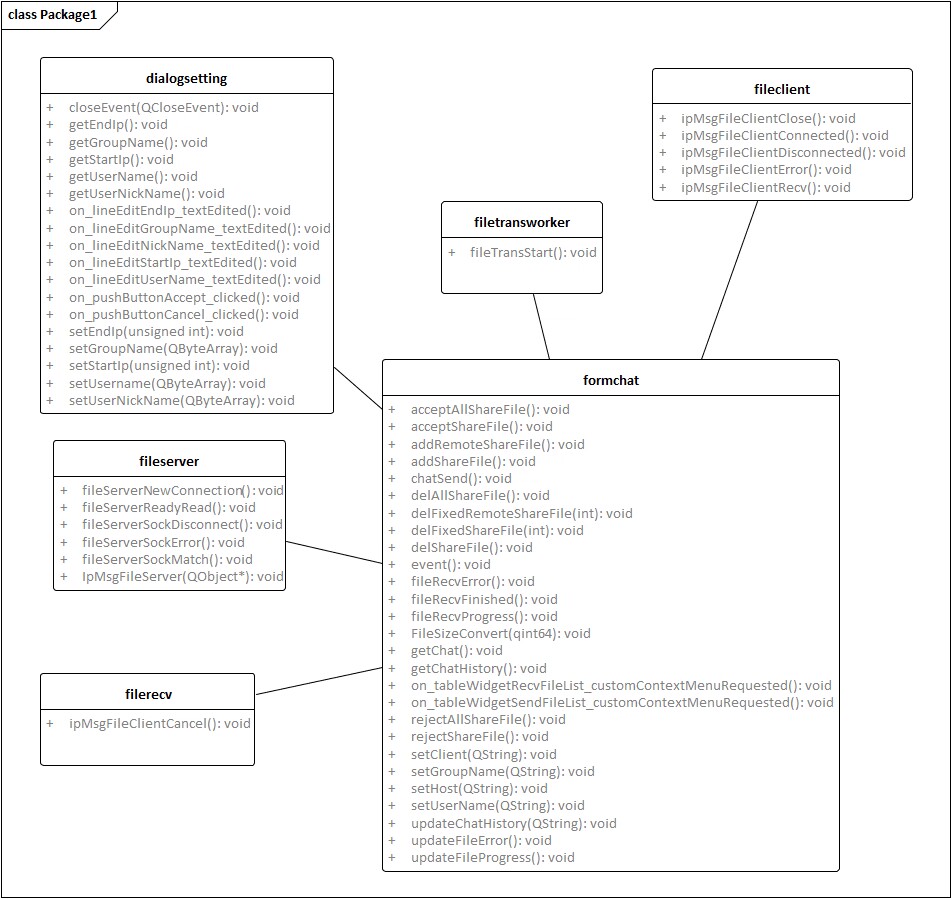
II.文件保存和打开：接收的文件可以保存到本地，并支持打开查看。

**3.UML模型图展示**

(1)用例图



(2)类图



类图描述：

dialogsetting: 该类管理与 IP 地址、用户名和组名相关的设置的用户界面元素。

filetransworker: 该类处理文件传输操作。

fileclient: 该类管理与文件服务器的通信。

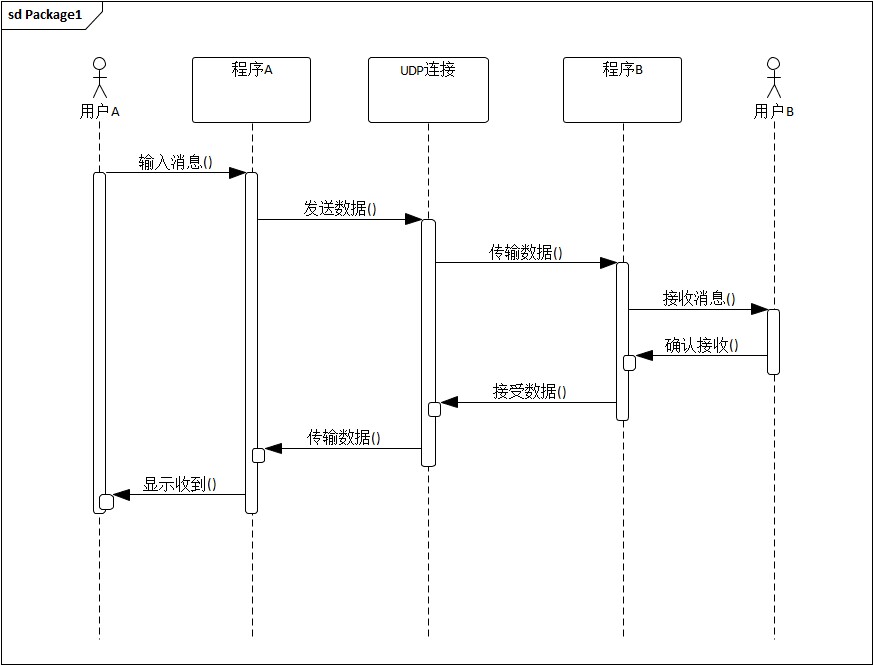
fileserver: 该类处理服务器端操作，以管理连接和文件传输。

filerecv: 该类处理客户端的文件接收。

formchat: 该类充当管理文件共享操作的中心枢纽，包括接受、拒绝和删除文件，以及更新状态信息。

该图强调了这些类之间的关系，展示了它们如何协作以在应用程序中启用文件共享功能。

(3)信息流图



信息流图描述：

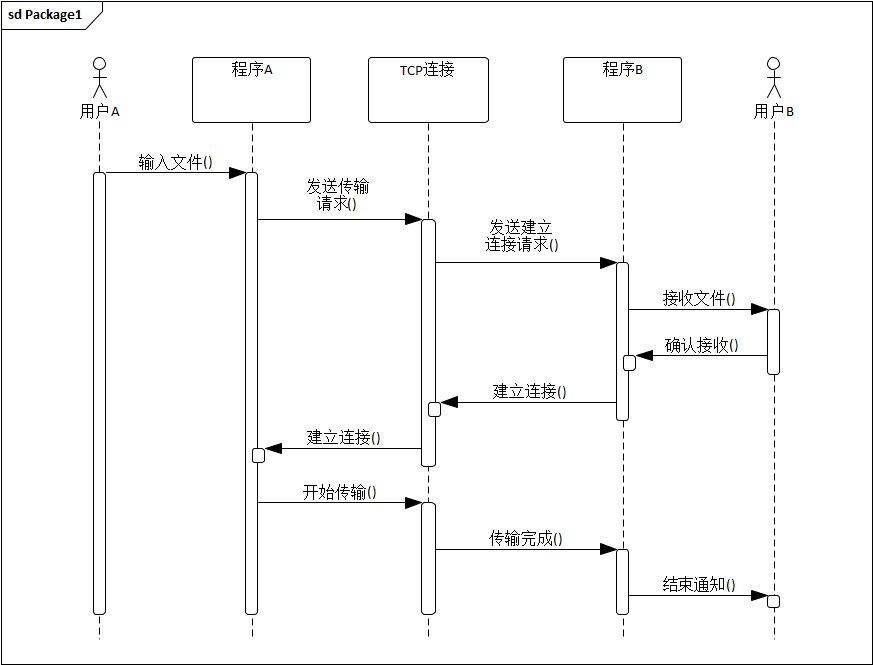
用户A向程序A发送消息，程序A将数据发送至UDP连接。

UDP连接将数据传输至程序B，程序B接收数据。

同时，用户B接收程序B发送的消息。

用户B向程序B发送确认接收的信号，程序B将确认接收信号发送至UDP连接。

UDP连接将确认接收信号发送至程序A，程序A将信号反馈给用户A，表明数据已成功收到。



信息流图描述：

用户 A 向程序 A 发送一个“输入文件”的指令。

程序 A 向 TCP 连接发送一个“发送传输请求”的消息。

TCP 连接向程序 B 发送一个“发送建立连接请求”的消息。

程序 B 接收该请求，并向 TCP 连接发送一个“建立连接”的消息。

TCP 连接将“建立连接”的消息发送回程序 A。

程序 A 向 TCP 连接发送一个“开始传输”的消息。

程序 B 接收“开始传输”的消息，并向用户 B 发送一个“接收文件”的指令。

用户 B 接收该指令，并向程序 B 发送一个“确认接收”的消息。

程序 B 将“确认接收”的消息发送至 TCP 连接。

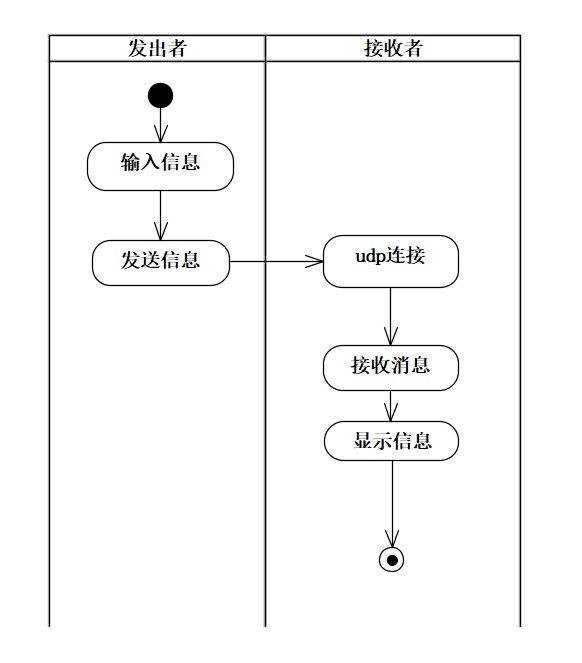
TCP 连接将“确认接收”的消息发送回程序 A。

程序 A 接收到“确认接收”的消息，并向 TCP 连接发送一个“传输完成”的消息。

TCP 连接将“传输完成”的消息发送至程序 B。

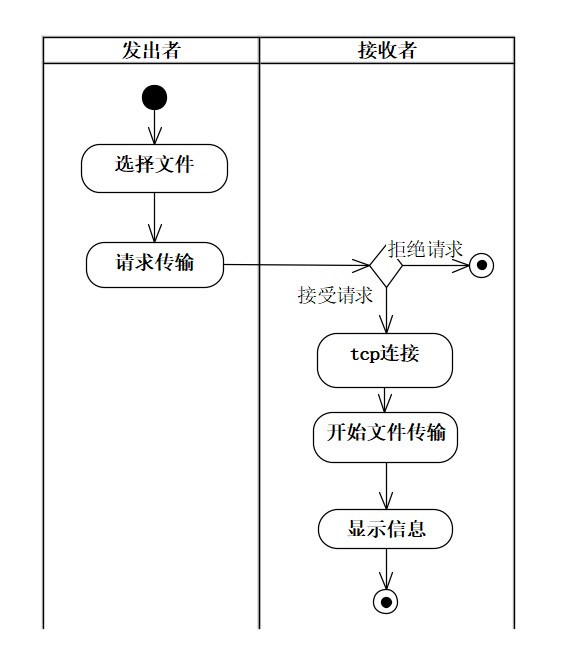
程序 B 接收到“传输完成”的消息，并向用户 B 发送一个“结束通知”的消息。

(4)活动图



活动图描述：

首先，发送者开始执行，并输入信息。然后，发送者将信息通过 UDP 连接发送至接收者。接收者通过 UDP 连接接收数据并解析。最后，接收者将接收到的信息显示出来。整个流程中，发送者负责发送信息，而接收者负责接收并显示信息，中间通过UDP连接进行数据传输。



活动图描述：

发送者流程:  
发送者首先选择要传输的文件。然后发送者向接收者发出请求传输的指令。  
接收者流程:  
接收者收到请求传输的指令后，可以选择接受或拒绝。如果接受，则建立 TCP 连接。建立连接后，接收者开始接收文件传输。文件传输完成后，接收者向发送者发送确认信息。该图展示了简单文件传输的流程，其中发送者发起请求，接收者接受请求并完成传输。

**4.系统实现**

(1) 开发环境

操作系统： Manjaro

编程语言： C++ ,Qt 6.7.1

GUI 库： Qt 6.7.1

开发工具： Qt Creator

(2) 关键技术实现

IPMSG 实现局域网聊天和文件传输：

I.消息格式定义： 明确定义消息的发送者、接收者和内容结构。

II.IPMSG 命令处理： 实现发送消息、接收确认和文件传输请求的 IPMSG 命令处理。

III.通信流程： 发送设备依据 IPMSG 协议创建消息，通过网络传送至接收设备。接收设备解析消息内容，并在用户界面上展示。