

Echoo开发文档

项目概述

目标

Echoo 是一个基于C++、Qt、QML、TCP/UDP、SQLite编程实现的简单的在Linux操作系统下，P2P模式的局域网通信聊天软件，支持用户注册、登录、发送消息、添加好友、删除好友、设置头像等用户信息等操作。该软件使用Qt信号与槽、C++与QML联动等技术，使用QML实现图形化操作界面并连接到C++逻辑端进行通信交流。

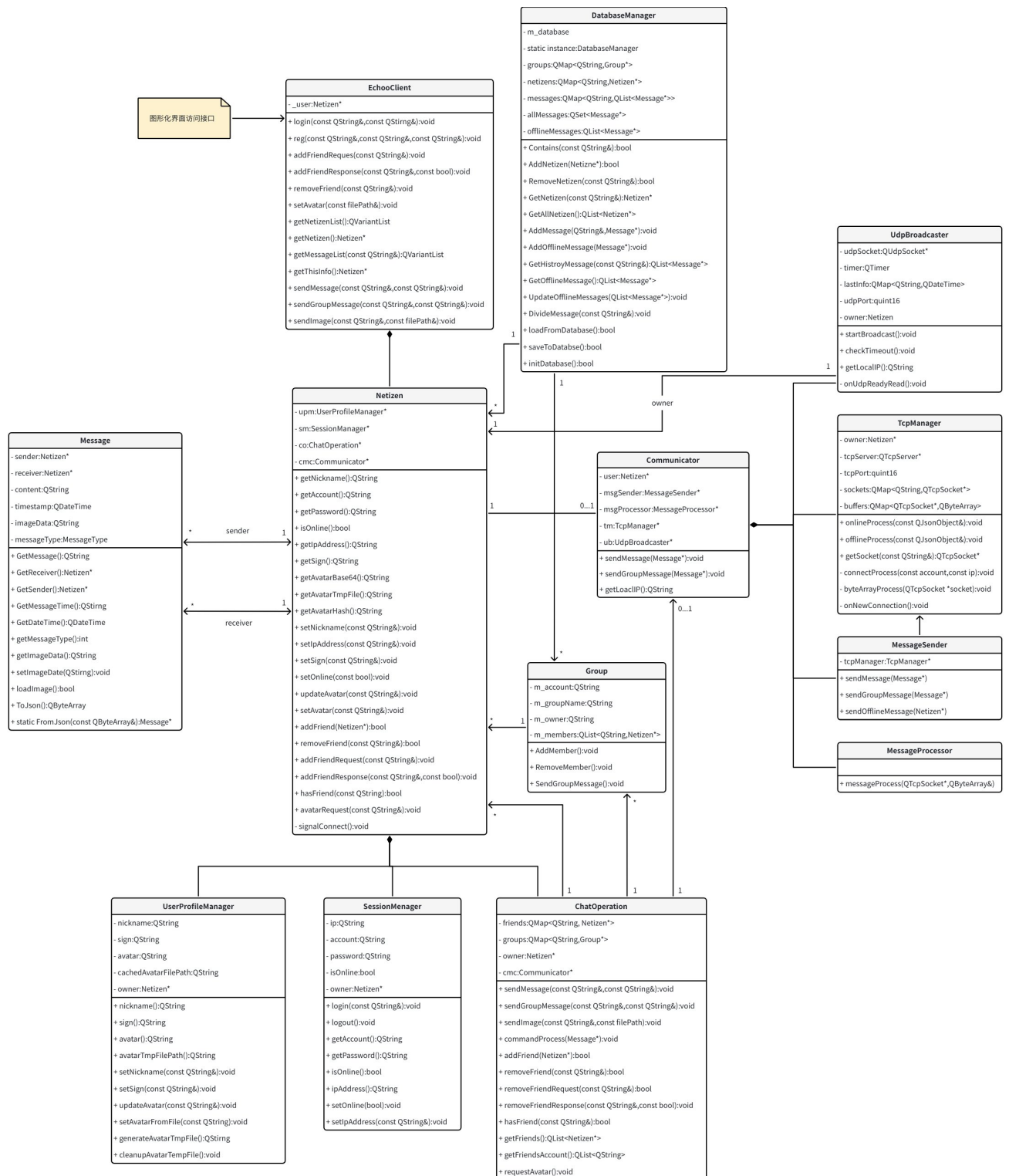
技术栈

- 编程语言：C++
 - 框架：Qt、QML
 - 数据库：SQLite
 - 网络协议：QTCPSocket/QUdpSocket、QTcpServer
 - 操作系统：Linux
-

项目结构

C++端

UML类图



QML集成 (echooclient.h)

- 为QML提供注册、登录、发送消息等操作接口

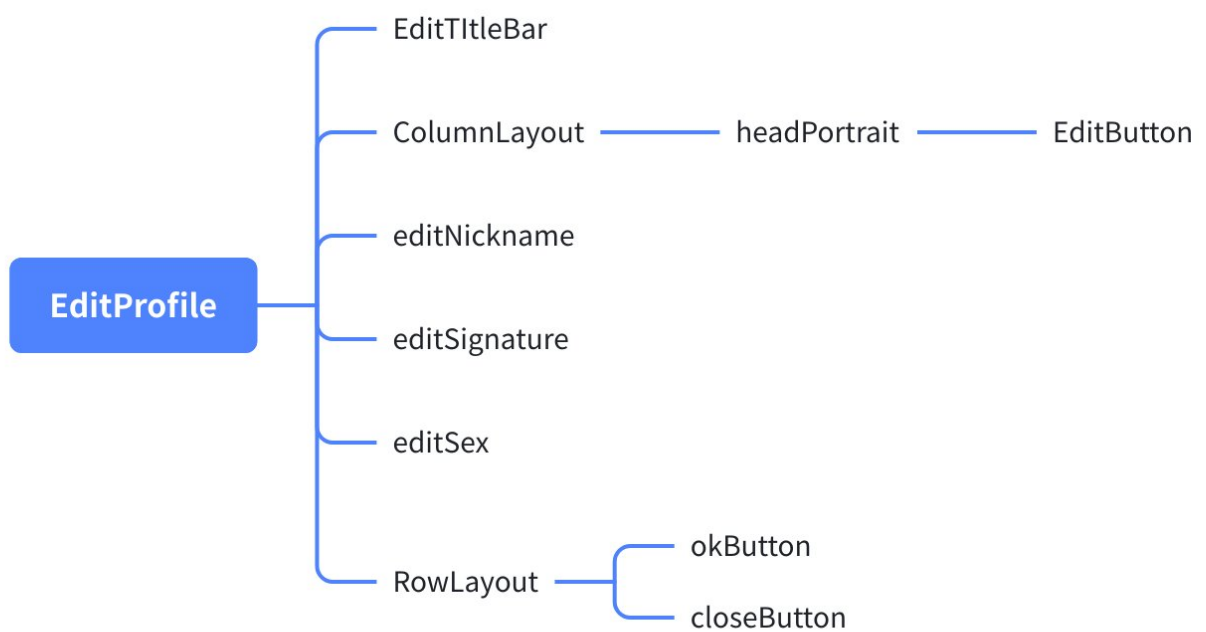
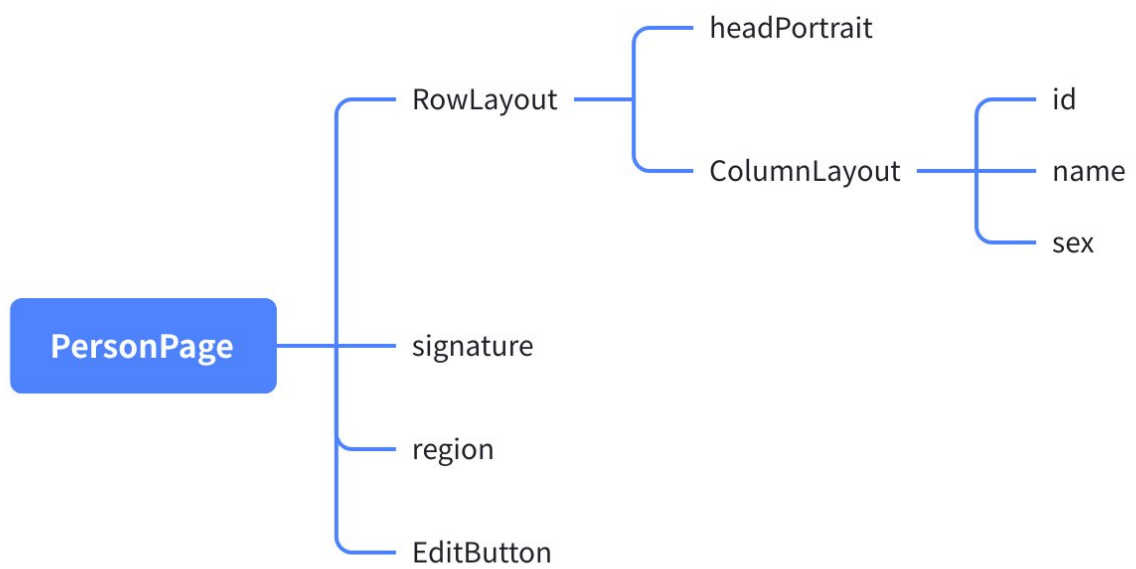
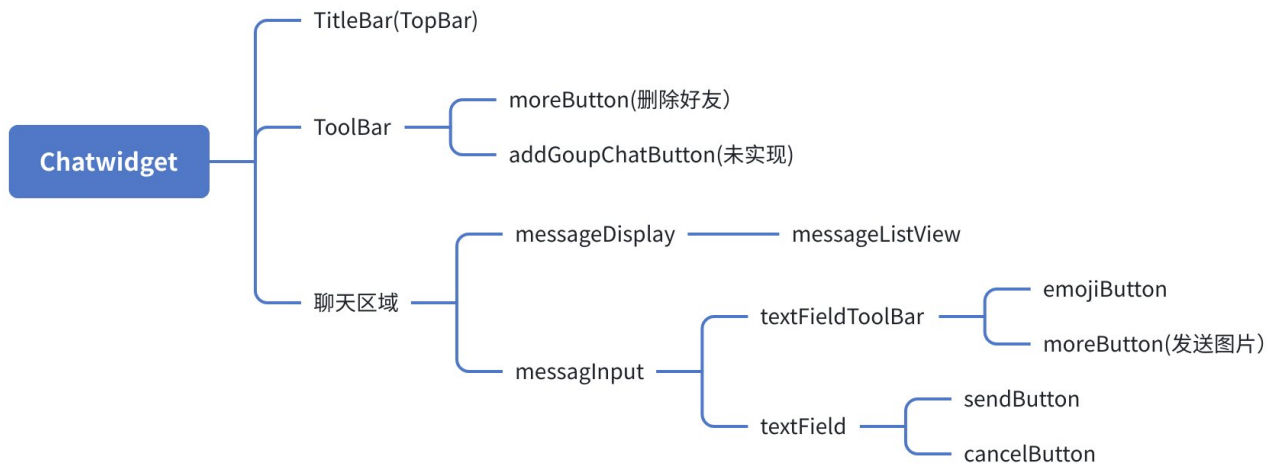
用户相关类

(netizen.h, userprofilemanager.h, sessionmanager.h, chatoperation.h, communicator.h)

- 用户实体类，用于实现用户的各类操作如登录、注册、发送消息等，并分化出各个子模块用于管理特定的操作

- netizen：集成各子模块的操作
- userprofilemanager：用于管理用户的昵称、签名、头像等信息
- sessionmanager：用于管理用户的ip地址、账号密码、在线状态等
- chatoperation：用于管理用户的好友、群组、聊天等操作
- communicator：用于建立局域网通信
- 局域网通信类
 - (communicator.h, messagesender.h, messageprocessor.h, tcpmanager.h, udpbroadcast.h)
 - 通信器实体类，用于发送UDP广播、建立TCP连接、传输消息等操作，并分化出各个子模块用于管理特定的操作
 - communicator：集成各子模块的操作
 - messagesender：用于处理消息发送等操作
 - messageprocessor：用于处理接受的消息并发出对应的消息信号
 - udpbroadcast：用于发送UDP广播，广播当前用户的信息以及IP地址等
 - tcpmanager：用于与收到的UDP广播的用户建立TCP连接
- 数据库管理类 (databasemanager.h)
 - 用于存储用户信息、消息信息等，使用SQLite存储在本机
- 实体对象类 (message.h, group.h, netizen.h)
 - message：消息实体类，用于存储消息的发送者、接收者等信息，并实现序列化与反序列化等操作进行网络传输中的字节流传输
 - group：群组实体类，用于记录群组中的用户信息（碍于时间问题，群组功能暂未实现）
 - netizen：用于实体类，用于记录用户的属性，实现用户的各种操作等

QML端





- 账号窗口

- LoginForm.qml

- 登陆界面，用户需输入账号和密码
 - 可通过点击密码旁的眼睛来查看自己输入的密码

- RegForm.qml

- 注册界面，用户需输入昵称，账号和密码
 - 注册成功与否会有相应的提示

- StartWindow.qml

- 通过点击左上角的图标切换状态（注册或登录）
 - 登录成功就关闭当前页面，并显示主页面
 - 登录失败会有相应的提示，就需重新输入

- 聊天窗口

- Chatwidget.qml

- 实现聊天记录，发送表情，文字消息（图片消息可选择，但部分功能未实现），删除好友
 - TopBar.qml
 - 显示对方昵称，缩放窗口
 - ToolBar.qml
 - 实现删除好友功能
 - 实现添加群聊（未完成）

- 删除双方消息列表和好友列表的对应好友
 - 如果聊天窗口或者好友主页未关闭，则一起关闭
- MessageWidget.qml
 - 显示聊天消息
 - 定位最后一条消息
 - 显示发送时间
 - 显示发送者
- EmojiPopup.qml
 - 发送表情
- MessageItem.qml
 - 显示单个消息，包括联系人昵称和头像，最后一条消息，时间戳，未读消息
- 好友窗口
 - AddFriend.qml
 - 在搜索栏输入想加的好友的好友完整账号，如数据库中有相应的账号则显示
 - 如果已有该好友，则提示已是好友，如果还不是好友，点击“加好友”按钮发送好友请求待处理
 - FriendItem.qml
 - 显示单个好友信息，包括好友昵称，头像，签名
 - Friends.qml
 - 显示分组里的总好友
 - 点击单个FriendItem时会创建对应的好友个人主页已显示好友信息
 - Friendpersonpage
 - 显示好友的个人信息，包括好友头像，昵称，账号，签名等
 - Friendnotification
 - 当有用户发送好友请求时显示出来
 - 点击“同意”按钮后会将该用户添加为好友，并将其加在好友列表和消息列表上
- 主窗口
 - Friendtotal.qml
 - 显示好友通知与好友
 - 点击“好友通知”会创建对应的好友通知窗口
 - 点击“我的好友”分组会展示该用户的所有好友

- Mainpagetitlebar.qml
 - 显示搜索栏（还没来得及实现）
 - 点击“+”号会创建加好友的窗口
 - 点击“-”号实现最小化页面
 - 点击“x”号实现关闭程序
 - Maintotal.qml
 - 点击工具栏不同的图标切换对应的主页面
 - 现阶段实现了好友总页面与消息总页面
 - Messagetotal.qml
 - 显示与所有好友的聊天组件
 - 点击对应好友的MessageItem就会创建对应的聊天框
 - 当添加或删除好友会更新好友列表
 - 当发消息或收到消息就会更新对应的MessageItem的时间戳，最后一条消息，以及将其移到消息列表的最上方
 - ToolBars.qml
 - 给每个主页面设置了一个切换的图标，当点击时会切换到对应的页面上
 - 目前只实现了查看个人主页，消息页面，好友列表三个页面
 - 个人主页
 - PersonPage.qml
 - 显示好友主页（通过点击ToolBar.qml的头像区域可查看，再次点击可以关闭）
 - EditProfile.qml
 - 编辑个人头像，签名，昵称等
 - EditTitle.qml
 - 操作窗口
 - 模板
 - FrameLessWindow.qml
 - 实现无边框窗口和全局拖拽功能
-

主要类和方法概述

EchooClient类

- login：登录账号
- reg：注册账号
- addFriendRequest：发送添加好友请求给接收方
- addFriendResponse：发送添加好友回应给请求方
- setAvatar：调用Netizen类方法，传入图片路径，设置用户头像，并转换为base64进行存储

Netizen类

- loginDetection：进行密码检测，有EchooClient调用访问需要登录检测的用户
- getAvatarBase64：返回base64格式的头像数据
- getAvatarTmlFile：由于Qt内存管理的问题，多处引用同一base64会导致崩溃，因此采用生成临时头像文件的形式供QML中Image组件使用，返回base64生成的头像文件的路径
- getAvatarHash：返回头像的哈希值，用于在UDP广播中传播，防止过量的数据传播，当接收方检测到对方发送的头像哈希值与本地储存的头像哈希值不同时，接收方会向发送更新头像请求，并通过TCP传送新的头像信息

UdpBroadcaster类

- startBroadcast：通过定时器调用，每隔5s发送一次广播信息，广播当前用户的各类信息表示用于在线
- checkTimeout：检测本地记录到的接收到广播的用户距上一次广播的时间是否超过了预定值（15s），若超过则视为超时掉线，并作出相应处理
- onUdpReadyRead：处理接收到的UDP广播信息，更新本地记录的用户上一次广播时间

TcpManager类

- onlineProcess：处理用户在线信息，建立TCP连接，并判断是否存在于本地数据库，更新相应的用户信息
- offlineProcess：用户断线处理，清楚在本地残留的各个数据

Communicator类

- signalsConnect：连接各个子模块之间的信号与槽，管理各个子模块，实现局域网通信

DatabaseManager类

- initDatabase：初始化数据库，从本地数据库读取信息
- loadFromDatabase：从数据库中读取信息载入内存

- saveToDatabase：析构时调用，将内存中的相关信息进行更新到数据库中

Message类

- toJson：将消息的各个属性序列化为字节序列，用于TCP传输
 - fromJson：静态方法，对传入的字节序列进行检测，若符合则反序列化为Message类，并返回对应指针
-

主要功能及需求

1.注册账号

输入: 用户名（文本） 账号（文本） 密码（文本）

处理:验证账号是否已被占用，保存用户信息到数据库

输出:注册成功或失败的消息提示

2.添加好友

输入:好友的账号

处理:定位到好友账号，发送好友请求，待好友同意之后保存好友关系到数据库

输出:好友列表更新

3.删除好友

在与该好友的聊天框里点击“删除”

处理:找到好友对应的消息列表和好友列表，从数据库中删除当前用户关于此好友的数据

输出:删除好友成功或失败的消息提示，好友列表更新

4.局域网通信

输入:消息内容（文本）

处理:发送消息到目标用户，保存消息到数据库

输出:消息发送成功或失败的消息提示，聊天窗口更新显示发送和接收的消息

5.离线消息接收

输入:消息内容（文本）

处理:发送消息到目标用户，但要发送方在线时接收方登陆后才能接收到离线消息

输出:聊天窗口更新显示发送和接收的消息

6.设置个人主页

输入:个人信息（如昵称、头像、签名等）

处理:更新用户个人信息到数据库

输出:个人主页信息更新显示

7. 连接数据库并存储功能

输入:连接本地数据库，若本地不存在则会自动生成新的数据库文件

处理:连接到数据库，存储用户信息、好友关系、聊天记录等数据

安装与配置

获取安装包

代码块

```
1 // 访问网址，在releases处下载软件安装包
2 https://github.com/SLSkuy/Echoo
```

根据安装包提示进行安装

安装完成后在菜单栏搜索Echoo即可运行程序

获取源码

代码块

```
1 git clone https://github.com/SLSkuy/Echoo.git
2 // 在Client目录执行cmake
```

编译完成后执行编译后的可执行程序即可