《C++程序课程设计》开发文档

题目：仿QQ聊天程序

班级：24-1班

组号：14

成员：2410120025 郭警豪

2410120034 陈晓豪

日期：2025年6月15日

1. 程序说明

本小组开发了一个仿QQ的聊天程序，主要实现了以下功能：

- **用户注册与登录**：用户可以通过输入用户名和密码进行注册和登录。
- **好友管理**：用户可以搜索其他用户并添加为好友，查看好友列表，同意或拒绝好友申请。
- **聊天功能**：用户可以与好友进行单聊，发送和接收文本消息，并查看历史聊天记录。

该程序使用Qt框架开发客户端界面，Sqlite数据库存储用户信息和聊天记录，通过Socket实现客户端与服务端的通信。

---

2. 程序分析

功能需求

- **用户管理**：支持用户注册、登录，存储用户信息。
- **好友管理**：支持搜索用户、添加好友、查看好友列表、处理好友申请。
- **聊天功能**：支持单聊，发送和接收消息，查看历史消息。

性能需求

- **响应速度**：用户操作（如登录、发送消息）应在短时间内得到响应。
- **数据存储**：用户信息和聊天记录需安全存储，支持历史消息查询。
- **并发处理**：支持多个用户同时在线，处理并发通信。

---

3. 程序设计

设计思路

- **客户端**：使用Qt框架开发图形用户界面，实现用户交互功能。
- **服务端**：处理客户端请求，与数据库交互，存储和查询数据。
- **通信**：通过Socket实现客户端与服务端的通信，传输JSON格式的数据。

数据库设计

- **用户表（`users`）**

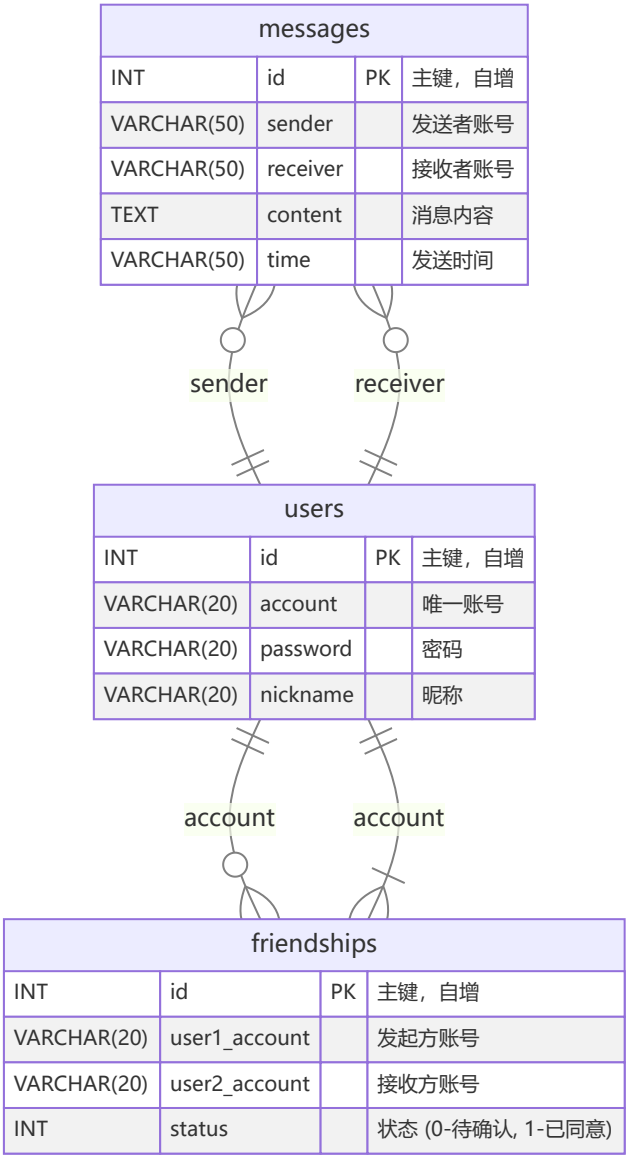
字段名	数据类型	描述
`account`	VARCHAR(20)	用户账号（主键）
`password`	VARCHAR(50)	密码
`nickname`	VARCHAR(50)	昵称

- **好友关系表（`friendships`）**

字段名	数据类型	描述
`user1_account`	VARCHAR(20)	用户1账号
`user2_account`	VARCHAR(20)	用户2账号
`status`	INT	好友状态（0：申请中，1：已通过）

- **消息表（`messages`）**

字段名	数据类型	描述
`sender`	VARCHAR(20)	发送者账号
`receiver`	VARCHAR(20)	接收者账号
`content`	TEXT	消息内容
`time`	DATETIME	发送时间



类与函数设计

- **客户端**
  - `Register`：注册界面类，处理用户注册逻辑。
  - `MainWindow`：主窗口类，处理用户登录。
  - `ChatWindow`：聊天窗口类，处理消息发送和接收。
  - `Index`:主界面类，处理界面转化。
  - `FriendManagement`：好友管理界面类，处理好友管理。
  - `FriendItemWidget`：好友元素类，用于好友显示。
  - `DragEvent`:拖动事件类，处理拖动事件。
  - `Information`：信息类，处理用户信息显示与发送好友申请。
  - `MessageBubbleWidget`：消息气泡类，处理聊天信息的显示。
- **服务端**
  - `Widget`：服务端主类，处理客户端请求，与数据库交互。

- `Mythread`：线程类，用于实现多线程。

---

4. 程序实现

4.1注册账号的实现

4.1.1客户端发送注册申请

```
void Register::on_RegisterIn_clicked()
{
    QString name =ui->Name->text();
    QString firstP =ui->PassWord->text();
    QString secondP =ui->ConfirmWord->text();
    if(firstP!=secondP){
        QMessageBox::warning(this,"警告","两次输入不同");
    }else{
        QJsonObject jsonObject;
        jsonObject["nickname"]=name;
        jsonObject["password"]=firstP;
        jsonObject["type"]="register";//数据类型
        QJsonDocument document(jsonObject);
        QByteArray byte =document.toJson();
        socket->write(byte);
    }
}
```

4.1.2服务端的处理

```
else if(type=="register"){

    QString account = QString::number(QRandomGenerator::global()-
>bounded(100000000000011)+10000000);
    while(isAccountExists(account)){
        account = QString::number(QRandomGenerator::global()-
>bounded(100000000000011)+10000000);
    }
    QString nickname =jsonObject.value("nickname").toString();
    QString password = jsonObject.value("password").toString();
    password=crypassword(password);

    QSqlQuery query;
    query.prepare("INSERT INTO users (account, password, nickname) VALUES
(:account, :password, :nickname)");
    query.bindValue(":account", account);
    query.bindValue(":password", password);
    query.bindValue(":nickname", nickname);

    response["type"] = "register_response";
    if (query.exec()) {
        //注册成功
        QString successMessage = QString("Registration successful: Account: %1,
Nickname: %2").arg(account, nickname);
        ui->messageList->addItem(successMessage);
```

```
        // 构造成功响应消息
        response["status"] = "success";
        response["message"] = "Registration successful";
        response["account"] = account; // 返回生成的账号

        //将自己与自己建立好友关系
        insertFrinend(account,account,1);

    }
```

4.1.2.1isAccountExists函数的实现

```
bool Widget::isAccountExists(QString account){
    QSqlQuery query;
    query.prepare("SELECT 1 FROM users WHERE account = :account"); // 查询常量值1
    query.bindValue(":account", account);
    if(query.exec()){
        return query.next(); // 如果找到了匹配的记录，返回 1，否则返回 空 即为假
    } else {
        qDebug() << "Query failed:" << query.lastError().text(); // 记录错误信息
        return false; // 查询失败，返回 false
    }
}
```

4.2登录账号的实现

4.2.1客户端发送登录申请

```
void MainWindow::on_LogButton_clicked() // 登录
{
    QString account = ui->accountEdit->text();
    QString password = ui->passwordEdit->text();

    // 创建一个 JSON 对象来存储账号和密码
    QJsonObject jsonObject;
    jsonObject["account"] = account;
    jsonObject["password"] = password;
    jsonObject["type"]="login";//数据类型
    // 将 JSON 对象转换为 QByteArray
    QJsonDocument document(jsonObject);
    QByteArray data = document.toJson();
    socket->write(data);
}
```

4.2.2服务端处理

```
if(type=="login"){
    QString account = jsonObject.value("account").toString();
    QString password = jsonObject.value("password").toString();
    password=crypassword(password);
    QSqlQuery query;
    query.prepare("SELECT * FROM users WHERE account = :account");
    query.bindValue(":account", account);
    if(query.exec()){
```

```
        response["type"] = "login_response";//消息类型
        if(query.next()){
            int storeId = query.value(0).toInt();
            QString storeAccount = query.value(1).toString();
            QString storePassword = query.value(2).toString();
            QString snickName = query.value(3).toString();

            QJsonObject jsonObject;//账号信息
            jsonObject["id"] = storeId;
            jsonObject["account"] = storeAccount;
            jsonObject["nickname"] = snickName;
            if(storePassword==password){
                //密码正确
                response["status"] = "success";
                response["message"] = "Login successful";
                //将账号信息发送到客户端
                response["account"]=storeAccount;
                response["nickname"]=snickName;
                response["id"]=storeId;
                currentThread->setAccount(storeAccount);
                threadInfo[storeAccount]=currentThread;//将登录成功的信息加入线程信息
                userSocketMap[storeAccount] = socket;
            }else{
                //密码错误
                response["status"] = "failure";
                response["message"] = "Incorrect password";
            }
        }else{
            //无账号
            response["status"] = "failure";
            response["message"] = "Account not found";
        }
    }

}
```

4.3添加好友的实现

4.3.1客户端向服务端发送搜索申请

```
void AddFriend::on_searchlindEdit()
{
    QString message=ui->searchlindEdit->text();
    QJsonObject object;
    object["message"]=message;//信息
    object["sendaccount"]=jsonOb.value("account").toString();
    object["sendnickname"]=jsonOb.value("nickname").toString();
    object["id"]=jsonOb.value("id").toString();
    object["type"]="addFriend_search";//添加好友搜索
    QByteArray byte =QJsonDocument(object).toJson();
    socket->write(byte);
}
```

4.3.2服务端处理

```
else if(type=="addFriend_search"){
    QString message=jsonObject.value("message").toString();
    //在数据库中搜索
    //提供两种搜索方式，nickname，account
    QSqlQuery query_n;
    query_n.prepare("SELECT account,nickname FROM users WHERE nickname =
:message");
    query_n.bindValue(":message", message);
    QSqlQuery query_a;
    query_a.prepare("SELECT account,nickname FROM users WHERE account =
:message");
    query_a.bindValue(":message", message);
    QJsonArray usersArray;
    if(query_n.exec()){//通过nickanme搜索
        while(query_n.next()){
            // 获取每一行的数据
            QString account = query_n.value("account").toString();
            QString nickname = query_n.value("nickname").toString();

            // 创建一个 JSON 对象表示一个用户
            QJsonObject userObject;
            userObject["account"] = account;
            userObject["nickname"] = nickname;

            // 将用户对象添加到数组中
            usersArray.append(userObject);
        }
    }
    if (query_a.exec()) {
        while (query_a.next()) {
            // 获取每一行的数据
            QString account = query_a.value("account").toString();
            QString nickname = query_a.value("nickname").toString();
            // 创建一个 JSON 对象表示一个用户
            QJsonObject userObject;
            userObject["account"] = account;
            userObject["nickname"] = nickname;

            // 将用户对象添加到数组中
            usersArray.append(userObject);
        }
    }
    response["type"]="addFriend_searcher_reponse";
    response["users"]=usersArray;
}
```

4.3.3客户端显示搜索结果

```
if(type=="addFriend_searcher_reponse"){
    model->clear();
    if(jsonobject.contains("users")&&jsonobject["users"].isArray()){
        QJsonArray users=jsonobject["users"].toArray();
```

```
        for(QJsonArray::const_iterator
it=users.constBegin();it!=users.constEnd();++it){
            const auto &item =*it;
            if (item.isObject()) {
                QJsonObject userObject = item.toObject();
                QString account = userObject["account"].toString();
                QString nickname = userObject["nickname"].toString();
                QString message=QString("账号：%1，昵称：
%2").arg(account,nickname);
                QStandardItem *item =new QStandardItem(message);
                model->appendRow(item);
            }
        }
    }
    ui->resultlistView->setModel(model);
}
```

4.3.4双击相应条目显示出相应信息

```
connect(ui-
>resultlistView,&QListView::clicked,this,&AddFriend::viewClickedSlots);
void AddFriend::viewClickedSlots(const QModelIndex& index){
    QString message=index.data().toString();
    delete info;
    info=new Information(socket,jsonOb,message,this);
    connect(this,&AddFriend::sendToINF,info,&Information::fromAD);
    info->show();
}
```

4.3.5从message中提取出账号和昵称

```
static QRegularExpression re("账号：(\\d+)，昵称：([^，]+)");//使用静态
QRegularExpression对象替代临时对象可以提高性能和效率
QRegularExpressionMatch match = re.match(message);
viewer=v;
if (match.hasMatch()) {
    account = match.captured(1); // 获取第一个捕获组的内容（账号）
    nickname = match.captured(2); // 获取第二个捕获组的内容（昵称）
} else {
    qDebug() << "未找到匹配的内容";
}
```

4.3.6向服务端验证搜索用户与用户的关系

```
QString v_account=viewer["account"].toString();//查看者的账号
//向服务端发送检查好友信息
QJsonObject jsonObject;
jsonObject["type"]="checkFriend";
jsonObject["v_account"]=v_account;
jsonObject["account"]=account;
ui->nicknamelabel->setText(nickname);
ui->accountlabel->setText(account);
QByteArray byte=QJsonDocument(jsonObject).toJson();
socket->write(byte);
```

4.3.7服务端处理

```
if(type=="checkFriend"){
    qDebug()<<"checkFriend";
    response["type"]="checkFriend_response";
    QString v_account=jsonObject["v_account"].toString();
    QString account=jsonObject["account"].toString();
    QSqlQuery query;
    query.prepare("select count(*) as is_friend from friendships where "
                  "(user1_account= :v_account and user2_account= :account and
status=1) or "
                  "(user2_account= :v_account and user1_account= :account and
status=1);");
    query.bindValue(":v_account",v_account);
    query.bindValue(":account",account);
    if(query.exec()){
        if(query.next()){
            bool is_friend=query.value(0).toBool();
            if(is_friend){
                response["result"]="is friend";
            }else{
                response["result"]="is not friend";
            }
        }
    }else{
        qDebug()<<query.lastQuery();
    }
}
```

4.3.8根据服务端返回的结果设置按钮的显示

```
if(type=="checkFriend_response"){
    QString is_friend=jsonobject["result"].toString();
    if(is_friend=="is not friend"){
        ui->sendpushButton->setText("发送好友申请");
    }
}
```

4.3.9客户端发送添加好友的申请

```
if(ui->sendpushButton->text()=="发送好友申请"){
    QString v_account=viewer["account"].toString();//查看者的账号
    //向服务端发送好友申请
    QJsonObject jsonObject;
    jsonObject["type"]="friend_request";
    jsonObject["v_account"]=v_account;
    jsonObject["account"]=account;
    ui->nicknamelabel->setText(nickname);
    ui->accountlabel->setText(account);
    QByteArray byte=QJsonDocument(jsonObject).toJson();

    socket->write(byte);
}
```

4.3.10服务端处理申请

```
if(type=="friend_request"){
    response["type"]="friend_request_response";
    QString v_account=jsonObject["v_account"].toString();
    QString account=jsonObject["account"].toString();
    if(insertFrinend(v_account,account,0)){
        response["result"]="insert_successed";
    }else{
        response["result"]="insert_not_successed";
    }
}
```

4.3.11insertFrinend的实现

```
bool Widget::insertFrinend(QString v_account, QString account, int status)
{
    QSqlQuery query;
    query.prepare("insert into friendships(user1_account,user2_account,status)
values(:v_account,:account,:status);");
    query.bindValue(":v_account",v_account);
    query.bindValue(":account",account);
    query.bindValue(":status",status);
    if(query.exec()){
        qDebug()<<"插入成功";
        return true;
    }else{
        qDebug()<<query.lastError();
    }
    return false;
}
```

4.3.12客户端回应服务端

```
if(type=="friend_request_response"){
    QMessageBox::information(this,"提示","已发送申请");
}
```

4.4查看所有好友和好友申请

4.4.1客户端

```
void FriendManagement::update(){
    QJsonObject jsonobect;
    jsonobect["type"]="View_friend_relationships";
    jsonobect["account"]=account;
    QByteArray byte=QJsonDocument(jsonobect).toJson();
    socket->write(byte);
}
```

4.4.2服务端处理

```
if(type=="View_friend_relationships"){
    QString account =jsonObject["account"].toString();
    findallfriend(account,response);
}
```

4.4.3findallfriend的实现

```
void Widget::findallfriend(const QString &account,QJsonObject &response)
{
    QSqlQuery query;
    QJsonArray allfriend;
    //user1_account为发送好友申请的账号
    //所以第一个搜索中加上status=1,进行限制,以此得到好友申请和好友关系
    query.prepare("select user2_account,status from friendships where
user1_account=:account and status =1 "
                  "union select user1_account,status from friendships where
user2_account=:account;");
    query.bindValue(":account",account);
    if(query.exec()){
        while(query.next()){
            QString friend_account=query.value(0).toString();
            QSqlQuery qu;
            QString friend_nickname;
            qu.prepare("select nickname from users where account = :account");
            qu.bindValue(":account",friend_account);
            if(qu.exec()){
                if(qu.next()){
                    friend_nickname=qu.value(0).toString();
                }
            }
            int status=query.value(1).toInt();
            QJsonObject one_friend;
            one_friend["friend_account"]=friend_account;
            one_friend["friend_nickname"]=friend_nickname;
            one_friend["status"]=status;
            allfriend.append(one_friend);
        }
        qDebug()<<allfriend;
    }
    response["type"]="View_friend_relationships_response";
    response["allfriend"]=allfriend;
}
```

4.4.5客户端显示所有好友和好友申请

```
if (type=="View_friend_relationships_response"){
    model->clear();
    if(jsonobject.contains("allfriend")&&jsonobject["allfriend"].isArray()){
        QJsonArray allfriend=jsonobject["allfriend"].toArray();
        qDebug()<<allfriend;
        for(QJsonArray::const_iterator
it=allfriend.constBegin();it!=allfriend.constEnd();++it){
```

```
            const auto &item =*it;
            if (item.isObject()) {
                QJsonObject cu_friend = item.toObject();
                qDebug()<<cu_friend;
                emit createItem(cu_friend);
            }
        }
    }

}
connect(this,&FriendManagement::createItem,this,&FriendManagement::addFriendItem)
;
```

4.4.6addFriendItem的实现

```
void FriendManagement::addFriendItem(const QJsonObject &js)
{
    qDebug()<<__func__<<js;
    // 创建自定义小部件
    QJsonObject newjs=js;
    newjs["account"]=account;
    newjs["type"]="FRIENDMANAGEMENT";//用于判断是否显示出删除好友按钮
    FriendItemWidget* widget = new FriendItemWidget(newjs,this);
    widget->show();
    // 连接按钮点击信号
    connect(widget,
&FriendItemWidget::deleteButtonClicked,this,&FriendManagement::onDeleteButtonClic
ked);
    connect(widget, &FriendItemWidget::agreeButtonClicked, this,
&FriendManagement::onAgreeButtonClicked);
    connect(widget, &FriendItemWidget::refuseButtonClicked, this,
&FriendManagement::onRefuseButtonClicked);
    connect(widget, &FriendItemWidget::sendMessageButtonClicked, this,
&FriendManagement::onSendMessageButtonClicked);


    // 创建一个不可见的项，用于占据空间
    QStandardItem* item = new QStandardItem();
    item->setEditable(false);
    QSize sizeHint = item->sizeHint();
    sizeHint.setHeight(40); // 设置高度为40
    item->setSizeHint(sizeHint);
    // 将小部件添加到视图中
    int row = model->rowCount();
    qDebug()<<"row:"<<row;
    model->insertRow(row, item);
    qDebug()<<model->index(row,0);
    ui->messagelistView->setIndexWidget(model->index(row, 0), widget);
}
```

4.4.7FriendItemWidget类的实现

```
FriendItemWidget::FriendItemWidget(const QJsonObject &js, QWidget *parent):
QWidget(parent) {
    qDebug()<<__func__<<js;
    jsonobject=js;
    QString account=js["friend_account"].toString();
    QString nickname=js["friend_nickname"].toString();
    QString v_account=js["account"].toString();
    QString type=js["type"].toString();
    if(v_account==account){
        nickname="self";
    }
    int status=js["status"].toInt();
    QString message = QString("账号: %1, 昵称: %2").arg(account, nickname);
    messageLabel=new QLabel(message);
    QHBoxLayout* layout = new QHBoxLayout(this);
    qDebug()<<"messageLabel:"<<message;

    // 设置布局的内边距和控件间距
    layout->setContentsMargins(10, 5, 10, 5);
    layout->setSpacing(10);

    // 设置文本标签大小
    messageLabel->setFixedSize(200, 30);
    messageLabel->setSizePolicy(QSizePolicy::Fixed, QSizePolicy::Preferred);
    layout->addWidget(messageLabel);

    // 添加伸展因子，将按钮推向右侧
    layout->addStretch();

    // 创建按钮
    agreeButton = new QPushButton("同意", this);
    refuseButton = new QPushButton("拒绝", this);
    sendMessageButton = new QPushButton("发送信息", this);
    deleteButton=new QPushButton("删除好友",this);

    // 设置按钮大小和大小策略
    agreeButton->setFixedSize(80, 30);
    refuseButton->setFixedSize(80, 30);
    //是否显示删除好友按钮
    if(type=="FRIENDMANAGEMENT"){
        sendMessageButton->setFixedSize(80, 30);
        deleteButton->setFixedSize(80,30);
        sendMessageButton->setVisible(status == 1);
        deleteButton->setVisible(status == 1);
    }else{
        sendMessageButton->setVisible(status == 1);
        deleteButton->setVisible(0);
    }


    agreeButton->setSizePolicy(QSizePolicy::Fixed, QSizePolicy::Preferred);
    refuseButton->setSizePolicy(QSizePolicy::Fixed, QSizePolicy::Preferred);
    sendMessageButton->setSizePolicy(QSizePolicy::Fixed, QSizePolicy::Preferred);
```

```
    deleteButton->setSizePolicy(QSizePolicy::Fixed, QSizePolicy::Preferred);


    // 根据状态设置按钮的可见性
    agreeButton->setVisible(status == 0);
    refuseButton->setVisible(status == 0);


    // 添加按钮到布局
    layout->addWidget(agreeButton);
    layout->addWidget(refuseButton);
    layout->addWidget(sendMessageButton);
    layout->addWidget(deleteButton);

    // 设置布局
    this->setLayout(layout);

    // 连接按钮的点击信号
    connect(agreeButton, &QPushButton::clicked, this, [this]() {
        emit agreeButtonClicked(jsonobject);
    });

    connect(refuseButton, &QPushButton::clicked, this, [this]() {
        emit refuseButtonClicked(jsonobject);
    });

    connect(sendMessageButton, &QPushButton::clicked, this, [this]() {
        emit sendMessageButtonClicked(jsonobject);
    });

    connect(deleteButton, &QPushButton::clicked, this, [this]() {
        qDebug()<<"deleteButton点击"<<jsonobject;
        emit deleteButtonClicked(jsonobject);
    });

}
```

4.4.8同意，拒绝/删除好友的实现

```
if(type=="Agree_the_friend"){
    QString account=jsonObject["account"].toString();
    QString friend_account=jsonObject["friend_account"].toString();
    qDebug()<<"account:"<<account<<"friend_account:"<<friend_account;
    QSqlQuery query;
    query.prepare("update friendships set status=1 where
user1_account=:friend_account and user2_account=:account;;");
    query.bindValue(":account",account);
    query.bindValue(":friend_account",friend_account);
    if(query.exec()){
        qDebug()<<"Agree_the_friend"<<query.lastQuery();
        findallfriend(account,response);
    }else{
        qDebug()<<query.lastError();
    }
    qDebug()<<"Agree_the_friend"<<query.lastQuery();
```

```
}else if (type=="Refuse_the_friend"||type=="Delete_the_friend"){
    QString account=jsonObject["account"].toString();
    QString friend_account=jsonObject["friend_account"].toString();
    QSqlQuery query;
    query.prepare("delete from friendships where user1_account=:friend_account
and user2_account=:account;");
    query.bindValue(":account",account);
    query.bindValue(":friend_account",friend_account);
    if(query.exec()){
        findallfriend(account,response);
    }else{
        qDebug()<<query.lastError();
    }
}
```

4.5发送信息

4.5.1显示聊天窗口

```
void FriendManagement::onSendMessageButtonClicked(const QJsonObject &js)
{
    qDebug()<<__func__<<js;
    QString friendAccount = js["friend_account"].toString();
    QString friendName = js["friend_nickname"].toString();
    if(account==friendAccount){
        friendName="self";
    }
    ChatWindow *chat = new ChatWindow(socket, account, friendAccount,friendName,
nullptr);
    connect(this,&FriendManagement::sendToCHAT,chat,&ChatWindow::onReadyRead);
    chat->show();
}
```

4.5.2聊天窗口类的实现

4.5.2.1客户端发送信息

```
void ChatWindow::on_SendButton_clicked()
{
    QString text = ui->EditArea->toPlainText();
    if (text.isEmpty()) return;

    QJsonObject object;
    object["type"] = "chat_message";
    object["from"] = this->selfAccount;
    object["to"] = this->friendAccount;

    object["content"] = text;
    object["time"] = QDateTime::currentDateTime().toString("yyyy-MM-dd
HH:mm:ss");
    QByteArray data = QJsonDocument(object).toJson();
    socket->write(data);

    QString messageLine = "[" + object["time"].toString() + "] 我: " + text;
    addMessageToList(messageLine, true);
```

```
    ui->EditArea->clear();
}
```

4.5.2.2服务端处理

```
if(type == "chat_message"){
    QString from = jsonObject["from"].toString();
    QString to = jsonObject["to"].toString();
    qDebug() << "to_user:::" << from << to;
    QString content = jsonObject["content"].toString();
    QString time = jsonObject["time"].toString(); // 可以加时间字段

    QJsonObject chatData;
    chatData["type"] = "chat_message";
    chatData["from"] = from;
    chatData["to"] = to;
    chatData["content"] = content;
    chatData["time"] = time;

    QJsonDocument doc(chatData);
    QByteArray data = doc.toJson();
    QSqlQuery saveChat;
    saveChat.prepare("INSERT INTO messages (sender, receiver, content, time)
VALUES (:from, :to, :content, :time)");
    saveChat.bindValue(":from", from);
    saveChat.bindValue(":to", to);
    saveChat.bindValue(":content", content);
    saveChat.bindValue(":time", time);
    if (!saveChat.exec()) {
        qDebug() << "Failed to save message:" << saveChat.lastError().text();
    }

    if(userSocketMap.contains(to)) {
        userSocketMap[to]->write(data); // 转发给对方
        response["status"] = "sent";
    } else {
        response["status"] = "offline"; // 对方不在线
    }
}
```

4.5.2.3客户端显示信息

```
if (type == "chat_message") {
    receiveMessage(obj);
}
```

4.5.3receiveMessage的实现

```
void ChatWindow::receiveMessage(const QJsonObject &js) {
    qDebug()<<__func__<<js;
    if (js["from"].toString() == friendAccount) {
        QString time = js["time"].toString();
        QString content = js["content"].toString();
        QString messageLine = "[" + time + "] " + js["name"].toString() + ": " +
content;
        addMessageToList(messageLine, false);
    }
}
```

4.5.4addMessageToList的实现

```
void ChatWindow::addMessageToList(const QString &text, bool isOwnMessage)
{
    QListWidgetItem *item = new QListWidgetItem(ui->MessageListWidget);
    MessageBubbleWidget *bubble = new MessageBubbleWidget(text, isOwnMessage);
    item->setSizeHint(bubble->sizeHint());
    ui->MessageListWidget->addItem(item);
    ui->MessageListWidget->setItemWidget(item, bubble);
    ui->MessageListWidget->scrollToBottom();
}
```

4.5.5MessageBubbleWidget的实现

```
#include "MessageBubbleWidget.h"

MessageBubbleWidget::MessageBubbleWidget(const QString &text, bool isOwnMessage,
QWidget *parent)
    : QWidget(parent)
{
    setupUi(text, isOwnMessage);
}

void MessageBubbleWidget::setupUi(const QString &text, bool isOwnMessage)
{
    // 创建主布局
    auto *mainLayout = new QHBoxLayout(this);
    mainLayout->setContentsMargins(10, 5, 10, 5);
    mainLayout->setSpacing(10);

    // 创建消息标签
    QLabel *messageLabel = new QLabel(text);
    // 设置文本样式
    QFont font;
    font.setPointSize(10);              // 字号
    font.setFamily("Microsoft YaHei");  // 字体（可换成任意常见字体）
    messageLabel->setFont(font);
    messageLabel->setStyleSheet("color: #1b1b1b;"); // 字体颜色（也可以是灰色等）
    messageLabel->setWordWrap(true);
    messageLabel->setTextInteractionFlags(Qt::TextSelectableByMouse);
    messageLabel->setContentsMargins(10, 6, 10, 6);
```

```
    messageLabel->setMaximumWidth(350);
    messageLabel->setStyleSheet("font-size: 14px;");

    // 气泡背景控件
    QWidget *bubble = new QWidget;
    QVBoxLayout *bubbleLayout = new QVBoxLayout(bubble);
    bubbleLayout->addWidget(messageLabel);
    bubbleLayout->setContentsMargins(0, 0, 0, 0);

    bubble->setLayout(bubbleLayout);
    bubble->setStyleSheet(isOwnMessage
                              ? "background-color: #d4f4dd; border-radius: 12px;"
 //  自己的消息为浅绿色
                              : "background-color: #fddde6; border-radius:
12px;"); //  对方消息为浅粉色
    bubble->setFixedWidth(320);

    // 气泡左右对齐
    if (isOwnMessage) {
        mainLayout->addStretch();
        mainLayout->addWidget(bubble);
    } else {
        mainLayout->addWidget(bubble);
        mainLayout->addStretch();
    }

    setLayout(mainLayout);
    setSizePolicy(QSizePolicy::Expanding, QSizePolicy::Minimum);
}
```

4.6得到历史消息

4.6.1客户端发送申请

```
void ChatWindow::getHistory()
{
    QJsonObject json;
    json["type"] = "get_history";
    json["from"] = this->selfAccount;
    json["to"] = this->friendAccount;
    QByteArray data = QJsonDocument(json).toJson();
    socket->write(data);
}
```

4.6.2服务端处理

```
if(type == "get_history") {
    QString a1 = jsonObject["from"].toString();
    QString a2 = jsonObject["to"].toString();

    QSqlQuery query;
    query.prepare("SELECT sender, receiver, content, time FROM messages WHERE "
                  "(sender = :a1 AND receiver = :a2) OR (sender = :a2 AND
receiver = :a1) "
```

```
                  "ORDER BY time ASC");
    query.bindValue(":a1", a1);
    query.bindValue(":a2", a2);

    QMap<QString, QString> nicknameMap;
    QSqlQuery nicknameQuery;
    nicknameQuery.prepare("SELECT account, nickname FROM users WHERE account =
:a1 OR account = :a2");
    nicknameQuery.bindValue(":a1", a1);
    nicknameQuery.bindValue(":a2", a2);
    if (nicknameQuery.exec()) {
        while (nicknameQuery.next()) {
            QString account = nicknameQuery.value("account").toString();
            QString nickname = nicknameQuery.value("nickname").toString();
            nicknameMap[account] = nickname;
        }
    }

    QJsonArray msgArray;
    if (query.exec()) {
        while (query.next()) {
            QString sender = query.value("sender").toString();
            QJsonObject msg;
            msg["name"] = nicknameMap.value(sender, sender);
            msg["from"] = sender;
            msg["to"] = query.value("receiver").toString();
            msg["content"] = query.value("content").toString();
            msg["time"] = query.value("time").toString();
            msgArray.append(msg);
        }
    }

    response["type"] = "get_history_response";
    response["messages"] = msgArray;


}
```

4.6.3客户端处理

```
if (type == "get_history_response") {
    QJsonArray history = obj["messages"].toArray();
    for (const QJsonValue &val : history) {
        QJsonObject msg = val.toObject();
        QString sender = msg["from"].toString();
        QString content = msg["content"].toString();
        QString time = msg["time"].toString();
        QString name = msg["name"].toString();
        QString displayMsg;

        QString messageLine = "[" + time + "] " + name + ": " + content;
        bool isOwn = (sender == selfAccount);
        addMessageToList(messageLine, isOwn);
    }
```

```
}
```

4.7获取并储存聊天记录

4.7.1获取聊天记录文件路径

```
QString ChatWindow::getHistoryFilePath() const {
    QString filename = QString("%1_%2.json").arg(selfAccount).arg(friendAccount);

    // 获取当前可执行文件所在目录
    QString currentPath = QDir::currentPath();
    QString dirPath = currentPath + "/chat_history";
    QDir dir;

    // 创建 chat_history 目录（如果不存在）
    if (!dir.exists(dirPath)) {
        qDebug() << "正在创建目录：" << dirPath;
        if (!dir.mkpath(dirPath)) {
            qDebug() << "创建目录失败！";
        }
    }

    QString fullPath = dirPath + "/" + filename;
    qDebug() << "create" << fullPath;
    return fullPath;
}
```

4.7.2储存聊天记录到本地

```
void ChatWindow::saveMessageToLocal(const QJsonObject &msg) {
    QString filePath = getHistoryFilePath();
    qDebug() << "save" << filePath;

    QFile file(filePath);
    QJsonArray historyArray;

    // 如果文件存在，读取已有内容
    if (file.exists()) {
        qDebug() << "✅ 文件已存在，尝试读取历史记录...";
        if (file.open(QIODevice::ReadOnly)) {
            QByteArray data = file.readAll();
            QJsonDocument doc = QJsonDocument::fromJson(data);
            if (doc.isArray()) {
                historyArray = doc.array();
            }
            file.close();
        }
    }

    // 过滤重复消息
    QString messageId = msg["id"].toString();
    if (messageId.isEmpty()) {
        QString content = msg["content"].toString();
        QString time = msg["time"].toString();
        messageId = QString("%1_%2").arg(time).arg(content);
```

```
    }

    bool exists = false;
    for (const QJsonValue &val : historyArray) {
        QJsonObject existingMsg = val.toObject();
        QString existingId = existingMsg["id"].toString();
        if (existingId.isEmpty()) {
            QString existingContent = existingMsg["content"].toString();
            QString existingTime = existingMsg["time"].toString();
            existingId = QString("%1_%2").arg(existingTime).arg(existingContent);
        }
        if (existingId == messageId) {
            exists = true;
            break;
        }
    }

    if (!exists) {
        historyArray.append(msg);

        if (file.open(QIODevice::WriteOnly | QIODevice::Truncate)) {
            QJsonDocument doc(historyArray);
            qint64 bytesWritten = file.write(doc.toJson());
            qDebug() << "✅ 写入文件成功，共写入" << bytesWritten << "字节";
            file.close();
        } else {
            qDebug() << "❌ 无法打开文件进行写入：" << file.errorString();
        }
    } else {
        qDebug() << "⚠️ 消息已存在，未写入文件：" << messageId;
    }

}
```

4.7服务端makfeile文件

```
#############################################################################
# Makefile for building: CmdServer
# Generated by qmake (3.1) (Qt 5.15.13)
# Project:  CmdServer.pro
# Template: app
# Command: /usr/lib/qt5/bin/qmake -o Makefile CmdServer.pro
#############################################################################

MAKEFILE      = Makefile

EQ            = =

####### Compiler, tools and options

CC            = gcc
CXX           = g++
DEFINES       = -DQT_NO_DEBUG -DQT_SQL_LIB -DQT_NETWORK_LIB -DQT_CORE_LIB
```

```
CFLAGS        = -pipe -O2 -Wall -Wextra -D_REENTRANT -fPIC $(DEFINES)
CXXFLAGS      = -pipe -O2 -std=gnu++1z -Wall -Wextra -D_REENTRANT -fPIC
$(DEFINES)
INCPATH       = -I. -I/usr/include/x86_64-linux-gnu/qt5 -I/usr/include/x86_64-
linux-gnu/qt5/QtSql -I/usr/include/x86_64-linux-gnu/qt5/QtNetwork -
I/usr/include/x86_64-linux-gnu/qt5/QtCore -I. -I/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/linux-g++
QMAKE         = /usr/lib/qt5/bin/qmake
DEL_FILE      = rm -f
CHK_DIR_EXISTS= test -d
MKDIR         = mkdir -p
COPY          = cp -f
COPY_FILE     = cp -f
COPY_DIR      = cp -f -R
INSTALL_FILE  = install -m 644 -p
INSTALL_PROGRAM = install -m 755 -p
INSTALL_DIR   = cp -f -R
QINSTALL      = /usr/lib/qt5/bin/qmake -install qinstall
QINSTALL_PROGRAM = /usr/lib/qt5/bin/qmake -install qinstall -exe
DEL_FILE      = rm -f
SYMLINK       = ln -f -s
DEL_DIR       = rmdir
MOVE          = mv -f
TAR           = tar -cf
COMPRESS      = gzip -9f
DISTNAME      = CmdServer1.0.0
DISTDIR = /home/server/.tmp/CmdServer1.0.0
LINK          = g++
LFLAGS        = -Wl,-O1
LIBS          = $(SUBLIBS) /usr/lib/x86_64-linux-gnu/libQt5Sql.so
/usr/lib/x86_64-linux-gnu/libQt5Network.so /usr/lib/x86_64-linux-
gnu/libQt5Core.so -lpthread
AR            = ar cqs
RANLIB        =
SED           = sed
STRIP         = strip

####### Output directory

OBJECTS_DIR   = ./

####### Files

SOURCES       = main.cpp \
        mythread.cpp \
        server.cpp moc_mythread.cpp \
        moc_server.cpp
OBJECTS       = main.o \
        mythread.o \
        server.o \
        moc_mythread.o \
        moc_server.o
DIST          = /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/spec_pre.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/unix.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/linux.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/sanitize.conf \
```

```
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/gcc-base.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/gcc-base-unix.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/g++-base.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/g++-unix.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/qconfig.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_accessibility_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_bootstrap_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_concurrent.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_concurrent_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_core.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_core_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_dbus.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_dbus_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_devicediscovery_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_edid_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_egl_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eglfs_kms_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eglfsdeviceintegration_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eventdispatcher_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_fb_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_fontdatabase_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_glx_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_gui.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_gui_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_input_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_kms_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_linuxaccessibility_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_network.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_network_private.pri
\
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_opengl.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_opengl_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_openglextensions.pri
\
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_openglextensions_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_platformcompositor_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_printsupport.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_printsupport_private.pri \
```

```
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_service_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_sql.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_sql_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_testlib.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_testlib_private.pri
\
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_theme_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_vulkan_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_widgets.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_widgets_private.pri
\
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_xcb_qpa_lib_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_xkbcommon_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xml.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xml_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt_functions.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt_config.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++/qmake.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/spec_post.prf \
        .qmake.stash \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/exclusive_builds.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/toolchain.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/default_pre.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resolve_config.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/default_post.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/lrelease.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/cmdline.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/warn_on.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resources_functions.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resources.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/moc.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/unix/thread.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qmake_use.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/file_copies.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/testcase_targets.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/exceptions.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/yacc.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/lex.prf \
        CmdServer.pro mythread.h \
        server.h main.cpp \
        mythread.cpp \
        server.cpp
QMAKE_TARGET  = CmdServer
DESTDIR       =
TARGET        = CmdServer


first: all
####### Build rules
```

```
CmdServer:  $(OBJECTS)
    $(LINK) $(LFLAGS) -o $(TARGET) $(OBJECTS) $(OBJCOMP) $(LIBS)

Makefile: CmdServer.pro /usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-
g++/qmake.conf /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/spec_pre.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/unix.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/linux.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/sanitize.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/gcc-base.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/gcc-base-unix.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/g++-base.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/g++-unix.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/qconfig.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_accessibility_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_bootstrap_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_concurrent.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_concurrent_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_core.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_core_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_dbus.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_dbus_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_devicediscovery_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_edid_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_egl_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eglfs_kms_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eglfsdeviceintegration_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eventdispatcher_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_fb_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_fontdatabase_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_glx_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_gui.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_gui_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_input_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_kms_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_linuxaccessibility_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_network.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_network_private.pri
\
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_opengl.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_opengl_private.pri \
```

```
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_openglextensions.pri
\
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_openglextensions_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_platformcompositor_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_printsupport.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_printsupport_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_service_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_sql.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_sql_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_testlib.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_testlib_private.pri
\
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_theme_support_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_vulkan_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_widgets.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_widgets_private.pri
\
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_xcb_qpa_lib_private.pri \
        /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_xkbcommon_support_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xml.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xml_private.pri \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt_functions.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt_config.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++/qmake.conf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/spec_post.prf \
        .qmake.stash \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/exclusive_builds.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/toolchain.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/default_pre.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resolve_config.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/default_post.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/lrelease.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/cmdline.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/warn_on.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resources_functions.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resources.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/moc.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/unix/thread.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qmake_use.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/file_copies.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/testcase_targets.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/exceptions.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/yacc.prf \
        /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/lex.prf \
        CmdServer.pro
    $(QMAKE) -o Makefile CmdServer.pro
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/spec_pre.prf:
```

```
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/unix.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/linux.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/sanitize.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/gcc-base.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/gcc-base-unix.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/g++-base.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/common/g++-unix.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/qconfig.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_accessibility_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_bootstrap_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_concurrent.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_concurrent_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_core.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_core_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_dbus.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_dbus_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_devicediscovery_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_edid_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_egl_support_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eglfs_kms_support_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eglfsdeviceintegration_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_eventdispatcher_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_fb_support_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_fontdatabase_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_glx_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_gui.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_gui_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_input_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_kms_support_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_linuxaccessibility_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_network.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_network_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_opengl.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_opengl_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_openglextensions.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_openglextensions_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_platformcompositor_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_printsupport.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_printsupport_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_service_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_sql.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_sql_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_testlib.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_testlib_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_theme_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_vulkan_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_widgets.pri:
```

```
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_widgets_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xcb_qpa_lib_private.pri:
/usr/lib/x86_64-linux-
gnu/qt5/mkspecs/modules/qt_lib_xkbcommon_support_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xml.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/modules/qt_lib_xml_private.pri:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt_functions.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt_config.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++/qmake.conf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/spec_post.prf:
.qmake.stash:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/exclusive_builds.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/toolchain.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/default_pre.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resolve_config.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/default_post.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/lrelease.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/cmdline.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/warn_on.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qt.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resources_functions.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/resources.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/moc.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/unix/thread.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/qmake_use.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/file_copies.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/testcase_targets.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/exceptions.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/yacc.prf:
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/lex.prf:
CmdServer.pro:
qmake: FORCE
    @$(QMAKE) -o Makefile CmdServer.pro

qmake_all: FORCE


all: Makefile CmdServer

dist: distdir FORCE
    (cd `dirname $(DISTDIR)` && $(TAR) $(DISTNAME).tar $(DISTNAME) && $(COMPRESS)
$(DISTNAME).tar) && $(MOVE) `dirname $(DISTDIR)`/$(DISTNAME).tar.gz . &&
$(DEL_FILE) -r $(DISTDIR)

distdir: FORCE
    @test -d $(DISTDIR) || mkdir -p $(DISTDIR)
    $(COPY_FILE) --parents $(DIST) $(DISTDIR)/
    $(COPY_FILE) --parents /usr/lib/x86_64-linux-
gnu/qt5/mkspecs/features/data/dummy.cpp $(DISTDIR)/
    $(COPY_FILE) --parents mythread.h server.h $(DISTDIR)/
    $(COPY_FILE) --parents main.cpp mythread.cpp server.cpp $(DISTDIR)/


clean: compiler_clean
    -$(DEL_FILE) $(OBJECTS)
    -$(DEL_FILE) *~ core *.core
```

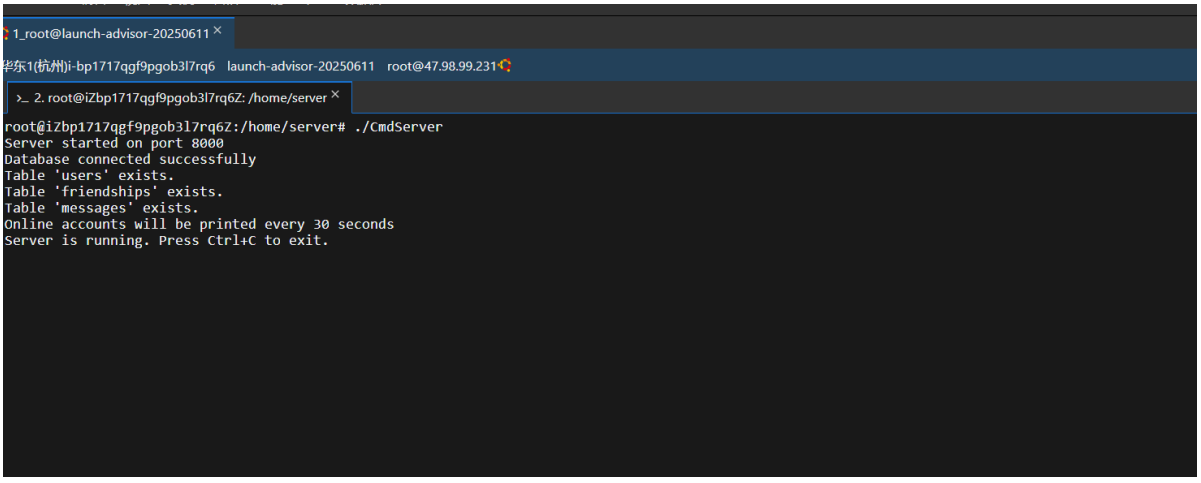
```
distclean: clean
	-$(DEL_FILE) $(TARGET)
	-$(DEL_FILE) .qmake.stash
	-$(DEL_FILE) Makefile


####### Sub-libraries

mocclean: compiler_moc_header_clean compiler_moc_objc_header_clean
compiler_moc_source_clean

mocables: compiler_moc_header_make_all compiler_moc_objc_header_make_all
compiler_moc_source_make_all

check: first

benchmark: first

compiler_lrelease_make_all:
compiler_rcc_make_all:
compiler_rcc_clean:
compiler_moc_predefs_make_all: moc_predefs.h
compiler_moc_predefs_clean:
	-$(DEL_FILE) moc_predefs.h
moc_predefs.h: /usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/data/dummy.cpp
	g++ -pipe -O2 -std=gnu++1z -Wall -Wextra -dM -E -o moc_predefs.h
/usr/lib/x86_64-linux-gnu/qt5/mkspecs/features/data/dummy.cpp

compiler_moc_header_make_all: moc_mythread.cpp moc_server.cpp
compiler_moc_header_clean:
	-$(DEL_FILE) moc_mythread.cpp moc_server.cpp
moc_mythread.cpp: mythread.h \
		moc_predefs.h \
		/usr/lib/qt5/bin/moc
	/usr/lib/qt5/bin/moc $(DEFINES) --include /home/server/moc_predefs.h -
I/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++ -I/home/server -
I/usr/include/x86_64-linux-gnu/qt5 -I/usr/include/x86_64-linux-gnu/qt5/QtSql -
I/usr/include/x86_64-linux-gnu/qt5/QtNetwork -I/usr/include/x86_64-linux-
gnu/qt5/QtCore -I/usr/include/c++/13 -I/usr/include/x86_64-linux-gnu/c++/13 -
I/usr/include/c++/13/backward -I/usr/lib/gcc/x86_64-linux-gnu/13/include -
I/usr/local/include -I/usr/include/x86_64-linux-gnu -I/usr/include mythread.h -o
moc_mythread.cpp

moc_server.cpp: server.h \
		mythread.h \
		moc_predefs.h \
		/usr/lib/qt5/bin/moc
```

```
    /usr/lib/qt5/bin/moc $(DEFINES) --include /home/server/moc_predefs.h -
I/usr/lib/x86_64-linux-gnu/qt5/mkspecs/linux-g++ -I/home/server -
I/usr/include/x86_64-linux-gnu/qt5 -I/usr/include/x86_64-linux-gnu/qt5/QtSql -
I/usr/include/x86_64-linux-gnu/qt5/QtNetwork -I/usr/include/x86_64-linux-
gnu/qt5/QtCore -I/usr/include/c++/13 -I/usr/include/x86_64-linux-gnu/c++/13 -
I/usr/include/c++/13/backward -I/usr/lib/gcc/x86_64-linux-gnu/13/include -
I/usr/local/include -I/usr/include/x86_64-linux-gnu -I/usr/include server.h -o
moc_server.cpp

compiler_moc_objc_header_make_all:
compiler_moc_objc_header_clean:
compiler_moc_source_make_all:
compiler_moc_source_clean:
compiler_yacc_decl_make_all:
compiler_yacc_decl_clean:
compiler_yacc_impl_make_all:
compiler_yacc_impl_clean:
compiler_lex_make_all:
compiler_lex_clean:
compiler_clean: compiler_moc_predefs_clean compiler_moc_header_clean

####### Compile

main.o: main.cpp server.h \
        mythread.h
    $(CXX) -c $(CXXFLAGS) $(INCPATH) -o main.o main.cpp

mythread.o: mythread.cpp mythread.h
    $(CXX) -c $(CXXFLAGS) $(INCPATH) -o mythread.o mythread.cpp

server.o: server.cpp server.h \
        mythread.h
    $(CXX) -c $(CXXFLAGS) $(INCPATH) -o server.o server.cpp

moc_mythread.o: moc_mythread.cpp
    $(CXX) -c $(CXXFLAGS) $(INCPATH) -o moc_mythread.o moc_mythread.cpp

moc_server.o: moc_server.cpp
    $(CXX) -c $(CXXFLAGS) $(INCPATH) -o moc_server.o moc_server.cpp

####### Install

install_target: first FORCE
    @test -d $(INSTALL_ROOT)/opt/CmdServer/bin || mkdir -p
$(INSTALL_ROOT)/opt/CmdServer/bin
    $(QINSTALL_PROGRAM) $(QMAKE_TARGET)
$(INSTALL_ROOT)/opt/CmdServer/bin/$(QMAKE_TARGET)
    -$(STRIP) $(INSTALL_ROOT)/opt/CmdServer/bin/$(QMAKE_TARGET)

uninstall_target: FORCE
    -$(DEL_FILE) $(INSTALL_ROOT)/opt/CmdServer/bin/$(QMAKE_TARGET)
    -$(DEL_DIR) $(INSTALL_ROOT)/opt/CmdServer/bin/


install: install_target  FORCE
```

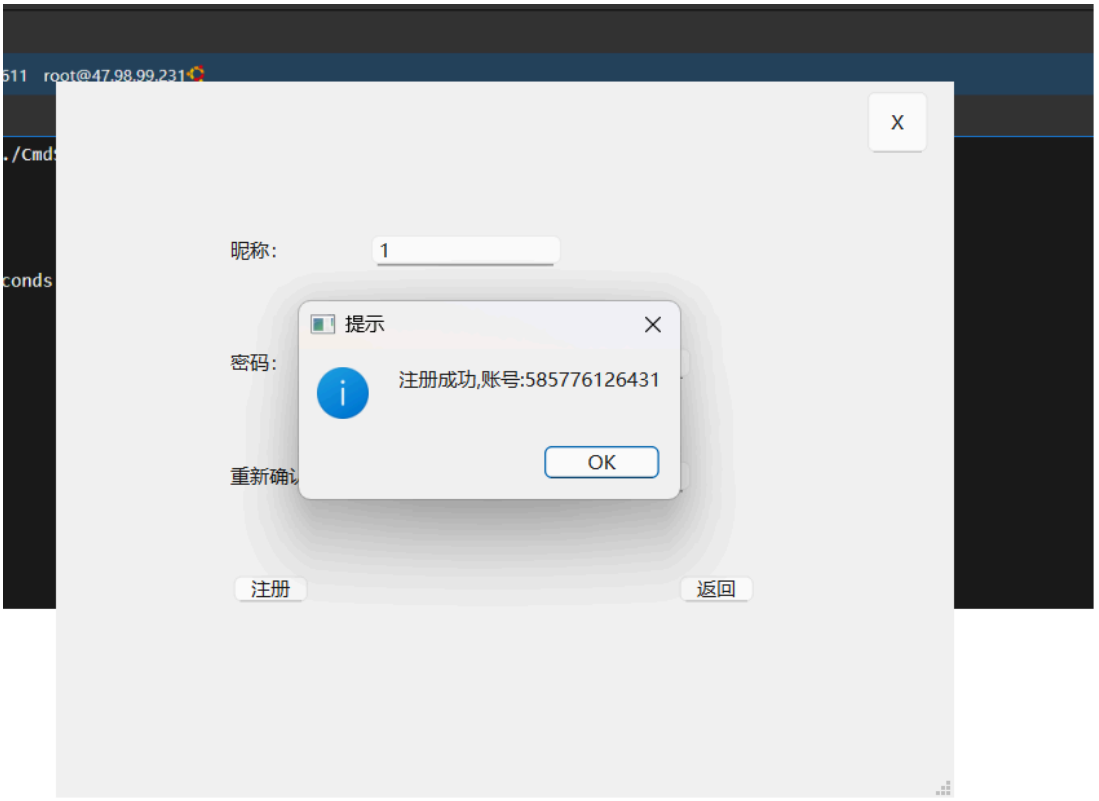
```
uninstall: uninstall_target  FORCE

FORCE:
```

5.程序测试

5.1云服务器

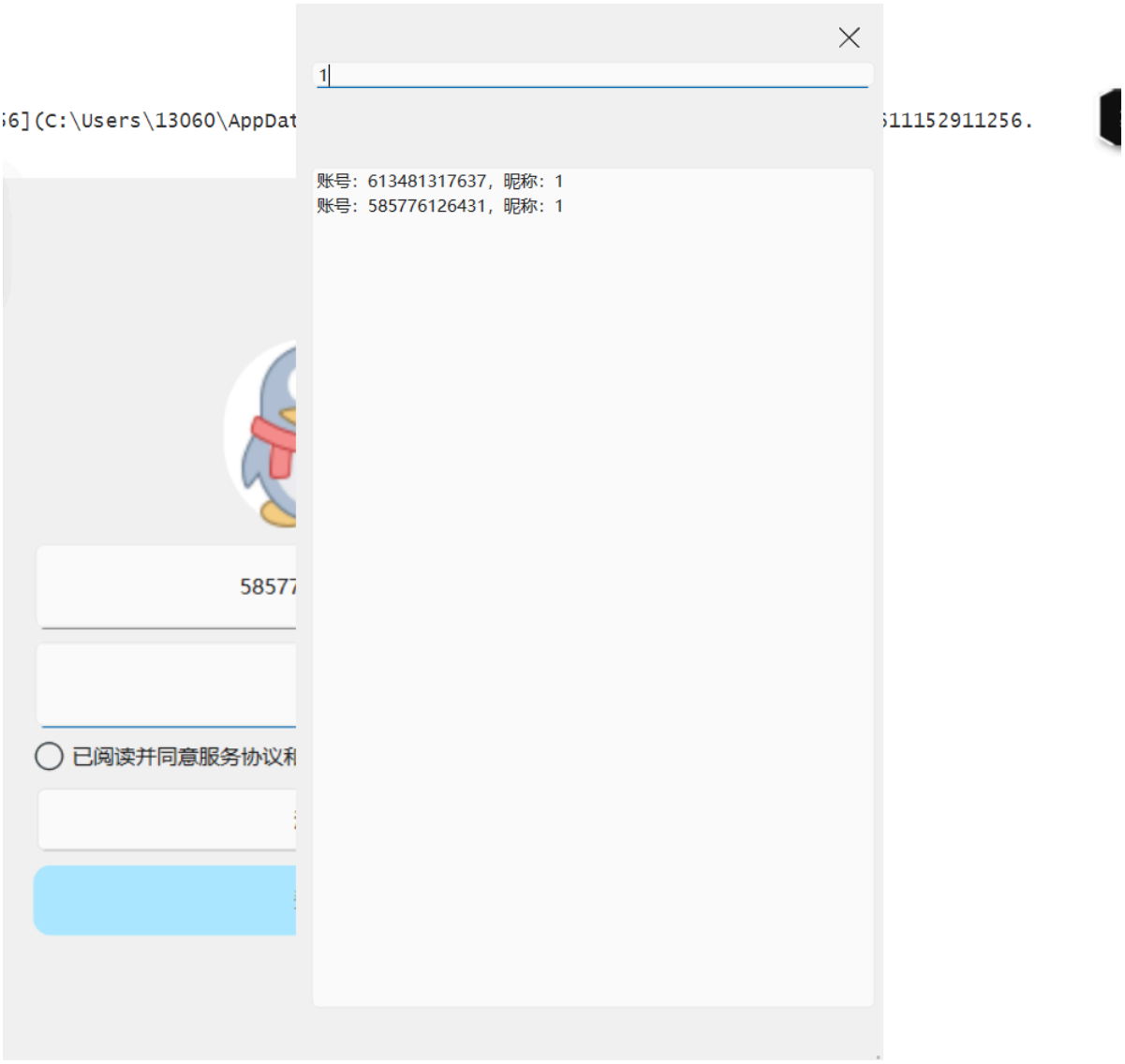


5.2注册与登录

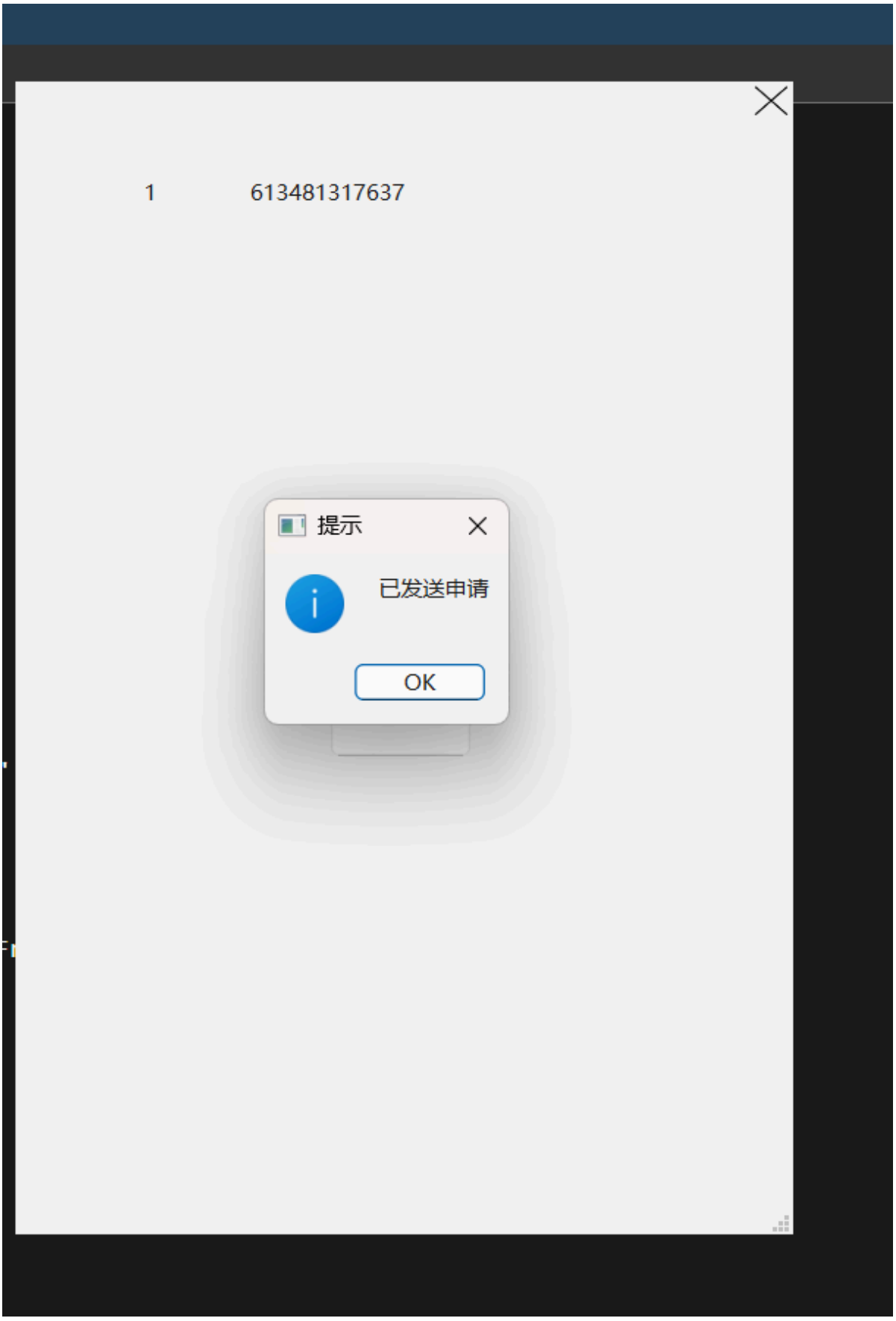



585776126431
1
已阅读并同意服务协议和QQ隐私服务
注册
登录

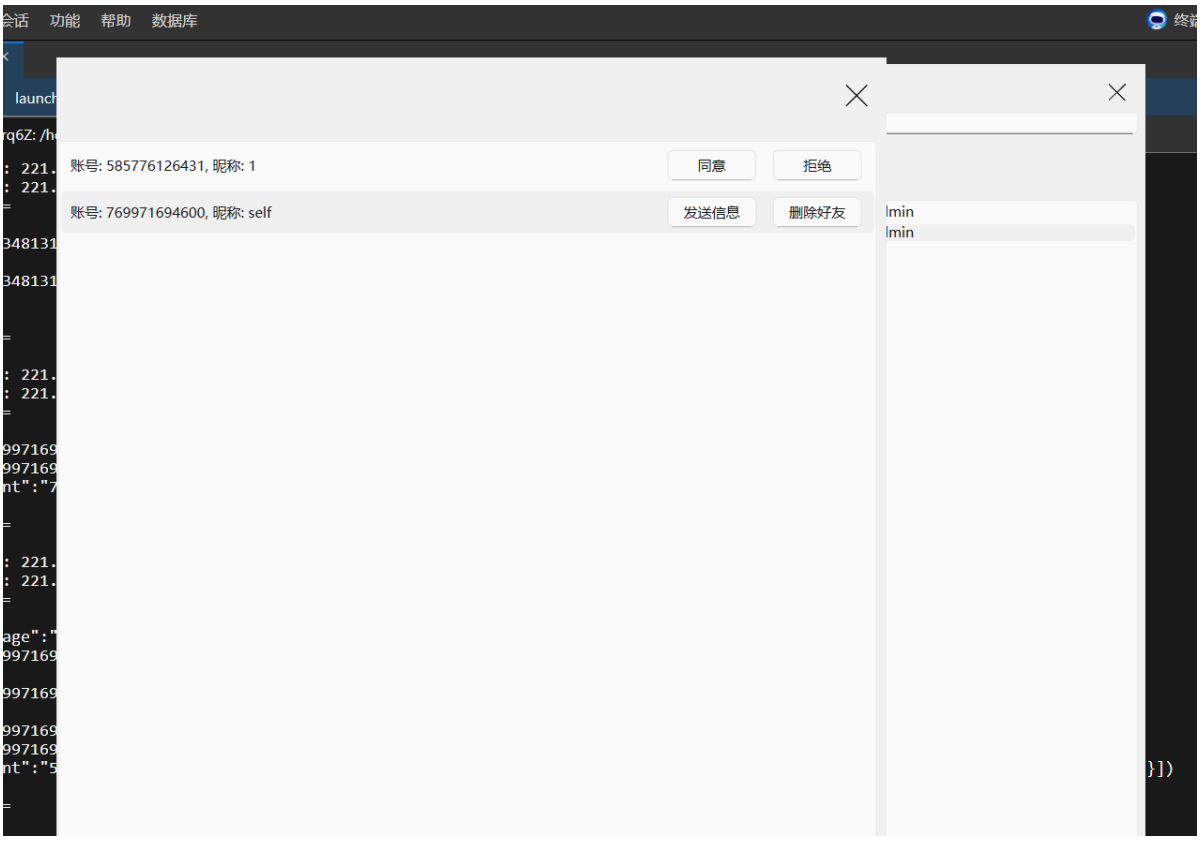
5.3搜索用户


6](C:\Users\13060\AppDat
511152911256.
1
账号：613481317637，昵称：1
账号：585776126431，昵称：1
58577
已阅读并同意服务协议和

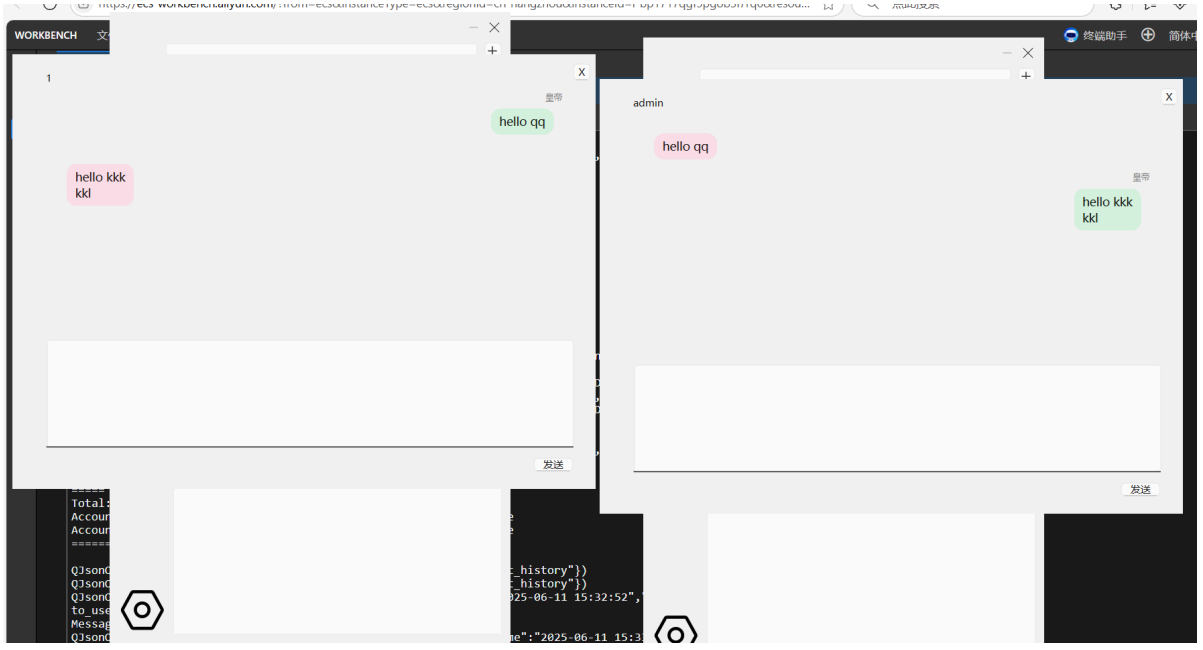
5.4好友申请



5.5好友管理



5.6聊天界面



6.任务分工

郭警豪 2410120025

- **注册与登录**

- **好友申请**
- **好友管理**

陈晓豪 2410120034

- **聊天界面**
- **消息气泡类**