项 目 名 称: QiuQiu实时聊天软件

学 校 名 称:北京理工大学

项 目 讲 师: 崔浩

姓 名: 何宇轩

学 号: 1120203174

实 训 日 期: 2022年8月15日-8月25日

**1 引言**

1.1项目概述

QiuQiu是一个实时通信软件，允许两人使用网络实时的传递文字消息和文件。用户通过邮箱进行注册，可以选择自己的头像和昵称，并以邮箱作为唯一的身份验证标准。

项目目的是构建Linux平台的编译环境，完成对实时通讯软件的讨论、设计全过程，实现即时聊天功能和文件传输功能，并使用户能对部分个人信息实现自定义。

项目使用c/s架构，由服务器（server）和客户端（client）组成。服务端进行数据库操作存储用户信息、聊天记录和传输文件副本等数据。同时进行用户注册、登入请求的处理与进行聊天信息和文件的转发工作。客户端提供了用户交互界面，提供有注册、登入、添加好友、即时聊天、修改头像、传输文件等功能。

项目使用Qt工具采用C++编写，即时通讯和文件传输使用TCP协议，编程主要使用Qt提供的QTcpServer和QTcpSocker类。数据存储使用了数据库Mysql。

UI界面采用模拟QQ的界面设计，通过Qt的UI界面设计和代码相结合的形式，旨在给用户新鲜和流畅的体验，通过自定义背景、控件避免给用户使用未经美化过的UI部件，界面颜色以蓝色和白色为主体，在主界面的右下角可以实现一键换肤，完成日间模式和夜间模式的切换，从而根据用户所处环境对眼睛进行保护。

1.2 需求分析

即时通讯是目前互联网上最时兴最流行的通讯形式，各种各样的即时通讯软件也层出不穷。从实际工程应用角度出发,以计算机网络原理为指导,结合当前网络中的一些常用技术,编程实现基于C/S架构的网络聊天工具是切实可行的。

1.3 运行环境

软件环境：Qt Creator 4.11.2 Linux系统

**2 项目设计**

2.1 设计思路

我作为小组中客户端UI的部分，主要与罗嘉宇同学共同负责该部分的设计和开发。以下主要介绍这部分的内容。

首先，在当前这个信息化时代，人们不通过面对面的交谈，也能进行交心交谈，彼此传递信息，但是，随着要求的不断增加与繁杂，国内最典型的社交软件qq已经成为了大部分人诟病的存在，从其用户的不断流逝与微信和tim（轻量版QQ）的广受欢迎中已经可以看到端倪。于是，我们首先确立设计目标是设计一款轻量级的、页面简约、使用轻便的简单聊天程序。

对于登录界面来说，因为只在登录时会看见，可以略微花哨一些，于是我们设置了动态图片以增加美感，另外，登录界面需要有注册功能，找回密码功能，参考qq我们同样认为可以拥有记住密码以及自动登录功能，这两个功能轻便好用，符合我们轻量化聊天软件的要求。

注册功能需要输入账号，以及密码。

记住密码和自动登录只有在成功登录时会被设置，记住密码会记住账号和密码，自动登录则会自动记住密码。

登陆界面由罗嘉宇同学完成，在这里不多赘述。

主界面主要由我完成。因为该界面会一直看见，所以我们设计了简单的纯色风格，采用扁平化的设计，给人看起来十分舒适。

在主界面的左下角为添加好友按钮，右下角为换肤按钮：日间与夜间自如切换。

主界面的上部分为用户的主要信息界面：左侧为头像，右侧上部为昵称，下部为个性签名。点击用户头像弹出修改头像窗口即可修改头像。

修改头像窗口可实现上传头像、头像旋转、头像缩放等功能、可让用户尽情自定义头像。

在消息列表模块设计了显示头像、显示昵称、显示最后一条消息及其时间，力求还原QQ的主界面布局风格。双击好友消息框即可进入聊天室。

聊天室由我完成。上大半部分为消息显示区域，下半部分为消息编辑区域。

消息显示为气泡形式，与QQ十分相似。

聊天室有发送文件按钮，点击即可弹出选择文件窗口，选择文件之后即可在聊天室右侧弹出发送文件窗口，可现实文件类型图标、文件名称以及发送进度条。

2.2 模块功能介绍

登录模块：

登录模块包括注册、找回密码、自动登录、记住密码、修改密码以及登录的功能。

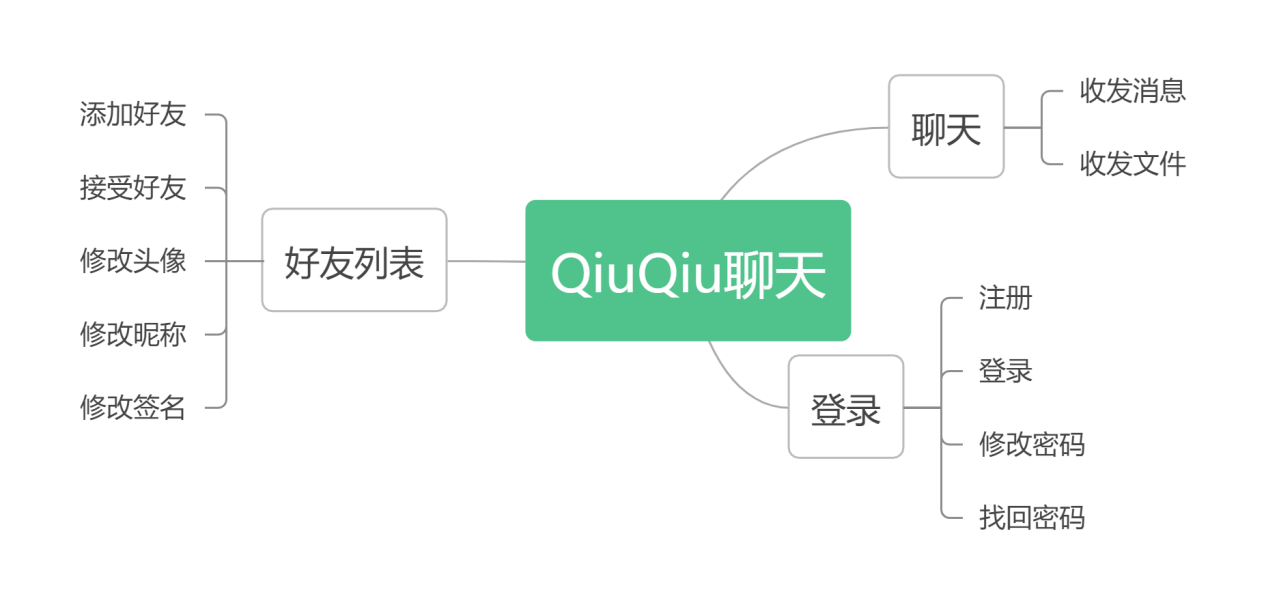
主界面模块：

好友列表模块包括六大功能：消息列表、好友列表、添加好友、接受好友、修改头像、切换皮肤。

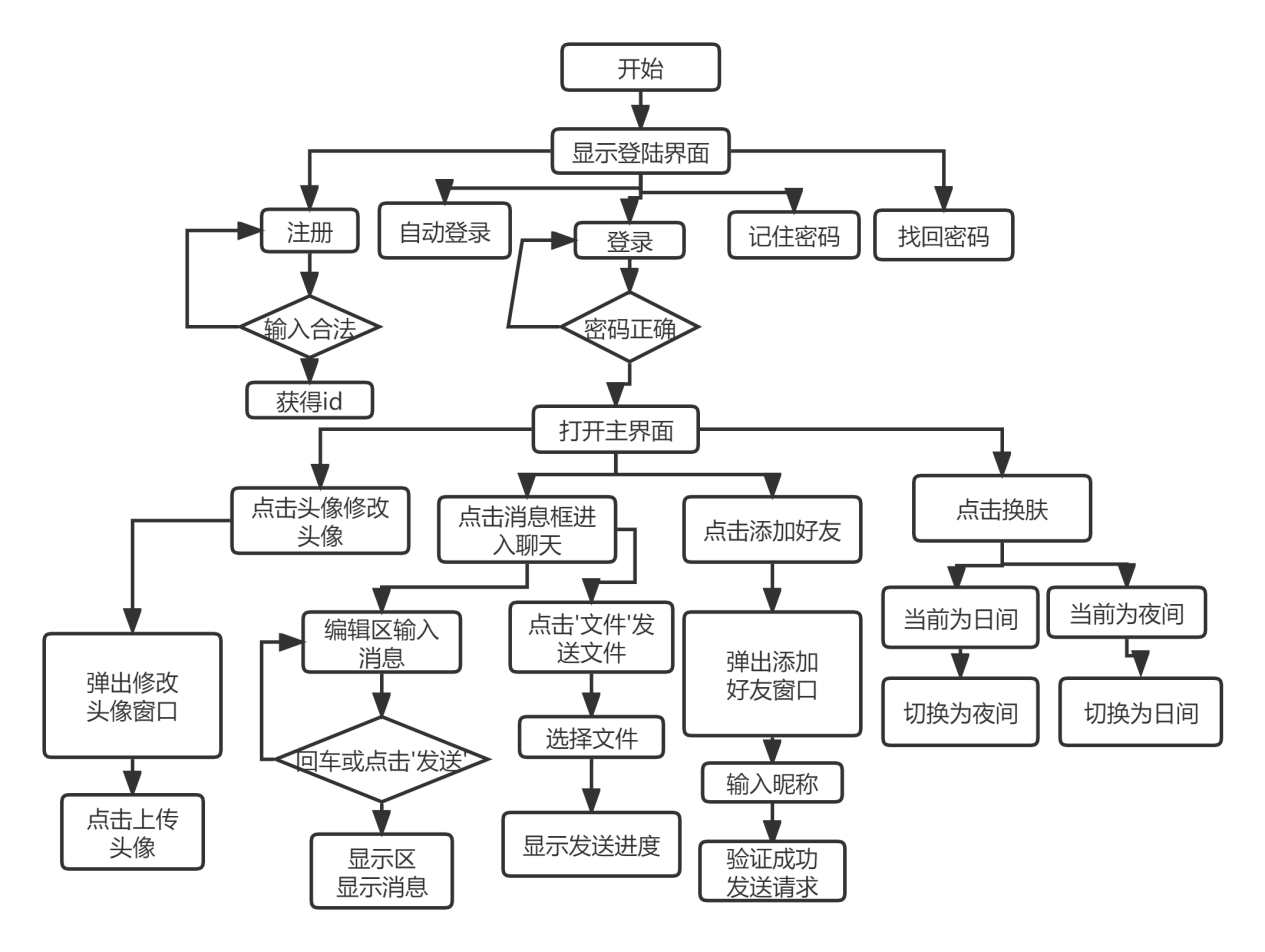
聊天模块：

聊天模块包括两大功能：好友聊天、文件传输（接受和发送）。

2.3 模块结构图



2.4 程序流程图



2.5 功能设计分工

何宇轩：

负责主界面消息列表、聊天室、修改头像、切换皮肤的ui设计。

罗嘉宇：

负责了登录界面、好友列表、添加好友窗口的ui设计。

周振翔、田东琦：

实现了登录、找回密码、注册、自动登录、记住密码、收发消息文件、修改头像昵称签名的功能。

周佳慧：

负责用户数据存储。

**3 详细设计**

①个人信息：从服务器获取个人信息，将头像昵称签名显示在主界面上部分。

②消息、好友列表：显示个人信息的同时获得好友信息并绘制消息列表和好友列表。

③好友聊天：双击消息框进入聊天，弹出相应好友的聊天窗口，发送和接收消息并显示，发送文件、接受文件并存储。

④添加好友：点击添加好友按钮弹出窗口，输入个人ID，发送个人ID给服务器，服务器查询是否有此人并告诉申请者，若有此人这传递好友请求，另一个客户端获得好友请求并显示。

⑤接受好友：对方向服务器发送ID后，服务器查询如有此人，则向本人发出弹窗。

⑥记住密码：在登陆成功时将账号与密码写入配置文件，并且在每次打开登录界面时从配置文件中调取账号密码，默认为空

⑦自动登录：在记住密码的基础上，每次打开登录界面时，在程序运行时直接执行登录按钮的后端逻辑，实现自动登录

⑧注册账号：点击注册按钮，打开注册信息填写菜单，其中保证输入框不为空，两次填写密码一致，密保用来找回密码，点击发送将数据发送给服务器

⑨找回密码：点击找回密码按钮，等待服务器响应，服务器响应后返回密码结果

⑩登录：在保证账号密码不为空时点击登录按钮，会将账号密码传给服务器进行验证，等待服务器响应，如果成功则打开好友列表并关闭登录界面

⑪修改个人信息：点击修改个人信息按钮，打开修改页面，点击提交后将修改的信息发给服务器和好友列表，实现实时修改

⑫修改头像：在配置文件中存储头像路径，每次启动登录界面和好友列表时将图片设为头像，点击头像会打开资源管理器选择图片，将该路径的文件传给主界面设为当前头像，并同时更改配置文件的头像路径，实现下次登录的头像变更。

⑬切换皮肤：点击换肤按钮，识别当前皮肤模式，并切换为另一个皮肤模式：标题栏和窗体背景色全部修改，按键正常、悬浮和按下的样式对应修改。

⑭收发文件：点击发送文件按钮弹出选择文件窗口，选择好文件后在聊天室右侧显示发送文件窗口并显示发送进度条，服务器将数据流传给对方。对方将会弹出接收文件窗口并显示接受进度。

**4 实现方案**

①个人信息：在登陆界面按下登录键时，从服务器获取个人信息，若信息符合则显示主界面，将头像昵称签名显示在主界面上部分。

②消息、好友列表：显示个人信息的同时获得好友信息并绘制消息列表。消息列表有好友消息盒子组成，显示好友头像 昵称 最后一条消息及其时间。

核心代码：

void loginWidget::**on\_loginpushButton\_clicked**()

{

qDebug()<<"login successed";

this->hide();

mainwin.show();

}

//用户信息初始化

void ListWidget::**initMessage**(const QString name, const QString icon, const QString sign)

{

//用户信息布局

head = new QPushButton(this);

QLabel \*username = new QLabel(this); //名字

QLabel \*usersign = new QLabel(this);

head->setMouseTracking(true);

head->setStyleSheet(QString("width:70px;height:70px;border:0px;border-radius:35px;"

"border-image:url(:/image/toolicon/%1);").arg(icon));

username->setText(name);

username->setStyleSheet("QLabel{font: 500 11pt Microsoft YaHei;"

"color: white;}");

usersign->setText(sign);

usersign->setStyleSheet("QLabel{font: 100 8pt Microsoft YaHei;"

"color: white;}");

//竖直布局

QWidget \*vlayout\_widget = new QWidget(this);

QVBoxLayout \*vLayout = new QVBoxLayout(vlayout\_widget);

//水平布局2

QWidget \*hlayout\_widget\_2 = new QWidget(vlayout\_widget);

QHBoxLayout \*hLayout\_2 = new QHBoxLayout(hlayout\_widget\_2);//水平布局2

//水平布局3

QWidget \*hlayout\_widget\_3 = new QWidget(vlayout\_widget);

QHBoxLayout \*hLayout\_3 = new QHBoxLayout(hlayout\_widget\_3);//水平布局3

ui->userLayout->addWidget(head);

ui->userLayout->addWidget(vlayout\_widget);

vLayout->addWidget(hlayout\_widget\_2);

vLayout->addWidget(hlayout\_widget\_3);

hLayout\_2->addWidget(username);

hLayout\_2->addStretch(2);

//hLayout\_2->addWidget(qqmessage);

hLayout\_3->addWidget(usersign);

hLayout\_2->setContentsMargins(0,10,0,0);

hLayout\_3->setContentsMargins(0,0,0,15);

}

//将好友加入消息列表

for(int i=0;i<FriNum;i++)

{

newsBox\* box = new newsBox(FriMess[i].name, FriMess[i].icon, FriMess[i].lastnews, FriMess[i].lastime, FriMess[i].direct, this);

box->installEventFilter(box);

box->setStyleSheet("QWidget::hover{background-color:rgb(235,235,235)}");

boxList.append(box);

vLayout->addWidget(boxList[i]);

}

QWidget\* pWidget = new QWidget(this);

pWidget->setLayout(vLayout);

ui->newslist->setWidget(pWidget);

//头像，qq名，qq信息，最后信息时间，发送方向

Fname = name;

Ficon = icon;

Fnews = news;

Ftime = time;

Fdir = dir;

qqicon->setStyleSheet(QString("width:50px;height:50px;border:0px;border-radius:25px;border-image:url(:/image/toolicon/%1);").arg(icon));

qqname->setText(name);

qqmessage->setText(news);

qqmessage->setStyleSheet("font: 100 8pt Microsoft YaHei;color: rgb(116, 120, 122);}");

msg\_date\_label->setText(time);

msg\_date\_label->setStyleSheet("font: 100 8pt Microsoft YaHei;color: rgb(116, 120, 122);}");

hLayout->addWidget(qqicon);

hLayout->addWidget(vlayout\_widget);

vLayout->addWidget(hlayout\_widget\_2);

vLayout->addWidget(hlayout\_widget\_3);

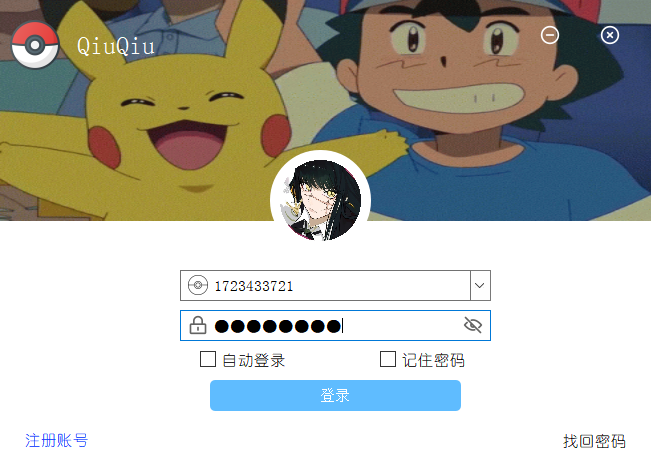
hLayout\_2->addWidget(qqname);

hLayout\_2->addStretch(2);

hLayout\_2->addWidget(msg\_date\_label);

hLayout\_3->addWidget(qqmessage);

实现结果：





③好友聊天：双击消息框进入聊天，弹出相应好友的聊天窗口，发送和接收消息并显示，发送文件、接受文件并存储。

核心代码：

//双击消息框进入聊天

bool newsBox::***eventFilter***(QObject \*obj, QEvent \*eve)

{

if(obj == this)

{

if (eve->type() == QEvent::MouseButtonDblClick)

{

ChatWindow\* chat = new ChatWindow(Fname, Ficon, Fnews, Fdir, nR, nG, nB, nr, ng, nb, mode);

chat->show();

return true;

}

}

return false;

}

//\*信息收发\*//

//“发送”按钮槽函数

connect(ui->Send,&QPushButton::clicked,this,&ChatWindow::sendMessage);

//发送文件按钮槽函数

connect(ui->file,&QPushButton::clicked,this,&ChatWindow::sendfiles);

//添加消息

void ChatWindow::**addMessage**(Message msg)

{

Bubble\* pFrame = new Bubble(msg, ui->scrollArea);

int count = vLayout->*count*();

//本人气泡显示在左侧，对方气泡显示在右侧

qDebug()<<msg.direct;

if(msg.direct == 'rece')

vLayout->insertWidget(count-1, pFrame, 0, Qt::AlignLeft);

else vLayout->insertWidget(count-1, pFrame);

verticalBar->setRange(0, ui->scrollArea->height());

//所有消息列表

pMessages.push\_back(pFrame);

verticalBar->setValue(verticalBar->maximum());

}

//发送消息功能

void ChatWindow::**sendMessage**()

{

QString msg = ui->textEdit->toPlainText();

// 把最后一个回车换行符删掉

while (msg.endsWith("\n")) {

msg.remove(msg.length() - 2, 2);

}

if(msg.isEmpty())

{

QMessageBox msgBox;

msgBox.setText("不能发送空白消息");

msgBox.*exec*();

return;

}

// 清除输入框

ui->textEdit->clear();

Message message;

message.content = msg;

addMessage(message);

}

实现结果：









⑫修改头像：在配置文件中存储头像路径，每次启动登录界面和好友列表时将图片设为头像，点击头像会打开资源管理器选择图片，将该路径的文件传给主界面设为当前头像，并同时更改配置文件的头像路径，实现下次登录的头像变更。修改头像界面仿照qq。

核心代码：

void iconChange::**LoadHeader**(const QPixmap &pixmap)

{

m\_sourceHeader = pixmap;

int nSliderMinmum = 0;

QPixmap scaledPixmap;

if (m\_sourceHeader.width() > m\_sourceHeader.height())

{

scaledPixmap = m\_sourceHeader.scaledToHeight(ui->scrollArea->height());

nSliderMinmum = scaledPixmap.height() \* ui->Slider->maximum() / m\_sourceHeader.height();

}

else

{

scaledPixmap = m\_sourceHeader.scaledToWidth(ui->scrollArea->width());

nSliderMinmum = scaledPixmap.width() \* ui->Slider->maximum() / m\_sourceHeader.width();

}

ui->scrollAreaWidgetContents->resize(m\_sourceHeader.size());

ui->Slider->setEnabled(true);

ui->Slider->setMinimum(nSliderMinmum);

ui->Slider->setValue(nSliderMinmum);

ui->head->setPixmap(scaledPixmap);

m\_pDivWidget->show();

}

bool iconChange::***eventFilter***(QObject\* watched, QEvent\* event)

{

if (m\_pDivWidget == watched && QEvent::Paint == event->type()) //绘制遮罩层

{

m\_pDivWidget->resize(ui->scrollArea->size());

QPainterPath path;

path.addRect(m\_pDivWidget->geometry());

path.addEllipse(m\_pDivWidget->geometry().adjusted(2, 2, -2, -2));

QPainter p(m\_pDivWidget);

p.setRenderHint(QPainter::Antialiasing);

p.setBrush(QColor(100, 100, 100, 200));

p.setClipPath(path);

p.drawRect(m\_pDivWidget->geometry());

}

return QWidget::eventFilter(watched, event);

}

void iconChange::**onOk**()

{

if (const QPixmap\* pHeader = ui->head->pixmap())

{

QFileInfo info(iconName);

usericon = info.fileName();

qDebug()<<usericon;

ui->labeltip->setText(QString/\*::fromLocal8Bit\*/("修改头像成功"));

emit updateHeader(usericon);

}

else

{

ui->labeltip->setText(QString/\*::fromLocal8Bit\*/("还未加载头像"));

}

}

void ListWidget::**headChange**()

{

iconchange->show();

}

void ListWidget::**recvsignal**(const QString &iconPath)

{

usericon = iconPath;

qDebug()<<usericon;

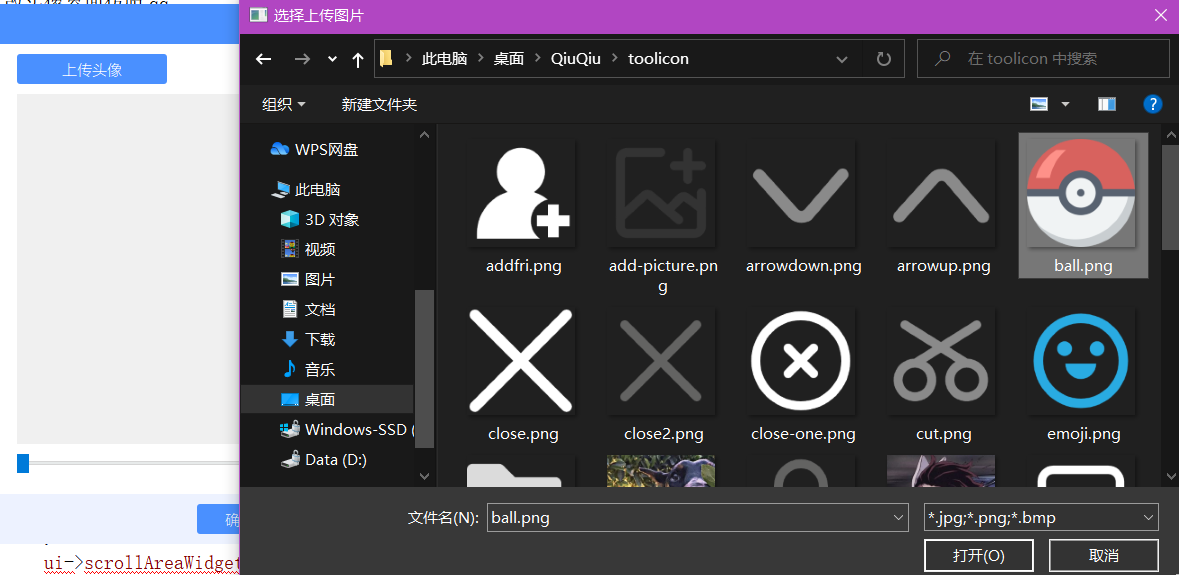
head->setStyleSheet(QString("width:70px;height:70px;border:0px;border-radius:35px;border-image:url(:/image/toolicon/%1);").arg(usericon));

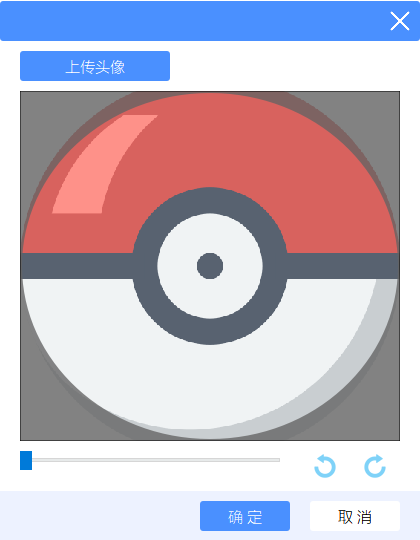
///客户端组实现将本地头像上传至云端

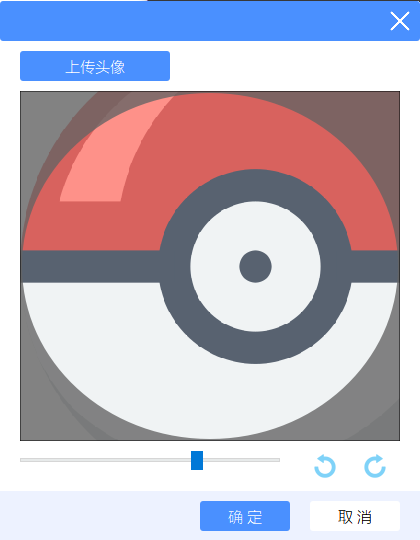
//\*...\*//

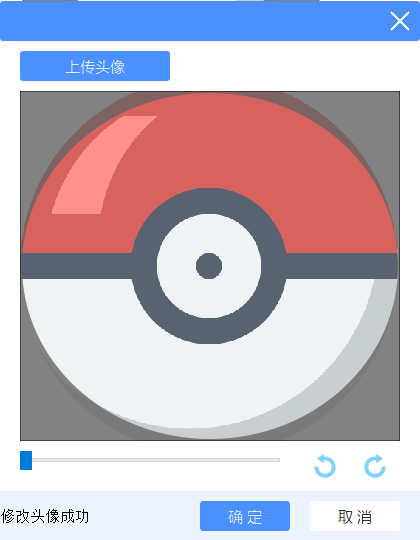
}

实现结果：











⑬切换皮肤：点击换肤按钮，识别当前皮肤模式，并切换为另一个皮肤模式：主界面和聊天室标题栏和窗体背景色全部修改，按键正常、悬浮和按下的样式对应修改。

核心代码：

void ListWidget::**on\_mode\_clicked**()

{

if(ui->mode->toolTip() == "切换夜间模式")

{

ui->mode->setStyleSheet(QString("border-image:url(:/image/toolicon/moon.png);"));

ui->mode->setToolTip("切换日间模式");

themeChange(72, 61, 120, 72, 61, 139, 1);

}

else if(ui->mode->toolTip() == "切换日间模式")

{

ui->mode->setStyleSheet(QString("border-image:url(:/image/toolicon/sun.png);"));

ui->mode->setToolTip("切换夜间模式");

themeChange(255, 255, 255, 74, 144, 255, 0);

}

}

void ListWidget::**themeChange**(int br, int bg, int bb, int tr, int tg, int tb, int mode)

{

R = br;

G = bg;

B = bb;

tR = tr;

tG = tg;

tB = tb;

bgR = br;

bgG = bg;

bgB = bb;

for(int i=0;i<FriNum;i++)

{

boxList[i]->nR = R;

boxList[i]->nG = G;

boxList[i]->nB = B;

boxList[i]->nr = tr;

boxList[i]->ng = tg;

boxList[i]->nb = tb;

boxList[i]->mode = mode;

}

setColor();

ui->userBox->setStyleSheet(QString("background-color: rgb(%1,%2,%3)").arg(tr).arg(tg).arg(tb));

ui->toolBar->setStyleSheet(QString("background-color: rgb(%1,%2,%3)").arg(tr).arg(tg).arg(tb));

m\_titleBar->setBackgroundColor(tr, tg, tb);

//"消息"按钮样式

int bgr, bgg, bgb;

if(mode == 0)

{

bgr = bgg = bgb = 255;

}

else

{

bgr = tr;

bgg = tg;

bgb = tb;

tr = tg = tb = 255;

}

...

}

void ChatWindow::**setTheme**(int r, int g, int b, int tr, int tg, int tb, int mode)

{

bgR = r;

bgG = g;

bgB = b;

setColor();

m\_titleBar->setBackgroundColor(tr, tg, tb);

if(mode == 0)

{

ui->scrollArea->setStyleSheet(QString("background-color: rgb(255, 255, 255);"));

ui->textEdit->setStyleSheet(QString("background-color: rgb(238, 242, 255);"));

ui->Close->setStyleSheet(QString("font: 25 12pt Microsoft YaHei;"

"color: rgb(255, 255, 255);"

"background-color: rgb(255, 129, 131);"

"border-radius:5px"));

ui->Send->setStyleSheet(QString("font: 25 12pt Microsoft YaHei;"

"color: rgb(255, 255, 255);"

"background-color: rgb(134, 170, 255);"

"border-radius:5px;"));

}

...

}

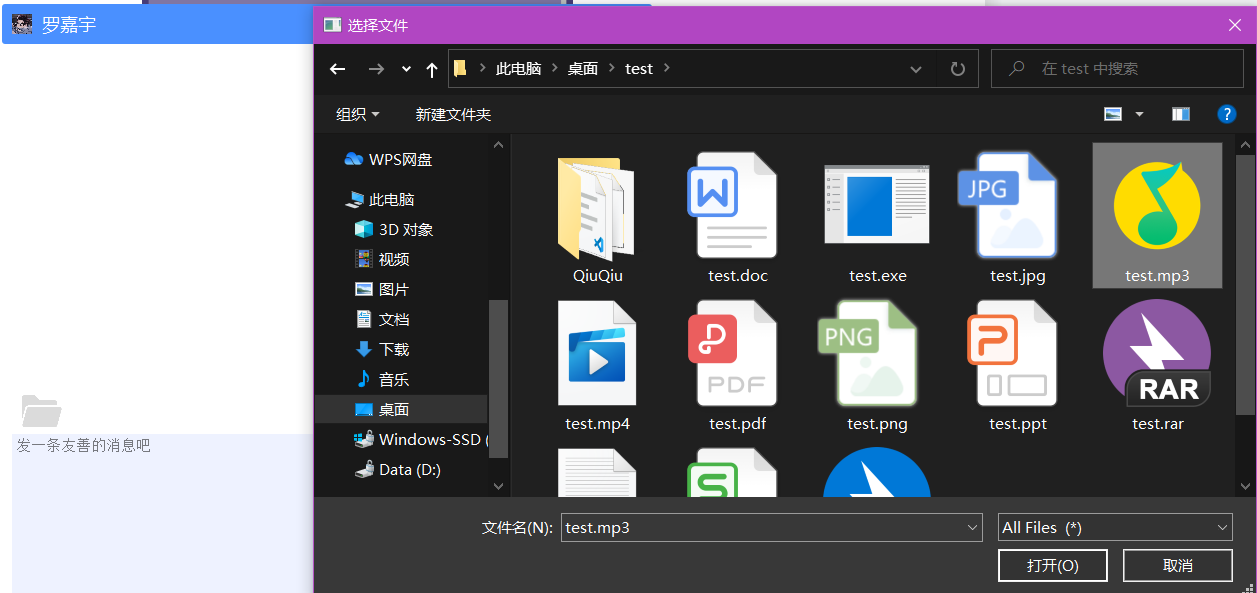
实现结果：

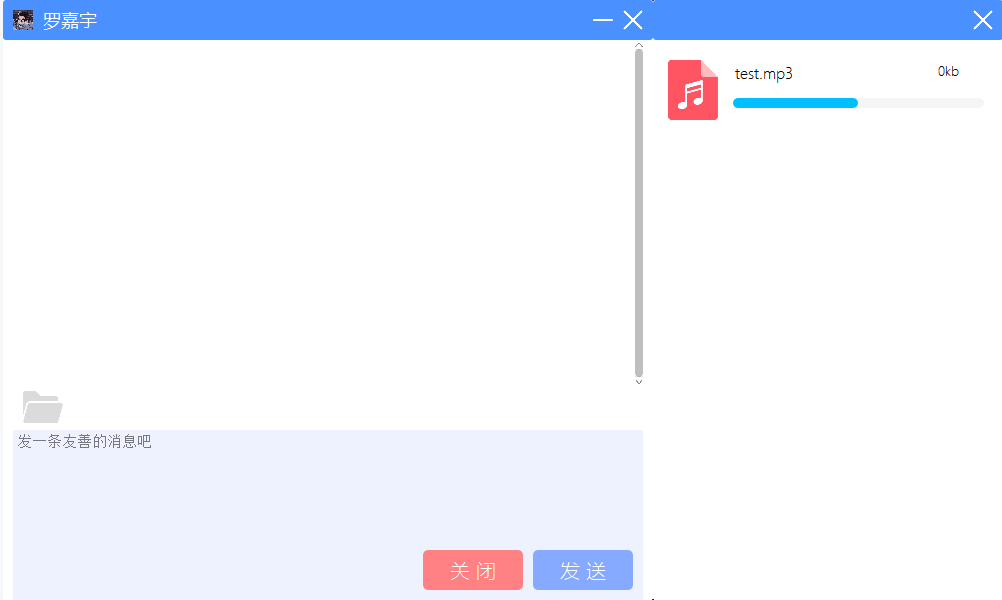






⑭收发文件：点击发送文件按钮弹出选择文件窗口，选择好文件后在聊天室右侧显示发送文件窗口并显示发送进度条，服务器将数据流传给对方。对方将会弹出接收文件窗口并显示接受进度。发送文件窗口与聊天室同步移动同步关闭。发送文件图标会根据文件类型不同而改变。





**5 系统测试**

①个人信息：登录账号后，观察主界面上半部分是否显示用户信息。

成功显示用户头像昵称签名；

无bug；

②好友列表：登录账号后，观察主界面主要部分是否显示好友消息盒子。

成功显示聊过天的好友的消息盒子的好友昵称头像和聊天信息；

无bug；

③好友聊天：双击好友消息盒子进入聊天，弹出相应好友的聊天窗口，并且发送消息时，自己和好友都需要看到。

无bug；

④发送文件：点击发送文件按钮时，要弹出选择文件窗口；选择文件后，右侧弹出文件传输窗口，并显示文件类型图标文件名文件大小和发送进度。

Bug：文件进度条不更新，一直显示为0；

修正方法：具体参见服务端组报告。

⑤修改头像：点击用户头像弹出修改头像页面，点击上传头像可弹出头像选择窗口，选择完成后显示图片，滚动条实现缩放图片大小，旋转按钮实现图像旋转。点击确定更新主界面用户头像；

成功显示上述事件；

无BUG。

⑥切换皮肤：点击换肤按钮，主界面和聊天框的窗口以及标题栏切换皮肤；

点击按钮时成功换肤；

无bug。

**6 心得体会**

这是我第一次亲自做一个工程，感触十分深刻。

在完成项目的过程中，需要学习许多新知识，学习新知识的过程比较折磨人，但做出效果之后，成就感往往是大于这些的。

同时，最令我感受深刻的是多人协同做一个项目时，Debug（）就成为了一个非常重要的任务。常常是客户端和服务端同时打断点调试，互相找错误，末尾的几天基本都是在找BUG，改BUG，一做就是一天一晚上还通宵，但是当把BUG改完，产品变得比较完善时，大家都可以安息了。

完成这个项目的过程我收获的感觉要大于收获的知识，我切身体会到了什么叫做完成一个项目，如何与他人协作完成项目，同时在高强度连续性的工作中如何坚持下来。