

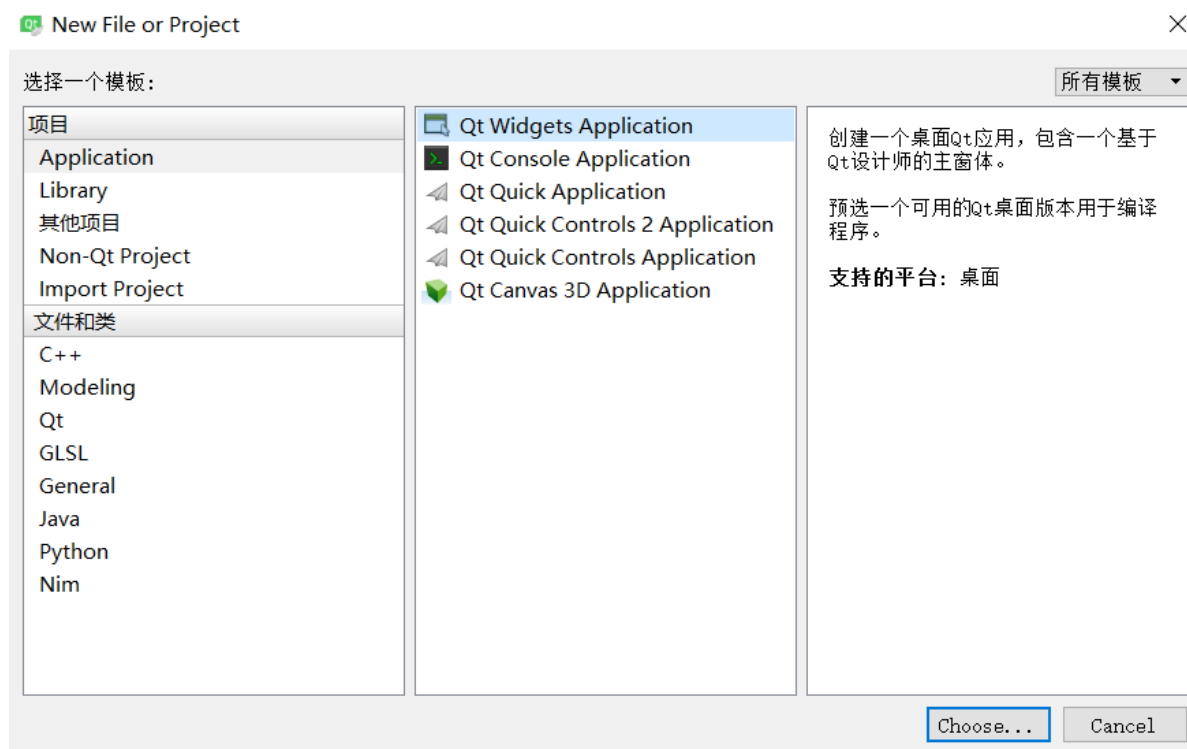
# Qt实现简易QQ

## 1 概述

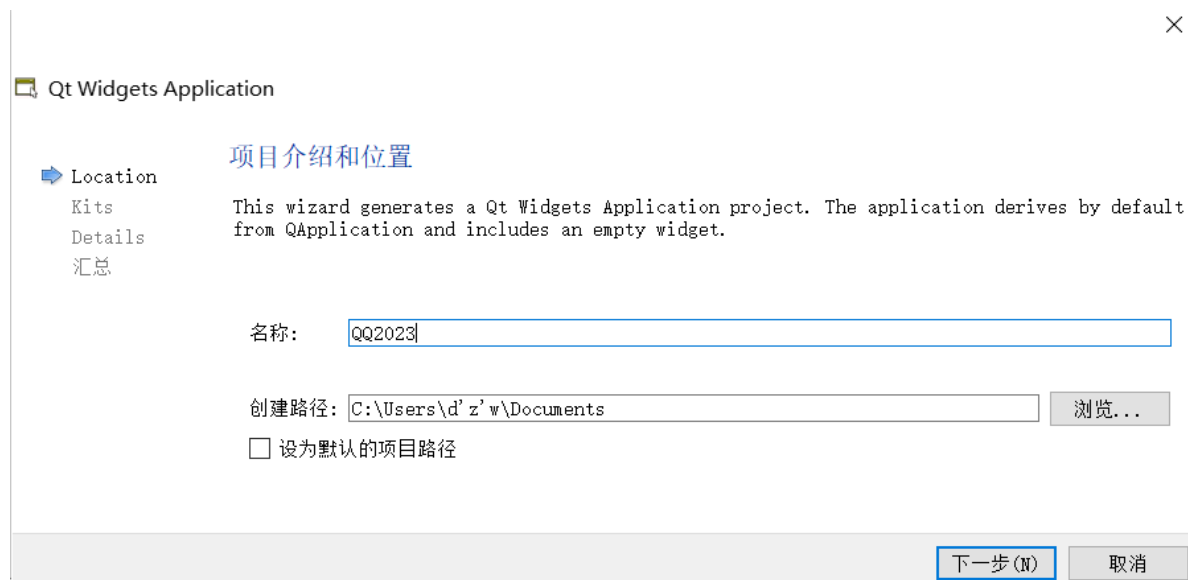
本项目通过Qt Creator以及socket编程实现了简易QQ，其中包括功能有通过账号密码登录、更改头像、在公共聊天室聊天、更换字体类型、更换字体颜色、保存聊天记录等。

## 2 创建新项目

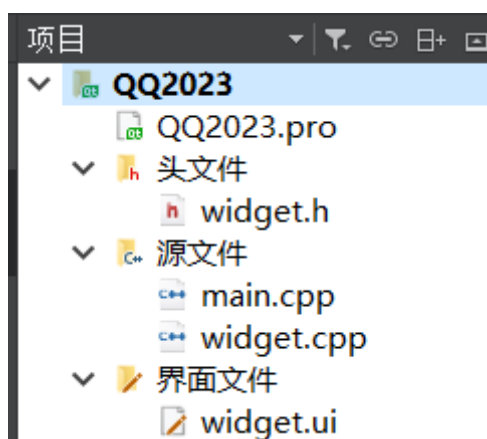
### 2.1 新建项目，以Qt Widgets Application为模板



## 2.2 选择 *QWidget* 为基类



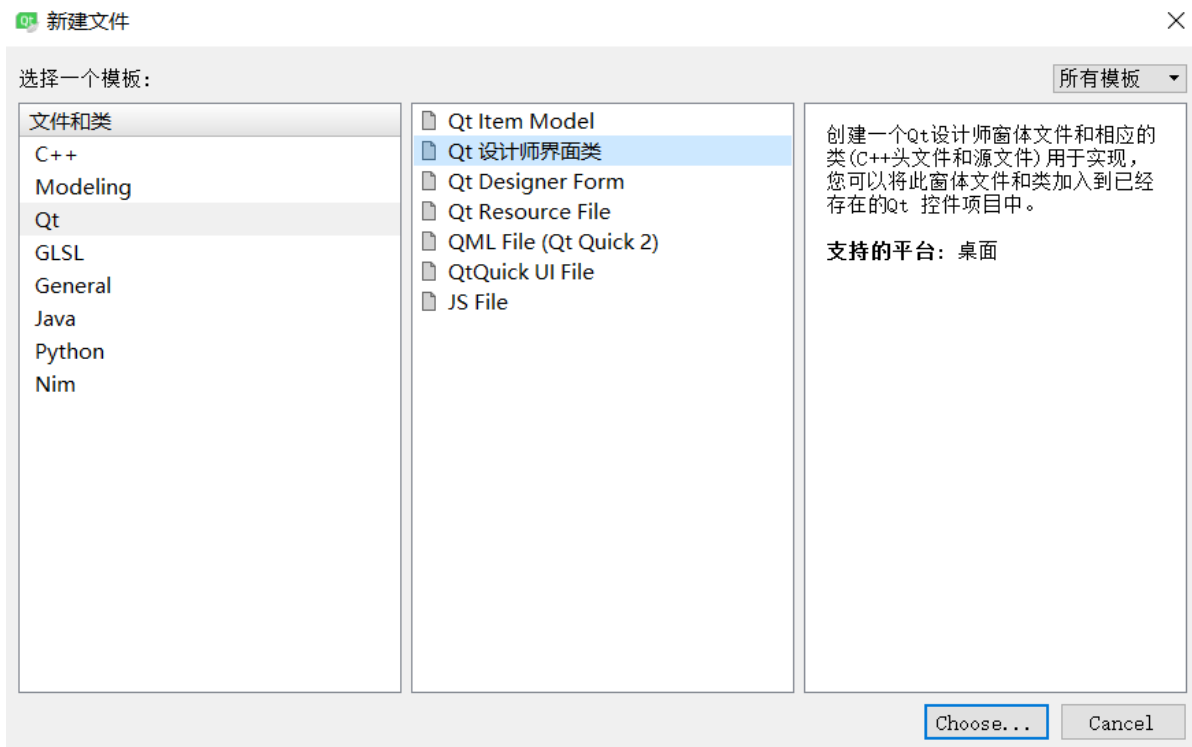
## 2.3 建立完成后，项目视图如下所示



# 3 创建对话列表

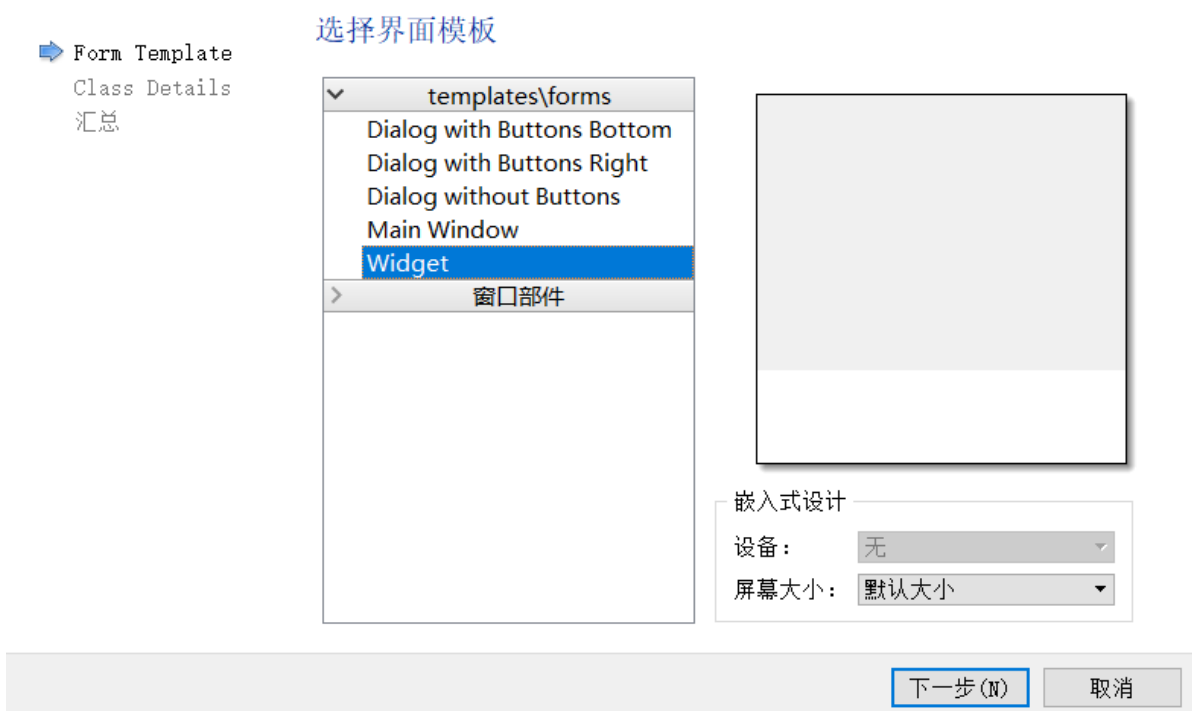
## 3.1 添加对话框列表

### 3.1.1 右击 *QQ2023*，选择添加新文件



3.1.2 选择Qt->Qt设计师界面类, 选择Widget

Qt 设计器界面类



## 3.2 设计对话框列表

3.2.1 在 *fdform.ui* 中修改 *fdform* 对象的宽度为300, 高度为650

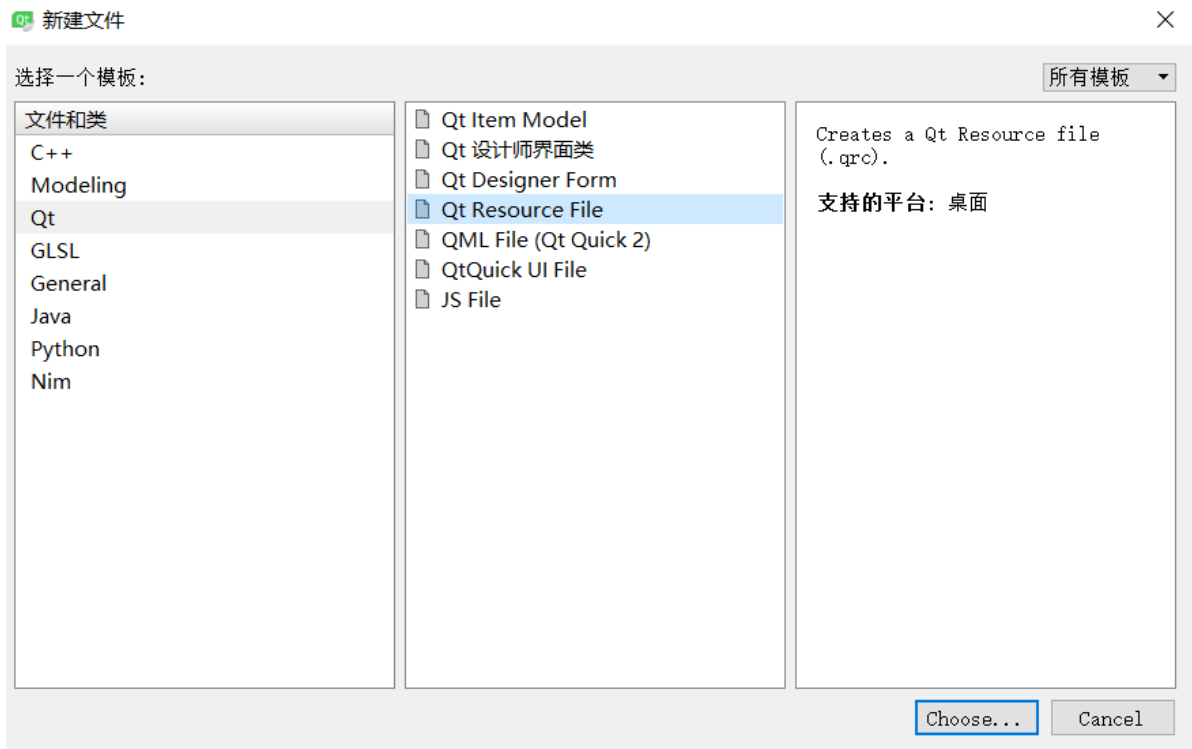
fdform : QWidget	
属性	值
▼ QObject	
objectName	fdform
▼ QWidget	
windowModal...	NonModal
enabled	<input checked="" type="checkbox"/>
▼ geometry	[(0, 0), 300 x 650]
X	0
Y	0
宽度	300
高度	650

3.2.2 拖拽 *QToolBox* 到 *fdform* 中, 删除页2, 对 *fdform* 进行垂直布局, 修改 *QToolBox* 的 *currentItemText* 为群成员, 修改 *fdform* 的布局边缘为0; 为 *QToolBox* 添加控件, 进行垂直布局, 然后删除控件, 从而为 *page* 添加布局, 修改 *page* 的布局名称为 *horizontalLayout*

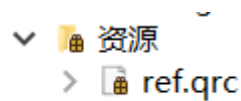
fdform : QWidget	
属性	值
> accessibleName	
> accessibleDes...	
layoutDirection	LeftToRight
autoFillBackgr...	<input type="checkbox"/>
styleSheet	
> locale	Chinese, China
> windowFilePath	
> inputMethod...	ImhNone
▼ Layout	
layoutName	horizontalLayout
layoutLeftMar...	11
layoutTopMar...	11
layoutRightM...	11
layoutBottom...	11
layoutSpacing	7
layoutStretch	0
layoutSizeCon...	SetDefaultConstraint

### 3.3 导入资源

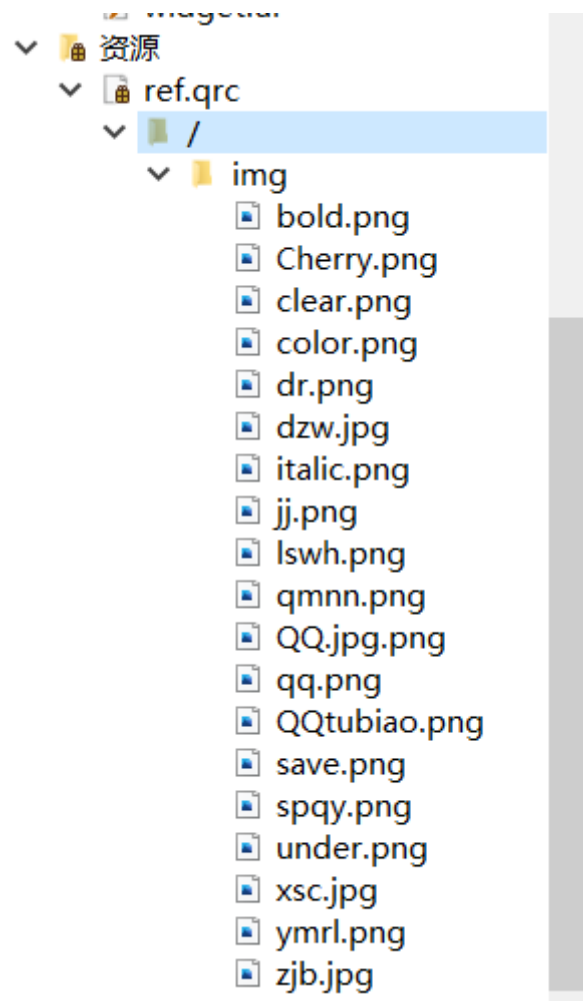
3.3.1 为Qt添加资源文件, 右击项目名称 *QQ2023*, 选择Qt->Qt Resource File, 下一步命名为 *ref*



3.3.2 确定后，项目视图如下：



3.3.3 添加前缀和文件，导入图片



### 3.4 显示fdform

3.4.1 *main.cpp*进行修改如下:

```
1  #include "widget.h"
2  #include "login.h"
3  #include "fdform.h"
4
5  #include <QApplication>
6  #include <QLocale>
7  #include <QTranslator>
8
9  int main(int argc, char *argv[])
10 {
11     QApplication a(argc, argv);
12
13     QTranslator translator;
14     const QStringList uiLanguages = QLocale::system().uiLanguages();
15     for (const QString &locale : uiLanguages) {
16         const QString baseName = "qq2023_" + QLocale(locale).name();
17         if (translator.load(":/i18n/" + baseName)) {
18             a.installTranslator(&translator);
19             break;
20         }
21     }
22     login b;
23     b.show();
24     return a.exec();
25 }
```

3.4.2 *fdform.cpp*修改如下

其中实现功能包括对各个成员用`toolbutton`按钮进行设置、呈现QQ图标、建立9个公共聊天室以及点击即弹出对话框等。

```
1  #include "fdform.h"
2  #include "ui_fdform.h"
3  #include <QIcon>
4  #include <QToolButton>
5  #include <QMessageBox>
6  #include "widget.h"
7
8  fdform::fdform(QWidget *parent, QString name) :
9      QWidget(parent),
10      ui(new Ui::fdform)
11  {
12      ui->setupUi(this);
13      //this->setWindowIcon(QIcon(":/img/Qtubiao.png"));
14      //this->setWindowTitle("QQ 2023");
15
16      QList<QString> nameList;
17      nameList << "公共聊天室1" << "公共聊天室2" << "公共聊天室3" << "公共聊天室4" << "公共聊天室5"
18          << "公共聊天室6" << "公共聊天室7" << "公共聊天室8" << "公共聊天室9";
19      QStringList iconNameList; //图标资源列表
20      iconNameList << "qq" << "qq" << "qq" << "qq" << "qq"
21          << "qq" << "qq" << "qq" << "qq";
```

```

22
23 QVector<QToolButton*> vToolBtn;
24 for(int i = 0; i < 9; i++){
25     QToolButton * btn = new QToolButton;
26     btn->setText(nameList[i]);
27     QString str = QString(":/img/%1.png").arg(iconNameList.at(i));
28     btn->setIcon(QPixmap(str));
29     btn->setIconSize(QPixmap(str).size());
30     btn->setAutoRaise(true);
31     //设置文字和图片一起显示
32     btn->setToolButtonStyle(Qt::ToolButtonTextBesideIcon);
33     ui->verticalLayout->addWidget(btn);
34     vToolBtn.push_back(btn);
35     //将标识默认初始化为false
36     isShow.push_back(false);
37 }
38 myName = name;
39 for(int i = 0; i < 9; i++){
40     connect(vToolBtn[i], &QToolButton::clicked, [=]() {
41         if(isShow[i]){
42             QString str = QString("%1窗口已经被打开了").arg(vToolBtn[i]->text());
43             QMessageBox::warning(this, "警告", str);
44             return;
45         }
46         isShow[i] = true;
47         //弹出聊天对话框
48         //构造聊天窗口时候, 告诉这个窗口他的名字, 参数1 顶层方式弹出
49         //注意! widget构造函数 并没有这两个参数
50         qint16 portnum = 9990 + i;
51         widget * widget = new widget(0, myName, portnum);
52         widget->setWindowTitle(vToolBtn[i]->text());
53         widget->setWindowIcon(vToolBtn[i]->icon());
54         //widget->port = 9999;
55         widget->show();
56         //如果窗口关闭将标志位设置回来
57         connect(widget, &widget::closewidget, [=]() {
58             isShow[i] = false;
59         });
60     });
61 }
62 }
63 void fdform::closeEvent(QCloseEvent * e){
64     emit this->closeFdform(); //发送关闭当前窗口的自定义信号
65     QWidget::closeEvent(e);
66 }
67
68 fdform::~fdform()
69 {
70     delete ui;
71 }
72

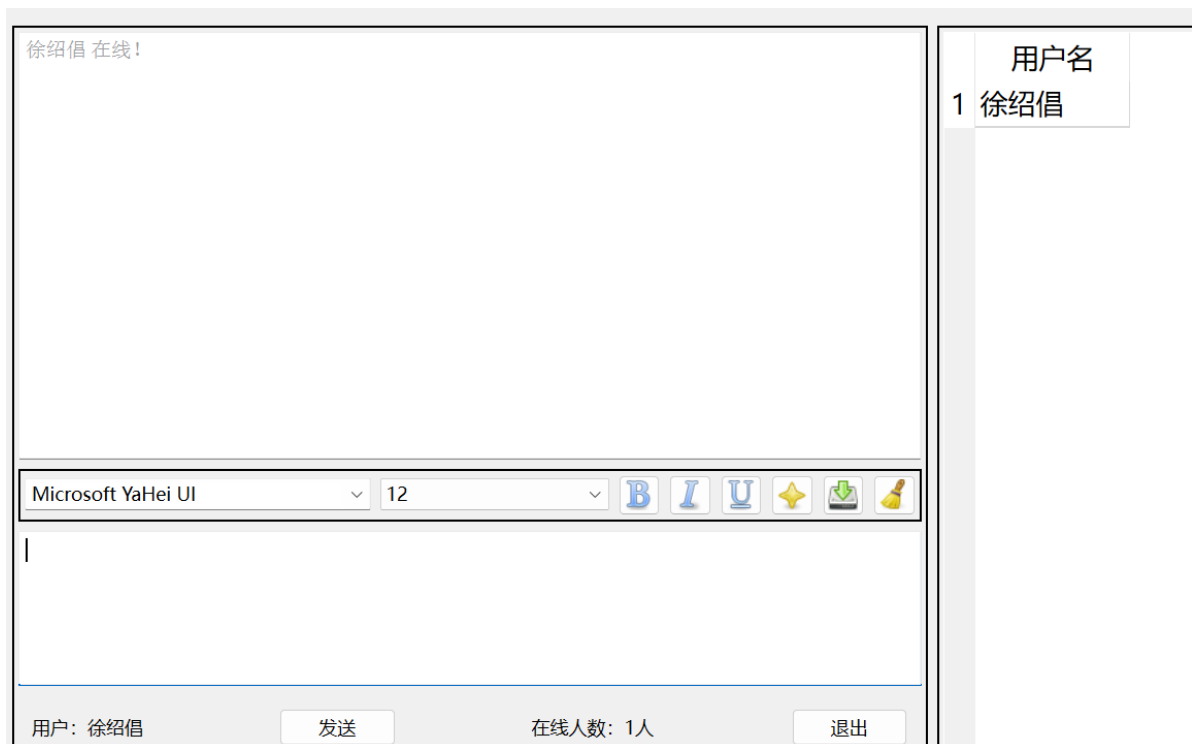
```

3.4.3 设置完成后, 运行效果如下:



#### 4 聊天窗口搭建





在用户登录成功后，会弹出几个已经建立好的群聊，用户可以随意选择群聊，发送或接受消息，在群聊的右方，会显示当前在线的成员。该模块具有修改字体，修改颜色，保存聊天记录的功能。在草稿框可输入将要发送的内容，完成后点击发送，用户使用完成后即可点击退出。

代码如下：

```

1  #include "widget.h"
2  #include "ui_widget.h"
3  #include <QMessageBox>
4  #include <QDataStream>
5  #include <QDateTime>
6  #include <QColorDialog>
7  #include <QFileDialog>
8
9  widget::widget(QWidget *parent,QString name,qint16 portn)
10     : QWidget(parent)
11     , ui(new Ui::Widget)
12 {
13     ui->setupUi(this);
14     myName = name;
15     this->port = portn;
16     this->udpSocket = new QUdpSocket(this);
17     ui->usrName->setText(QString("用户: %1").arg(myName));
18
19     udpSocket->bind(port, QUdpSocket::ShareAddress
20 |QUdpSocket::ReuseAddressHint);
21
22     //
23     connect(udpSocket,&QUdpSocket::readyRead,this,&widget::ReceiveMessage);
24
25     //连接发送按钮
26     connect(ui->sendBtn,&QPushButton::clicked,[=]() {
27         sndMsg(Msg);
28     });
29     //连接退出按钮

```

```

29     connect(ui->exitBtn,&QPushButton::clicked,this,[=]() {
30         this->close();
31     });
32     //新用户进入
33     sndMsg(UsrEnter);
34
35     //设置字体
36     connect(ui->fontCbx,&QFontComboBox::currentFontChanged,[=](const QFont
    &font){
37         ui->msgTxtEdit->setFontFamily(font.toString());
38         ui->msgTxtEdit->setFocus();
39     });
40     //设置大小
41     void (QComboBox:: * sizebtn)(const QString &text) =
    &QComboBox::currentTextChanged;
42     connect(ui->sizeCbx,sizebtn,[=](const QString &text){
43         ui->msgTxtEdit->setFontSize(text.toDouble());
44         ui->msgTxtEdit->setFocus();
45     });
46     //加粗
47     connect(ui->boldTBtn,&QToolButton::clicked,this,[=](bool checked){
48         if(checked){
49             ui->msgTxtEdit->setFontWeight(QFont::Bold);
50         }else{
51             ui->msgTxtEdit->setFontWeight(QFont::Normal);
52         }
53         ui->msgTxtEdit->setFocus();
54     });
55     //倾斜
56     connect(ui->italicTBtn,&QToolButton::clicked,this,[=](bool checked){
57         ui->msgTxtEdit->setFontItalic(checked);
58         ui->msgTxtEdit->setFocus();
59     });
60     //下划线
61     connect(ui->underlineTBtn,&QToolButton::clicked,this,[=](bool checked){
62         ui->msgTxtEdit->setFontUnderline(checked);
63         ui->msgTxtEdit->setFocus();
64     });
65     //设置文本颜色
66     connect(ui->colorTBtn,&QToolButton::clicked,this,[=]() {
67         QColor color = QColorDialog::getColor(color,this);    //color对象可
    以在widget.h中定义私有成员
68         ui->msgTxtEdit->setTextColor(color);
69     });
70     //保存聊天记录
71     connect(ui->saveTBtn,&QToolButton::clicked,this,[=]() {
72         if(ui->msgBrowser->document()->isEmpty()){
73             QMessageBox::warning(this,"警告","聊天记录为空，无法保
    存!",QMessageBox::Ok);
74         }
75         else{
76             QString fName = QFileDialog::getSaveFileName(this,"保存聊天记
    录","聊天记录","(*.txt)");
77             if(!fName.isEmpty()){
78                 //保存名称不为空再做保存操作
79                 QFile file(fName);
80                 file.open(QIODevice::writeOnly | QFile::Text);
81                 QTextStream stream(&file);

```

```

82         stream << ui->msgBrowser->toPlainText();
83         file.close();
84     }
85 }
86 });
87 //清空聊天记录
88 connect(ui->clearTBtn,&QToolButton::clicked,[=]() {
89     ui->msgBrowser->clear();
90 });
91 }
92
93 void Widget::closeEvent(QCloseEvent * e){
94     emit this->closeWidget(); //发送关闭当前窗口的自定义信号
95     sndMsg(UsrLeft);
96     //断开套接字
97     udpSocket->close();
98     udpSocket->destroyed();
99     QWidget::closeEvent(e);
100 }
101
102 Widget::~Widget()
103 {
104     delete ui;
105 }
106
107 void Widget::sndMsg(MsgType type)
108 {
109     QByteArray array;
110     //创建流 可以分段 参数1流入地址 参数2只能写
111     QDataStream stream(&array, QIODevice::WriteOnly);
112     stream<<type<<this->getName(); //流入类型和用户名
113     switch(type){
114         case Msg:
115             if(this->ui->msgTxtEdit->toPlainText()==""){
116                 QMessageBox::warning(this,"警告","聊天内容不能为空");
117                 return;
118             }
119             stream<<this->getMsg(); //发送的是聊天信息    发送格式    消息类型 +
            用户名    + 发送内容
120             break;
121         case UsrEnter://发送的是新用户进入    发送格式    消息类型 + 用户名
122             break;
123         case UsrLeft://用户离开
124             break;
125         default:
126             break;
127     }
128     //书写报文
129     udpSocket->writeDatagram(array.data(),array.size(),QHostAddress::Broadcast,this->port);
130 }
131
132 void Widget::usrEnter(QString username)
133 {
134     bool isEmpty = ui->usrTblWidget->findItems(username,Qt::MatchExactly).isEmpty();
135     if(isEmpty)

```

```

136     {
137         QTableWidgetItem *usr = new QTableWidgetItem(usrname);
138         //插入行
139         ui->usrTblWidget->insertRow(0);
140         ui->usrTblWidget->setItem(0,0,usr);
141         //追加聊天记录
142         ui->msgBrowser->setTextColor(Qt::gray);
143         ui->msgBrowser->setCurrentFont(QFont("Times New Roman",10));
144         //在线人数更新
145         ui->msgBrowser->append(tr("%1 在线! ").arg(usrname));
146         ui->usrNumLbl->setText(tr("在线人数: %1人").arg(ui->usrTblWidget-
>rowCount()));
147         //已经在线的各个端点也要告知新加入的端点他们自己的信息, 若不这样做, 在新端点用
户列表中就无法显示其他已经在线的用户
148         sndMsg(UsrEnter);
149     }
150 }
151
152 void Widget::usrLeft(QString username, QString time)
153 {
154     bool isEmpty = ui->usrTblWidget->findItems(username,
Qt::MatchExactly).isEmpty();
155     if(!isEmpty){
156         ui->usrTblWidget->removeRow(isEmpty);
157         //追加聊天记录
158         ui->msgBrowser->setTextColor(Qt::gray);
159         ui->msgBrowser->setCurrentFont(QFont("Times New Roman", 10));
160         ui->msgBrowser->append(QString("%1 于 %2 离
开! ").arg(username).arg(time));
161         ui->usrNumLbl->setText(QString("在线人数: %1人").arg(ui-
>usrTblWidget->rowCount()));
162     }
163 }
164
165 void Widget::ReceiveMessage()
166 {
167     qint64 size = udpSocket->pendingDatagramSize();
168     int mysize = static_cast<int>(size);
169     QByteArray array = QByteArray(mysize,0);
170     udpSocket->readDatagram(array.data(),size);
171     QDataStream stream(&array,QIODevice::ReadOnly);
172     int msgtype;
173     stream>>msgtype;
174     QString name,msg;
175     QString time = QDateTime::currentDateTime().toString("yyyy-MM-dd
hh:mm:ss");
176     switch(msgtype){
177     case Msg:
178         stream>>name>>msg;//流出用户名 聊天内容
179         ui->msgBrowser->setTextColor(Qt::blue);
180         ui->msgBrowser->setCurrentFont(QFont("Times New Roman",12));
181         ui->msgBrowser->append("[ " + name + " ]" + time);
182         ui->msgBrowser->append(msg);
183         break;
184     case UsrEnter:
185         stream>>name;
186         usrEnter(name);
187         break;

```

```

188         case UsrLeft:
189             stream>>name;
190             usrLeft(name,time);
191             break;
192     }
193 }
194
195 QString Widget::getName()
196 {
197     return this->myName;
198 }
199
200 QString Widget::getMsg()
201 {
202     QString msg = ui->msgTxtEdit->toHtml();
203     ui->msgTxtEdit->clear();
204     ui->msgTxtEdit->setFocus();
205     return msg;
206 }
207
208

```

## 5 创建登录界面

### 5.1 login.cpp

QStringList usrNameList; //账号 用来输入登录账号

QStringList passwordList; //登录密码

1. 点击登录按钮时，如果账号和密码匹配成功，则弹出信息对话框，则登录成功
2. 匹配失败，则弹出警告对话框，信息为账号密码错误，是否重新登录，该对话框中有一个按钮，点击ok则重新输入登录信息

```

1  #include "login.h"
2  #include "ui_login.h"
3  #include <QMessageBox>
4  #include "widget.h"
5  #include "fdform.h"
6
7
8
9  login::login(QWidget *parent) :
10      QWidget(parent),
11      ui(new Ui::login)
12  {
13      //setWindowFlags(Qt::WindowType::FramelessWindowHint);
14
15      ui->setupUi(this);
16      ui->username->lineEdit()->setPlaceholderText("QQ号码/手机/邮箱");
17      this->setWindowIcon(QIcon(":/img/QQtubiao.png"));
18      this->setWindowTitle("QQ 2023");
19
20
21  }

```

```

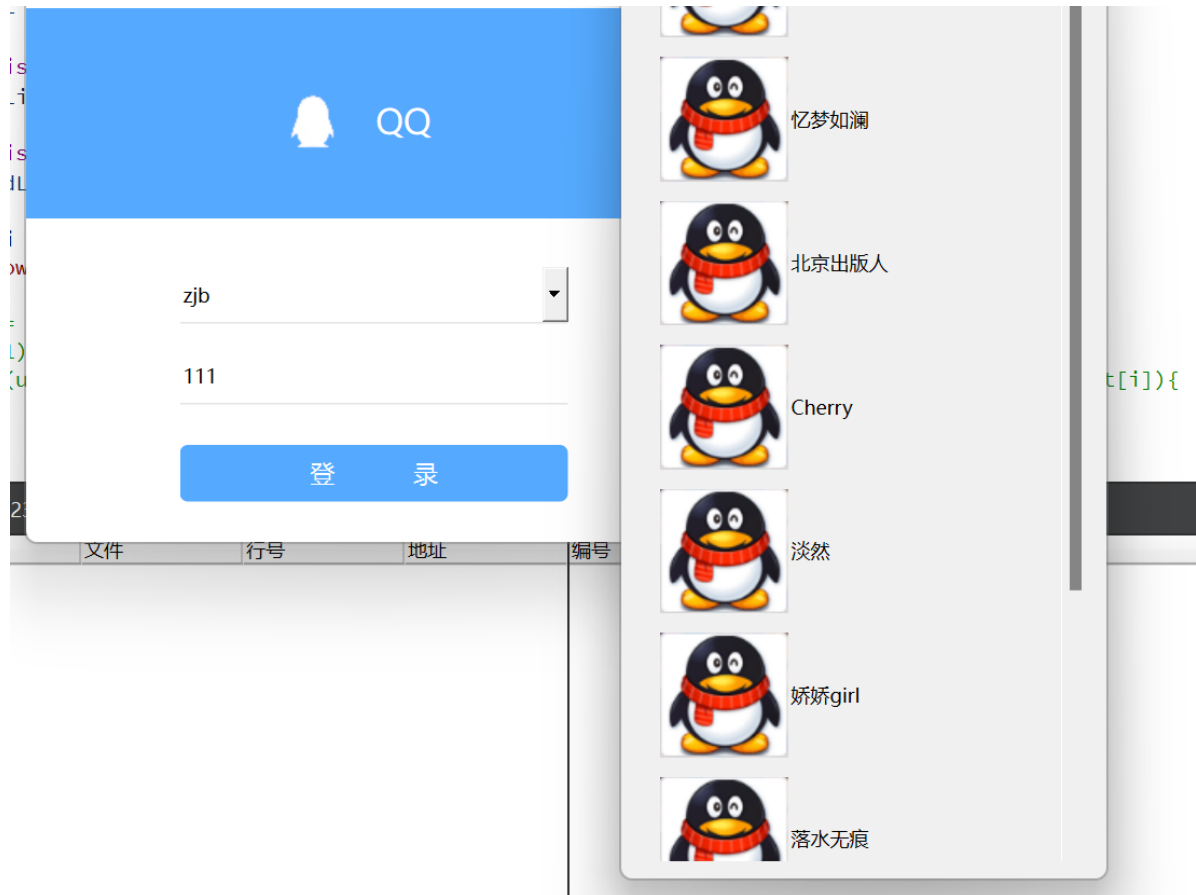
22
23 login::~login()
24 {
25     delete ui;
26 }
27 void login::on_btnClose_clicked()
28 {
29     this->close();
30 }
31
32
33 void login::on_btnMin_clicked()
34 {
35     this->showMinimized();
36 }
37
38
39 void login::on_btnLogin_released()
40 {
41     QList<QString> nameList;
42     nameList << "周骏博" << "邓卓文" << "徐绍倡" << "潘星宇" << "淡然"
43         << "娇娇girl" << "落水无痕" << "青墨暖暖" << "无语";
44     QStringList usrNameList; //账号
45     usrNameList << "zjb" << "dzw" << "xsc" << "pxy" << "qq"
46         << "qq" << "qq" << "qq" << "qq";
47     QStringList passwordList; //密码
48     passwordList << "111" << "222" << "333" << "444" << "qq"
49         << "qq" << "qq" << "qq" << "qq";
50     for(int i = 0; i < 4; i++){
51         isShow.push_back(false);
52     }
53     // int i = 0;
54     // while(1){
55     //     if(ui->username->currentText() == usrNameList[i] && ui->password-
56     >text() == passwordList[i]){
57         //     }
58         // }
59         int flag = 0;
60         for(int i = 0; i < 5; i++){
61
62             if(ui->username->currentText() == usrNameList[i] && ui->password-
63 >text() == passwordList[i]){
64                 if(isShow[i]){
65                     QString str = QString("%1已登录").arg(nameList[i]);
66                     QMessageBox::warning(this, "警告", str);
67                     return;
68                 }
69                 isShow[i] = true;
70                 flag = 1;
71                 fdform * fdlist = new fdform(0, nameList[i]);
72                 //widget * widget = new widget(0, usrNameList[i]);
73                 fdlist->setWindowTitle(usrNameList[i]);
74                 QString str = QString(":/img/%1.jpg").arg(usrNameList.at(i));
75                 fdlist->setWindowIcon(QPixmap(str));
76                 fdlist->show();
77                 //widget->show();
78                 connect(fdlist, &fdform::closeFdform, [=]() {

```

```

78         isShow[i] = false;
79     });
80 }
81 else if(i > 3 && flag == 0){
82     QMessageBox::warning(this,"提示","账号或密码错误");
83 }
84 }
85 }
86
87

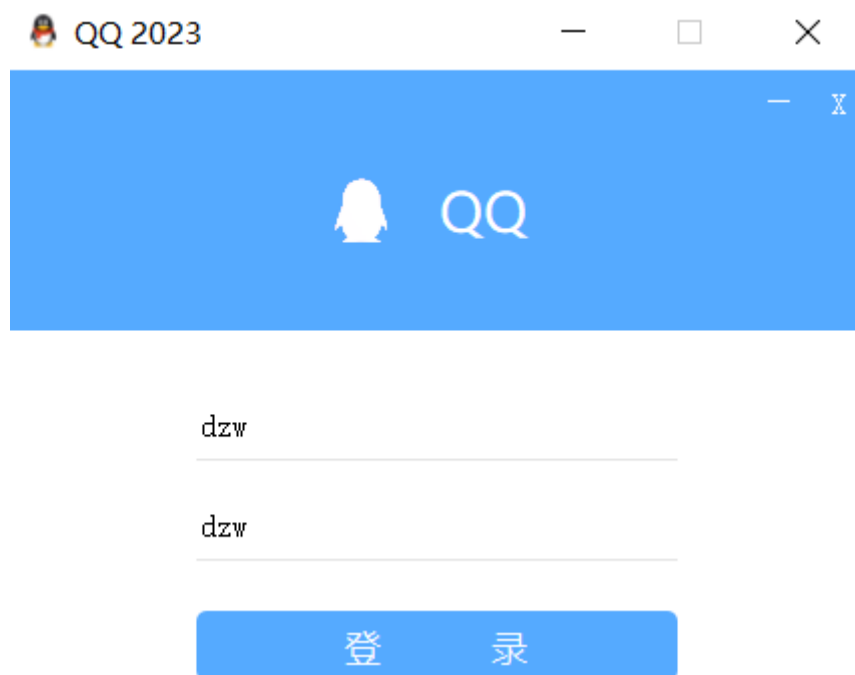
```





## 6 项目成果展示

### 6.1 登录QQ账户



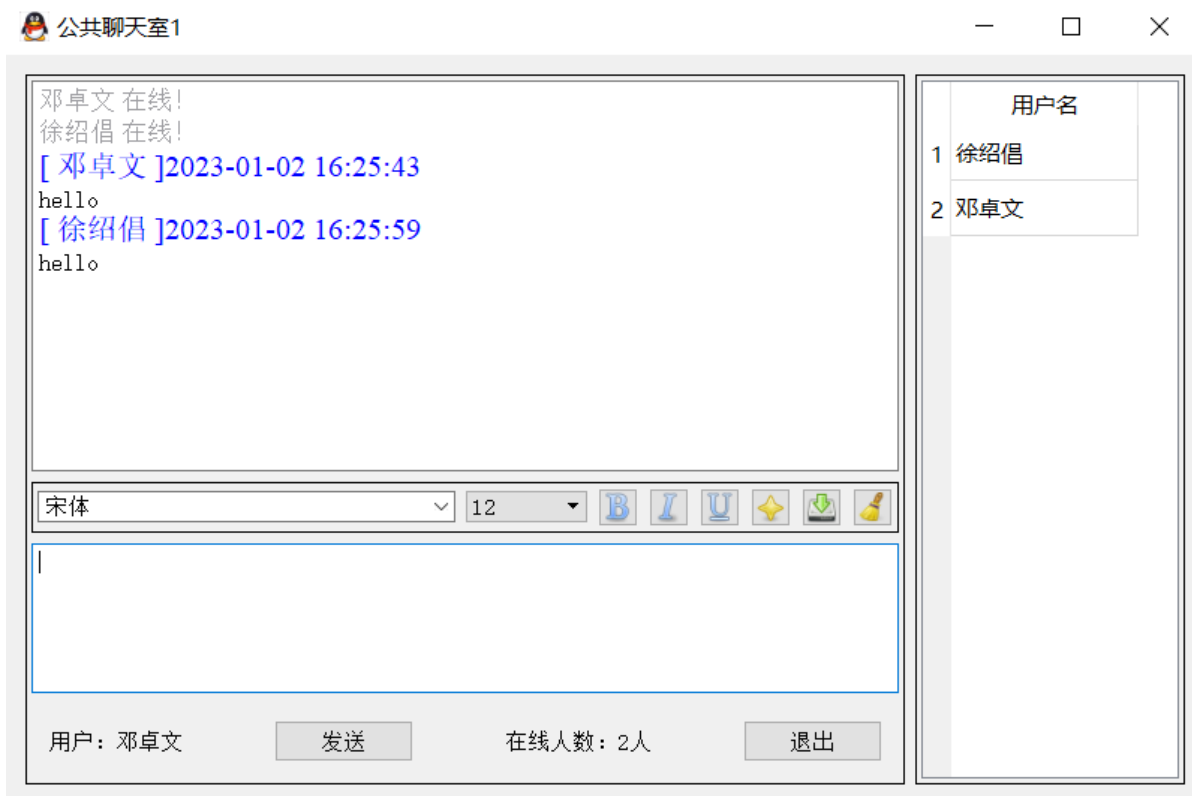


## 6.2 进入对话列表

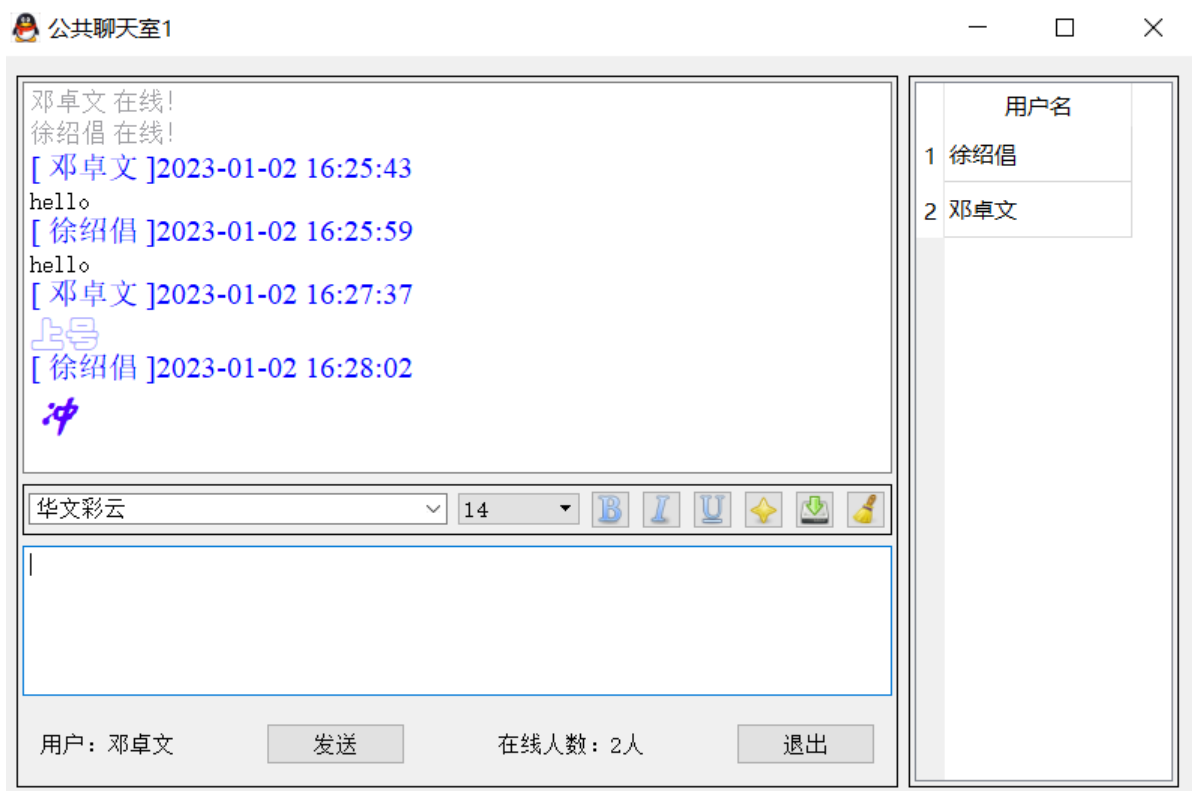


## 6.3 进入聊天窗口

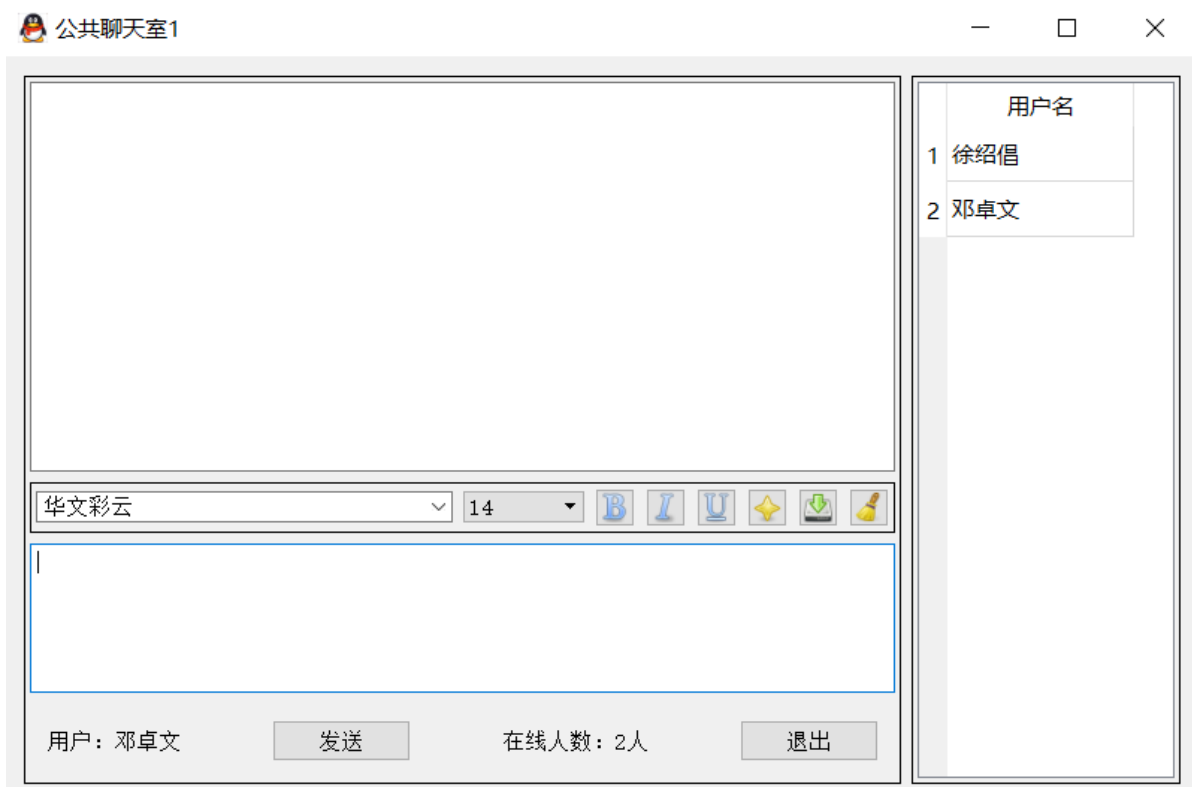
### 6.3.1 进行聊天



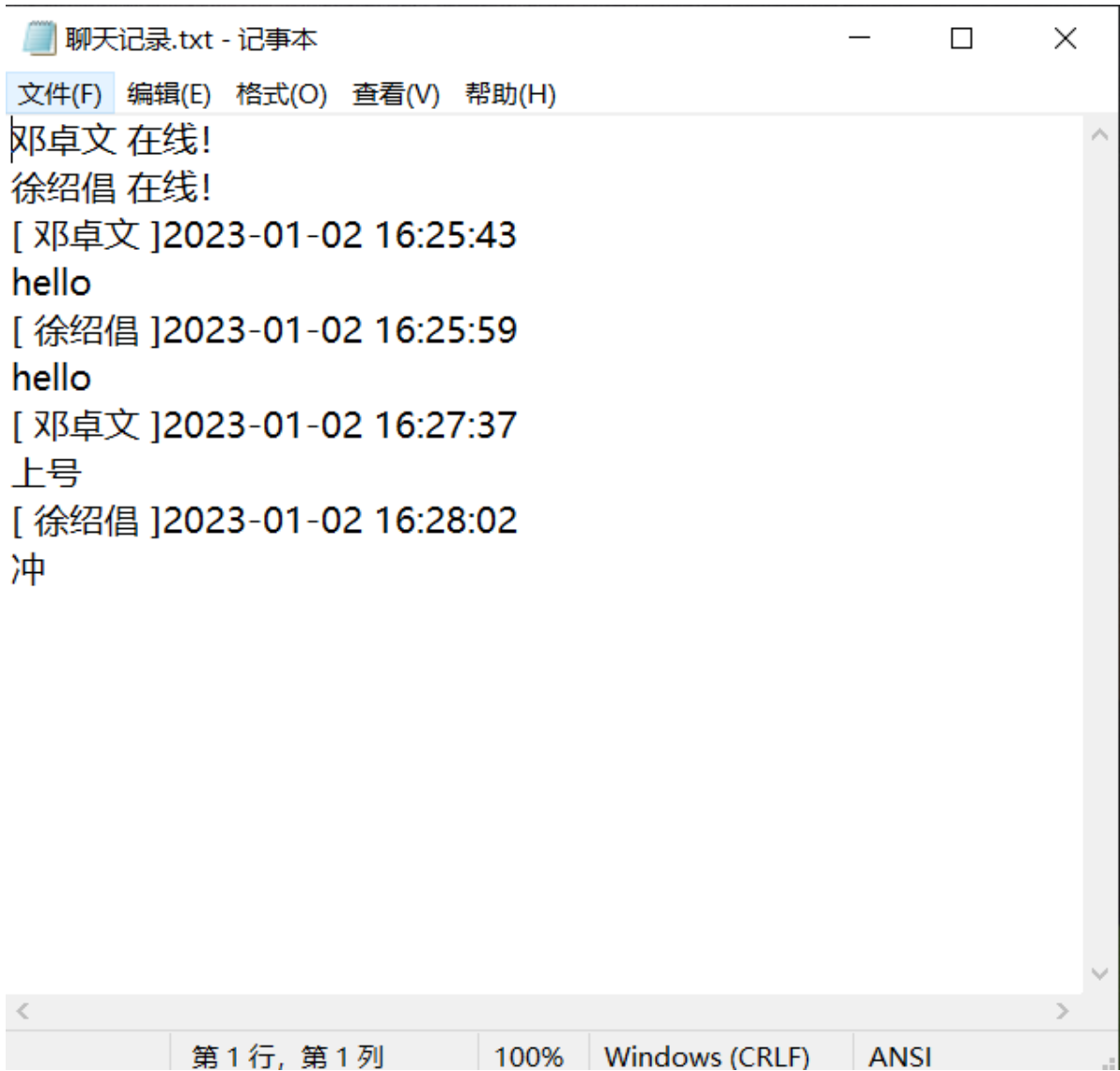
### 6.3.2 更改字体类型、大小、加粗、颜色等



### 6.3.3 删除聊天记录



#### 6.4 保存聊天记录



## 7 实验收获

由人的关注特点，在设计人机交互界面时应做到：信息的显示应醒目，以便执行任务时使用，可使用动画图形、彩色、下划线，对条目及不同的信息进行排序，在条目之间使用间隔符等。避免在界面上安排过多的信息。尤其要谨慎使用色彩、声音和图像，人们倾向于使用过多的这类表示，而导致界面混杂，分散用户的注意力，让用户反感。朴实的界面更容易使用。

一个交互式系统在设计时，应该开发一个易于用户理解的系统映像，应该做到及时响应用户的输入并给出有用的反馈；提供易于理解的、直观的反馈信息。此外，一个好的交互系统还需要提供正确的信息类型以及正确的信息层次，以针对不同层次的用户，提供不同层次的系统透明度。这方面包括：①有条理的、易于理解的说明②合适的在线帮助和自学教程上下文③相关的用户指南，即，针对不同层次的用户，解释在不同的任务阶段应如何处理各种情况。