小组项目报告

<table>
<tr><td>项目名称</td><td colspan="6">ChatMore 聊天工具</td></tr>
<tr><td>完成时间</td><td>2018.7.18</td><td>指导教师</td><td colspan="4">于红</td></tr>
<tr><td>小组自评
（A/B/C）</td><td></td><td>其他
说明</td><td colspan="4"></td></tr>
<tr><td rowspan="2">小组成员
（姓名学号）</td><td rowspan="2">成员分工
（角色及任务）</td><td rowspan="2">自评
A/B/C</td><td colspan="4">个人成绩</td></tr>
<tr><td>3-③</td><td>4-③</td><td>5-②</td><td>总分</td></tr>
<tr><td>田健鲲
201673010</td><td>角色：组长
任务：UI 界面编程以及聊天机器人的网络编程</td><td></td><td>20</td><td>20</td><td>10</td><td>50</td></tr>
<tr><td>于靖达
201673011</td><td>角色：组员
任务：UDP/TCP 网络编程，实现局域网聊天与文件上传功能。</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

大连理工大学软件学院

评分细则及标准

考察项目	总分	评分细则	分数	得分
问题规模	10分	创新超额完成指定任务，工作量饱满	9-10分	
		基本完成指定任务，工作量一般	6-8分	
		少量指定任务未完成，工作量不足	3-5分	
		大量指定任务未完成，工作量明显不足	0-2分	
技术难度	10分	模型设计合理优化	9-10分	
		模型设计基本正确	6-8分	
		模型设计存在一些问题	3-5分	
		模型设计存在很多问题	0-2分	
实现程度	10分	系统实现完整，界面友好，测试全面，没有错误	9-10分	
		系统实现完整，界面友好，存在少许错误	6-8分	
		系统实现不够完整，界面不够友好，存在一些错误	3-5分	
		系统实现不够完整，界面不够友好，存在明显错误	0-2分	
报告质量	10分	报告完整、格式统一、结构清晰、图表正确	9-10分	
		报告完整、格式较为统一、结构较为清晰、图表基本正确	6-8分	
		报告完整、格式有问题、结构不够清晰、图表有错误	3-5分	
		报告不完整、格式问题较多、结构较混乱、图表错误较多	0-2分	
个人工作	10分	团队核心成员，积极参与项目设计与实现	9-10分	
		团队主要成员，参与项目部分的设计或实现	6-8分	
		团队其他成员，参与项目不够积极	3-5分	
		基本不参与项目的成员	0-2分	

目　　录

第一部分　学习内容及总结 3
1 田健鲲学习内容及体会 3
2 于靖达学习内容及体会 4
第二部分 项目开发报告 5
1 项目简介 5
2 需求分析 5
3 总体设计 7
3.1　系统类图 7
3.1.1　主要类的类图 7
3.1.2　各个类的详细属性 7
3.2　功能模块图 11
4 详细设计与实现 13
4.1　登陆模块 13
5.1.1　程序流程图 13
5.1.2　模块设计文字说明 13
4.1.3　核心界面截图 14
4.2　多人聊天模块 15
4.2.1　程序流程图 15
4.2.2　模块设计文字说明 16
4.2.3　核心界面截图 17
4.3　双人私聊模块 18
4.3.1 程序流程图 18
4.3.2 模块设计文字说明 19
4.3.3 核心界面截图 19
4.4　文件传输模块 20
4.4.1 程序流程图 20
4.4.2 模块设计文字说明 21
4.4.3 核心界面截图 21
5　系统测试 22
5.1 登录模块 22
5.2 聊天室模块 22

5.3 文件收发模块 23
5.4 私聊模块 23
6 设计总结 24
6.1 项目总体完成情况 24
6.2 技术难点 24
6.3 解决方法 24
6.4 不足 25

第一部分　学习内容及总结

（要求：每个组员单独撰写，体现每个阶段学习内容、安排及收获，每人 1-2 页，不少于 1000 字）

1 田健鲲学习内容及体会

（1）　第一阶段：完成 QT 的基本学习，确定项目课题，明确项目难点。懂得了 QT 的设计思想，了解了界面编程的基本流程，熟悉了基本常用类。根据团队成员的兴趣点和技能，最终确定项目为聊天软件。难点为网络编程（UDP, TCP）方面。学习收获：在短暂的时间内，快速了解了一个工具的使用，并且能够从一个项目中，根据自己的知识，明确项目难点，开始资料收集与进一步的学习。懂得了不重复造轮子的道理，要学会利用前人的成果，去创造更大的成就

（2）　第二阶段：需求分析，功能模块划分。明确每个人的任务安排。功能模块基本确定为：界面（登录，群聊，私聊），登录模块，群聊模块（有文件上传，消息文字形式的更改等扩展功能，也为其他可扩展功能预留了按钮），私聊模块。我负责界面模块、登录模块以及私聊模块中的聊天机器人功能。学习收获：懂得了团队分工合作，模块化任务划分的本领，知道了在面临一个大的任务时，如何划分任务，达成有效的合作。通过对常用软件的分析，不仅仅模仿，更是加入了扩展功能，比如聊天机器人的私聊功能。

（3）　第三阶段：完成个人任务。通过在网上收集相关资料，查看官方文档，网上爬取图片，学会使用 photoshop 等绘图工具，调用网络 API，加强了解决问题的能力，进一步培养了对于软件界面的审美，学会了解决问题，在互联网上寻求解决方案的方法。在解决问题的过程中，尤其是 ui 编程模块，加深了对于 QT 常用类以及内部模式的了解，如对于样式表以及 designer 的使用。

（4）　第四阶段：代码融合并进行系统测试。两个人根据之前的模块划分，进行代码融合，如相关变量的统一设置等，对于不兼容的部分，进行取舍。进一步进行代码重构，使得代码清晰，更加具有可扩展性，添加注释，方便进一步修改。学习收获：体会到了规范制定的重要性，会极大地方便团队协作，避免之后不知名的麻烦。同时，懂得了代码注释的重要性，尤其是面临大项目时。对于软件设计模式有了更加深刻的认识，懂得了代码重构的重要性。系统测试：根据软件的基本功能，设计测试样例，并且考虑了用户的异常操作，重点对网络部分，即文件上传和聊天功能进行了详尽的测试，进一步加强了程序的鲁棒性。

（5） 第五阶段：撰写文档，录制视频和制作 PPT。根据每个人的任务划分模块，分别撰写文档相关部分，需求分析、总体设计等部分，经过商讨后共同撰写。根据软件具有的基本功能和扩展功能，录制少于两分钟的视频用于展示。制作 PPT 用于答辩。收获：明白了一个项目，程序重要，但是功能的展示，界面，易用性，用户友好性更加重要。

2 于靖达学习内容及体会

（1） 第一阶段，为期三天，从一个在线教程入手 QT，开发一个简单的记事本软件，习得 QT 的基本操作，以及界面的基本开发流程。“纸上得来终觉浅，绝知此事要躬行”，只有从实际的项目开始编程，而不是从最基础的理论开始，才能在短时间内习得实用的技能。

（2） 第二阶段，为期一天，小组内商讨项目类型，我提出做输入法，组长提出做聊天软件，我们各自分析利弊，发现输入法的难点在于数据库后端的处理以及存储，以我们现阶段的算法知识和理论很难实现，最终选择了做通讯软件，除了基本功能外，还增加了聊天机器人的陪聊功能，并且难点在于网络编程，可以通过搜索文档以及从博客上搜索信息得到解决方案。最终确定项目为聊天工具，并且我负责网络编程模块，组长负责界面设计以及总体规划。

（3） 第三阶段，开始根据任务搜集资料。首先，我根据示例代码，在 QT 上运行，逐行理解，最终理解了网络编程的机制，明确了 UDP 和 TCP 的区别，还有单播、组播、广播的区别以及应用场景。之后，开始搭建局域网，不仅尝试了宿主机和虚拟机之间构成局域网，也尝试了不同主机之间通过同一路由器或者手机热点搭建局域网。这一阶段，我对于虚拟机的网络模式和局域网有了更深的理解，为下一阶段奠定了厚实的基础。

（4） 第四阶段，小组内协调类的设计和功能模块的划分，在背景知识具备的情况下，开始真正做项目。由于界面部分由另一人负责，我搭建了简陋的测试环境，用于网络编程，最终确定用 UDP 广播实现群聊功能，UDP 单播实现私聊功能，TCP 协议实现文件收发功能，并且实践成功。

（5） 第五阶段，代码的融合与重构。我的网络相关的逻辑与另一人的界面进行融合重构。本以为很简单，结果却有变量命名和编写习惯等问题，使融合变得有些困难，但最终实现了目标。从中体会到了规范的重要性，虽然代码本身不会影响最终的结果，但是好的代码却会方便之后的维护。

（6） 第六阶段，集思广益进行系统测试样例的收集，主要针对用户的反常规操作，鼠标乱点等行为。最常见的错误就是网络错误，在这一阶段，锻炼了我的单步调试能力，面对异常的未定义行为，只能通过逻辑去分析，不能凭感觉判断，而是要通过错误日志，局部单元测试进行调试。

总之，这次项目实践，不仅让我更加了解了虚拟机，网络编程，QT 界面编程，更加珍贵的是，让我体会到了团队的强大，团队协作的重要性，以及团队协作的规范制定的必要性。只有通过团队协作，才能在更短的时间内实现更加强大的功能，好队友很重要，好的团队制度也很重要。

第二部分 项目开发报告

1 项目简介

项目名称为 ChatMore，类型为通讯软件。使用软件前，需要输入产品密钥，默认用户名为计算机名。在同一局域网内的用户，可以进入聊天室，开始群聊，提供给指定用户传输文件，更改字体，保存聊天记录等功能。在聊天室中，选定指定用户进入私聊模式。私聊模式中，提供聊天机器人，可以进行百科式问答或者闲聊。本软件是局域网内群聊私聊软件，在联网状态下，可以实现与聊天机器人闲聊功能。界面美观，音效恰当，是一款实用的局域网通信软件。

2 需求分析

本聊天软件的功能需求如下：

1. 登陆部分

（1） 软件具有密钥功能，必须输对密钥才能进入主界面。

（2） 显示用户信息，以及系统时间。

2. 进行多人聊天：本软件事实上以多人聊天为基础功能，为处于同一个局域网中的所有用户搭建起一个多人聊天室。

（1） 用户可以接收同一局域网内其他用户所发送的信息。

（2） 用户可以向同一局域网内的其他用户发送信息。

（3） 当有用户进入或者退出聊天室时会出现消息提示。

（4） 用户可以看到当前使用此软件且与其处于同一局域网内的其他用户的用户名、主机名、IP 地址。

（5） 用户可以对于自己所发送消息的文字进行设置，比如设置字号、字体、颜色、加粗等等。

（6） 在聊天的时候用户可以播放音乐。

（7） 当消息发送的时候会有发送的提示音，当有新消息的时候会有接受的提示音。

（8） 可以对于聊天记录进行保存。

（9） 可以清除当前屏幕的全部信息。

3. 与其他用户进行私聊：当用户处于群聊页面时，可以选择一个用户进行私聊

（1） 用户可以与任意一个用户进行私聊

（2） 用户可以接收或者发送私聊信息

（3） 当有用户退出私聊时会出现消息提示

（4） 用户可以对于自己所发送消息的文字进行设置，比如设置字号、字体、颜色、加粗等等

（5） 当消息发送的时候会有发送的提示音，当有新消息的时候会有接受的提示音

（6） 可以对于聊天记录进行保存。

（7） 可以清除当前屏幕的全部信息。

4. 用户之间可以进行文件的传输

（1） 在群聊的时候，用户可以选择一个在线的用户并向其发送文件。

（2） 在私聊的时候，用户可以向对方发送文件。

（3） 当发现有用户向自己传输文件时，会进行提示，并决定是否接受

（4） 可以显示文件传输的进度，文件发送的速度，已经消耗的时间等参数。

（5） 双方可以随时决定停止文件的传输

5. 聊天机器人

（1） 用户在群聊界面中选择聊天机器人，进入与机器人的私聊模式。

（2） 在私聊界面中，用户发送日常用语，机器人回复相应的文字，合情合理，并且同样的消息也会收到不同的答复。

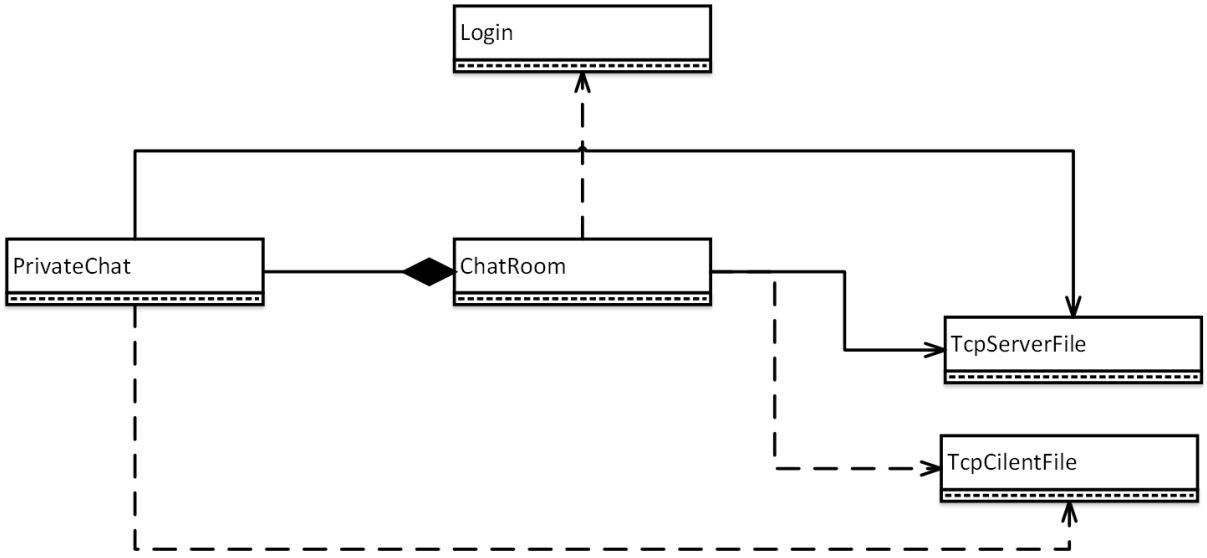
（3） 用户发送名人的名字或者专有名词，聊天机器人会进行百科式回复，向用户科普知识，类似于知识图谱功能。

（4） 用户输入“成语接龙”，会进行成语接龙游戏

3 总体设计

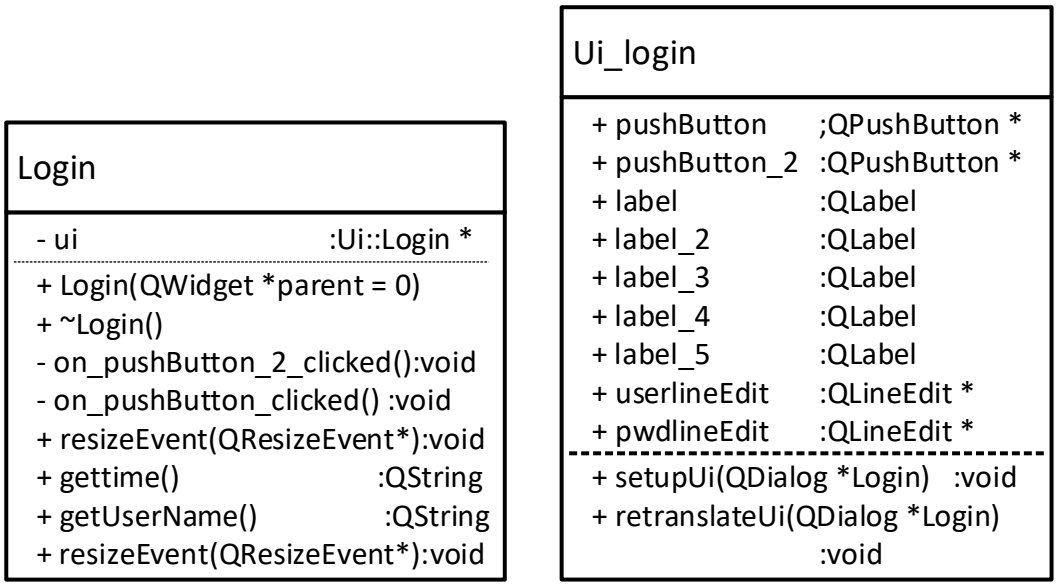
3.1 系统类图

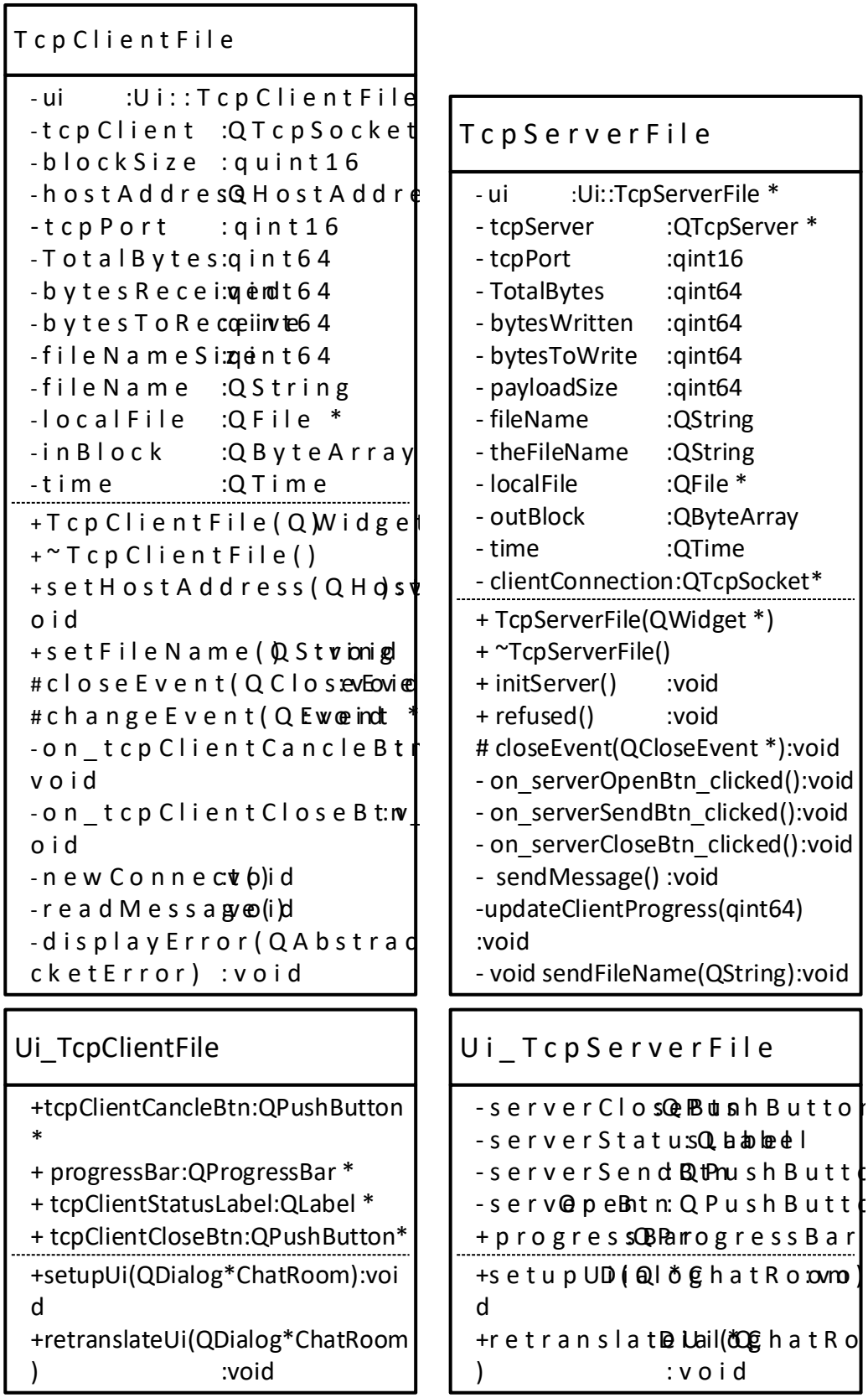
3.1.1 主要类的类图

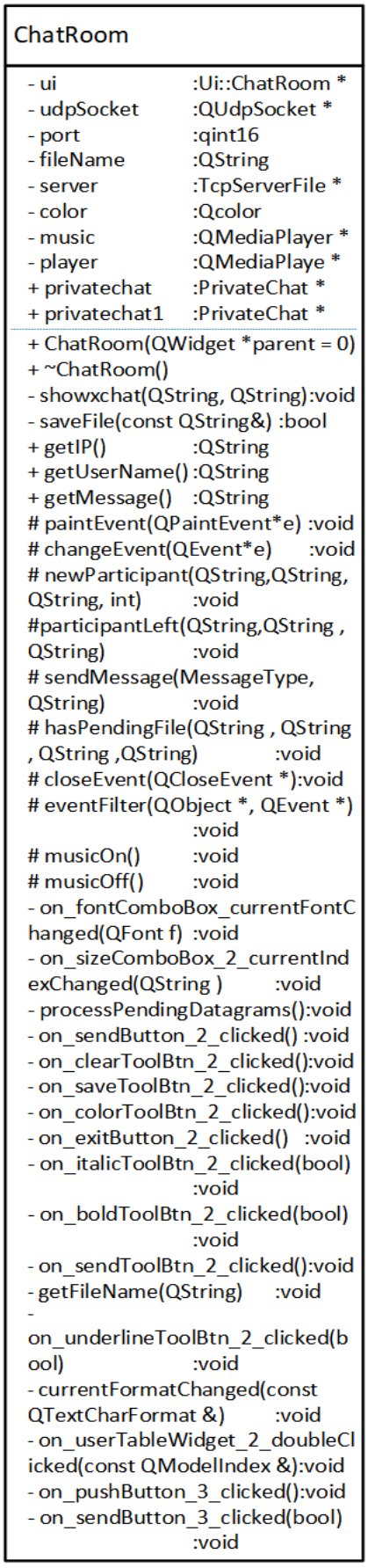


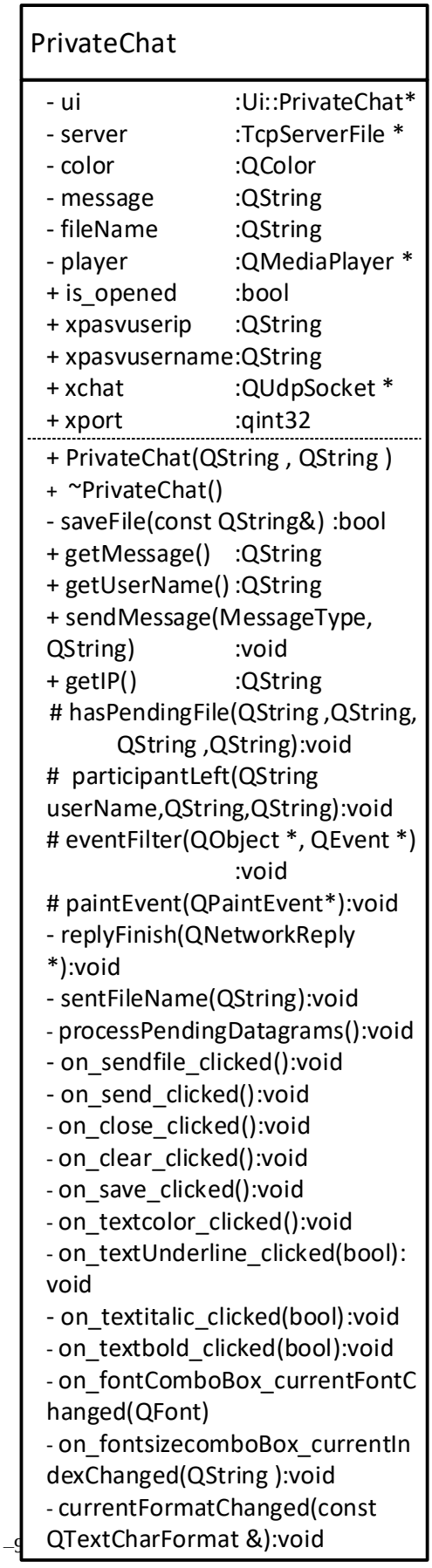
3.1.2 各个类的详细属性

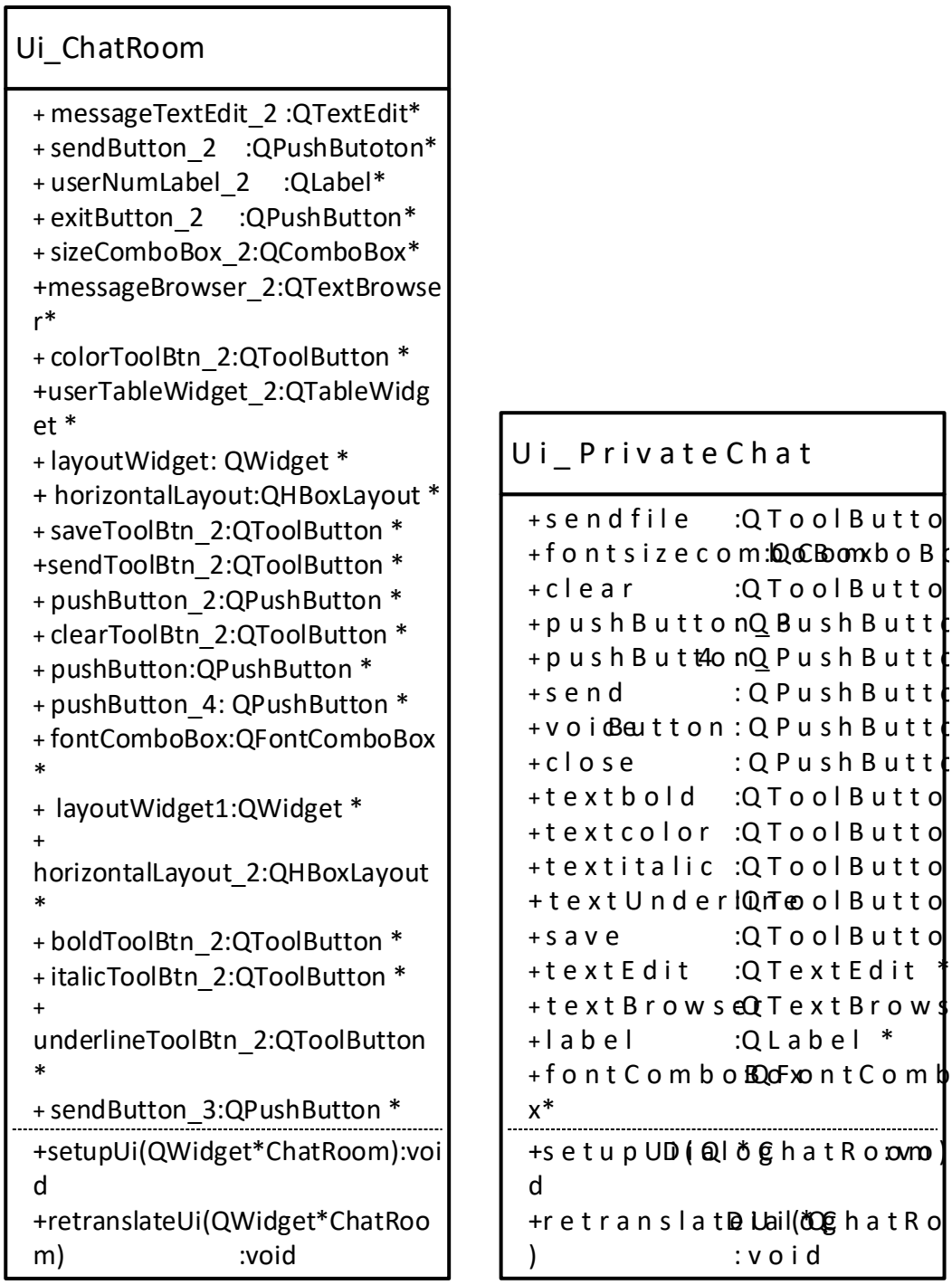
本聊天软件项目的核心功能的类就是在上图中的五个类。而使用 QT 时系统会自动为新建立的界面创建出相应的 Ui 类，因此在这里我也列举出了与核心类相关的 QT 自动创建的五个 Ui 类。




TcpClientFile
-ui :Ui::TcpClientFile
-tcpClient :QTcpSocket
-blockSize :quint16
-hostAddress:QHostAddre
-tcpPort :qint16
-TotalBytes:qint64
-bytesReceived:qint64
-bytesToReceive:qint64
-fileNameSize:qint64
-fileName :QString
-localFile :QFile *
-inBlock :QByteArray
-time :QTime
+TcpClientFile(QWidget
+~TcpClientFile()
+setHostAddress(QHostAddress)
void
+setFileName(QString)
#closeEvent(QCloseEvent
#changeEvent(QEvent *)
-on_tcpClientCancleBtn_
void
-on_tcpClientCloseBtn_
oid
-newConnect()
-readMessage()
-displayError(QAbstrac
cketError) :void
TcpServerFile
-ui :Ui::TcpServerFile *
-tcpServer :QTcpServer *
-tcpPort :qint16
-TotalBytes :qint64
-bytesWritten :qint64
-bytesToWrite :qint64
-payloadSize :qint64
-fileName :QString
-theFileName :QString
-localFile :QFile *
-outBlock :QByteArray
-time :QTime
-clientConnection:QTcpSocket*
+TcpServerFile(QWidget *)
+~TcpServerFile()
+initServer() :void
+refused() :void
#closeEvent(QCloseEvent *):void
-on_serverOpenBtn_clicked():void
-on_serverSendBtn_clicked():void
-on_serverCloseBtn_clicked():void
- sendMessage() :void
-updateClientProgress(qint64)
:void
-void sendFileName(QString):void
Ui_TcpClientFile
+tcpClientCancleBtn:QPushButton
*
+progressBar:QProgressBar *
+tcpClientStatusLabel:QLabel *
+tcpClientCloseBtn:QPushButton*
+setupUi(QDialog*ChatRoom):voi
d
+retranslateUi(QDialog*ChatRoom
) :void
Ui_TcpServerFile
-serverCloseBtn:QPushButton
-serverStatusLabel:QLabel
-serverSendBtn:QPushButt
-serverOpenBtn:QPushButt
+progressBar:QProgressBar
+setupUi(QDialog*ChatRoom):vo
d
+retranslateUi(QDialog*ChatRo
) :void

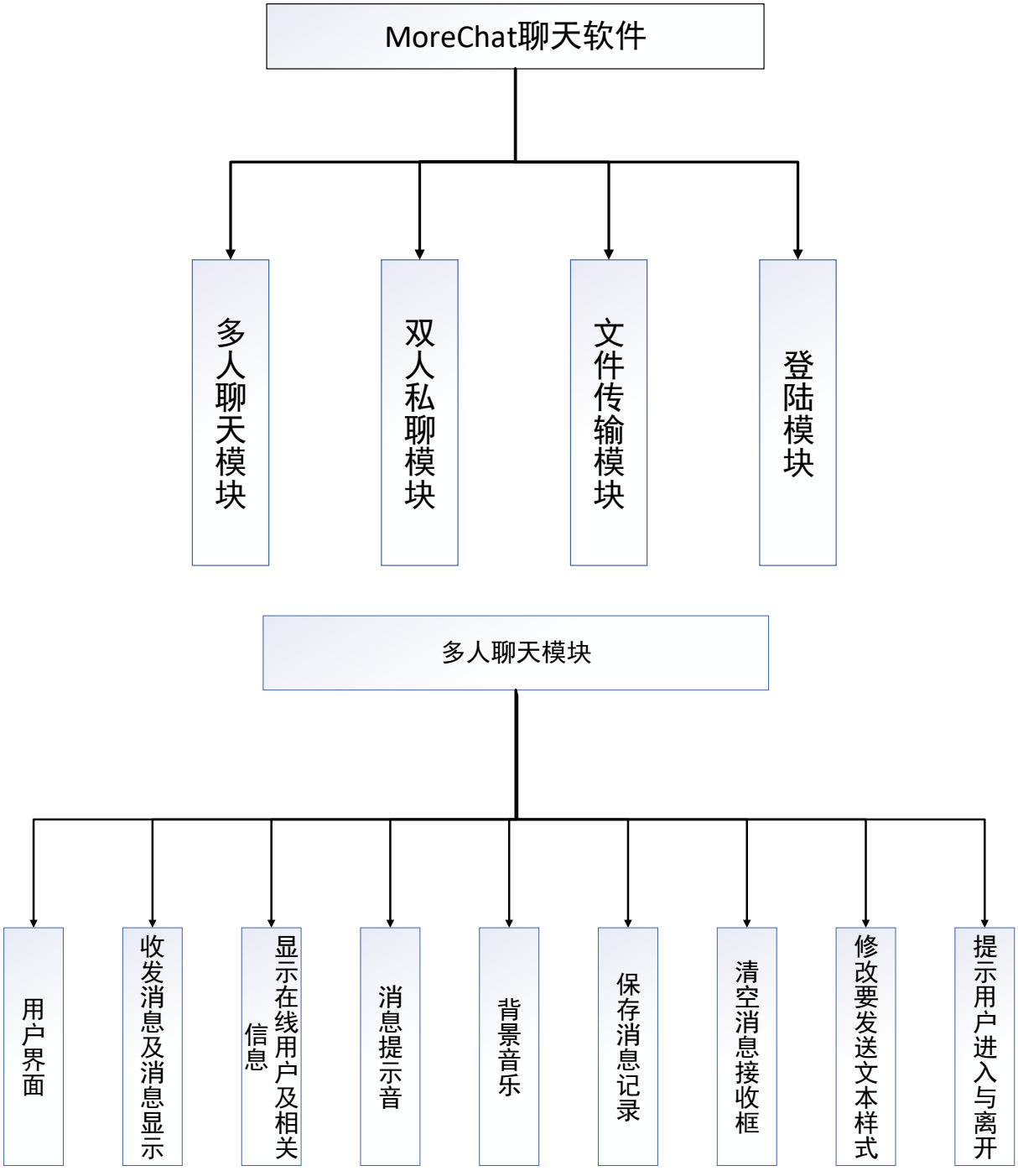

ChatRoom
- ui :Ui::ChatRoom *
- udpSocket :QUdpSocket *
- port :qint16
- fileName :QString
- server :TcpServerFile *
- color :Qcolor
- music :QMediaPlayer *
- player :QMediaPlaye *
+ privatechat :PrivateChat *
+ privatechat1 :PrivateChat *
+ ChatRoom(QWidget *parent = 0)
+ ~ChatRoom()
- showxchat(QString, QString):void
- saveFile(const QString&) :bool
+ getIP() :QString
+ getUserName() :QString
+ getMessage() :QString
# paintEvent(QPaintEvent*e) :void
# changeEvent(QEvent*e) :void
# newParticipant(QString,QString, QString, int) :void
#participantLeft(QString,QString , QString) :void
sendMessage(MessageType, QString) :void
hasPendingFile(QString , QString , QString ,QString) :void
closeEvent(QCloseEvent *):void
eventFilter(QObject *, QEvent *) :void
musicOn() :void
musicOff() :void
- on_fontComboBox_currentFontChanged(QFont f) :void
- on_sizeComboBox_2_currentIndexChanged(QString) :void
- processPendingDatagrams():void
- on_sendButton_2_clicked() :void
- on_clearToolBtn_2_clicked():void
- on_saveToolBtn_2_clicked():void
- on_colorToolBtn_2_clicked():void
- on_exitButton_2_clicked() :void
- on_italicToolBtn_2_clicked(bool) :void
- on_boldToolBtn_2_clicked(bool) :void
- on_sendToolBtn_2_clicked():void
- getFileName(QString) :void
- on_underlineToolBtn_2_clicked(bool) :void
- currentFormatChanged(const QTextCharFormat &) :void
- on_userTableWidget_2_doubleClicked(const QModelIndex &):void
- on_pushButton_3_clicked():void
- on_sendButton_3_clicked(bool) :void

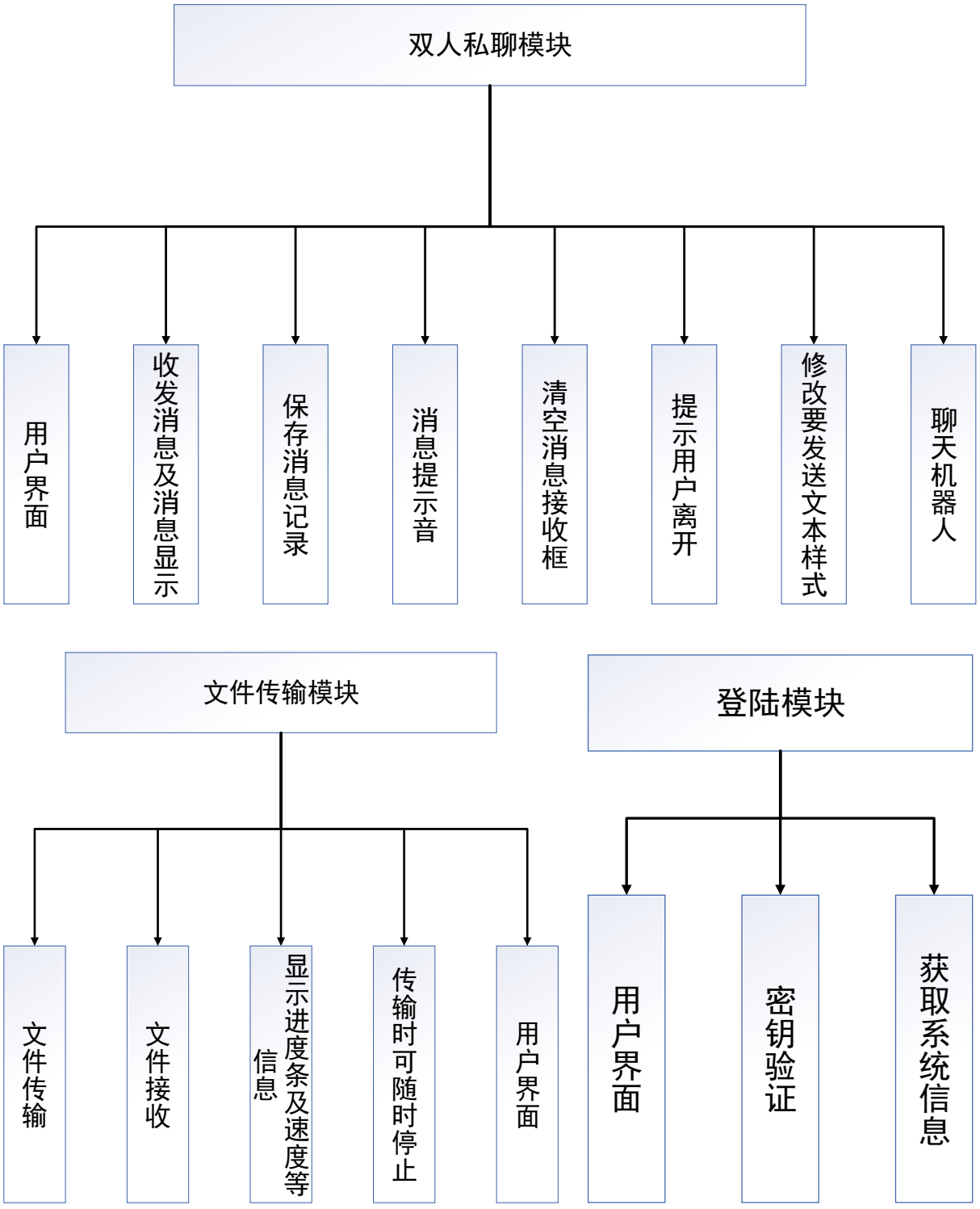

PrivateChat
- ui :Ui::PrivateChat*
- server :TcpServerFile *
- color :QColor
- message :QString
- fileName :QString
- player :QMediaPlayer *
+ is_opened :bool
+ xpasvuserip :QString
+ xpasvusername:QString
+ xchat :QUdpSocket *
+ xport :qint32
+ PrivateChat(QString , QString)
+ ~PrivateChat()
- saveFile(const QString&) :bool
+ getMessage() :QString
+ getUserName() :QString
+ sendMessage(MessageType, QString) :void
+ getIP() :QString
# hasPendingFile(QString ,QString, QString ,QString):void
# participantLeft(QString userName,QString,QString):void
# eventFilter(QObject *, QEvent *) :void
# paintEvent(QPaintEvent*):void
- replyFinish(QNetworkReply *):void
- sentFileName(QString):void
- processPendingDatagrams():void
- on_sendfile_clicked():void
- on_send_clicked():void
- on_close_clicked():void
- on_clear_clicked():void
- on_save_clicked():void
- on_textcolor_clicked():void
- on_textUnderline_clicked(bool): void
- on_textitalic_clicked(bool):void
- on_textbold_clicked(bool):void
- on_fontComboBox_currentFontChanged(QFont)
- on_fontsizecomboBox_currentIndexChanged(QString):void
- currentFormatChanged(const QTextCharFormat &):void


Ui_ChatRoom
+ messageTextEdit_2 :QTextEdit*
+ sendButton_2 :QPushButoton*
+ userNumLabel_2 :QLabel*
+ exitButton_2 :QPushButton*
+ sizeComboBox_2:QComboBox*
+messageBrowser_2:QTextBrowser*
+ colorToolBtn_2:QToolButton *
+userTableWidget_2:QTableWidget *
+ layoutWidget: QWidget *
+ horizontalLayout:QHBoxLayout *
+ saveToolBtn_2:QToolButton *
+sendToolBtn_2:QToolButton *
+ pushButton_2:QPushButton *
+ clearToolBtn_2:QToolButton *
+ pushButton:QPushButton *
+ pushButton_4: QPushButton *
+ fontComboBox:QFontComboBox *
+ layoutWidget1:QWidget *
+ horizontalLayout_2:QHBoxLayout *
+ boldToolBtn_2:QToolButton *
+ italicToolBtn_2:QToolButton *
+ underlineToolBtn_2:QToolButton *
+ sendButton_3:QPushButton *
+setupUi(QWidget*ChatRoom):void
+retranslateUi(QWidget*ChatRoom) :void
Ui_PrivateChat
+sendfile :QToolButto
+fontsizecombox :QComboBox
+clear :QToolButto
+pushButton_3 :QPushButt
+pushButton_4 :QPushButt
+send :QPushButt
+voiceButton :QPushButt
+close :QPushButt
+textbold :QToolButto
+textcolor :QToolButto
+textitalic :QToolButto
+textUnderline :QToolButto
+save :QToolButto
+textEdit :QTextEdit *
+textBrowser :QTextBrows
+label :QLabel *
+fontComboBox :QFontComb
x*
+setupUI(QDialog *ChatRoom) :void
+retranslateUi(QDialog *ChatRoom) :void

3.2 功能模块图

总模块图以及各个子模块

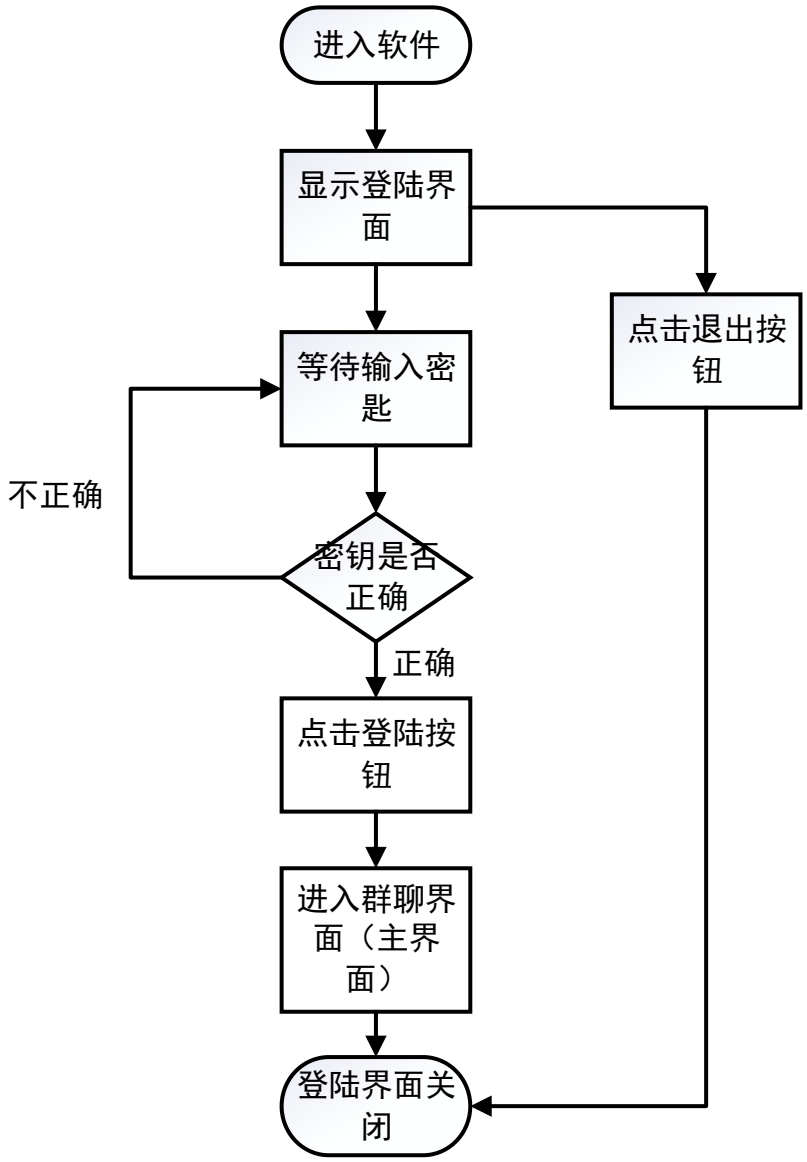



双人私聊模块
用户界面
收发消息及消息显示
保存消息记录
消息提示音
清空消息接收框
提示用户离开
修改要发送文本样式
聊天机器人
文件传输模块
文件传输
文件接收
显示进度条及速度等信息
传输时可随时停止
用户界面
登陆模块
用户界面
密钥验证
获取系统信息

4 详细设计与实现

4.1 登陆模块

4.1.1 程序流程图



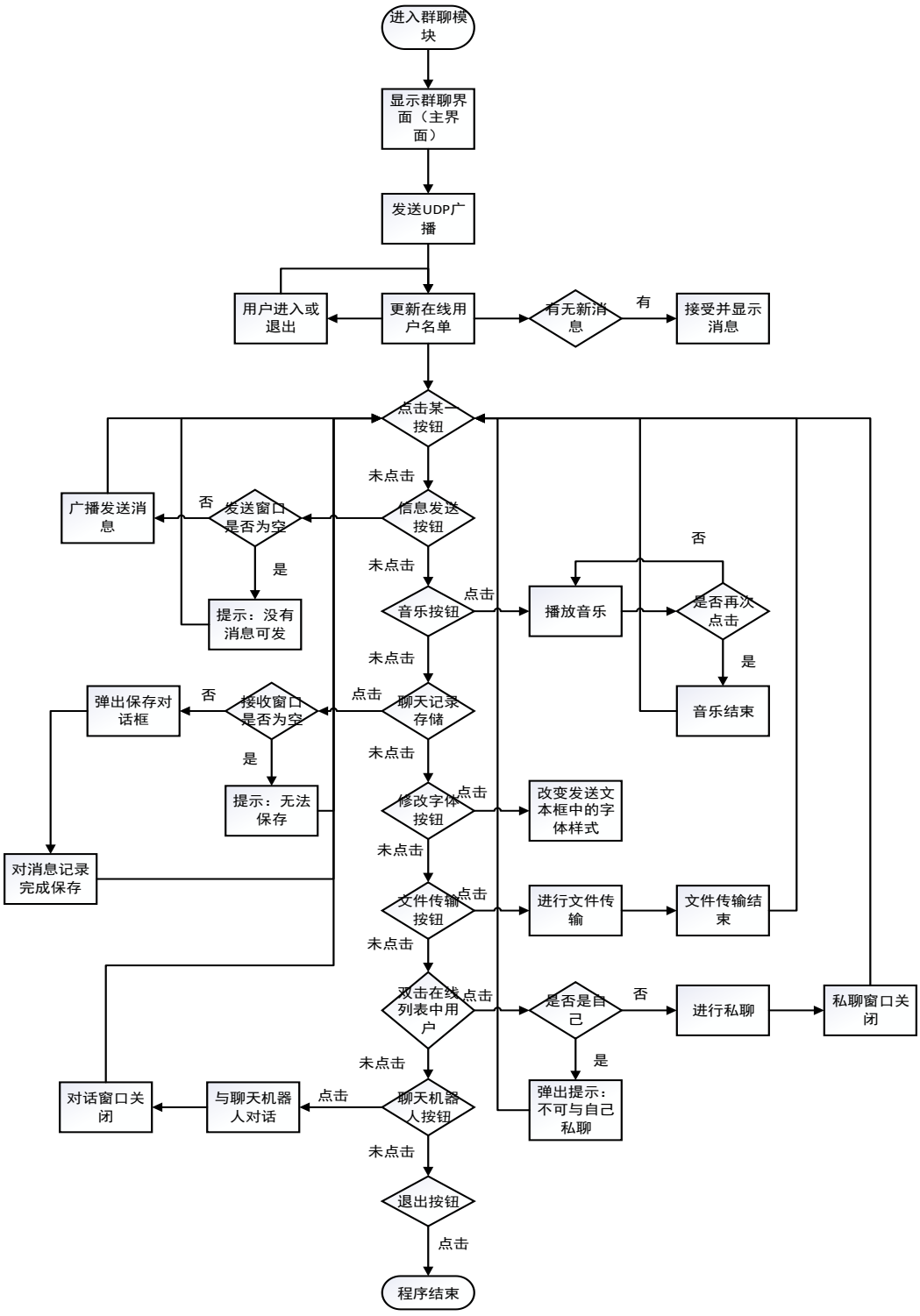
4.1.2 模块设计文字说明

登陆界面的核心功能是进行产品密钥的检测功能，首先显示当前系统时间，获取当前主机的用户名，作为默认用户名，只有输入正确密码才能进入聊天室界面，否则弹出警告窗口，按钮有两个，一个是登陆按钮，一个是退出按钮。

4.1.3 核心界面截图

4.2 多人聊天模块

4.2.1 程序流程图



4.2.2 模块设计文字说明

本聊天软件的群聊功能设计思路是：为保证能够在局域网内群聊，因此采用 UDP 广播的方式。在 ChatRoom 类内，定义了一个 massageType 的枚举类型作为我所发送或者接收消息时的标志量，在调用 messageSend(massageType, QString)方法的时候，通过 messageType 的不同，来区分我所要发送的不同的消息，messageType 能够表示的消息为：发送消息，用户加入，用户离开，发送文件，进行私聊，拒绝接收文件。而接收方根据收到的 UDP 报文中的标识位的不同，而采用不同的处理方法。

当出现新用户后，新用户会调用 messageSend 方法，通过传入的枚举类型变量 NewParticipant，向局域网内进行一次广播，并在 UDP 报文内放入自己的主机名，用户名，IP 地址，而接收到广播的用户则在获得这些信息之后，将其添加到自己的用户列表当中，而后同样发送一个 UDP 广播包，而新用户在接收到这些回复的 UDP 广播包之后，就也会提取出信息，添加进自己的用户列表中。从而完成用户列表的构建。

当用户在发送消息时，在输入完消息，单击完发送按钮之后，发送按钮所对应的槽就会判断输入文本框是否为空，如果为空就弹出提示对话框，如果不为空，就读取文本框中的文本，装入到 UDP 报文中，此外还会装入主机名、IP 地址等信息，而后进行 UDP 广播。接收方在接收报文之后，根据标识位发现是发送的消息，就会读取消息以及主机名、IP 地址等信息，显示在消息接收框中。

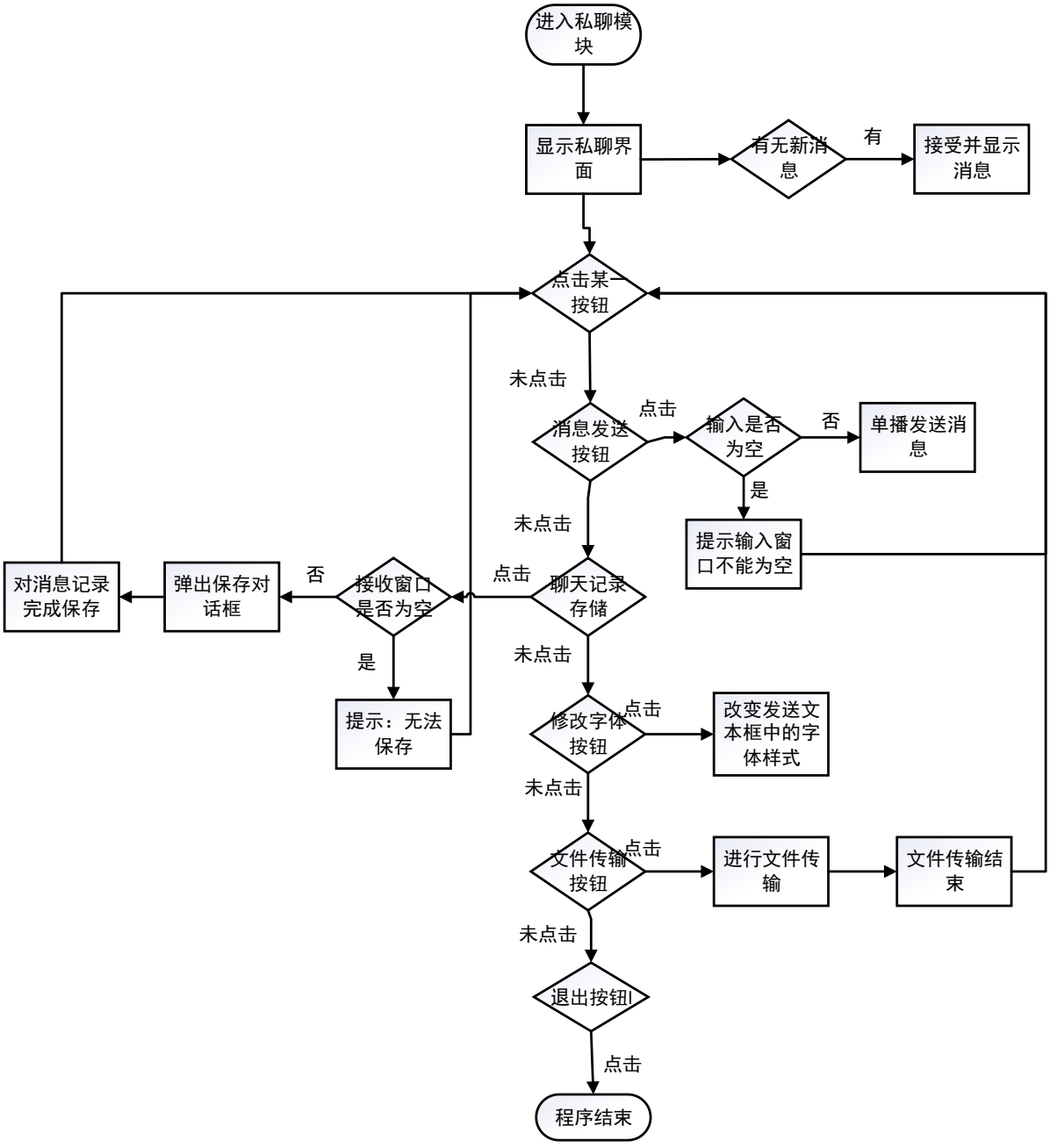
当用户双击用户列表中的其他用户的时候，会调用槽函数，首先检查用户是否双击自己进行私聊，如果不是的话，就建立一个私聊界面，并且向其发送 UDP 报文，而对方接受 UDP 报文之后，就也会打开一个私聊的对话框，从而进行私聊。

4.2.3 核心界面截图


now online: 3 person(s)
用户名
主机名
IP地址
1 Alice Alice 0.0.0.0
2 jk.tian WIN-I6BGOQVDU73 192.168.218.129
3 jk.tian LAPTOP-N95B2I79 192.168.218.1
hi
[LAPTOP-N95B2I79] 2018-07-18 09:16:43
你好
[LAPTOP-N95B2I79] 2018-07-18 09:16:52
你好
华文隶书
17
B
I
U
音乐
退出
发送

4.3 双人私聊模块

4.3.1 程序流程图



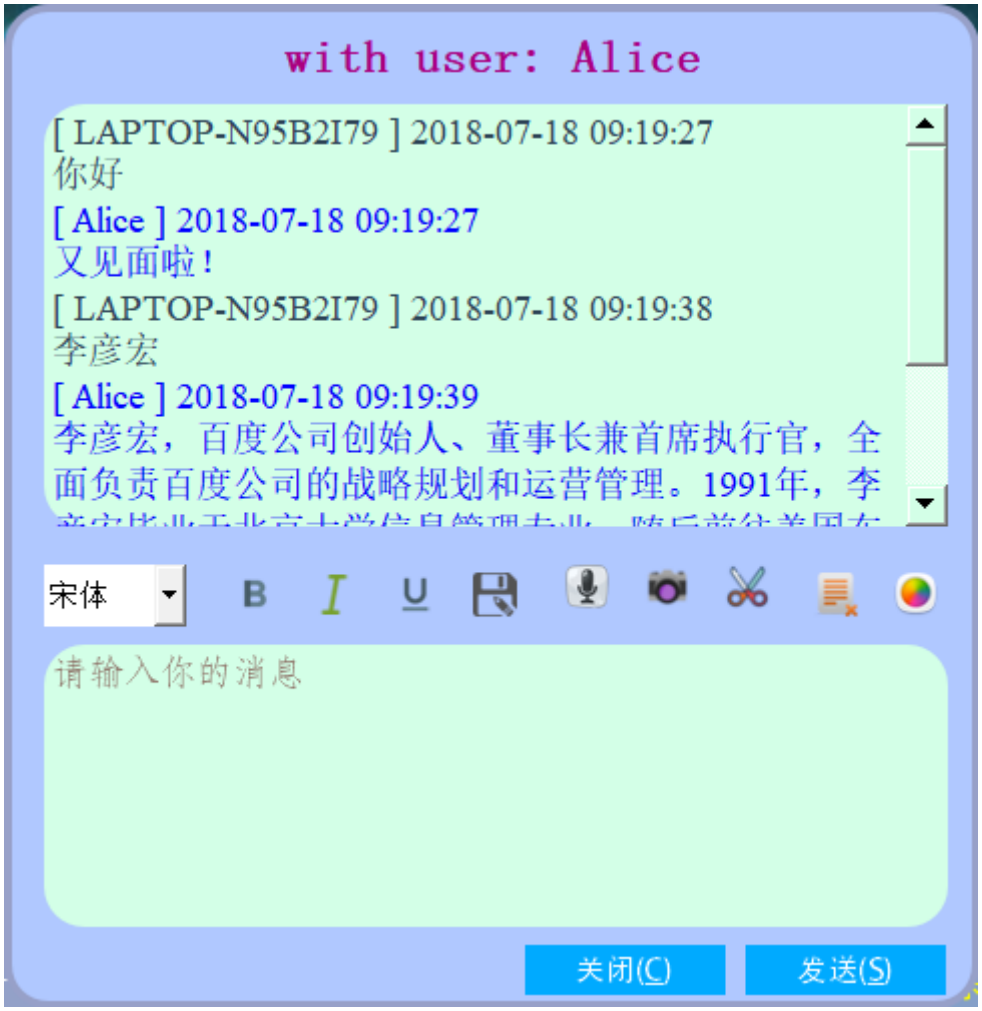
4.3.2 模块设计文字说明

双人私聊模块的设计思路与群聊模块基本相似，也是通过设置一个枚举量，作为UDP 报文里的标志位，从而使得发送方与接收方能够区分不同的 UDP 报文，并进行不同的处理。

私聊模块与群聊模块的区别主要在于，由于建立私聊模块时已经存储了对方的 IP 地址与主机名，因此双方采用 UDP 单播而不是广播来交换信息。此外也会使用与群聊时不同的端口，来进行私聊。

在私聊中，还加入了聊天机器人的功能，设计思路为调用图灵机器人的官网 API，前提是已经在官网创建了自己的聊天机器人，通过机器人密钥、URL 与发送消息的字符串连接得到新的 URL，提出请求访问新的 URL，从网络上得到机器人的 json 格式的消息回复，解析 json 格式文本，从中提取出有效信息，并且在消息框中显示。

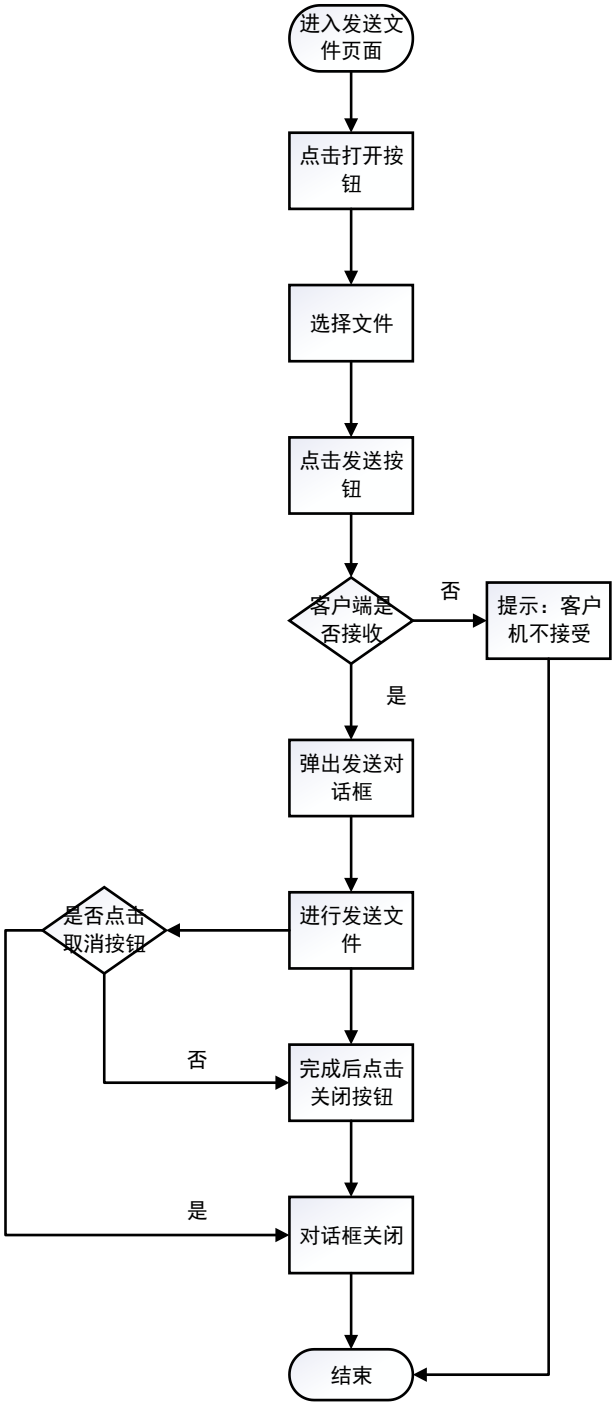
4.3.3 核心界面截图

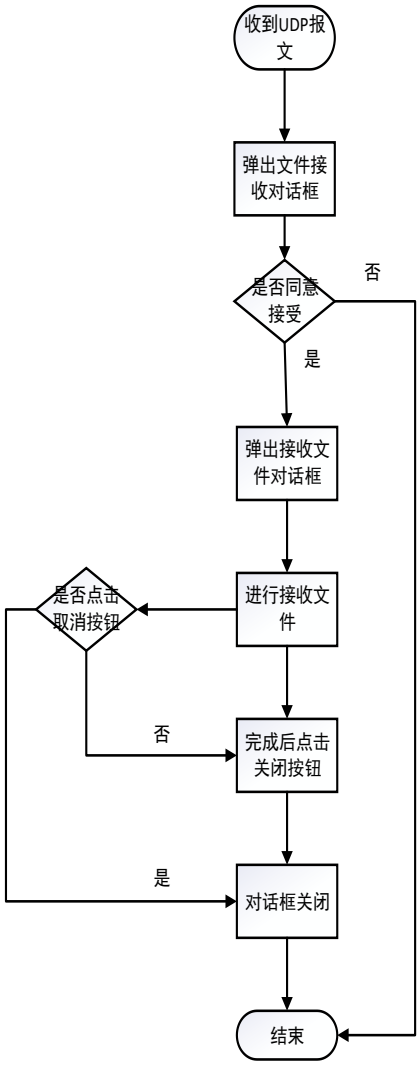


4.4 文件传输模块

4.4.1 程序流程图

左侧是发送方在进行文件传输时的流程图，右边是接收方在进行文件传输时的流程图



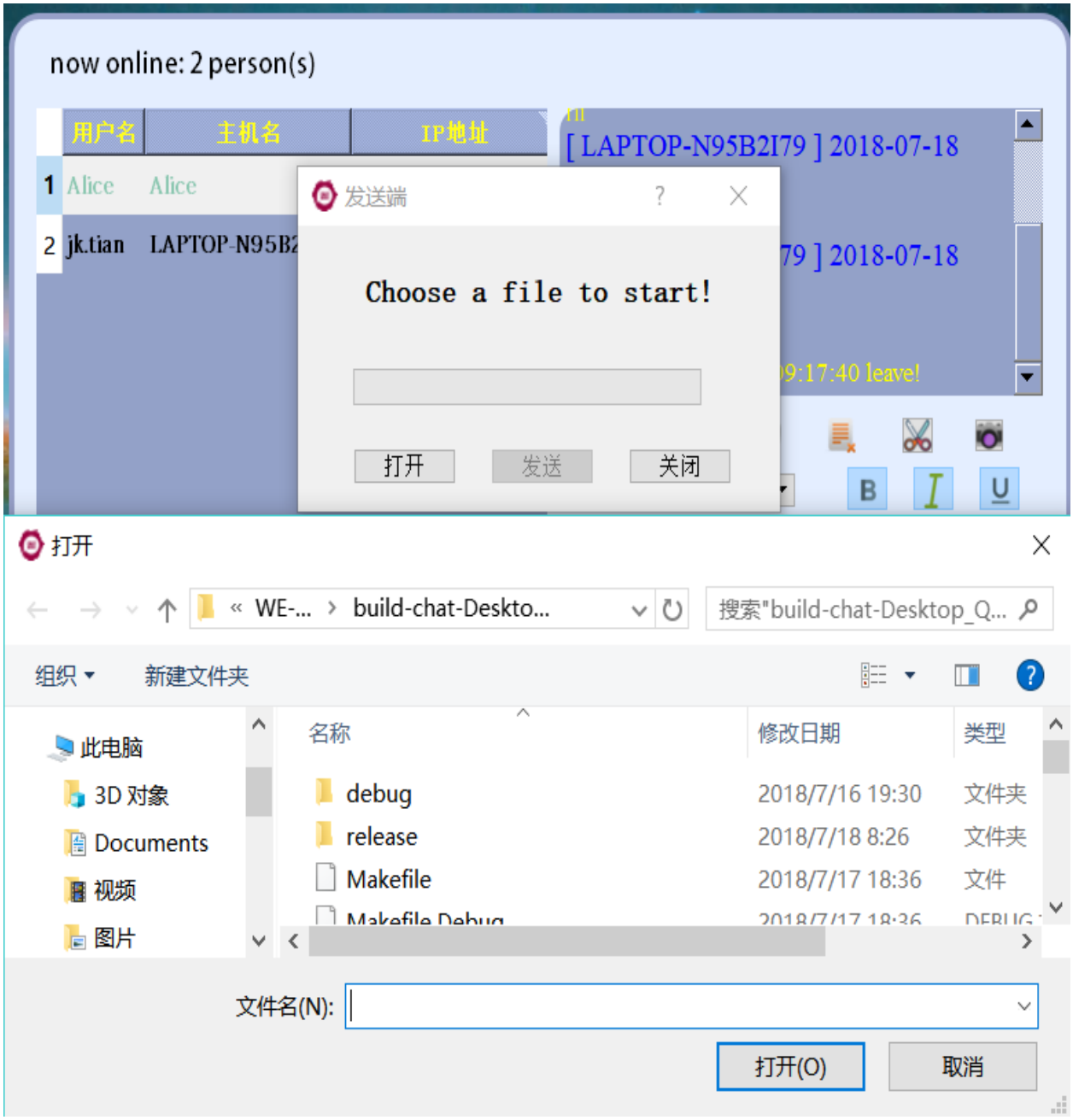


4. 4. 2 模块设计文字说明

文件传输模块的设计思路主要为：

当用户决定向其他用户发送文件之后，就会发送 UDP 广播，客户端收到广播之后弹出一个对话框，点击同意之后会新建一个客户端，服务端使用 TCP 进行文件传输，如果客户机选择拒绝，那么服务器端就会关闭 TCP 连接。

4. 4. 3 核心界面截图



5 系统测试

5.1 登录模块

序号	测试项	前提条件	操作步骤	预期结果	测试结果
1	登录界面	进入登录界面	输入密码为空，按下登录按钮	弹出警告框	通过
2		进入登录界面	输入错误密码，并按下登录按钮	弹出警告框	通过
3		进入登录界面	按下退出按钮	程序结束运行	通过
4		进入登录界面	输入正确密码，并按下登录按钮	进入聊天室界面	通过

5.2 聊天室模块

序号	测试项	前提条件	操作步骤	预期结果	测试结果
5	背景音乐播放功能	进入聊天室界面	点击右下角播放按钮	开始播放背景音乐	通过
6		进入聊天室界面	1. 点击播放按钮 2. 一段时间后，再点击播放按钮	背景音乐先播放，后暂停	通过
7		进入聊天室界面	连续三次点击播放按钮	背景音乐播放，停止，重播	通过
8	用户列表	处于聊天室界面	只有一个人进入聊天室界面	用户列表默认显示两个用户（机器人和自己）	通过
9		处于聊天室界面	局域网内另一个用户进入聊天室	用户列表新增一员	通过
10		处于聊天室界面	聊天室内另一名用户退出	用户列表减少一员	通过
11		处于聊天室界面	在主界面输入数字 4 并按下回车	界面不变化，等待下一个输入	通过
12	消息收发	处于聊天室界面	另一主机向此电脑发消息	响起消息接受提示音，消息列表出现新消息	通过
13		处于聊天室界面	此电脑向另一主机发消息	响起消息发送提示音，消息列表出现新消息	通过

14		处于聊天室界面	此电脑更改消息文本样式后，向另一主机发送消息	消息列表出现样式变化了的新消息	通过
15	消息记录保存	处于聊天室界面	点击消息记录保存按钮	出现“保存路径选择”对话框	通过
16	清空消息列表	处于聊天室界面	点击清空消息列表按钮	消息列表清空	通过
17	文本格式转换	处于聊天室界面	点击字号调整，下划线，斜体等任意按钮，并输入消息	文本格式发生变化	通过
18	退出功能	处于聊天室界面	点击退出按钮	退出程序	通过
19	进入私聊模块	处于聊天室界面	双击用户列表中某一用户（自己除外）	进入私聊界面	通过

5.3 文件收发模块

序号	测试项	前提条件	操作步骤	预期结果	测试结果
20	收发文件	至少两个主机处于聊天室	1. 主机 a 在用户列表选择主机 b，点击文件传送按钮，选择文件 2. 主机 b 点击接受按钮，选择保存路径	文件保存到主机 b 的指定位置	通过

5.4 私聊模块

序号	测试项	前提条件	操作步骤	预期结果	测试结果
21	聊天机器人	聊天室界面中选择“Alice”进入私聊界面	发送消息“你好”	收到回复：“好久不见”等问候语	通过
22		聊天室界面中选择“Alice”进入私聊界面	发送消息“李彦宏”	回复消息为李彦宏的个人信息，达到百科功能	通过

23		聊天室界面中选择“Alice”进入私聊界面	发送消息“成语接龙”	收到回复：“我先来，跃跃欲试”等	通过
24	文本格式转换	处于私聊界面	点击字号调整，下划线，斜体等任意按钮，并输入消息	文本格式发生变化	通过
25	消息收发	处于聊天室界面	另一主机通过私聊界面向此电脑发消息	此用户弹出私聊窗口，并收到消息	通过

6 设计总结

6.1 项目总体完成情况

本项目总共有四个模块。分别为界面模块（登录、群聊、私聊）、群聊模块、私聊模块、文件上传模块。

现在此软件可以实现登录时的身份验证功能、群聊正常收发消息、私聊、收发文件四个基础功能。在基础之上，添加了字体样式更换、音效播放、聊天机器人陪聊功能，且界面美观。此项目已经实现了基础以及扩展功能。

团队协作下，在经过任务划分、代码融合以及极具多样性的系统测试之后，此软件已经具有很好的鲁棒性和用户体验。

6.2 技术难点

界面的样式覆盖以及美观问题。在 QT 内置窗口、控件类的样式中，若之间存在继承、组合等关系，需要注意样式覆盖，导致编程困难。

消息通信以及文件上传的网络编程难点。对于 UDP 和 TCP 的使用是否合理的问题。要对计算机网络的基本知识有一定了解，如子网掩码、网关等。局域网的设置和环境的搭建。

聊天机器人如何实现的问题。如何调用图灵机器人官网的 API，进行信息的获取与展示。

6.3 解决方法

UI 编程中 QPalette 和 styleSheet 的混合使用，且要格外注意覆盖问题。

最终使用 UDP 广播实现局域网内部通信功能，用 TCP 实现文件收发功能。

聊天机器人调用图灵机器人官方给的 API 接口实现，前提是联网。要引入 QScript 等库，对网站进行 push, get, post 等操作，并且对取得的 JSON 格式数据再次进行处理，得到目标信息。

6.4 不足

可改良的功能：用注册功能取代产品密钥功能，可以加入数据库编程，用于管理用户信息。

可以扩展的功能：音视频的录制以及截屏功能没有实现，但是已经预留了按钮。