



# 学士学位论文

## 基于 Qt 的通信软件

---

姓 名 何宇轩

学 号 202090716

院 系 信息科学与工程学院

专 业 计算机科学与技术

指导教师 周晓宇

二 〇 二 〇 年 六 月 一 日

基于 Qt 的通信软件

Qt based communication software

## 学位论文原创性声明

本人所提交的学位论文《基于 Qt 的通信软件的设计与实现》，是在导师的指导下，独立进行研究工作所取得的原创性成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中标明。

本声明的法律后果由本人承担。

论文作者（签名）： 何宇轩 指导教师确认（签名）： 周晓宇  
2022 年 6 月 1 日 2022 年 6 月 1 日

## 学位论文版权使用授权书

本学位论文作者完全了解河北北方学院有权保留并向国家有关部门或机构送交学位论文的复印件和磁盘，允许论文被查阅和借阅。本人授权河北北方学院可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其它复制手段保存、汇编学位论文。

保密的学位论文在 2022 年解密后适用本授权书。

论文作者（签名）： 何宇轩 指导教师确认（签名）： 周晓宇  
2022 年 6 月 1 日 2022 年 6 月 1 日

## 摘 要

通信软件发展很快，由开始的仅仅少部分人使用到如今发展到数十亿人在使用，已经离不开我们的生活，但在一些特殊领域，比如军工企业或拥有保密性质的研究所，没法使用手机，人与人之间的通讯仍然依靠口头交流，为此开发一款满足日常通信需求的国产局域网通信软件。

本基于Qt的网络通信软件是一个可以支持网络通信的软件，数据库采用了Mysql，运输层采用的协议是UDP协议，开发主要使用C++和Qt框架，软件是P2P架构。

此软件主要实现一个用于局域网的通讯软件，主要功能有好友间通讯以及群组内通讯，进程间通信采用套接字通信，信息直接由客户端到客户端，为了满足一些军工企业或者拥有保密机构的企业的日常通信需求而开发，能满足办公人员日常交流的同时可以提高工作效率与质量。

**关键词：**Qt；通信软件；P2P 架构

## ABSTRACT

Communication software has developed rapidly, from being used by only a few people at the beginning to being used by billions of people now, which has become inseparable from our lives. However, in some special fields, such as military enterprises or research institutes with confidential nature, mobile phones are not available, and communication between people still depends on oral communication. To develop a domestic LAN communication software to meet the daily communication needs.

This network communication software based on Qt is a software that can support network communication. The database uses Mysql, the transport layer uses UDP protocol, the development mainly uses C++ and Qt framework, and the software is P2P architecture.

This software mainly realizes a communication software for LAN, the main function is communication between friends and communication within the group, interprocess communication using socket communication, information directly from client to client, in order to meet the daily communication needs of some military enterprises or enterprises with security agencies and develop, Also hope to meet the daily communication of office staff can improve work efficiency and quality.

**Keywords:** Qt; Communication software; P2P architecture

# 目 录

第 1 章 绪论.....	1
1.1 研究背景.....	1
1.2 国内外发展现状.....	1
1.3 本文研究内容.....	1
第 2 章 项目开发环境简介.....	2
2.1 开发工具和开发环境简介.....	2
2.2 开发技术简介.....	2
2.2.1 C++简介 .....	2
2.2.2 MySQL 简介.....	2
2.2.3 Qt 简介 .....	2
2.2.4 MSVC 简介.....	2
2.2.5 Git 简介.....	3
2.2.6 P2P 简介 .....	3
第 3 章 软件分析.....	4
3.1 软件目标.....	4
3.2 需求分析.....	4
3.2.1 性能需求分析.....	4
3.2.2 功能需求分析.....	4
3.3 可行性分析.....	5
第 4 章 软件设计.....	6
4.1 软件设计原则和目标.....	6
4.2 前端设计.....	7
4.3 软件功能模块设计.....	8
4.3.1 登录注册.....	9
4.3.2 用户间通信与群组内通信.....	11
4.3.3 添加好友与添加群功能.....	14
4.3.4 创建群聊.....	15
4.3.5 获取好友与群组列表.....	16
4.4 数据库设计.....	18
4.4.1 数据库设计步骤.....	18
4.4.2 数据库概念设计.....	18
4.4.3 数据库逻辑设计.....	19

第 5 章 系统实现.....	21
5.1 数据库连接和操作的实现.....	21
5.2 系统功能的实现.....	22
5.2.1 登录注册功能实现.....	22
5.2.2 主界面功能实现.....	24
5.2.3 添加好友与添加群功能实现.....	26
5.2.4 创建群聊功能实现.....	26
第 6 章 软件测试.....	27
6.1 系统测试准备.....	27
6.2 系统测试用例设计.....	27
结论.....	29
参考文献.....	30
致谢.....	31

## 第 1 章 绪论

### 1.1 研究背景

随着互联网的逐步发展，电信业长期存在的“垄断性”将被淡化，人们的主要通信方式大部分都由付费的电话改为使用免费通信软件了，主要是昂贵的电话费用无法满足人们日益增长的交流欲望，而免费的通信软件已经占据我们日常通信的主要部分。现如今，在公网通信方面，通信软件由最初的仅有聊天的功能逐渐演变成如今拥有丰富功能的软件。而在局域网通信方面，仍然没有一个普及的通信软件。

在一些特殊领域，比如开发军工软件的企业，或者一些研究所，由于保密性质，日常工作人员无法携带手机到办公地点，而且使用的电脑都不允许连接公网，因此一个普及的局域网通信软件有着广阔的市场需求。

### 1.2 国内外发展现状

国内成熟的通信软件发展态势最好的就是 QQ 和微信，1998 年 1 月腾讯公司在深圳成立，1999 年 QQ 的前身 OICQ 正式诞生。在最初通讯软件提供即时通讯服务，当时的 OICQ 凭借设计合理操作简单的界面击败其他公司推出的即时通讯软件，在当年的 11 月份注册的用户量就达到了 100 万，占领当时中国在线即时通讯 80%以上的 IM 市场。现如今，QQ 和微信已经占据了国内通信市场的大部分业务，是最火热的通信软件。

国外最早的通信软件是 ICQ，由几个犹太人研发，至今还被很多俄语区的人们使用。而国外的通讯软件也在随后逐渐成熟，比如微软的 MSN，之后微软又收购了 Skype。目前很流行的通信软件还有 Line、WhatsApp 等等，使用人数也达到了上亿人。

目前国内成熟的商业化软件中没有一款使用范围广泛的局域网通信软件。在军工企业或者是研究所，由于保密性要求办公环境不允许连接公网，而且日常使用的软件都要求国产化，因此开发一款用于局域网的通信软件是一件很有必要的事。

### 1.3 本文研究内容

基于 Qt 的网络通信软件由 C++ 开发，使用 mysql 数据库来保存账号和群组信息以及存储好友间的关系和群组与用户之间的关系。软件使用 P2P 架构，通信是直接由客户端连接到客户端。本文详细记述了此软件的开发设计流程与细节步骤。在功能方面，此软件实现了在局域网内进行好友间通信以及群组内通信、用户登录注册、创建群组、添加好友与群组等功能。此软件用到 Qt 和 VS 为开发工具，对上述逻辑设计进行功能实质化的转换，实现一个拥有基础功能的网络通信软件。



## 第 2 章 项目开发环境简介

### 2.1 开发工具和开发环境简介

开发工具主要为 Visual Studio Community 2022, 以及 QtCreator 和 QtDesigner。数据库可视化工具使用的 Navicat。客户端适配的系统是 Window10。项目在 github 上建立了远程仓库<sup>[1]</sup>, 在 github 进行后续的开发更新以及下载源码。

### 2.2 开发技术简介

数据库使用的 Mysql5. 6. 45, 开发语言为 C++, 使用了 Qt 框架, 编译器是 Microsoft Visual C++(MSVC)。版本管理工具使用的分布式管理工具 git。

#### 2.2.1 C++简介

C++是一种被广泛使用的计算机程序设计语言。它是一种通用程序设计语言, 支持多重编程范式, 例如过程化程序设计、数据抽象、面向对象程序设计、泛型程序设计和设计模式等。

#### 2.2.2 MySQL 简介

MySQL 是一个开放源码的关系数据库管理系统, 它由 C 和 C++编写, 并使用了多种编译器进行测试, 这保证了源代码的可移植性。

#### 2.2.3 Qt 简介

Qt 是一个跨平台的 C++应用程序开发框架。广泛用于开发 GUI 程序, 在各种软硬件平台(如 Linux、Windows、macOS、Android 或嵌入式系统)的底层代码库中几乎没有或没有变化, 相同的代码可以在任何支持的平台编译与执行, 而不需要修改源代码。会自动依平台的不同, 表现平台特有的图形界面风格。它也是自由且开放源代码的软件, 在 GNU 宽通用公共许可证(LGPL)条款下发布, 本软件也支持此协议。

#### 2.2.4 MSVC 简介

Microsoft Visual C++(MSVC), 它是 C、C++和 C++/CX 编程语言的编译器。MSVC 是专有软件, 它最初是一个独立的产品, 但后来成为 Visual Studio 的一部分, 并以试用软件和免费软件的形式提供。它具有用于开发和调试 C++代码的工具, 尤其是为 Windows API、DirectX 和 .NET 编写的代码。

### 2.2.5 Git 简介

Git 是用于 Linux 内核开发的版本控制工具。与 CVS、Subversion 一类的集中式版本控制工具不同，它采用了分布式版本库的作法，不需要服务器端软件，就可以运作版本控制，使得源代码的发布和交流极其方便。git 最为出色的是它的合并追踪（merge tracing）能力。

### 2.2.6 P2P 简介

对等(P2P)计算或网络是一种分布式应用程序架构，对等方将其部分资源（例如处理能力、磁盘存储或网络带宽）直接提供给其他网络参与者，而不需要服务器或稳定主机的中央协调。对等点既是资源的提供者，也是资源的消费者。

## 第3章 软件分析

### 3.1 软件目标

软件可以实现登录注册，显示当前好友，添加好友，以及与好友进行通讯等功能。之外还有创建群聊，添加群聊，以及在群组内通讯等功能。

### 3.2 需求分析

#### 3.2.1 性能需求分析

##### (1)硬件性能

有 200MB 以上的硬盘空间，1GB（32 位）或 2GB（64 位）RAM。

##### (2)安全性

用户只接收与用户有关的数据报文，报文要安全加密，防止被截断篡改。

##### (3)易用性

软件的操作界面需简单容易使用。

##### (4)可移植性

由 Qt 开发的软件可以编译运行于各种平台。

##### (5)可扩展性和可维护性

目前版本管理采用了 git，git 有详细的日志，能记住你的每一次提交、每一次改动，并且能够比较查看不同版本之间的异同，可以恢复到之前的任一版本。而且所有类的成员函数和成员变量都有详细的注释，函数也尽量功能明确，一个函数只做对应该做的事。

#### 3.2.2 功能需求分析

软件主要有用户的登录注册，添加好友，好友间通讯，创建与添加群聊，以及群聊内通信功能。用户主要包括以下功能模块：

(1)用户登录，用户能登录，能正确的验证账户和密码即可。

(2)注册，用户能注册即可，需要防止空字符以及重复的用户名进行注册。

(3)添加好友，需要判断是否已经成为好友了，如果是好友需给出对应的提示，需判断所添加的好友账户是否存在等情况，根据不同的情况需要给出对应的提示。

(4)好友间通信，需要考虑同设备同时登录多个程序的情况，消息不能接收混乱，应做到能准确的接收信息，用户间能正常的通信即可。

(5)创建群聊与加入群聊，能正确的创建群聊，并且群聊的账号不允许重复。加入群聊时需判断用户是否已经加入该群聊，并给出对应的提示。

(6)群聊内通信，确保加入群聊的用户都可以收到信息并且消息能准确的被接收。

### 3.3 可行性分析

软件可行性分析的目的就是为了在尽量短的时间内，去确定软件中存在的问题能否快速解决。软件的可行性分析主要从经济可行性，技术可行性，社会可行性，安全可行性，法律可行性去分析。

#### (1)经济可行性

目前投入的时间成本已经达到了 2 月。所用到的软件以及开发环境基本都是免费的。经济上来说是可行的。

#### (2)技术可行性

本文的设计是基于 Qt 框架的开发，数据库为 Mysql 数据库，编程语言使用的 C++，Qt 框架的优势是模块化程度高，可重用性好，对于用户的开发来说是十分方便的。Qt 提供了信号槽来代替回调函数，这使得各个元件间的协同工作十分方便。

#### (3)社会可行性

在一些保密机构或者军工企业，由于行业限制，这样的工作环境不允许连接公网，而一款局域网的通信软件有着广阔的市场。

#### (4)法律可行性

本软件在开发过程中，使用到的开发软件和框架都是开源免费的，关于代码全部为本人所敲，所以不会有任何法律问题。

## 第 4 章 软件设计

### 4.1 软件设计原则和目标

本软件为 P2P 架构，在开发过程中严格采用面向对象的程序设计方法作为基本原则，采用了数据抽象的概念可以在保持外部接口不变的情况下改变内部实现，从而减少甚至避免了外界的干扰，外界创建对象时仅可访问所提供的公有函数。通过继承大幅减少了冗余的代码，并可以方便地扩展现有代码，提高编码效率，也减低了出错概率，降低软件维护的难度。

开发过程中保持着高内聚低耦合的特点，任何可复用的功能模块都被编写成了单独的函数，同时采用了简单工厂模式进行开发，定义一个创建产品对象的工厂接口，并将实际创建工作推迟到子类中。考虑到用户需要的功能设计的对应的类，工厂模式 UML 类图如图 4.1 所示。

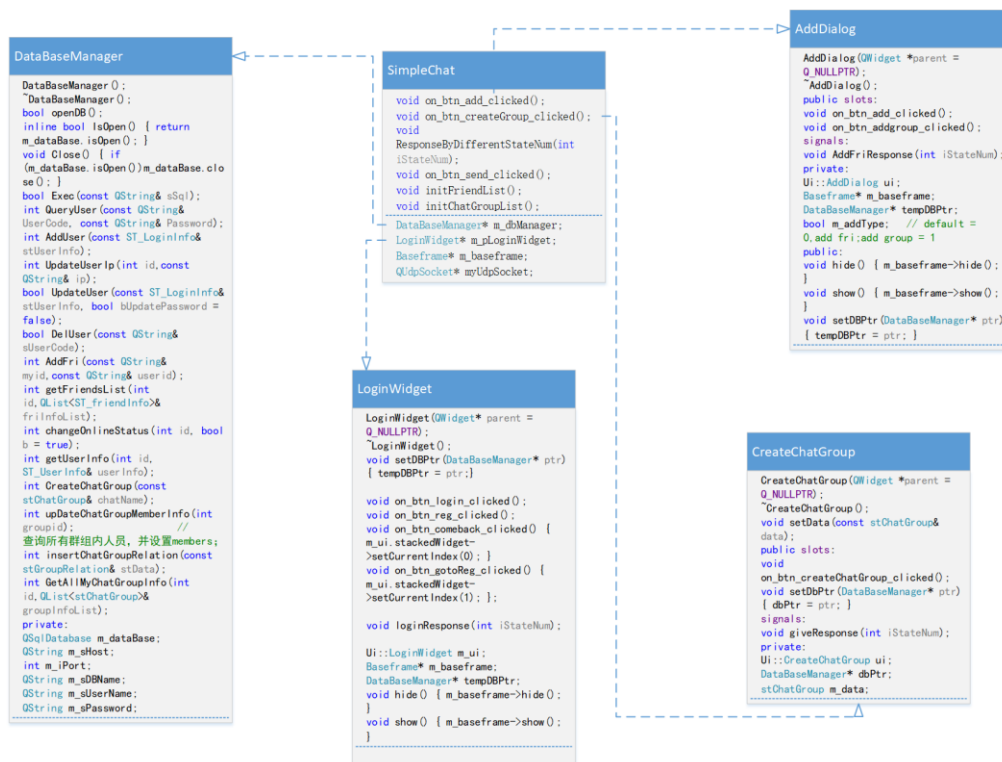


图 4.1 工厂模式 UML 类图

为了接口的可扩展性，所有的主要对象或者方法都是定义在 SimpleChat 类里，它作为工厂类分别创造了 LoginWidget、DataBaseManager、AddDialog 和 CreateChatGroup 这些类的对象，借此组成程序的功能界面。如果后续添加更多的功能只需要修改 SimpleChat 类和创建对应的功能类即可。

## 4.2 前端设计

前端由 QtDesigner 开发, 细节样式由 qss 来处理。Qss 是 Qt 中一种比较轻松的设置前端的方式, 具体主要使用 `setStyleSheet` 这个方法去设置的。qss 中有几种特殊的选择器, 比如一个自定义类你想给其中的一部分设置特殊的样式, 定义一个枚举值分别表示不同的情况, 再设定一个公有函数来设定不同的情况。比如在代码中使用 `setProperty` 来设定属性, 借此可以使用 Qss 中的属性选择器来设置对应的控件达到你想要的样式。为了表示用户的在线状态, 在用户的控件类 `userWidget` 中设置了 `setProperty` 方法, 借此可以实现登录时的状态改变, 当获取到的用户在线状态为离线时, 可以设置 `userWidget` 的属性为一个自定义值, 在 qss 文件中写好实现定义好的对应的离线和在线状态即可修改对应具体的样式。`setProperty` 设置的属性可以是一个自定义值, 但每次调用 `setProperty` 时需重新加载整个 qss 文件。

QtDesigner 中也可以编辑 Qss, 但企业开发中会把 Qss 集中放到一个文件中, 进而使得样式可以集中编辑, 并且可以写好注释。Qss 没有像 vscode 中的 `liveServer` 那样的插件可以方便的实时调试, 为了进行实时调试, 本文的做法是借助 Qt 提供的文件监视类 `QFileSystemWatcher`, 在 `SimpleChat` 中定义一个 `QFileSystemWatcher` 的成员变量, 将它绑定一个文件, 当这个文件发生变动时便会触发 `fileChanged` 信号, 借助 Qt 的信号槽, 你可以借此将调用 `setStyleSheet` 方法的函数与该信号绑定, 来达到实时修改实时调试样式的功能。借此你可以实现像 vscode 的插件 `LiveServer` 那样的效果。

为了实现无边框窗口, 本文创建了一个类 `BaseFrame`, 它继承自 `QWidget`, 几乎每个窗体类都被他所包含, 而这导致了窗口无法被拖动而且无法修改大小, 这需要你自己取重写一些鼠标的拖动事件来令这可行。你需要重写 `mousePressEvent`, `mouseMoveEvent` 和 `mouseReleaseEvent`, 这个窗体中设置一个 `layout` 来放置其他的控件即可, 本文的做法是在 `baseframe` 中提供一个函数接口, 命名为 `addWidget2Content`, 它的参数可以是一个 `QWidget` 或者是 `QWidget` 的任何派生类, 功能是把对应的控件添加到自身的 `Layout` 中, 举个例子, `SimpleChat` 中他是一个成员变量去存在的, 但调用了 `addWidget2Content` 方法后将 `SimpleChat` 的整个控件放到 `BaseFrame` 中了, 此时 `BaseFrame` 的关闭按钮就负责关闭整个软件了, 但它也会对应的进行一些特殊处理, 在 `baseframe` 中创建一个设定窗口类型的公有函数接口, 它被命名为 `setWindowType`, 该函数可以用来进行设置自身的属性, 它有一个 `int` 型的成员变量叫 `m_iWinType`, 定义一些枚举型来对应不同的窗体, 然后借此可以进行特殊的处理, 比如 `m_iWinType == 0` 时为默认窗体, 在关闭时会关闭整个软件, 而且会发射关闭信号交给 `SimpleChat` 去处理, `SimpleChat` 会借助信号槽去进行更新用户在线状态的操作, 然后再关闭软件, 而 `AddDialog` 的关闭则是直接调用 `QWidget` 的 `Close()`, 这是因为 `setWindowType` 去进行了特殊的处理而去对应不同的窗体。本文设置的 `setWindowType` 的几种枚举型类型, 窗体枚举型对应码表如表格 4.1 所示

表 4.1 窗体枚举型对应码表

枚举值	含义	对应码
defaultType	默认窗口	0
hideBtnZoom,	隐藏缩放窗口	1
DialogType,	对话框窗口	2
MainWindow	主界面窗口	3

在每个窗体类的构造函数中中可以调用 **BaseFrame** 类的 **setWindowType** 中去设置不同的窗口。

### 4.3 软件功能模块设计

用户登录时会进行对用户账户密码的校验判断，用户注册时会判断是否当前用户名是否存在后，如不存在才可注册成功。登录注册功能主要由 **LoginWidget** 类来处理。

用户间聊天即可以和已经添加了的好友进行聊天，群组内聊天即用户可以在加入的群组内进行聊天，这两项功能在 **SimpleChat** 类中来处理。

用户会在程序运行后每间隔 10s 获取一次数据库中好友列表和所加入群组列表，获取成功即在主功能界面显示出当前用户的好友列表和群组列表。这两项功能在 **SimpleChat** 类中来处理。

用户在创建群组时会判断群组是否已存在，然后创建该群组。创建群组由 **CreateGroupDialog** 类来处理。

用户在添加好友时会判断所添加用户是否已经是好友以及判断该用户是否存在后才允许添加好友，加入群组同理，添加的好友和群组会显示到主功能界面上。这两项功能由 **ChatDialog** 类来处理。将以上功能绘制了功能模块图，功能模块图如图 4.2 所示。

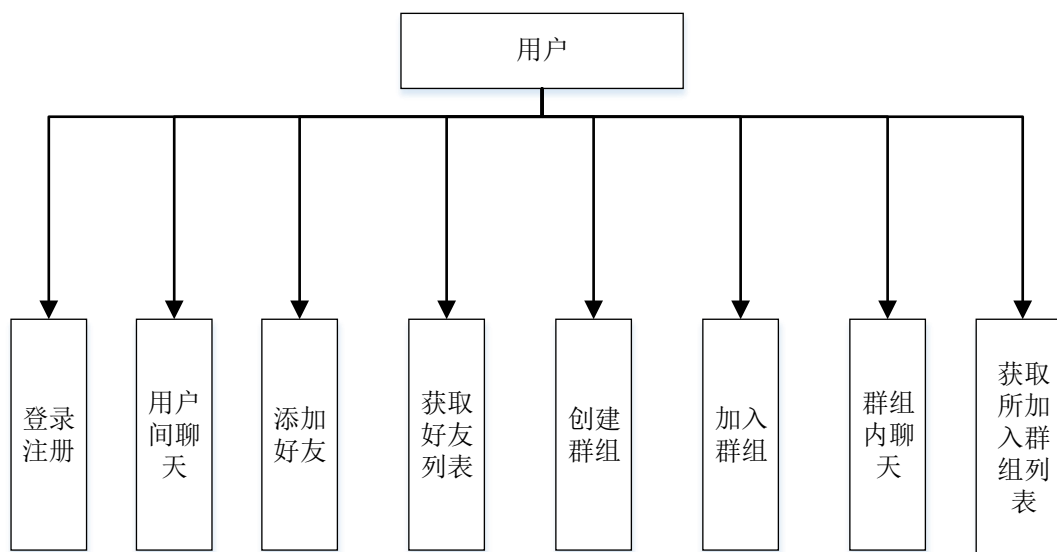


图 4.2 软件的功能模块图

### 4.3.1 登录注册

登录注册界面是由 LoginWidget 这个类实现的，Qt 中的界面是一个单独的成员变量，一般命名为 m\_ui，通过编译 Qt Designer 设计好的 ui 文件，进行声明定义，令 ui 生效就在 LoginWidget 的构造函数中调用 setUpUi() 方法即可。

登录界面中的主体部分是一个 QStackWidget，点击注册会跳转到 QStackWidget 的其他页面，当点击登录时，首先会进行简单的空值判断，然后根据传入的数据库操作类的指针调用查询用户的方法即可，定义一系列的状态码可以帮助你判断错误信息，登录有关的状态码有 LoginSuccess，QueryExecFailed 和 DBIsNotOpen，第一个是成功的状态码。数据库操作相关状态码表如表 4.2 所示。

表 4.2 数据库操作相关状态码表

枚举值	含义	状态码
Success,	成功	0
Failed,	失败	1
QueryExecFailed,	Sql 执行失败	2
LoginSuccess,	登陆成功	3
DBIsNotOpen,	数据库未打开	4
QueryCountLessThan0,	查询结果数量大于 0	5
AddUserSuccess,	添加用户成功	6
WannaAddUserButThereHasBeen,	想要添加用户好友但已经存在该好友关系了	7
CannotFindThidUser,	无法找到用户	8
AlreadyHaveThisRelation,	已经有关系了	9
AddFriSuccess,	添加好友成功	10
CreateGroupSuccess,	创建群组成功	11
insertGroupRelationSuccess,	加入群成功	12
updateChatGroupMembersInfoSuccess,	更新群组成员信息成功	13
AlreadyHaveThisGroupRelation,	已经加入群组	14
CouldNotFindTheGroup,	无法找到群组	15

主功能界面的类是 SimpleChat，SimpleChat 的构造函数中为一个 LoginWidget 变量分配内存，于是就有了登录页，SimpleChat 作为一个工厂类，是软件的主要界面部分，DataBaseManager 是一个专用于数据库操作的类，作为工厂类的成员变量来使用，LoginWidget 也是他的一个成员变量，当软件初始化时，构造函数中会为 LoginWidget 分配内存，此时即出现登录功能，LoginWidget 设置了 setDBPtr 方法，专门用来把工厂类中的 m\_DatabaseManager 用指针传递进来，借此 LoginWidget 可以进行跟数据库的一些操作，比如使用 QueryUser 来查询账户密码的正确性来实现验证登录，使用 AddUser 来注册账户等等，



当结束了LoginWidget的操作时，它会发射一个LoginSuccess的自定义信号，参数是一个枚举型的状态码，SimpleChat工厂类会调用ResponseDifferentStateNum去响应处理对应的状态码要做的事情。登录成功后，用户名称会被显示到左上角的按钮中，然后会修改数据库中当前用户的在线状态，修改成在线，然后会查询所有的好友信息并显示到界面中，而且还会查询一下当前用户的所有信息，并存入一个Static结构体变量中，该变量记录者当前用户的所有信息，比如id, username, password, ipaddress以及createtime等字段。

Qt 的信号槽机制有很多地方值得学习，信号槽是一种用函数指针的形式去设定的某种规则，Qt 提供了 connect 这个函数来绑定信号槽，而且它可以有很多的不同参数的组合，最基础的用四个参数，首先是一个对象，比如 QPushButton，然后是他的某个信号的文本值，信号可以被我们用 emit 指令触发，也可以被系统触发，比如一个按钮点击的时候会触发它自身的 clicked 信号，第三个参数是槽函数的对象，一般是 this 指针，第四个参数是槽函数的文本值。还有一个有缺省值的第五参数，是一个枚举值，用来表示连接类型。

除此之外还可以用函数指针和 lambda 表达式的形式，比如第一个值是一个对象的引用，第二个值是一个信号的地址，第三个值是一个对象的引用，第四个是这个对象的函数。而 lambda 表达式在一定程度上让这更简便了，有些时候这个槽函数的存在意义仅仅是相应这个信号而已，可以不用特意去声明并定义一个特别的槽函数而直接在表达式中来写要做的内容。lambda 表达式在很多时候都是一种很便捷的方式，但会让你的代码看起来乱一些。

Qt 的函数不只可以像这样被触发，还可以直接定义出发特定信号的函数：

比如用 on+控件的 objectname+ 信号名称，比如注册按钮点击事件槽函数 on\_btn\_login\_clicked()，这连信号槽的连接都省略了。借此方法定义的函数因为名字的特殊性，如果 btn\_send 的点击被触发了，便会调用该函数。从而完成自己想做的事。

避免定义槽函数和绑定信号槽的语句，你可以这样定义一个像 on\_btn\_reg\_clicked() 的函数，来实现注册的点击事件。当点击信号触发时，会调用传入的 DataBaseManager 指针的插入用户函数，注册的流程如图 4.3 所示。

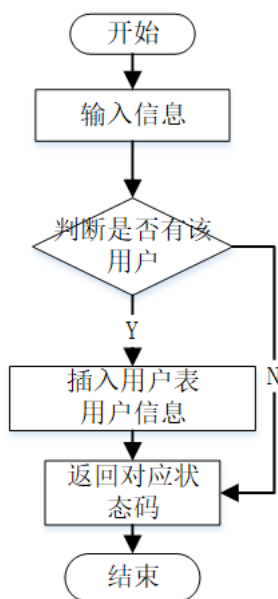


图 4.3 注册的流程图

也有另外一种方法，由于设置了 user 表中 id 和 username 的 unique 属性，即使有该用户也会插入失败，不会出现同时出现两个相同用户名的账户，也可以根据 sql 的执行失败来返回对应的状态码。这样可以省去一步查询的时间，在时间上更提高了程序的效率。

#### 4.3.2 用户间通信与群组内通信

通讯的部分，进程间的通信方式本文选择了套接字。Qt 有提供不同的传输层协议所对应的类。QTCPServer, QTcpSocket, 以及 QUdpSocket。本文使用的 QUdpSocket。

当点击发送信息时，首先在这之前 SimpleChat 的构造函数中就初始化了一个 QUdpSocket 的成员变量 m\_udpSocket，这个成员变量首先会监听当前的端口。并且它的 readyread 信号会被绑定到一个读取信息的槽函数，用来处理接受信息，当它的 hasPendingDatagrams 有值时边可进行读取。而发送信息调用 writeDatagram，参数时一个 ByteArray 的数据段，数据段的长度，一个 QHostAddress 类型的 ip 和一个整形端口号。

qt 项目支持使用网络编程的话，必须选择 network 模块，老版本的 qt 需要在 pro 文件中配置，然后使用该模块，新版本只要勾选即可。若要做到及时通信，则必须使用 UdpSocket，TCP 是建立在稳定的基础上的，而实际情况是用户有可能会下线，TCP 的三次建立连接的过程就无法完成。TCP 建立在双方都确实在线的情况下的，而如果一方离线则无法进行通讯，甚至无法发送，因为无法建立连接，而 UDP 则在这样的情况下仍然可以发送，UDP 只负责发而不关心接收。

UDP 建立连接的操作很简单，它是面向数据报的传输层协议，流程是：初始化一个 QUdpSocket 的变量，然后调用他的 bind 方法，来监听一个端口，由于考虑到在同一台电脑上可能会有多用户会登录，在绑定端口的时候可以选择绑定模式为共享地址。

为了不让消息混乱，在传送的数据报文中添加了一个自定义报文头数据，有三个数据段：第一个是发送的数据类型，这是一个枚举型的类型，但用 `uint32` 来强制转换这个数据，这用来表示报文类别，目前仅有群聊和用户间聊天两种类别，如果以后想增加更多的类型只要顺序往下定义即可；第二种是发送者 `id`，是一个 `uint32` 类型，第三个是接收者的 `id`，也是 `uint32` 类型。

群聊的通信也和好友间通信一样，你首先需要选择当前群聊，右边的对话框会显示当前群聊的信息，群聊的通信报文头结构也有三个 `uint32` 的字段，因为发送的报文是 `QByteArray` 类型的，就是 C 语言中的 `char*` 字符串，设定固定的报文头部，在接受也提取前固定位的值，便可据此判断信息。发送信息的流程图如图 4.4 所示。

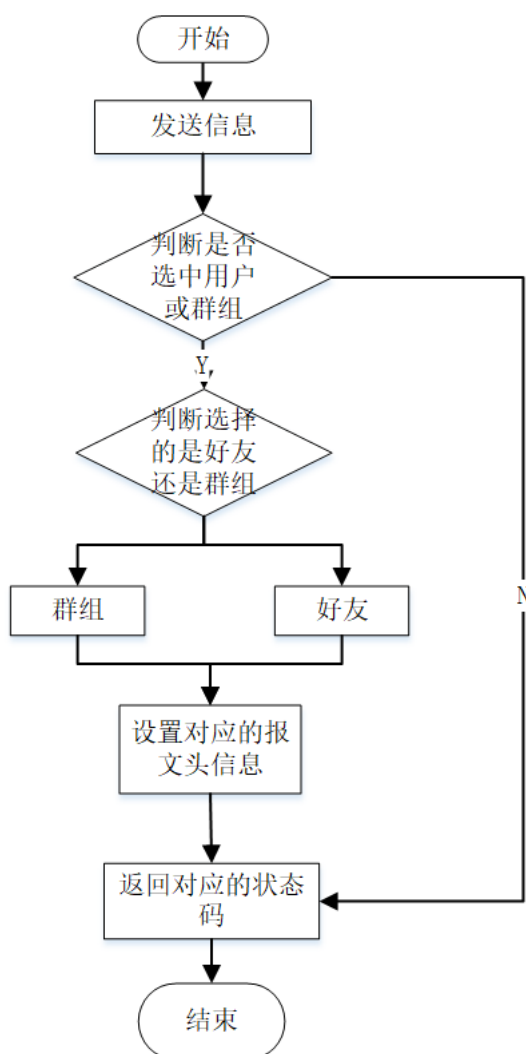


图 4.4 发送信息的流程图

发送的信息中包括报文头和报文内容，报文头字段需要定义一个结构体，来记录发送者与接收者的信息，发送的数据都是 `ByteArray` 类型，一般是用流去进行读取，所以需要左移右移操作符进行重载以进行读写。读写用 Qt 提供的 `QDataStream` 即可读写。

当接受信息时，会进行一个判断，首先判断是否是用户间通信，然后根据接收者 `id` 判断接受者是否是当前用户，如果发送者是当前选择的用户，消息会显示到控件上，否则会提示你说有其他用户在给你发消息，但消息不保存仅通知你一下。你必须选择那个用户才能看到它发送的信息。接收报文的流程图如图 4.5 所示。

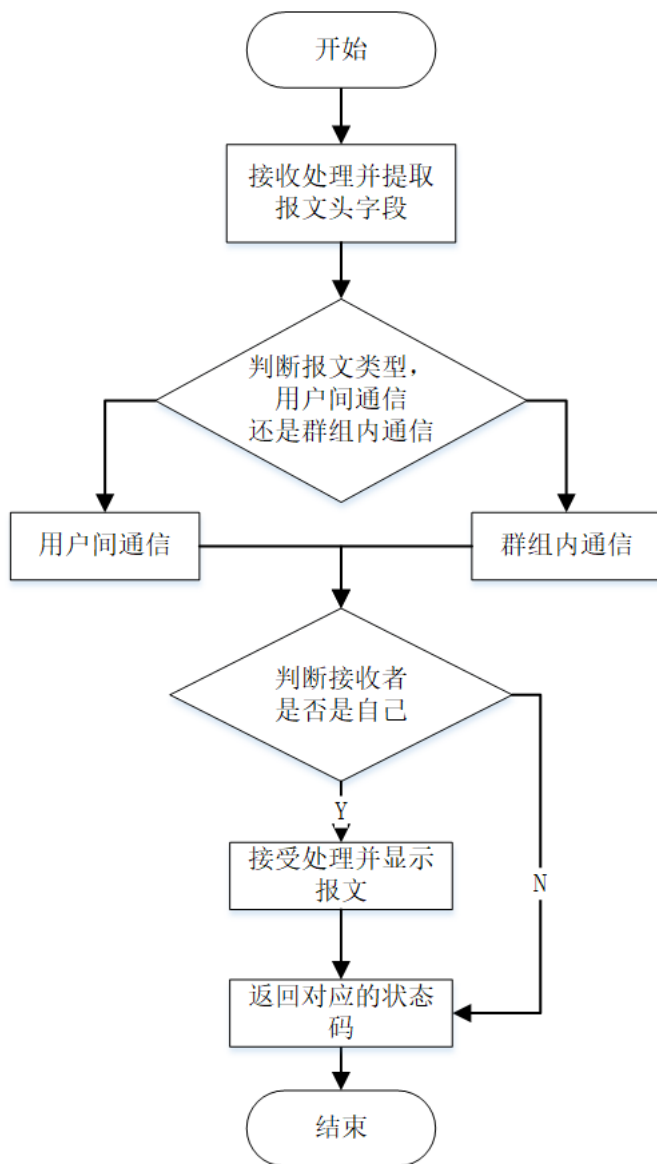


图 4.5 接受报文的流程图

### 4.3.3 添加好友与添加群功能

添加好友的功能，当点击添加时会弹出 AddDialog 界面。AddDialog 用来执行添加用户好友的功能，当主功能界面点击添加好友时会创建一个临时的 AddDialog 变量，AddDialog 会在分配内存后也传入 DataBaseManager 的指针以进行数据库操作。同样会发射一个携带枚举型的信号，交给 SimpleChat 去判断相应状态，是否提示报错等等。

添加好友时，首先会查询当前用户的 id 是否存在，如果存在，则返回对应的状态码。添加好友的流程如图 4.6 所示。

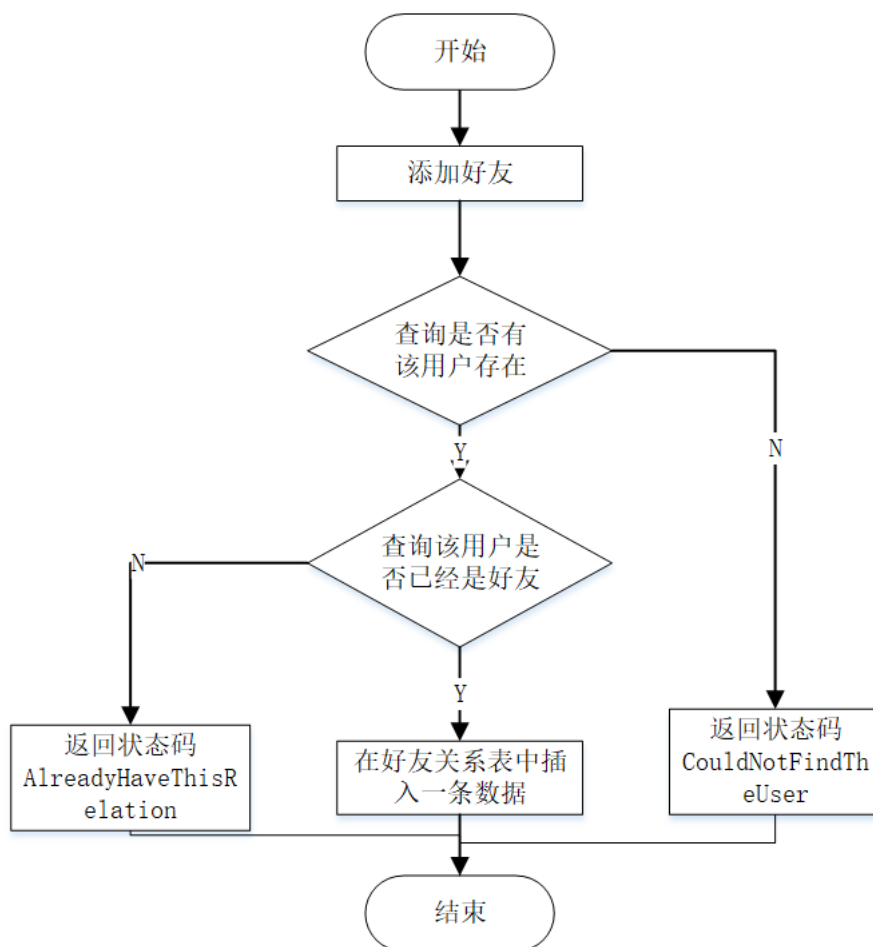


图 4.6 添加好友的流程图

添加群的功能也同理，首先判断该群 id 是否在数据库中存在，如果存在，查询是否已经加入该群聊了，如果以上都不满足，则加入群，加入群就是在 chatGroup\_relation 中插入一条数据来记录已经加入当前群。添加群的流程图如图 4.7 所示。

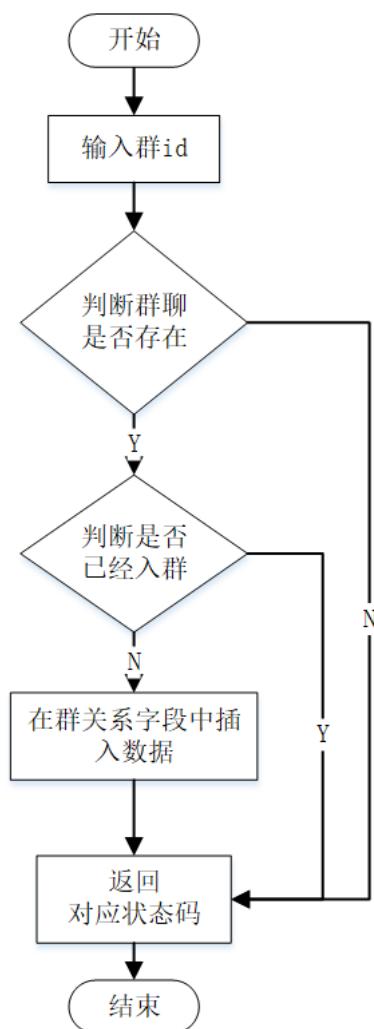


图 4.7 添加群的流程图

#### 4.3.4 创建群聊

群聊的昵称允许重复，但群有一个唯一的 id 值作为群号，在数据库中是自增的 int 型变量，当创建群聊后会对应的创建群关系，即在群关系表中插入一条数据来记录这段关系。之后返回对应的状态码。创建群聊流程如图 4.8 所示。

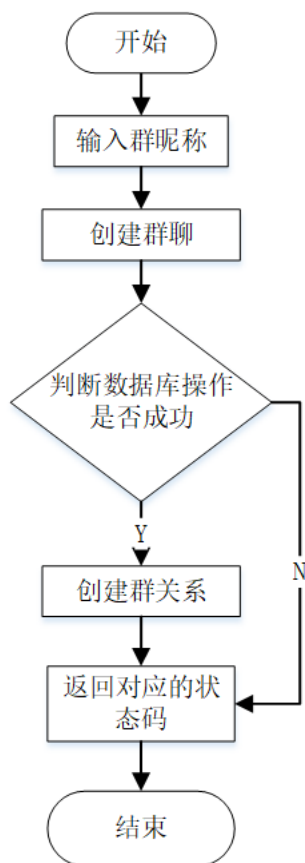


图 4.8 创建群聊流程

#### 4.3.5 获取好友与群组列表

本文创建了一个自定义类 `userWidget`，它继承自一个 `QWidget` 类，它会显示好友的信息并会被添加或删除在 `SimpleChat` 的界面上，当 `SimpleChat` 每次初始化时都会调用 `DataBaseManager` 的 `getAllMyFriends()` 来获取所有的好友信息，它传入一个结构体列表 `List<ST_friendInfo>` 的变量，还有当前用户的 `id`，根据用户 `id` 在数据库中查询相关的所有好友信息，在赋值给这个 `List<ST_friendInfo>` 类型的变量。

然后将得到的信息返回给 `SimpleChat`。得到信息后工厂类 `SimpleChat` 添加对应的 `userWidget` 到本身界面上，每次更新都会删除原先的 `userWidget`，用户信息中也有关于在线的信息，通过在 `SimpleChat` 中添加一个定时器，定时的去获取用户的信息，然后判断用户状态是否在线。在线的用户再调用 `setStyleSheet` 去设置为绿色，表示当前在线。

当点击某个用户时，这是一个自定义控件 `userWidget`，他没有 `clicked` 信号，但可以使用自定义信号，重写他的 `mousePressEvent` 事件，然后发射一个自定义信号 `clicked` 即可，然后在 `SimpleChat` 中将此信号和对应的处理事件连接在一起，比如点击某好友时需要把右边对话框上面的 `QLabel` 赋值为该好友的昵称与 IP 以及在线状态等等，同时也有一个 `Static` 的结构体变量来存储当前好友的信息数据，为了通信做准备工作。当点击一个

好友之后右边的发送按钮才可用，此时才可以进行发送信息。获取用户以及群组信息的流程图如图 4.9 所示。

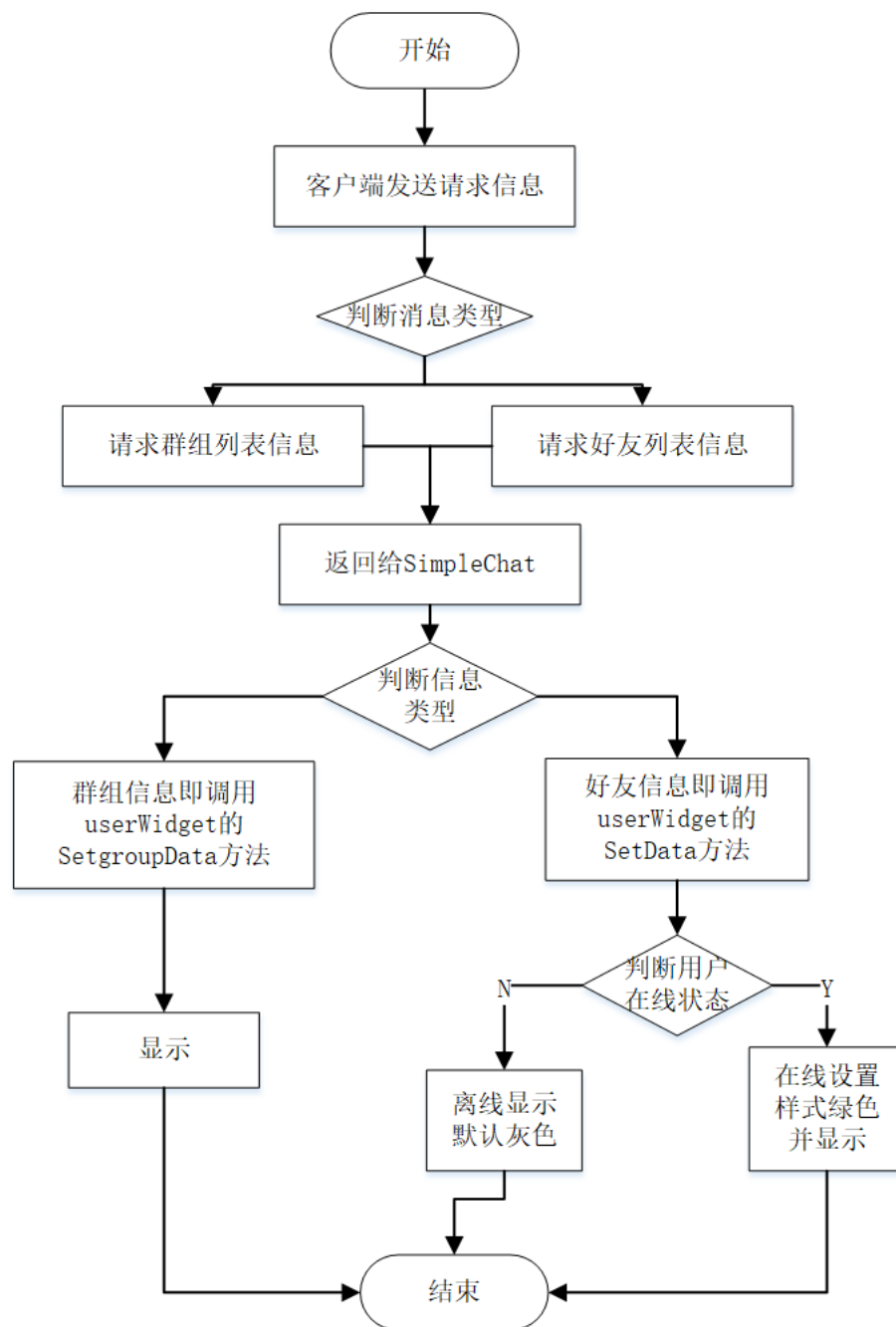


图 4.9 获取用户以及群组信息的流程图



当获取群组列表时，会调用 DataBaseManager 的 GetAllMyChatGroupInfo（）方法，它会传入当前用户的 id，然后还有一个 List<stChatGroup>的变量，查询结果会被赋值给他，来获取当前用户所添加的群组信息。群组信息也会用 userWidget 显示，userWidget 有两个对外的公有函数，setData（）和 setGroupData（），分别将好友或群组的个人信息赋值给 userWidget 的对象，然后显示出来。当向好友列表控件中添加 userWidget 时，如果用户在线，会设置其样式为绿色。获取到的群组信息和用户信息都会存在一个全局变量中，所有的全局变量和结构体定义都被放到一个 pubStruct 的头文件中，以方便全局使用。

## 4.4 数据库设计

### 4.4.1 数据库设计步骤

数据库需要用来存储用户信息，以及用户的 IP 和在线状态等信息，所以需要 user 表。为了存储用户好友间的关系，所以创建了 friend\_relation 表，它需要存储两个用户的 id 来存储这段关系。为实现群聊还需要一个表来记录群组信息，所以创建了 chatGroup 表，为了实现加入群组的功能，还需要创建一个 chatGroup\_relation 来记录用户和群组的关系。

### 4.4.2 数据库概念设计

概念设计用 ER 图来表示，数据库的实体有用户、群组、好友关系和群组关系，系统数据库总体 E-R 图如图 4.10 所示。

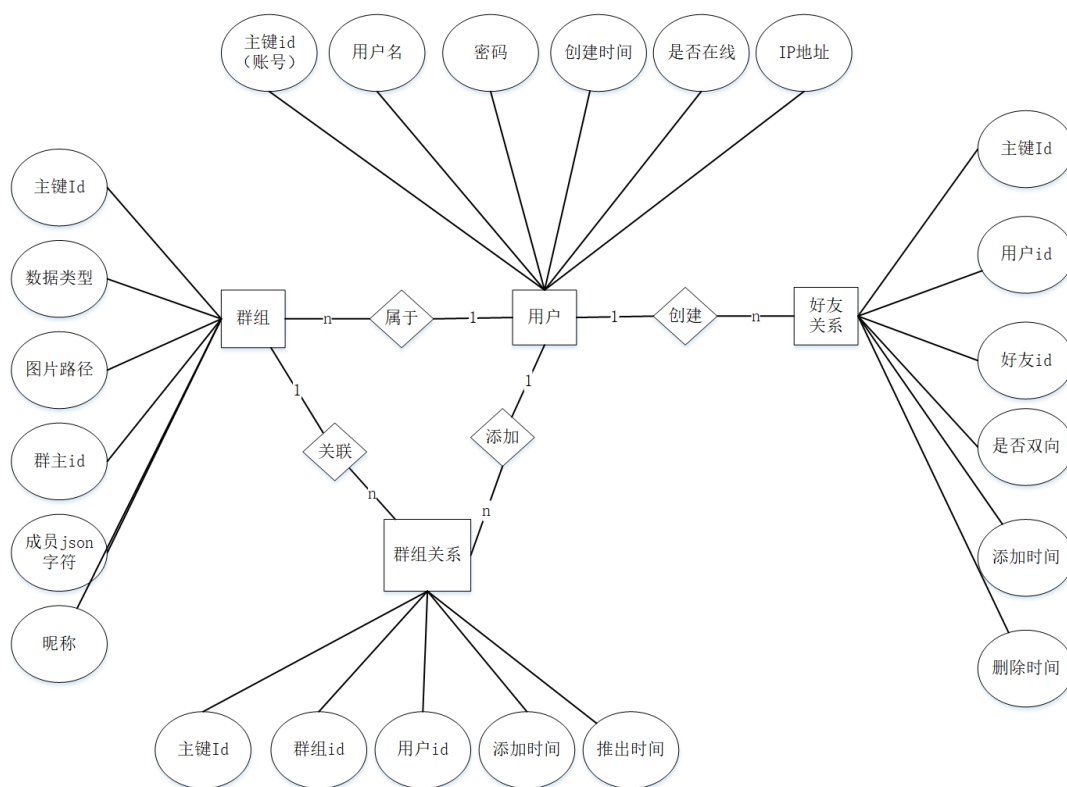


图 4.10 系统数据库总体 E-R 图

首先 user 表与 friend\_relation 表之间的关系是创建关系，用户添加一个好友就会在 friend\_relation 表中插入一条数据。然后 user 表与 chatGroup 表之间的关系既有属于也有创建，首先用户是群组的一员，而用户也可以创建群组。user 表和 chatGroupRelation 表之间的关系是添加，用户加入一个群聊时，即添加一条字段到 chatGroupRelation 表中。chatGroupRelation 表和 chatGroup 表的关系是关联，群组可以借此群组关系来查找群组内的用户以及对应的用户信息。

#### 4.4.3 数据库逻辑设计

用户（user）表，id 是 int 型自增的主键，并且是用户账号。username 是用户名，设置了 unique 约束防止创建相同的用户名，登陆时需要验证的是用户名和密码这两项。ifOnline，数据库中没有 bool 值，所以用了 tinyint，用来记录用户的在线状态，当用户登录成功后会修改这一项值为 1，正常退出软件后会修改这一项值为 0。用户表如表 4.3 所示。

表 4.3 用户表

列名	数据类型	约束类型	注释
id	int(10) unsigned	PRI	主键 id
username	varchar(255)	UNI	用户名
password	varchar(255)		密码
createtime	datetime		创建时间
ifOnline	tinyint(1)		0-offline, 1-online
ipAddress	varchar(40)		

好友关系（friend\_relation）表，id 是 int 型自增的主键，userid 和 friendid 用来记录成为好友的两个用户的 id，也设置了外键与 user 表中的 id 相关联。为避免重复添加，userid 和 friendid 的组合设置了 unique 索引，用来限制好友关系中仅能存在一项来表示两个人为好友，userid 为添加者 friend\_id 为被添加者。好友关系表如表 4.4 所示。

表 4.4 好友关系表

列名	数据类型	约束类型	注释
id	bigint(20) unsigned	PRI	
userid	int(10) unsigned	MUL	用户-id
friend_id	int(10) unsigned	MUL	好友-id
is_out	tinyint(1) unsigned		0 单向, 1=双向
addFriTime	timestamp		加好友时间
deleteFriTime	timestamp		删除好友时间

用户主界面显示自己好友列表时，需要根据用户 id，使用 Sql 的连接和 union 关键字

来进行多表查询，去好友关系表中查找所有的 `user_id` 和 `friend_id`，并根据查找到的 `id` 去 `user` 表中查询对应的用户信息，并将用户信息列表返回给主功能界面。

群组（`chatGroup`）表，`id` 是 `int` 型自增的主键，`GroupOwnerID` 用来表示群主 `id`，当创建群聊的时候会把当前用户设置为此群群主。`members` 是用来保存群成员 `id` 的 `json` 字符串给，每当用户加入群聊时会修改此字段，首先将所有群成员存储进 `map` 中，再准换成 `json` 字符串修改更新 `member` 字段的值。一个群的独立性标志是它的 `id`，就是群号，用户可以通过群号加入群。`nickname` 是群昵称，它允许重复，你可以创建同样名字的群但他们的群号是不同的。

`chatGroup` 表设置了触发器，当进行插入操作后会同样的在 `chatGroup_relation` 表中插入一条数据来记录群关系，表示当创建群组时当前用户便与该群组有了联系。新插入的群组信息是无法得到的，借助触发器可以使用其中的 `new` 关键字，来获取新添加的群组信息。添加的触发器用来确保每次创建群组时即插入一条数据到群组关系。群组表如表 4.5 所示。

表 4.5 群组表

列名	数据类型	约束类型	注释
<code>id</code>	<code>bigint(20) unsigned</code>	<code>PRI</code>	
<code>data_type</code>	<code>varchar(16)</code>		接口状态码
<code>GroupHeadPic</code>	<code>varchar(512)</code>		群头像
<code>GroupOwnerID</code>	<code>int(10) unsigned</code>	<code>MUL</code>	群所有者 <code>id</code>
<code>members</code>	<code>longtext</code>		群成员 <code>json</code>
<code>nickName</code>	<code>varchar(255)</code>		群名称

群组关系（`chatGroup_relation`）表，`id` 是 `int` 型自增的主键，`group_id` 设置了外键与 `chatGroup` 表中的 `id` 字段关联，`user_id` 设置了外键与 `user` 表中的 `id` 字段关联。用户主界面显示自己所加入群聊的信息时，需要根据用户 `id`，去群组关系中查找所有的 `group_id`，并根据 `group_id` 查找对应的群组信息，并将群组信息列表返回给主功能界面。群组关系如表 4.6 所示。

表 4.6 群组关系表

列名	数据类型	约束类型	注释
<code>id</code>	<code>bigint(20) unsigned</code>	<code>PRI</code>	
<code>group_id</code>	<code>bigint(20) unsigned</code>	<code>MUL</code>	群 <code>id</code>
<code>user_id</code>	<code>int(10) unsigned</code>	<code>MUL</code>	成员 <code>id</code>
<code>addTime</code>	<code>timestamp</code>		加群时间
<code>exitTime</code>	<code>timestamp</code>		退群时间

## 第 5 章 系统实现

### 5.1 数据库连接和操作的实现

下面是数据库可视化工具 Navicat 生成的模型图，其中 friend\_relation 表中的 userId 和 friend\_id 字段都设定了是外键，参考表是 user 表中的 id 字段，数据库设置了这样一层保护，可以防止添加未注册的账户为好友。软件在添加好友前也会为了验证去查询一下 user 表中是否有此用户，这也是为了防止误操作，比如开发人员的测试误操作添加一个 user 表中不存在的用户为好友的情况。

同理 chatGroup 中的 GrounOwnerID 字段也是，群组关系中的 group\_Id 和 user\_id 也一样道理。数据库模型图是为了直观地查看外键约束。模型图如 5.1 所示。

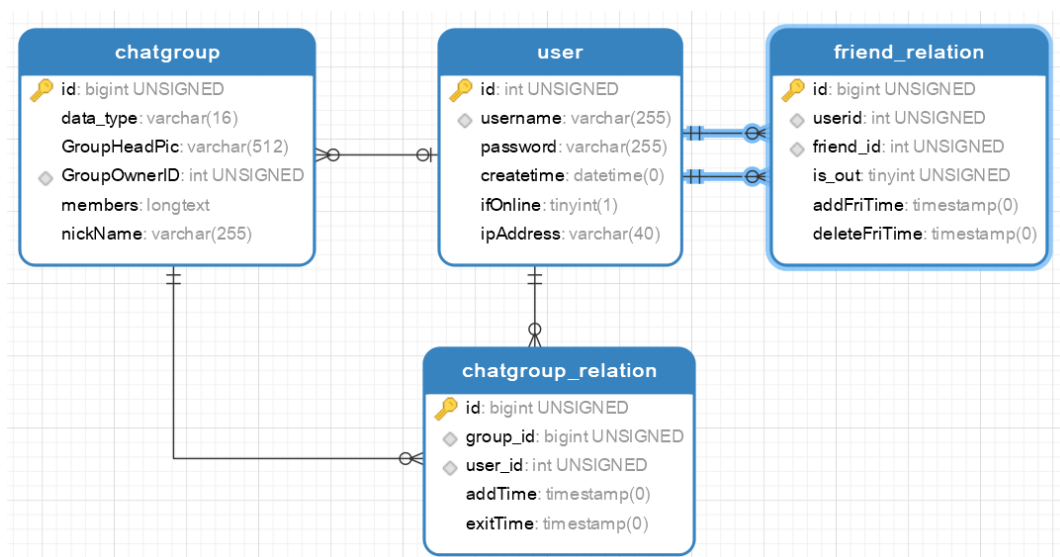


图 5.1 数据库模型图

软件的数据库使用的 MySQL，Qt 版本是 6.2.4，官方下载的开源版本没有 MySQL 驱动库，要想使用需要自行编译 Qt 的源码以生成 MySQL 的驱动库文件，代码都是用链接库的形式使用，使用 CMake 编译出驱动库文件使用即可。

本文创建了一个数据库类 DataBaseManager，这个类是本文专门用来进行数据库处理所创建，数据库需要五项配置参数，在构造函数可以配置好这几项的值。有数据库服务的主机 IP（这里用的是部署本地的服务器，是“192.168.171.1”），端口号一般是 3306，如果想任何用户都可以访问这个数据库，在服务器端命令行用 grant 指令授权给所有人，并且打开你的防火墙权限即可。然后是数据库的名称 simplechatdb，账户默认是 root，密码是安装 MySQL 时设置的密码。

数据库的连接用到了 Qt 中提供的 QSqlDatabase 类，m\_database 是此类型的一个成员

变量，在 `DataBaseManager` 中，首先打开数据库，先判断 `m_database` 是否打开，如果已经打开直接返回即可，否则就设置数据库驱动类型为 `QMYSQL`，然后调用 `m_database` 的 `open` 方法，打开成功会返回 `true`，否则会打印 `lastError()`，并借此来判断错误原因。

Qt 中提供了 QSqlQuery 方法，它的构造函数的参数是一个 QSqlDatabase 类型的值，执行对应的操作时，先定义一个 QSqlQuery 的变量，然后准备 QString 类型的 sql 语句，其中可以添加变量，但变量的值要由 QSqlQuery 的 bindValue 来添加，然后执行 exec () 函数，如果成功返回 true，否则为 false。如果是查询语句的话，query 的 next () 可以得到返回结果，可以根据 query.next () 是否为空来判断 sql 语句是否执行成功，并返回对应的状态码。Insert、update 和 delete 语句的返回值是被影响的行数，如果值大于零那就意味着 sql 语句执行成功了。

## 5.2 系统功能的实现

总的类图显示了本软件所用到的所有的类和结构体类型，包括类的成员变量和实现具体功能的成员函数。VisualStudio 生成的总的类图如图 5.2 所示。

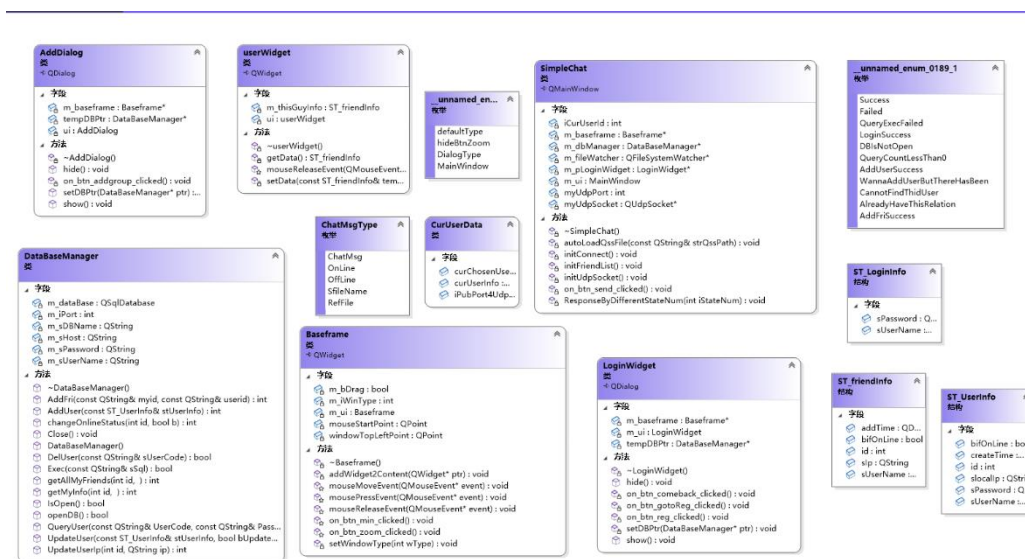
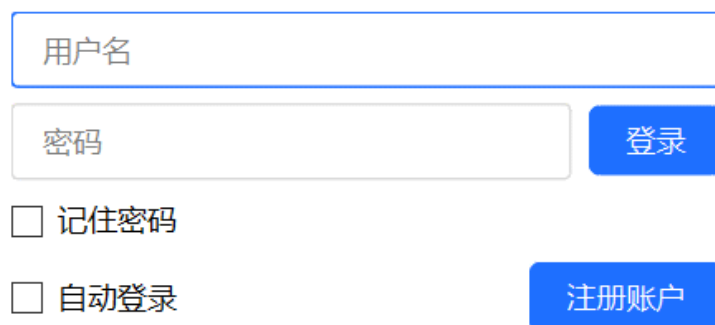


图 5.2 VisualStudio 生成的总类图

### 5.2.1 登录注册功能实现

用户输入用户名和密码后，在数据库中查询结果，并返回对应的状态码，借此来显示对应的提示。登录窗口的功能界面如图 5.3 所示。

普通聊天软件



The login interface features a title '普通聊天软件' at the top. Below it is a form with three input fields: '用户名' (Username), '密码' (Password), and a checkbox for '记住密码' (Remember password). To the right of the password field is a blue '登录' (Login) button. Below the checkbox is another checkbox for '自动登录' (Auto login), and to its right is a blue '注册账户' (Register account) button. A close button (X) is located in the top right corner.

图 5.3 登录的功能界面

注册时，首先进行判空，再判断是否已经存在该用户名，然后再进行注册，成功与否会给出对应的提示。注册界面如图 5.4 所示。



The registration interface has a title '普通聊天软件' at the top. It contains three input fields: '用户名' (Username), '密码' (Password), and another '密码' (Password) field. At the bottom are two blue buttons: '注册' (Register) and '返回' (Return). A close button (X) is in the top right corner.

图 5.4 注册功能界面

### 5.2.2 主界面功能实现

主界面会实时的请求当前用户的好友列表信息以及群组信息，当有好友上线时，则设置颜色为绿色，否则为灰色。主功能界面可以进行好友间通信和群组内通信。点击好友或群组则分别显示对应的列表。主功能界面如图 5.5 所示。



图 5.5 主功能界面

好友间如果进行通信，必须选中列表中的一个好友，发送按钮会判断当前是否有一个显示好友的 `userWidget` 是否被点击，当某一个 `userWidget` 被点击时才可用，并且会设置一个 `static` 的全局结构体变量来表示当前选择通信对象并存储其中的用户信息，每个 `userWidget` 在创建时即被赋值并且绑定了对应的点击事件，点击后即可发送信息。当前的通信好友的名称等信息会显示在消息框之上。聊天功能如图 5.6 所示。

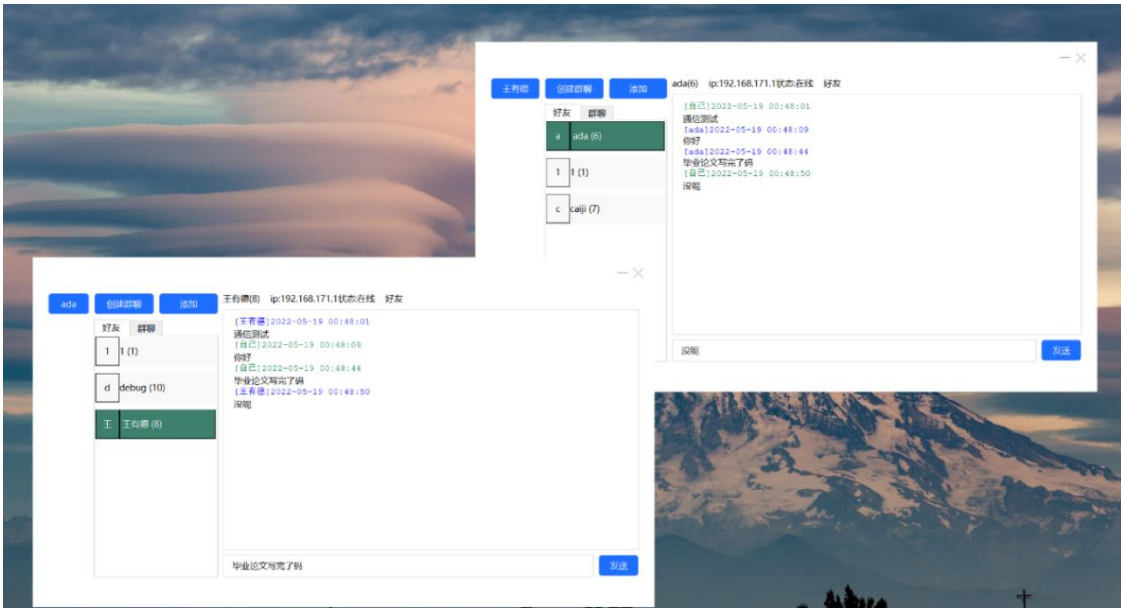


图 5.6 聊天功能示例

群组内聊天需选中当前群组，处于同一群组内的用户都会接收到该信息。选中群组后群组的名称会显示在右边的消息框之上。接收到消息时，消息框中会显示信息发送时间和发送者的 id 账号。聊天功能如图 5.7 所示。

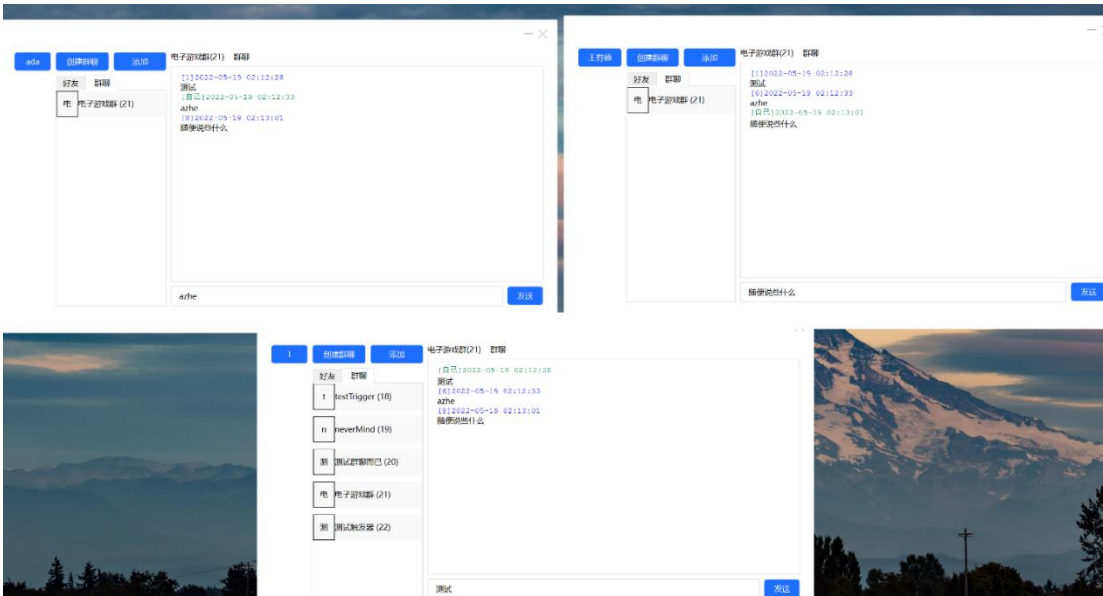


图 5.7 群组聊天功能示例



### 5.2.3 添加好友与添加群功能实现

加好友与加群是一组互斥的按钮，当点击某个按钮后（默认是添加好友），输入账号即可添加成功，并根据不同的情况给出不同的提示。添加好友与添加群的对话框功能界面如图 5.8 所示。



图 5.8 添加好友与添加群的对话框功能界面

### 5.2.4 创建群聊功能实现

CreateChatGroup 也是作为一个临时变量存在的，它也有自定义的 setData 和 setDbPtr 公有函数，当点击创建群聊调用 DataBaseManager 中的创建群聊方法。由于创建群聊的人本身也加入了群聊，所以也需要在 chatgroupRelation 中插入一条字段来证明这段关系，由于无法知道刚刚创建的群组 id，所以需要使用触发器来解决这件事。创建群聊的对话框功能界面如图 5.9 所示。

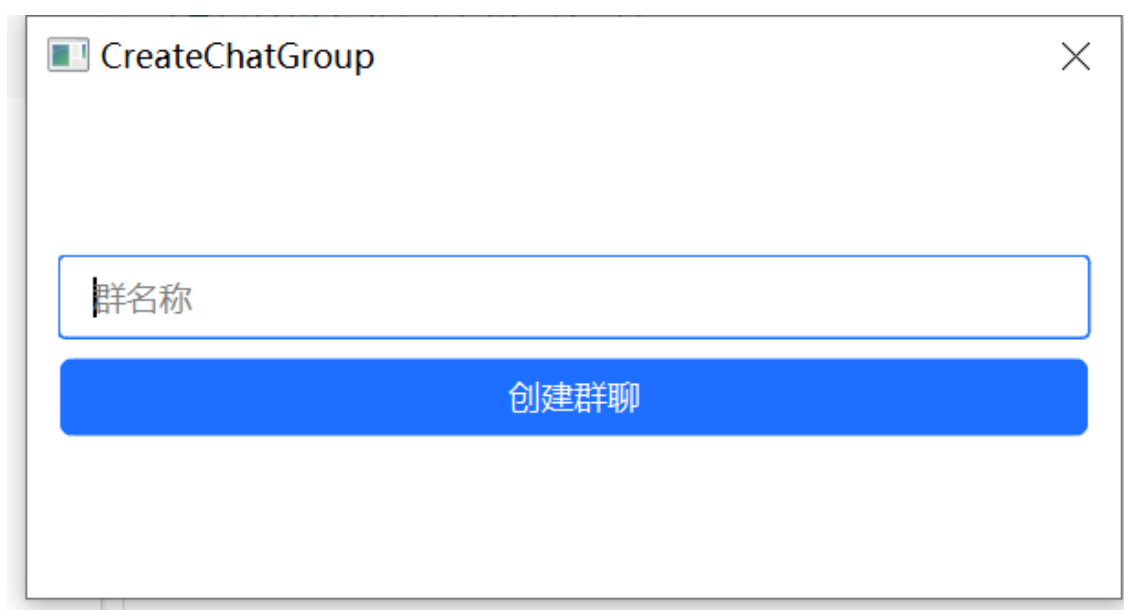


图 5.9 创建群聊的对话框功能界面

## 第 6 章 软件测试

### 6.1 系统测试准备

软件的运行需要配置对应的环境，程序已经选择了 `release` 编译，但生成的可执行文件打包放在单独的环境仍然无法运行。若要软件能独立够运行起来，这需要配备好程序运行的库，比如安装 Qt 或者 MSVC 并配置好环境变量，但实际上软件依赖的只是对应的链接库文件而已，只需要把所需的链接库文件拷贝到软件所在目录即可。

Qt 提供了补全库文件的指令 `windeploymt`，但如果是一些第三方库比如 `mysql` 的驱动包也需要你手动将库文件拷贝软件的 `release` 文件夹下。

### 6.2 系统测试用例设计

登录模块测试用例，用来检验登录功能是否完善，登录模块测试用例如表 6.1 所示。

表 6.1 登录功能测试用例

测试点	期望结果	测试结果	结果
账号: ada 密码: 空	提示密码不能为空	提示密码不能为空	通过
账号: 空 密码: 123	提示账号不能为空	提示账号不能为空	通过
账号: zxc 密码: 111	账号不存在，无法登录	账号或密码错误	通过

登录模块测试用例，用来检验注册功能是否完善，注册模块测试用例如表 6.2 所示。

表 6.2 注册功能测试用例

测试点	期望结果	测试结果	结果
账号: ada 密码: 空 重复密码	提示密码不能为空	提示密码不能为空	通过
账号: ada 密码: 1 重复密码: 空	提示重复密码不能为空	提示密码不能为空	通过
账号: ada 密码: 1 密码: 1	提示已注册该账户	提示已注册该账户	通过

添加好友模块测试用例，用来检验添加好友功能是否完善，添加好友模块测试用例如表 6.3 所示。

表 6.3 添加好友功能测试用例

测试点	期望结果	测试结果	结果
账号：1	提示已经有该好友	提示已经有该好友	通过
账号：2304	提示无法找到用户	提示无法找到用户	通过
账号：7	成功	成功	通过

添加群组模块测试用例，用来检验添加群组功能是否完善，添加群组模块测试用例如表 6.4 所示。

表 6.4 添加群组功能测试用例

测试点	期望结果	测试结果	结果
账号：1	提示已经有该群组	提示已经有该群组	通过
账号：2304	提示无法找到群组	提示无法找到群组	通过
账号：7	成功	成功	通过

好友间聊天模块测试用例，用来检验好友间聊天功能是否完善，好友间聊天模块测试用例如表 6.5 所示。

表 6.5 测试好友间聊天用例

测试点	期望结果	测试结果	结果
发送：你好，你在做什么	发送成功	发送成功	通过
发送：<h1 style="color:red;">duck</h1>	发送成功	发送成功	通过
发送：&*(^(*)*)*&^&*	成功	成功	通过

群组内聊天模块测试用例，用来检验群组内聊天功能是否完善，群组内聊天模块测试用例如表 6.6 所示。

表 6.6 测试群组内聊天用例

测试点	期望结果	测试结果	结果
发送：你好，你在做什么	发送成功	发送成功	通过
发送：<h1 style="color:red;">duck</h1>	发送成功	发送成功	通过
发送：&*(^(*)*)*&^&*	成功	成功	通过

## 结论

局域网通信软件的主要用户是军工企业或者是一些研究所，由于保密性质而不允许连接公网，了解这一情况后，对本软件进行了需求分析。

本软件实现了通信软件的一些简单的功能，主要功能就是用户可以进行好友聊天和群组内聊天，以及添加好友和群组，创建群组等功能，能在一定程度上能满足用户的基础通信需求。

软件虽然实现了基础的通信功能，但在功能上仍然有很多需要添加的地方，比如能记录聊天记录，或者是进行文件传输，随着通信软件的发展，如今的主流通信软件都有着丰富繁多的功能。本软件与商业化软件还有着很大的区别，在软件测试方面，真正的商业化软件要经过大量的系统性的测试，这不是几张功能测试表所能完成的，因此本软件还需要很大的提高，来使细节更加完善。

此次软件的开发过程中，我学习了很多东西，并且发现了自身很多的不足，从软件设计到开发阶段，涉及到很多不了解的内容，在这过程中涉及到的很多问题都通过自己解决，这锻炼了我解决问题的能力以及增加了将设计模式在实际中使用的经验。

## 参考文献

- [1]何宇轩. Simple-chat-software[EB/OL]. 2022[2022. 6. 1]. <https://github.com/sleepy2con/Simple-chat-software>.
- [2]侯捷. STL 源码剖析. 华中大学出版社. [M]. 2011:1-1.
- [3]Scott Meyers. More Effective C++. [M]. 2011:1-1.
- [4]Stephen Prata. C++ Primer Plus[M]. 6. 人民邮电出版社, 2012 :1-2.
- [5]詹姆斯·F. 库罗斯. 计算机网络: 自顶向方法[M]机械工业出版社, 2018 :1-2.
- [6]王维波, 栗宝鹃, 侯春望 Qt 5.9 C++开发指南[M]人民邮电出版社, 2018 :1-2.
- [7]Scott Meyers. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. [M]. 2011 :1-2.
- [8] 陆文周 Qt5 开发及实例(第四版) [M]中国工信出版社, 2019 :1-2.
- [9]杨志超, 张玉芳. 习近平关于力戒形式主义重要论述的逻辑理路[J]. 中共山西省委党校学报, 2021, 44(05):3-7.
- [10]张近山. 对新的形式主义要露头就打[N]. 衢州日报, 2021-11-05(005).
- [11]石磊. 力戒形式的历史经验及当代启示[N]. 济南日报, 2021-09-30(A10).
- [12]周敬青. 论靶向整治形式主义官僚主义的制度导向[J]. 中共中央党校(国家行政学院)学报, 2021, 25(05):55-63.
- [13]卫涛. 纠治“四风”不止步 激浊扬清树新风[N]. 抚州日报, 2021-09-24(A01).
- [14]邓斌, 龚照绮. 基层减负的治理困境及梳理——以 C 市 Q 区市场监督管理局“形式主义”整治为例[J]. 重庆社会科学, 2021(09):111-121.
- [15]梁新. 防止层层加码沦为形式主义[N]. 中国劳动保障报, 2021-11-05(008).
- [16]王士明. 思想汇报没必要“统一格式”[N]. 解放军报, 2021-11-10(007).
- [17]中国社会科学院政治学研究所研究员 房宁. “顶格管理”逼得基层搞形式主义[N]. 北京日报, 2020-06-08(013).
- [18]关严. 通信软件的发展现状与分析[J]. 通信世界, 2005, (14):37-38.
- [19]新疆维吾尔自治区纪委监委政策法规研究室. 坚决整治形式主义 切实减轻基层负担[N]. 中国纪检监察报, 2021-10-28(007).

## 致谢

首先，感谢周晓宇老师在我论文撰写时候给我的细致批注，论文的格式是很重要的部分。其次，感谢 C++ Primer Plus 这本书，这是我的 C++入门书籍，它非常通俗易懂引领我学习了 C++这门编程语言并且让我决定把 C++作为以后的编程职业的主要语言。感谢北京中航安为科技有限公司的同事们，在这家公司的实习经历让我学习了 Qt 框架，并且接触了商业化项目的开发流程，这些经验对于这个软件的开发发挥了很重要的作用。这个软件目前仍有很多地方值得去优化完善，它还不能称作优秀的软件，但开发它的过程中仍然让我学习了一些新的东西。

开发软件的过程中会有很多正向反馈，你会有一种创作欲望，还有边学习边解决问题的过程，我很享受解决问题的成就感，但当遇到某些重复性的东西时难免也会给人带来厌倦，包括一些充满形式主义的东西，我很讨厌做这些东西，但这也是必不可少的一部分，它也有着一定的作用。

再次感谢我周围对我有所帮助的人，你们的优秀品质影响了我，我的学生时代结束了，即将步入正式长久的工作状态，我已经做好了准备。