

西安科技大学

学位论文诚信声明书

本人郑重声明：所呈交的学位论文（设计）是我个人在导师指导下进行的研究（设计）工作及取得的研究（设计）成果。除了文中加以标注和致谢的地方外，论文（设计）中不包含其他人或集体已经公开发表或撰写过的研究（设计）成果，也不包含本人或其他人在其它单位已申请学位或为其他用途使用过的成果。与我一同工作的同志对本研究（设计）所做的任何贡献均已在论文中做了明确的说明并表示了致谢。

申请学位论文（设计）与资料若有不实之处，本人愿承担一切相关责任。

学位论文（设计）作者签名：

日期：

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：在校期间所做论文（设计）工作的知识产权属西安科技大学所有。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文（设计）被查阅和借阅；学校可以公布本学位论文（设计）的全部或部分内容并将有关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和汇编本学位论文。

保密论文待解密后适用本声明。

学位论文（设计）作者签名：

指导教师签名：

年 月 日

论文题目：基于 Qt 的即时通信系统的设计与实现

专业：电子信息工程

学生：凤园迪

签名：_____

指导教师：黄健

签名：_____

摘 要

本即时通信系统采用 Qt 开发平台，服务端数据库采用 Mysql，利用 MVC 设计模型搭建客户端和服务端的代码框架，开发出一套基于 Qt 的 C/S 架构的即时通信系统软件，为用户提供快速便捷的信息交流方式。客户端与服务器通过 TCP 进行连接，如在线保持、发送消息、更改信息等等都是通过服务器的转发；文件收发模块单独采用 UDP 连接，一方作为服务器，一方作为客户端，且该功能在私聊模块中有。

本即时通信系统实现了注册登录、个人信息管理、联系人管理、即时通信、群功能、文件收发等主要功能。用户通过简单的信息注册，便可通过系统分配的 ID 账号登录系统，随即可以进行联系人，群的添加等操作，就可以同其他在线用户进行即时通信。该即时通信系统不但可以处理传统的通信需求，而且也能够扩展以适应新型的网络应用，如文件传输。同时也将联系人、个人信息、群功能细化，使得用户方便管理。

即时通信系统软件拥有广泛的应用前景，通过不断的丰富聊天方式、增强服务性能和安全性，可以使得用户有更优质的应用体验。

【关键词】 即时通信，Qt, Mysql，MVC 设计模型，C/S 架构

【论文类型】 应用型

Title: Design and implementation of instant messaging system based on Qt

Major: Electronic and information engineering

Name: Feng YuanDi

Signature:_____

Supervisor: Huang Jian

Signature:_____

Abstract

This instant communication system uses Qt development platform, using Mysql server database, using the MVC design model to build the client and server code framework, developed a set of C/S architecture based on Qt instant communication system software, to provide users with fast and convenient information communication. The client and the server are connected through TCP, for example, the online request for sending messages is forwarded by the server. The file transceiver module adopts UDP connection separately. One party ACTS as the server and the other as the client, and this function is available in the private chat module.

This instant messaging system has realized the main functions of registration, login, registration, registration, login, personal information management, focal contact management, etc. Users through a simple registration information, you can through the system assigned ID to login system, then can make contacts, such as group to add operation, can communicate with other online users by instant. The instant messaging system can not only deal with the traditional communication needs, but also be extended to meet the new network applications, such as file transfer. At the same time, the contact, personal information and group functions will be refined to make it easy for users to manage.

The instant messaging system software has a wide application prospect. Through enriching the chat method continuously, it can enhance the service performance and security, so that users can have a better application experience.

【Key words】 Instant messaging, Qt,Mysql, MVC design model, C/S architecture

【Type of Thesis】 Application

目录

1 绪论.....	1
1.1 系统的开发背景.....	1
1.1.1 即时通信系统的发展.....	1
1.1.2 即时通信系统的现状.....	1
1.2 系统实现的目标.....	2
1.3 系统开发的意义.....	2
2 即时通信系统分析.....	3
2.1 系统的设计目标.....	3
2.2 系统的可行性分析.....	3
2.2.1 用户群体与市场分析.....	3
2.2.2 技术能力分析.....	3
2.2.3 可行性分析总结.....	4
3 系统总体设计.....	5
3.1 系统整体功能设计.....	5
3.2 通信协议的制定.....	5
3.2.1 基于 socket 的网络编程.....	5
3.2.2 TCP 和 UDP 协议的简单介绍.....	6
3.2.3 套接字建立网络连接的简单过程.....	6
3.2.4 本系统通信协议的设计.....	7
3.3 服务端设计.....	10
3.3.1 代码结构.....	10
3.3.2 数据库设计.....	11
3.3.3 流程图.....	13
3.4 客户端设计.....	14
3.4.1 代码结构.....	14
3.4.2 流程图.....	14
4 详细设计与实现.....	16
4.1 登录与注册功能的设计与实现.....	16
4.1.1 登录模块.....	16
4.1.2 注册模块.....	17
4.2 个人管理的设计与实现.....	18
4.2.1 个人信息查看和更改.....	18
4.2.2 密码更改.....	18
4.2.3 登录状态更改.....	19
4.2.4 新消息管理.....	19
4.3 联系人管理的设计与实现.....	20
4.3.1 添加/删除联系人.....	20
4.3.2 好友分组管理.....	21
4.3.3 联系人备注更改.....	21
4.3.4 搜索好友.....	22
4.3.4 即时通信.....	22
4.4 群管理的设计与实现.....	24

4.4.1 创建/解散群.....	24
4.4.2 添加/删除群成员.....	25
4.4.2 添加/删除群.....	26
4.4.3 查看群信息.....	27
4.4.4 群成员备注更改.....	28
4.4.5 群成员信息查看.....	29
4.4.5 群成员添加为联系人.....	29
4.4.6 即时通信.....	30
5 系统测试.....	31
5.1 系统测试分析.....	31
5.2 功能测试.....	31
5.2.1UDP 丢包问题.....	31
5.2.2 服务端并发连接问题.....	32
5.3 测试总结.....	33
结论.....	34
致谢.....	35
参考文献.....	36

1 绪论

1.1 系统的开发背景

1.1.1 即时通信系统的发展

即时通信（Instant Messaging）是一个实时通信系统，允许两人或多人使用网络实时的传递文字消息、文件、语音与视频交流。是目前网络上最为流行的通讯方式，各种各样的社交服务软件也层出不穷；服务商为了吸引用户也开发出越来越丰富的通讯服务功能。毫无疑问，即时通信网络已经成为真正的信息交流的高速公路。

IM 最早的创始人是三个以色列青年，是他们在 1996 年开发出来的，取名叫 ICQ。1998 年当 ICQ 注册用户数达到 1200 万时，被 AOL 看中，以 2.87 亿美元的天价买走。2008 年 CQ 有 1 亿多用户，主要市场在美洲和欧洲，已成为世界上最大的即时通信系统。

目前，中国市场上的企业级即时通信工具主要包括：信鸽、视高科技的视高可视协同办公平台、263EM、群英 CC2010、通软联合的 GoCom、腾讯公司的 RTX、IBM 的 Lotus Sametime、点击科技的 GKE、中国互联网办公室的 imo、中国移动的企业飞信、华夏易联的 e-Link、擎旗的 UcStar 等。相对于个人即时通信工具而言，企业级即时通信工具更加强调安全性、实用性、稳定性和扩展性。

1.1.2 即时通信系统的现状

而对于企业级用户来说，一个重要问题就是大多数即时通信系统是公开的，这意味着用户只要知道另一个用户的即时通信地址，他就可以直接向对方发送信息，这对于员工向外界泄露企业的商业秘密非常便利。而且即时通信的主要特点是两台终端之间可以直接进行交流，而不必通过任何第三方服务器中转。这就使得网络监管对即时通信用户的数据交换进行监控的难度增加，这让企业管理者大为头疼。IDC 公司的数据显示，全球超过 2000 万的商务人士都在使用即时通信工具，而且这一数字将在 2005 年之前猛升至 3 亿人。即时通信的广泛使用以及它本身缺乏安全功能的特性，为向它添加加密、归档和日志功能的产品创造出了很大需求。

1.2 系统实现的目标

根据即时通信的文档需求及选题报告要求，本系统主要实现以下的功能：

- 1) 支持用户注册、登录验证功能。
- 2) 支持客户端之间的消息传输和文件传输。
- 3) 支持多用户之间的群聊。
- 4) 支持用户添加好友、查看好友信息、好友分组管理、备注更改。
- 5) 支持用户创建群、解散群、添加群和退出群，同样也支持添加群成员，备注更改等功能。
- 6) 支持用户查看和更改自己信息，更改密码，密保答案找回密码，搜索好友等功能。
- 7) 支持用户密码加密存储和登录时的记住密码功能。
- 8) 支持新消息接收提示功能。

1.3 系统开发的意义

本系统旨在开发一套能够即时通讯的软件，用户可以通过此软件系统加强网络之间的信息沟通，有效的节省沟通双发的时间和经济成本。相比于现在流行的 QQ 和微信软件来说还有巨大的差距，更多体现在安全性、聊天方式多样性、业务拓展性能和服务器性能等，但是开发本系统更多的意义在于了解即时通信系统的开发过程，以及提升相关专业应用的经验，该即时通信系统在后期的开发中也需要增加功能特性以满足用户需求。

2 即时通信系统分析

2.1 系统的设计目标

该即时通信系统的实现需要设计三大部分：客户端、通信协议、服务端，。

服务端需要设计数据库表及其相应的增删改查服务，比如创建用户信息表、用户联系人表、群信息表、群成员表等等。

通信协议重点需要设计数据包格式和消息类型，数据包格式设计是通信的基础，而消息类型重点在于描述数据包所携带内容。同时传输层协议的选取也很重要，TCP 相比 UDP 更为安全可靠，因此使用 TCP 协议做消息传输，UDP 相比 TCP 传输效率高，因此使用 UDP 协议做文件收发。

客户端主要需要设计与用户操作相关的操作界面和相应的控制层，控制层通过方法调用或者信号槽来获取用户操作的数据，打包整合，通过设计的通信协议完成客户端和服务端的交互。

2.2 系统的可行性分析

2.2.1 用户群体与市场分析

本系统经过此次开发只实现简单的局域网即时通信，比如私聊和群聊，操作界面简单实用，适用于一些追求简洁美用户群体。

本系统可以提高聊天方式的丰富性，比如增加表情，动态图，视频或者语音通话等，功能性能提高同样会吸引更多的用户使用，因此该系统经过后期再开发同样会拥有更广泛的使用人群和市场前景。

2.2.2 技术能力分析

根据本系统设计的功能需求，采用 Qt 开发平台和 MySQL 数据库进行系统开发，由于 Qt 开发平台 UI 控件很多，因此使得基于 C/S 架构的应用软件开发变得方便快捷，MySQL 作为快速、多线程、多用户和健壮的 SQL 数据库服务器，使得系统能够在大量访问量的情况下依然保持较高的运行速度。因此，本系统的设计在技术层面上是完全可行的。

2.2.3 可行性分析总结

综合考虑信息沟通方面的需求、先进技术以及在市场前景等方面，即时通信系统有着巨大的发展前景，而即时通信系统正是基于这种考虑所出现，因此本系统的开发是具有较强的可行性。

3 系统总体设计

3.1 系统整体功能设计

本系统的功能模块主要包括：注册模块、登录模块、个人信息管理模块、联系人管理模块、群管理模块等五大功能模块，其各个功能模块下又细分了一些小模块。系统的各个功能模块的功能框图，如图 3-1 所示。

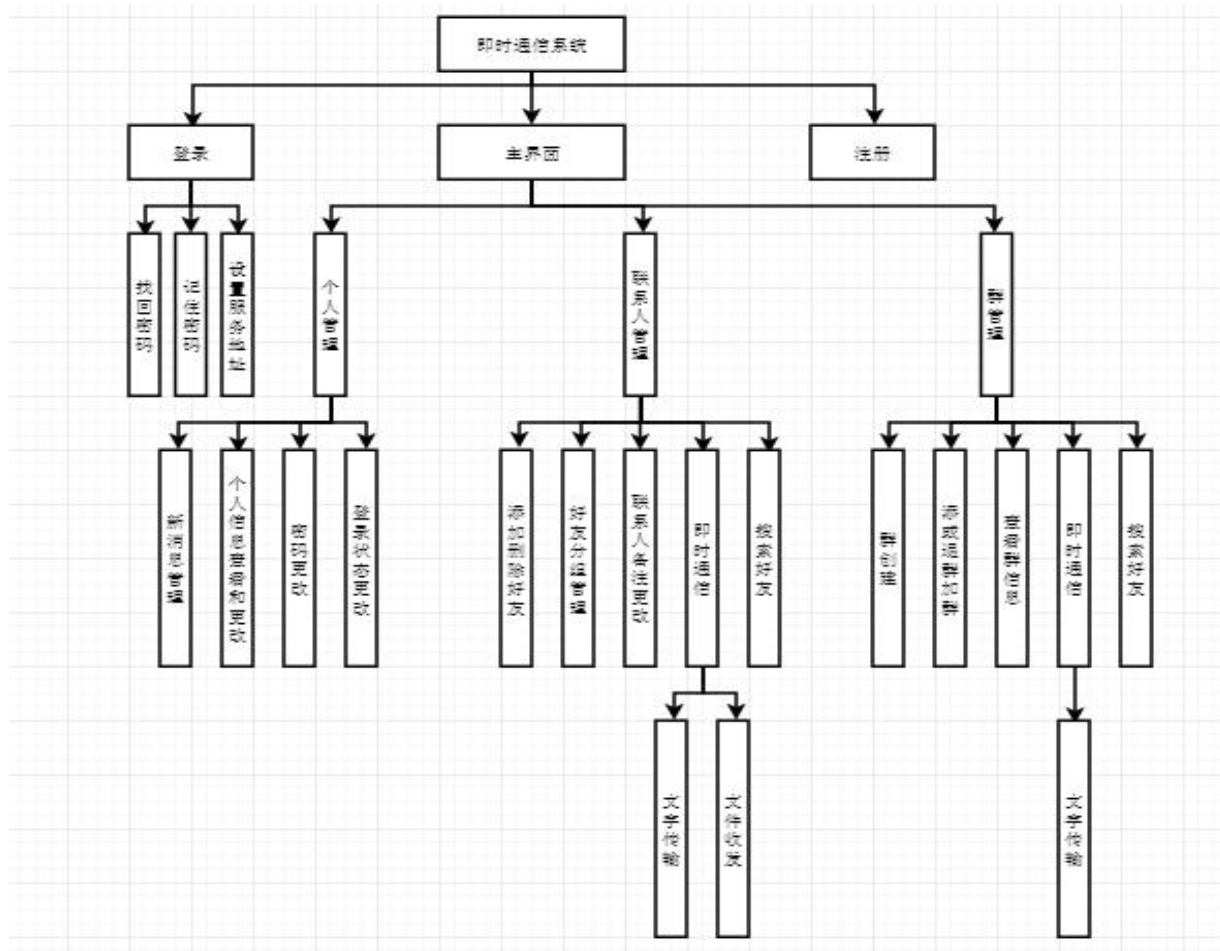


图 3-1 系统功能框图

3.2 通信协议的制定

3.2.1 基于 socket 的网络编程

Socket 是介于应用层与 TCP/IP 协议族通信的中间软件抽象层，它是一组接口。网络层的 IP 地址可以唯一的标识网络中的主机，而传输层的协议和端口可以唯一标识主机应

用程序（进程），这样可以利用三元组（IP 地址，协议，端口号）就可以标识网络中的进程了，网络中的进程通信就可以利用这个标志与其他进程进行通信。

由于传输层的协议有 TCP 和 UDP 两种，因此 socket 连接方式分为两种，一种是面向连接的流式套接字，一种是无连接的数据报套接字。下一节将重点介绍这两种协议。

3.2.2 TCP 和 UDP 协议的简单介绍

TCP 是一种面向连接的，可靠的基于字节流的传输层通信协议，TCP 层位于 IP 层之上，应用层之下的中间层。不同主机的应用层之间经常需要可靠的，像管道一样连接，但是 IP 层不提供这样的流机制，而是提供不可靠的包交换。

(1) 连接建立

TCP 是传输层协议，使用三次握手建立连接，才可以开始传输数据。

① 客户端发送 SYN (SEQ=x) 报文给服务器端，进入 SYN_SEND 状态。

② 服务器端收到 SYN 报文，回应一个 SYN (SEQ=y) ACK(ACK=x+1) 报文，进入 SYN_RECV 状态。

③ 客户端收到服务器端的 SYN 报文，回应一个 ACK(ACK=y+1) 报文，进入 Established 状态。

(2) 连接终止

终止一个连接需要经过四次挥手，这是由于 TCP 的半关闭造成的。

① 某个应用进程首先调用 close，称该端执行“主动关闭”（active close）。该端的 TCP 于是发送一个 FIN 分节，表示数据发送完毕。

② 接收到这个 FIN 的对端执行“被动关闭”（passive close），这个 FIN 由 TCP 确认。

③ 一段时间后，接收到这个文件结束符的应用进程将调用 close 关闭它的套接字。这导致它的 TCP 也发送一个 FIN。

④ 接收这个最终 FIN 的原发送端 TCP（即执行主动关闭的那一端）确认这个 FIN。既然每个方向都需要一个 FIN 和一个 ACK，因此通常需要 4 个分节。

UDP 是一种无连接协议，UDP 不提供数据包分组，组装和不能对数据包进行排序的缺点，也就是说，当报文发送之后，无法得知其是否安全到达对端，UDP 报文没有可靠性保证、顺序保证和流量控制字段等，可靠性较差。但是正因为 UDP 协议的控制选项较少，在数据传输过程中延迟小、数据传输效率高，适合对可靠性要求不高的应用程序，或者可以保障可靠性的应用程序。

3.2.3 套接字建立网络连接的简单过程

本即时通信系统采用 QNetWork 库的应用来设计和实现基于 TCP 的消息传输等功能，这里用简单的表示出 TCP 建立连接的过程。面向连接的流方式的调用过程如图所示。

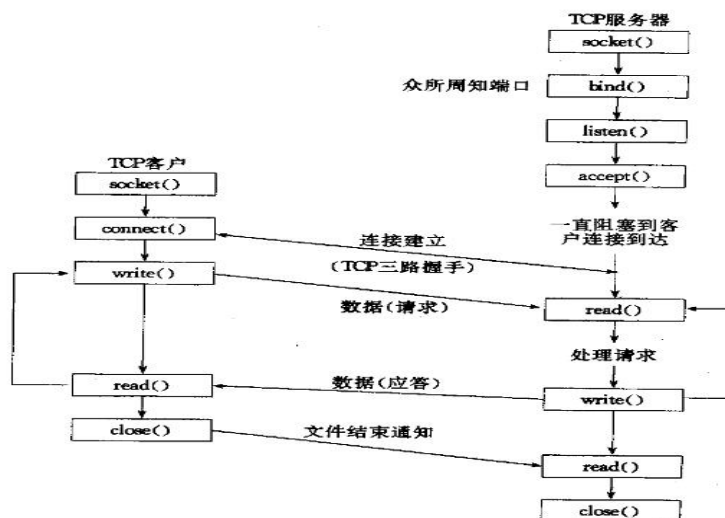


图 3-2

3.2.4 本系统通信协议的设计

(1) 应用层协议设计

数据包格式定义如图表表 3-3。

图表 3-3 数据包格式表

版本号	头长度	消息类型	数据长度
数据内容			

对于上述所述数据包格式中的版本号，定义为 `QDataStream::Qt_4_7`，而消息类型定义为图表 3-4。

序号	消息类型	类型解释
1.	CHECK_CONNECTION	检查是否能与服务器连接
2.	LOGIN	登录
3.	REGISTER	注册
4.	GET_QUESTION_ANSWER	获取密保问题以及答案
5.	GET_QUESTION_ANSWER_SUCCESS	获取密保问题以及答案成功
6.	GET_QUESTION_ANSWER_FAIL	获取密保问题以及答案失败
7.	CHECK_MESSAGE	检查有无给自己的信息
8.	GET_ALL_FRIENDS	获取所有好友信息
9.	GET_ALL_FLOCKS	获取所有群信息
10.	GET_FLOCK_MEMBERS	获取群成员
11.	TALK	发送对话
12.	ADD_FRIEND	添加好友
13.	ADD_FLOCK	用户加群

14.	FLOCK_ADD_MEMBER	群加成员
15.	GET_FRIEND_INFORMATION	获取好友信息
16.	DELETE_FRIEND	删除好友
17.	GET_MYSELF_INFORMATION	获取自己信息
18.	CHANGE_INFORMATION	CHANGE_INFORMATION
19.	CHANGE_REMARK	改变（好友）备注
20.	CHANGE_PASSWORD	修改密码
21.	CHANGE_STATUE	改变自身登录状态
22.	QUIT	退出
23.	LOGIN_SUCCESS	登录成功
24.	LOGIN_FAIL	登录失败
25.	HAVE_LOGINED	已经登录
26.	REGISTER_SUCCESS	注册成功
27.	REGISTER_FAIL	注册失败
28.	REMOVE_BOX	删除分组
29.	REMOVE_BOX_SUCCESS,	删除分组成功
30.	REMOVE_BOX_FAIL	删除分组失败
31.	RENAME_BOX	重命名分组
32.	RENAME_BOX_SUCCESS	重命名分组成功
33.	RENAME_BOX_FAIL	重命名分组失败
34.	CREATE_BOX	创建分组
35.	CREATE_BOX_SUCEESS	创建分组成功
36.	CREATE_BOX_FAIL	创建分组失败
37.	MOVE_FRIEND_BOX	移动好友至其他分组
38.	MOVE_FRIEND_BOX_SUCCESS	移动好友至其他分组成功
39.	MOVE_FRIEND_BOX_FAIL	移动好友至其他分组失败
40.	GET_ALL_FRIENDS_SUCCESS	获取所有好友信息成功
41.	GET_ALL_FRIENDS_FAIL	获取所有好友信息失败
42.	NO_FRIEND	没有好友
43.	GET_ALL_FLOCKS_SUCCESS	获取所有群信息成功
44.	GET_ALL_FLOCKS_FAIL	获取所有群信息失败
45.	NO_FLOCK	没有群
46.	GET_FLOCK_MEMBERS_FAIL	获取群成员失败
47.	GET_FLOCK_MEMBERS_SUCCESS	获取群成员成功
48.	NO_FLOCK_MEMBER	没有群成员
49.	NO_MESSAGE	没有信息
50.	HAVE_MESSAGE	有信息
51.	HAVE_TALK_MESSAGE	有聊天信息

52.	FRIEND_REQUESTED	好友请求
53.	FRIEDN_NO_ACCOUNT	没有这个用户名
54.	ALREAD_FRIENDS	回应好友请求
55.	GET_FRIEND_FAIL	获得好友资料失败
56.	GET_FRIEND_SUCCESS	获得好友资料成功
57.	DELETE_FRIEND_FAIL	删除好友失败
58.	DELETE_FRIEND_SUCCESS	删除好友成功
59.	GET_MYSELF_FAIL	获得自己详细资料失败
60.	GET_MYSELF_SUCCESS	获得自己详细资料成功
61.	CHANGE_INFORMATION_FAIL	改变自身资料失败
62.	CHANGE_INFORMATION_SUCCESS	改变自身资料成功
63.	CHANGE_REMARK_FAIL	修改备注失败
64.	CHANGE_REMARK_SUCCESS	修改备注成功
65.	OLD_PWD_IS_WRONG	旧密码是错的
66.	CHANGE_PWD_FAIL	修改密码失败
67.	CHANGE_PWD_SUCCESS	修改密码成功
68.	REQUEST_FRIEND	请求添加好友
69.	AGREE_FRIEND	同意好友请求
70.	REFUSE_FRIEND	不同意好友请求
71.	ADD_FRIEND_FAIL	添加好友失败
72.	ADD_FRIEND_SUCCESS	添加好友成功
73.	END_ADD_FRIEND	好友添加结束
74.	BE_DELETED	自己被好友删除了
75.	MESSAGE	普通信息
76.	TALK_MESSAGE	聊天信息
77.	REQUEST_FLOCK	用户请求加入群
78.	FLOCK_AGREE	群主同意用户加入
79.	FLOCK_REFUSE	群主拒绝用户加入
80.	ADD_FLOCK_FAIL	加入群失败
81.	ADD_FLOCK_SUCCESS	加入群成功
82.	NEW_FLOCK_MEMBER	新成员入群
83.	LEAVE_FLOCK_MEMBER	成员离群
84.	FLOCK_REQUEST	群主请求用户加入群
85.	AGREE_FLOCK	用户同意加入群
86.	REFUSE_FLOCK	用户拒绝加入群
87.	CREATE_FLOCK	创建群
88.	CREATE_FLOCK_FAIL	创建群失败
89.	CREATE_FLOCK_SUCCESS	创建群成功

90.	DROP_FLOCK	解散群
91.	DROP_FLOCK_FAIL	解散群失败
92.	DROP_FLOCK_SUCCESS	解散群成功
93.	LEAVE_FLOCK	退出群
94.	LEAVE_FLOCK_SUCCESS	退出群成功
95.	LEAVE_FLOCK_FAIL	退出群失败
96.	CHANGE_FLOCK	修改群信息
97.	CHANGE_FLOCK_SUCCESS	修改群信息成功
98.	CHANGE_FLOCK_FAIL	修改群信息失败
99.	CHANGE_FLOCK_REMARK	修改群成员备注
100.	CHANGE_FLOCK_REMARK_FAIL	修改群成员备注失败
101.	CHANGE_FLOCK_REMARK_SUCCESS	修改群成员备注成功
102.	TALK_FLOCK	群对话
103.	REQUEST_FILE	请求发送文件
104.	AGREE_FILE	同意接收
105.	REFUSE_FILE	拒绝接收
106.	END_FILE	接收完毕
107.	GET_HISTORY_MESSAGE	获取历史聊天消息（未使用）
108.	GET_HISTORY_MESSAGE_SUCCESS	获取历史聊天消息成功(未使用)
109.	GET_HISTORY_MESSAGE_FAIL	获取历史聊天消息失败(未使用)
110.	GET_OFFLINE_MESSAGE	获取离线聊天消息（未使用）
111.	GET_OFFLINE_MESSAGE_SUCCESS	获取离线聊天消息成功(未使用)
112.	GET_OFFLINE_MESSAGE_FAIL	获取离线聊天消息失败(未使用)
113.	NO_OFFLINE_MESSAGE	没有离线消息（未使用）

图表 3-4 消息类型表

3.3 服务端设计

3.3.1 代码结构

在即时通信系统服务端设计阶段，由于 Qt 除了可以创建漂亮的界面，还有很多其他的控制组件，所以对于设计服务端应用程序提供很大的便利。服务端系统代码采用 MVC 设计模型，采用模块化设计，并且各个模块根据“高内聚，低耦合”的设计思想，使得各个模块相互独立，减少模块间的数据耦合，便于系统后期的开发和维护。服务端系统的代码结构如图 3-12 所示。

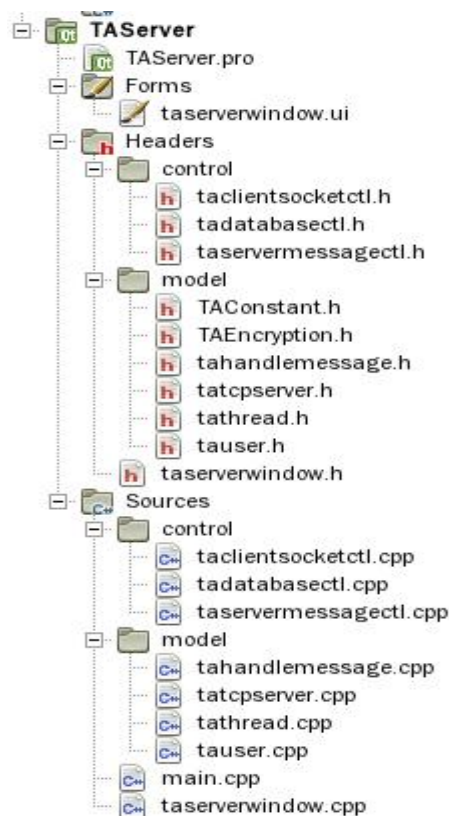


图 3-5 服务端代码结构

3.3.2 数据库设计

本即时通信系统的后台数据库表结构设计如下：

(1) 用户信息表：表名为 UserInformation，用户信息表用于记录注册用户信息，userID 字段值的产生来源于 DateCounter 时间计数表，通过当前日期和当天的注册数目来产生用户唯一的 ID 号码，表结构如表 3-6 所示。

表 3-6 表 UserInformation 的结构

Field	Type	Null	Key	comment
userID	varchar(15)	NO	PRI	用户 ID 号码
head	int(11)	NO		用户头像标识
nickname	varchar(20)	NO		用户昵称
password	varchar(10)	NO		用户密码
sex	varchar(2)	NO		用户性别
question	varchar(50)	NO		密保问题
answer	varchar(50)	NO		密保答案
phone	varchar(12)	YES		用户手机号
regDate	datetime	NO		注册时间

status	int(3)	NO		用户状态信息
--------	--------	----	--	--------

(2) 用户联系人表：表名 UserFriends，用户联系人表是用于描述用户和该用户好友的关系，插入数据表示用户添加好友，删除数据表示用户删除好友，groupName 字段用于用户好友分组管理，remarkName 字段用于用户标识好友，表结构如表 3-7 所示。

表 3-7 表 UserFriends 的结构

Field	Type	Null	Key	comment
friendNum	int(11)	NO	PRI	标识列
userID	varchar(15)	NO	MUL	用户 ID
friendID	varchar(15)	NO	MUL	好友 ID
groupName	varchar(50)	NO		好友所在组
remarkName	varchar(50)	YES		好友备注

(3) 群信息表：表名 FlockInformation，群信息表用户创建群所需收录信息，插入数据表示用户创建群，删除数据表示用户解散群，flockID 字段数据的产生同样来源于 DateCounter 时间计数表，通过当前日期和当天注册数目来产生，creatorID 表示群创建者，即就是群主 ID，createDate、flockName 和 theme 描述该群信息。表结构如表 3-8 所示。

表 3-8 表 FlockInformation 的结构

Field	Type	Null	Key	comment
flockID	varchar(15)	NO	PRI	群 ID
creatorID	varchar(15)	NO	MUL	群主 ID
createDate	date	NO		创建时间
flockName	varchar(50)	NO		群名
theme	varchar(100)	NO		群主题

(4) 群成员表：表名 UserFlocks，群成员表主要作用是为了表示用户和群的关系，remark、status 和 joinDate 描述群成员信息。表结构如表 3-9 所示。

表 3-9 表 UserFlocks 的结构

Field	Type	Null	Key	comment
userFlockNum	int(11)	NO	PRI	标识列
flockID	varchar(15)	NO	MUL	群 ID
userID	varchar(15)	NO	MUL	成员 ID
remark	varchar(20)	YES		成员备注

status	int(3)	NO		成员身份标识
joinDate	datetime	NO		加入时间

(5) 时间计数表：表名 DateCounter，时间计数表主要作用是为了产生唯一用户 ID 和群 ID，表结构如表 3-10 所示。

表 3-10 表 DateCounter 的结构

Field	Type	Null	Key	comment
date	date	NO		时间点
userTimes	int(11)	NO		用户 ID 时间序
flockTimes	int(11)	NO		群 ID 时间序

3.3.3 流程图

本服务端的流程图如图 3-11 所示。

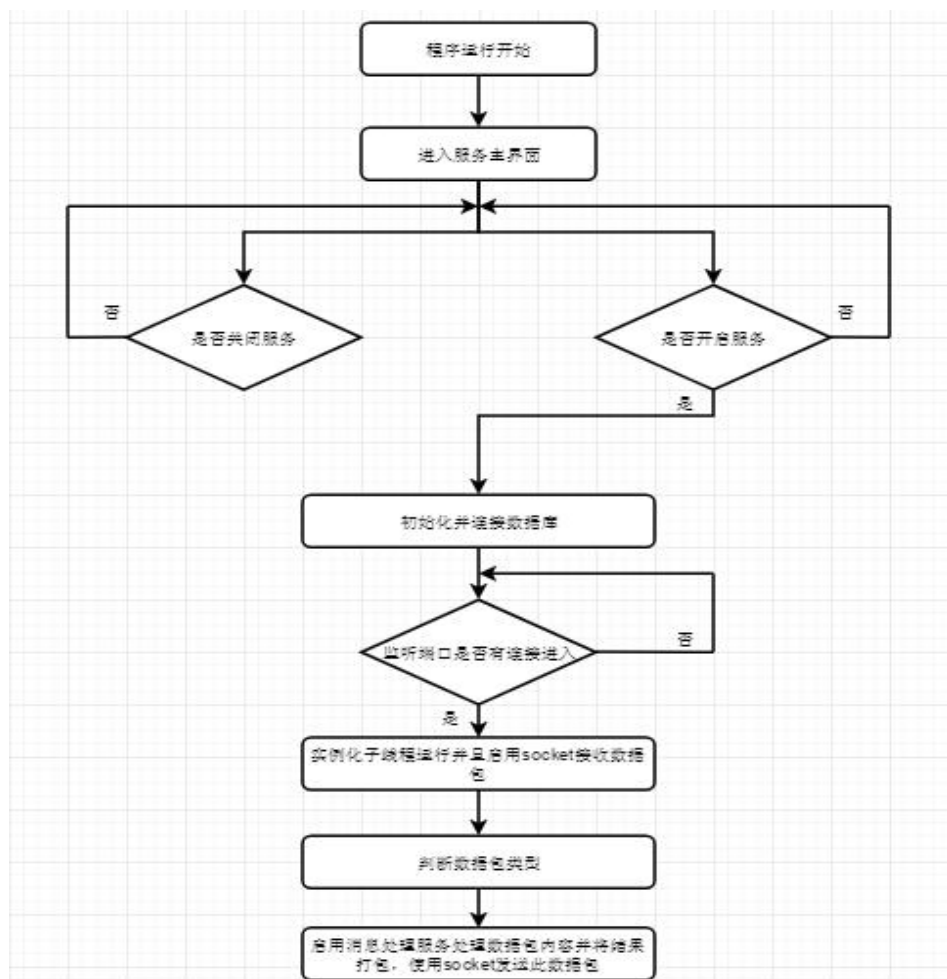


图 3-11 服务端运行流程图

3.4 客户端设计

3.4.1 代码结构

在即时通信系统设计阶段，对客户端系统的代码进行合理规划和设计，便于系统后期的开发和维护。本客户端系统代码采用 MVC 设计模型，系统整体设计采用模块化的结构，并且各个模块根据“高内聚，低耦合”的设计思想，使得各个模块相互独立，减少模块间的数据耦合，以提高系统的代码质量。客户端系统的代码结构如图 3-12 所示。

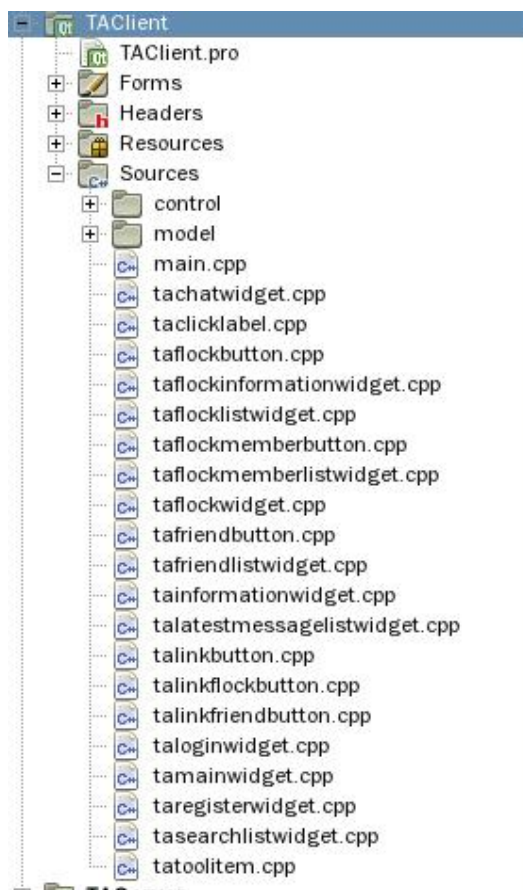


图 3-12 客户端系统的代码结构

3.4.2 流程图

程序运行流程图如图 3-13 所示。

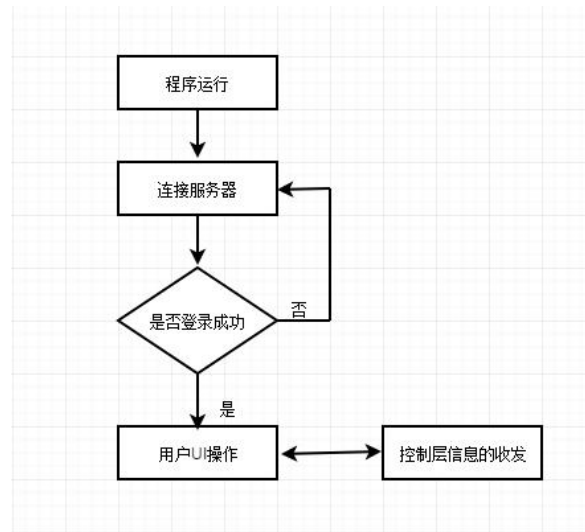


图 3-13 客户端程序运行流程图

4 详细设计与实现

4.1 登录与注册功能的设计与实现

4.1.1 登录模块

登录模块是进入即时通信系统的入口，通过对用户身份的验证，判断用户输入的用户名和密码是否与服务端的数据库中的信息相匹配，如果匹配，那么当前用户可以进入系统，如果不匹配，则无法进入系统，并显示相应的错误提示信息。

如果点击设置，则客户端会提示用户输入服务端的 IP 地址以及端口号。设置服务端地址如图 4-1 所示



图 4-1 设置服务端地址

如果点击记住密码，则客户端系统会把用户名、密码都保存在 auto_login.im 文件中，下次同用户登录系统就不需要再次输入 ID 和密码，直接点击登录即可进入通信系统。记住密码实例如图 4-2 所示。



图 4-2 记住密码实例

如果点击忘记密码，则客户端系统会提示用户输入 ID，同样服务端会在数据库中查询此用户的密保问题以及答案和密码，并将此查询结果发送到客户端，客户端系统提示用户输入此密保问题的答案，如果密保答案验证通过，直接将密码显示给用户。找回密码

码过程如图 4-3 所示。



图 4-3 找回密码过程

如果点击登录状态，用户可以选择在线的方式：在线、隐身、离开、忙碌、请勿打扰。登录状态更改如图 4-4。



图 4-4 登录状态更改

4.1.2 注册模块

注册模块是该通信系统收录用户信息并且分配唯一用户账号的入口，用户可以简单的输入信息，客户端系统首先进行简单的信息数据格式验证，如果验证通过，则使用异或运算将密码加密，则将注册信息打包，发送到服务端进行数据表信息的录入，并且把此注册信息重新打包发送到客户端，客户端对注册成功的用户信息进行消息提示。注册过程如图 4-5 所示。



图 4-5 注册过程

4.2 个人管理的设计与实现

4.2.1 个人信息查看和更改

在用户主界面上，一旦用户点击个人头像，则从服务端获取个人信息，并且通过一个 widget 进行信息展示，当然，用户可以点击编辑按钮进行信息更改，如昵称和手机号和性别，一旦更改完成并且点击保存，更改后的信息重新发送到服务端，进行数据库信息更改，同时获取该用户在线好友的 socket, 向每个 socket 发送该用户的新信息，进行好友列表的更新。个人信息查看和更改如图 4-6 所示。



图 4-6 个人信息查看和更改

4.2.2 密码更改

密码更改可以提高个人信息的安全性。在用户在界面上，用户一旦左键点击系统管理，创建出来的菜单选项中便有密码更改项目，用户点击此项，客户端会提示用户输入原密码，新密码。然后客户端将此信息发送到服务端，服务端将原密码验证，如果通过则立即更改用户信息表的密码字段。如果未验证通过，将失败结果发送到客户端，对用户进行友好提示。修改密码过程如图 4-7 所示。



图 4-7 修改密码过程

4.2.3 登录状态更改

用户之间的通信可以根据在线状态来人性化的通信,比如用户 A 看到联系人 B 的登录状态是忙碌,用户 A 就可以更根据消息内容的重要性,来向 B 发起通信。登录状态分为在线、隐身、忙碌、离开、请勿打扰,用户可以点击主界面上的 comBox 来选择登录状态。客户端软件将用户更改后的状态,发送到服务端,进行数据库信息更改,同时获取该用户在线好友的 socket,向每个 socket 发送该用户更改后的状态信息,进行好友列表更新。

假如刘 xin 用户选择请勿打扰状态,而他的联系人风园迪的联系人列表所显示刘 xin 用户状态为请勿打扰。登录状态更改如图 4-8 所示。



图 4-8 登录状态更改

4.2.4 新消息管理

新消息管理对于用户及时通信有着极其重要的作用,用户一旦接收到联系人或者群信息,就在用户主界面上出现新消息提示,并且新消息列表出现新消息提示来源。

用户一旦点击新消息提示按钮,便弹出新消息列表,用户可以对新消息选择查看或者忽略。新消息管理如图 4-9 所示。



图 4-9 新消息管理

4.3 联系人管理的设计与实现

4.3.1 添加/删除联系人

添加/删除联系人模拟现实生活中的人际交往过程。

在用户主界面上,用户可以点击添加好友工具按钮或者在联系人空白地方鼠标右击在创建的菜单选项中选择点击添加好友选项,当点击添加好友后,客户端系统弹出提示框,提示用户输入好友 ID,验证信息内容,客户端系统将加好友信息发送到服务端,服务端在数据库中查找是否有该用户的注册信息,如果有,则在服务端查找是否该用户在线,若在线发送加好友信息,好友收到此信息可以判断是否同意,同意则双方互为好友,若不同意则对双方进行消息提示。添加联系人操作过程如图 4-10 所示。

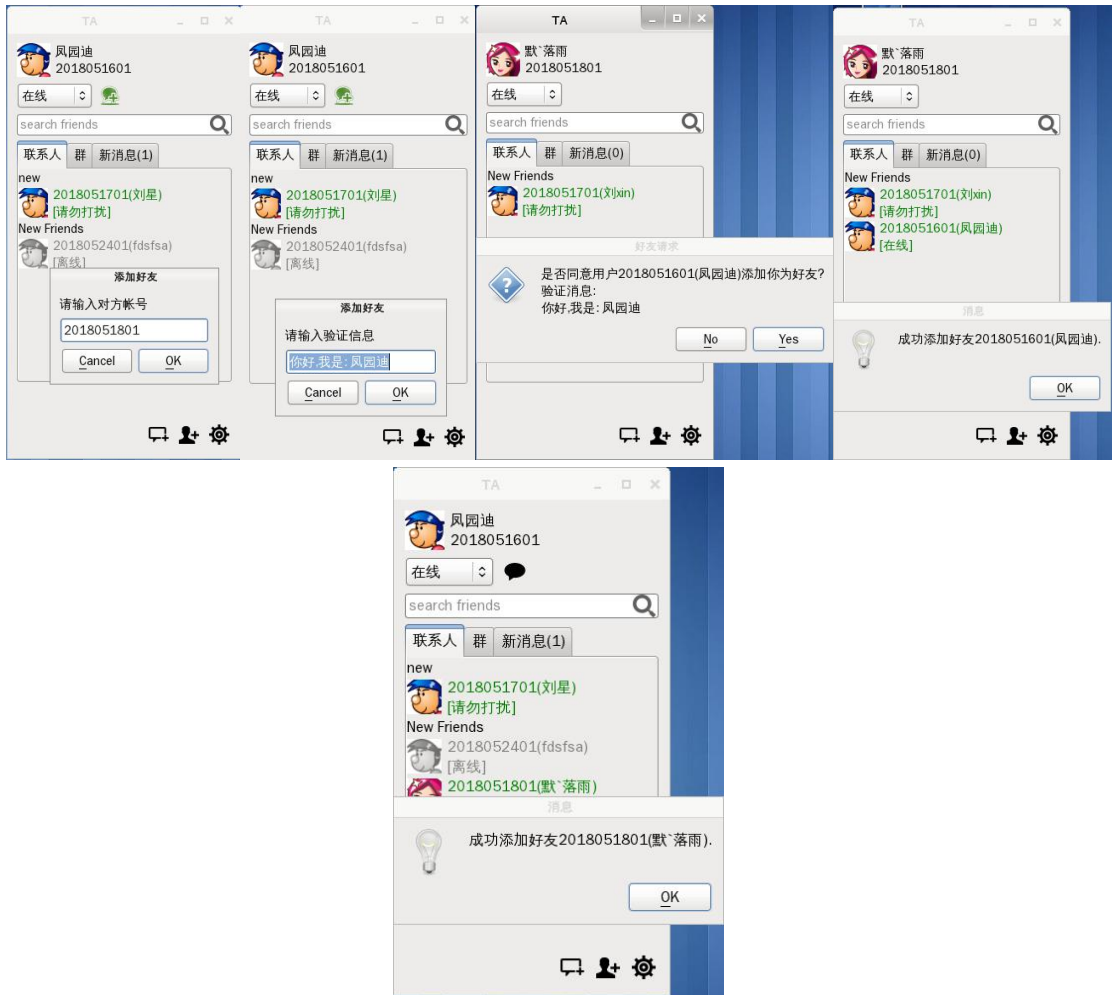


图 4-10 添加联系人操作过程

如果用户需要删除联系人,在联系人列表找到相应的联系人名片,鼠标右键选择删除好友即可,则双方都删除里联系人名片。删除联系人的操作过程如图 4-11 所示。



图 4-11 删除联系人的操作过程

4.3.2 好友分组管理

好友分组管理模拟现实生活中的人际交往圈,比如高中同学是一部分好友和同事是另一部分好友。用户可以创建分组来区分社交圈。

用户可以在联系人列表空白处鼠标右键点击选择创建分组,用户根据系统提示输入分组名称即可,创建好的分组,用户可以选择在该分组中添加好友或者从其他分组中移动好友到该分组。联系人分组管理过程如图 4-12 所示。



图 4-12 联系人分组管理过程

4.3.3 联系人备注更改

联系人备注更改模拟现实中人与人之间一种亲密度的表现,比如如用户默·落雨把联系人刘 xin 取备注为刘星,以方便该用户的识别。具体操作如图

用户在联系人列表中选中联系人并鼠标右键,选择点击修改备注名称,用户根据系统

弹出提示框输入备注名称即可。联系人备注更改过程如图 4-13 所示。



图 4-13 联系人备注更改过程

4.3.4 搜索好友

搜索好友可以帮助用户快速找到联系人，在用户界面上，用户可以点击搜索条框输入搜索关键字，便可以快速搜索出与之相关关键词的好友列表，关键字可以输入用户 ID、昵称、备注名。该功能的实现依靠 MAP 键值队数据结构，过程为：用户登录系统后，从服务端获取好友列表信息，在客户端系统创建好友 MAP<id, FriendButton>键值对，将每个好友插入到此 map 中，此后的搜索功能依靠此 map 成员变量。通过相应的迭代器，遍历此 map，一旦发现账号、昵称、备注信息匹配的，将此 FriendButton 加入到 Searchlist 的布局中，进行显示。搜索好友过程如图 4-14 所示。



图 4-14 搜索好友过程

4.3.4 即时通信

(1) 文字传输：

文字传输是该系统的最基础的功能，客户端系统从信息输入框中读取聊天内容，然后将发送端的账号和 IP、接收端账号和 ip 和消息类型打包发送到服务端，服务端接收到此数据包读取接收端账号，服务端检测接收端用户账号是否在线，如果在线将此包重新发送到接收端。接收端将此包解析进行消息提示或者信息显示。从而达到即时通信的目的。

用户通过鼠标双击联系人名片，进入聊天室，在输入信息框中输入聊天信息然后点击发送即可。文字传输的过程如图 4-15 所示。

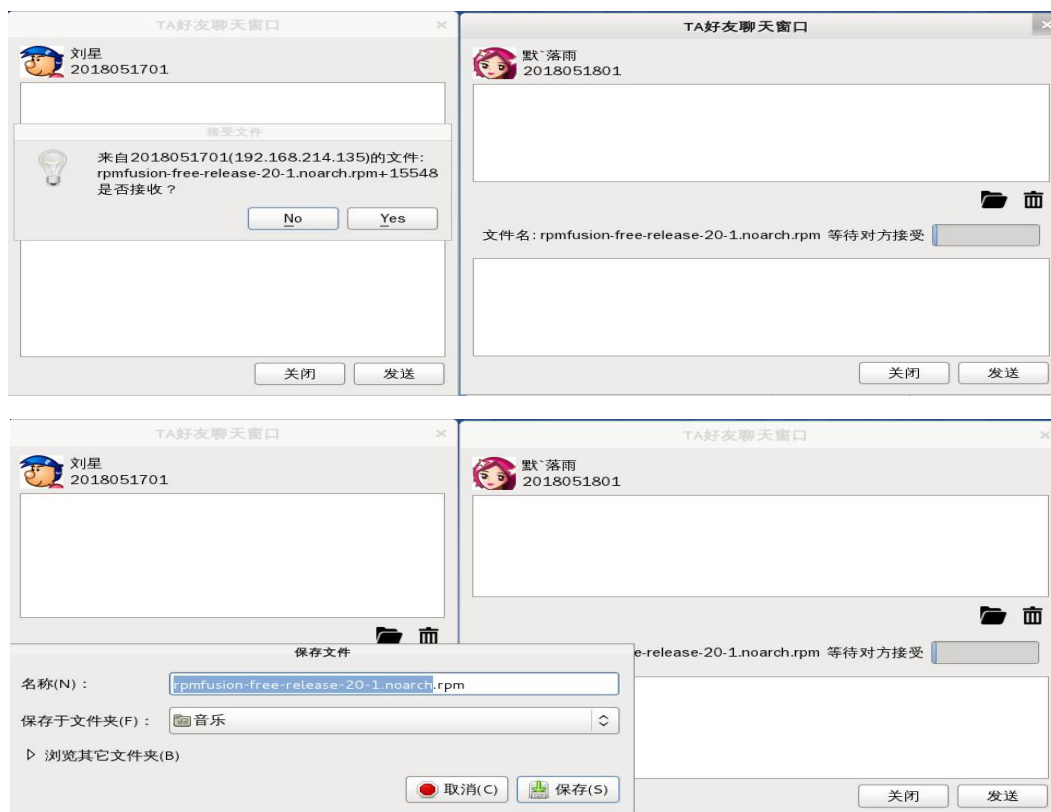


图 4-15 文字传输的过程

(2) 文件收发:

该功能是系统拓展的功能，用户之间可以通过此功能进行文件信息交流，用户 A 通过点击文档按钮选择文件，用户 B 可以根据提示框内容选择是否接收。如果选择接收，用户 B 接着选择文件的保存位置，最后等待用户 A 发送文件数据。

文件的收发功能使用 UDP 协议，而且不经过服务器的转发，一方用户作为服务端，一方作为客户端，进行文件的收发。文件收发的过程如图 4-16 所示。



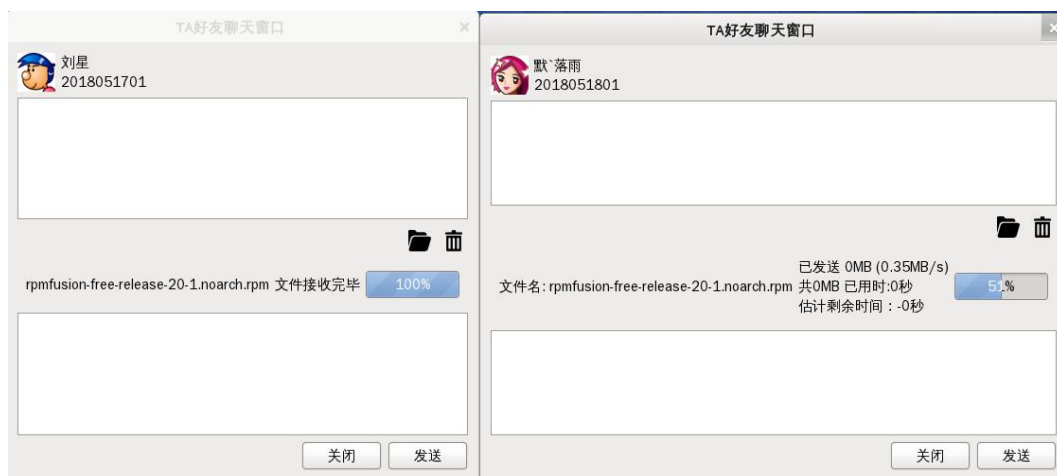


图 4-16 文件收发过程

4.4 群管理的设计与实现

4.4.1 创建/解散群

群类似于现实生活中的一个学校的班级，而班级里的同学可以发言也可以和班级里的同学说悄悄话，因此群对于即时通信系统也是最为重要的功能。

用户可以在群列表空白处鼠标右击，选择点击创建群，然后根据消息提示输入群名和群主题，此用户就拥有该群，成为群主。创建群的过程如图 4-17 所示。



图 4-17 创建群的过程

只有群的创建者（群主）才可以拥有解散群的权限，用户可以在群列表中选择要解散的群，然后鼠标右击，选择点击解散群，然后根据消息提示选择确定解散，此用户就成功解散该群。解散群的过程如图 4-18 所示。



图 4-18 解散群的过程

4.4.2 添加/删除群成员

只有群的创建者（群主）才可以拥有添加或者删除群成员的权限的权限。

用户可以在群列表中选择要添加成员的群，然后鼠标右击，选择点击添加群成员，然后根据消息提示输入该用户账号和验证信息，如果该用户同意入群，即成为群成员。添加群成员的过程如图 4-19 所示。

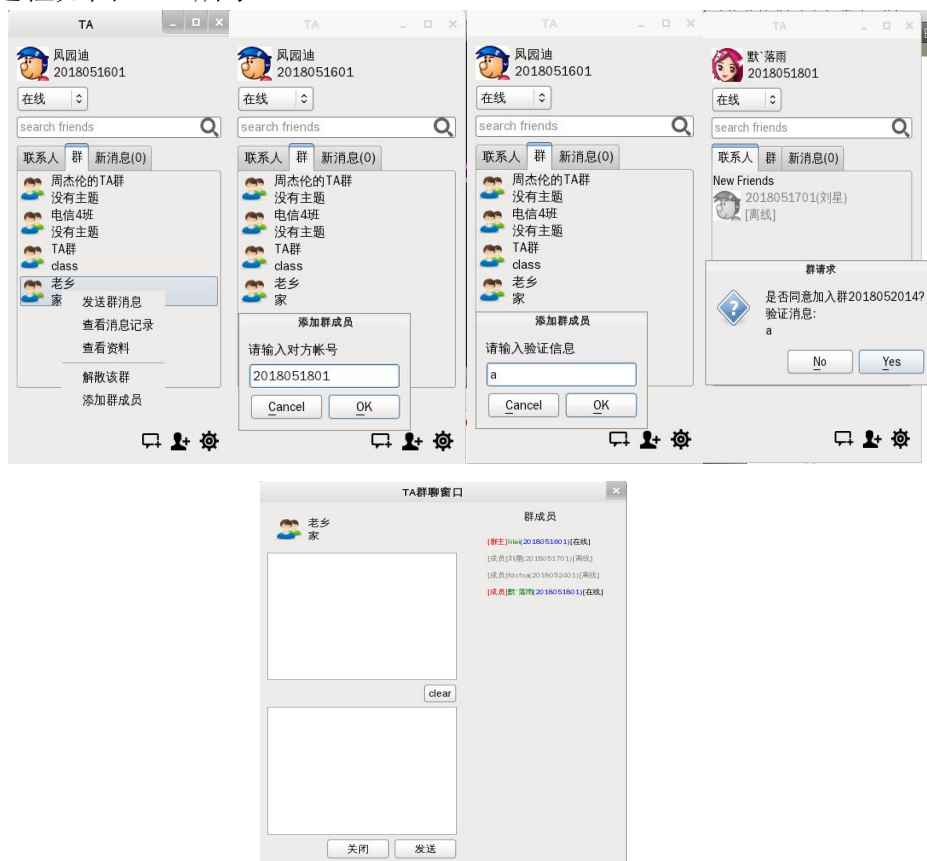


图 4-19 添加群成员的过程

只有群身份为群主的用户才拥有移除成员的权限，用户可以在群列表中选择群，然后双击进入聊天室，在群成员列表中选择要删除的群成员，右键点击选择移除成员选项即可，删除群成员的过程如图 4-20 所示。



图 4-20 所示删除群成员的过程

4.4.2 添加/删除群

用户可以在主界面下方点击添加群按钮，然后根据系统提示框输入群账号和验证消息，群主通过验证后，该用户成为群成员。添加群的过程如图 4-21 所示。

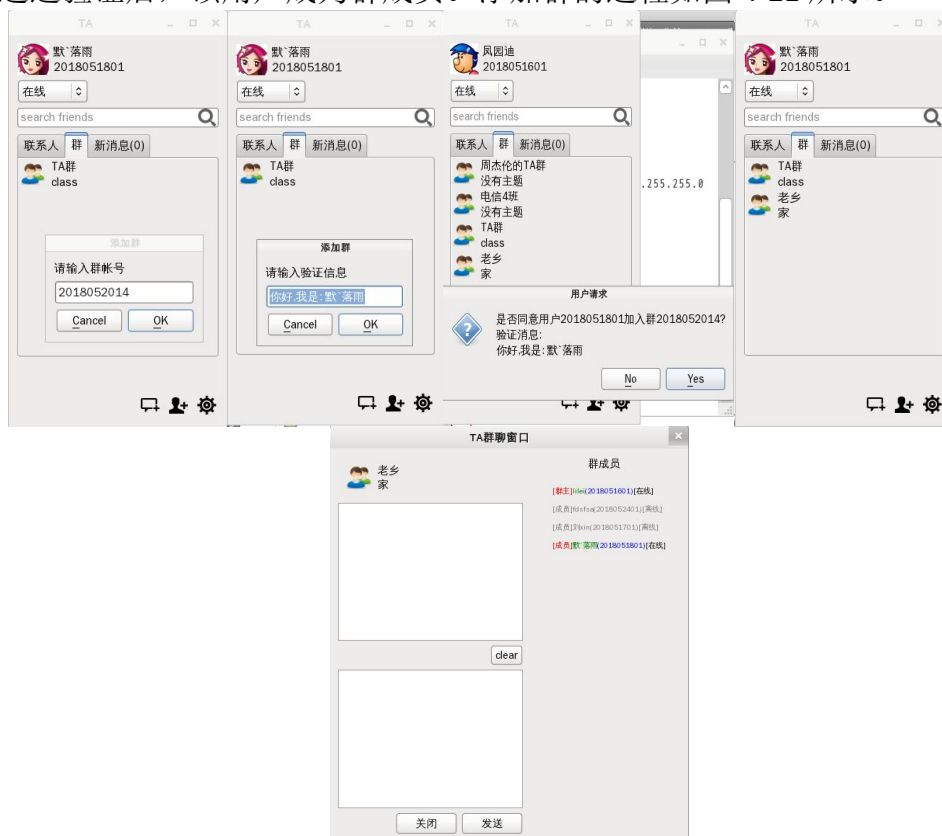


图 4-21 添加群的过程

用户可以在群列表中选择要退的群，然后鼠标右击选择退出该群，用户根据系统提示选择是否退群即可。退群的过程如图 4-22 所示。

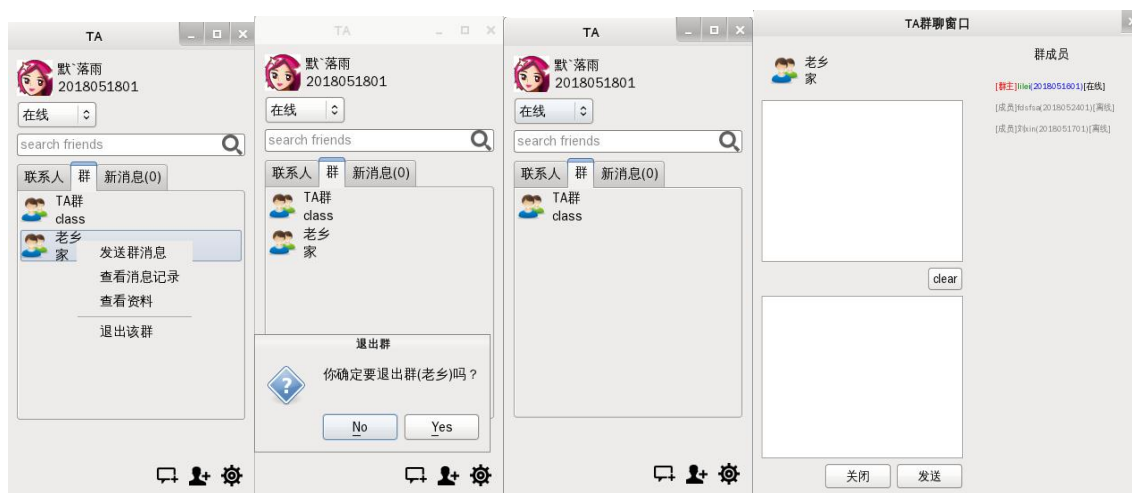


图 4-22 退群的过程

4.4.3 查看群信息

用户在群列表中选择要查看的群名片，然后鼠标右击选择查看群信息。用户即可查看群账号，名称，主题，群主名称，创建时间信息。群成员查看群信息如图 4-23 所示。

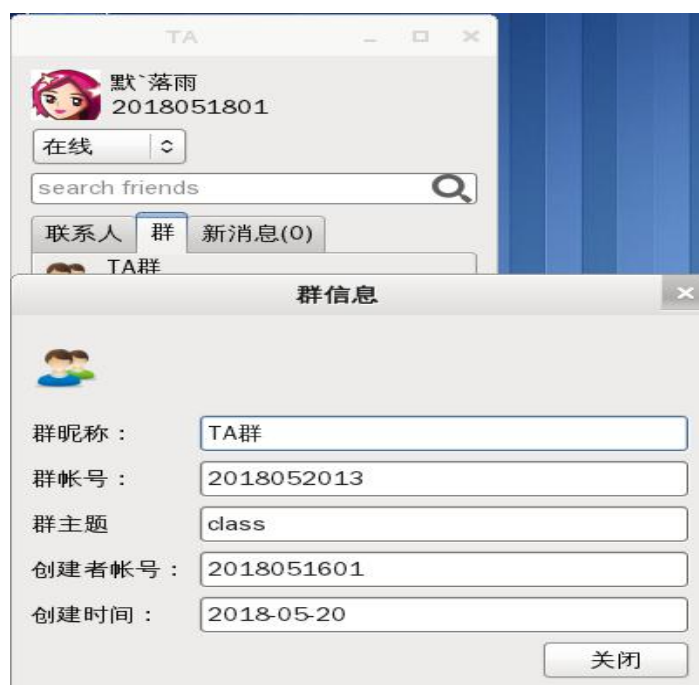


图 4-23 群成员查看群信息

而群主查看信息具有编辑权限，群主可以编辑更改群名称和群主题，然后点击保存按钮即可更改群信息。群主查看及更改群信息过程如图 4-24 所示。



图 4-24 群主查看及更改群信息过程

4.4.4 群成员备注更改

用户点击群名片进入聊天室，然后在群成员列表中找到自己的名片鼠标右击选择点击更改群名片，用户根据系统提示输入备注名称。群成员备注更改过程如图 4-25 所示。

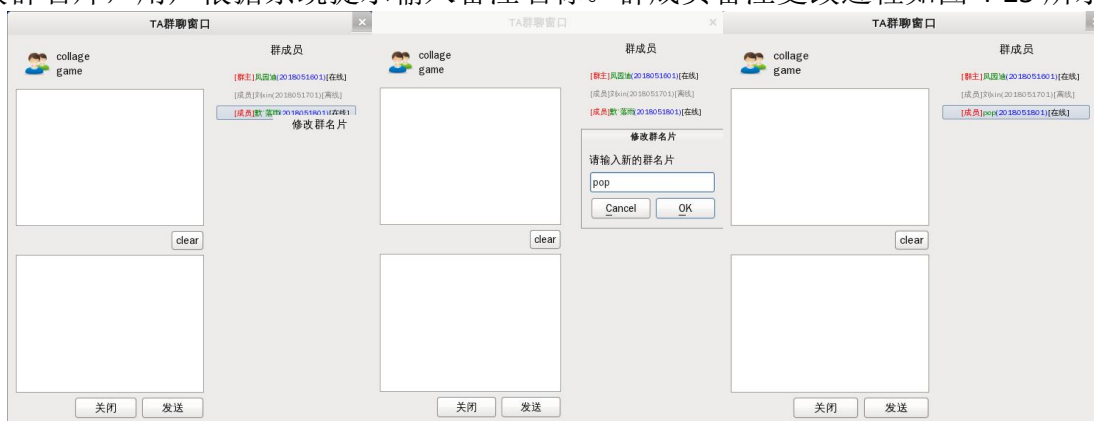


图 4-25 群成员备注更改过程

同样群主也有更改备注的权限，群主点击群名片进入聊天室，然后在群成员列表中选择任意群成员，然后鼠标右击选择修改群名片，群主根据提示输入群名片即可。群主更改备注过程如图 4-26 所示。



图 4-26 群主更改备注过程

4.4.5 群成员信息查看

用户点击群名片进入聊天室，在群成员列表中找到需要查看的用户名片，然后鼠标右击选择点击查看资料即可。群成员信息查看如图 4-27 所示。



图 4-27 群成员信息查看

4.4.5 群成员添加为联系人

用户点击群名片进入聊天室，在群成员列表中找到需要添加为联系人的用户名片，然后鼠标右击选择点击添加好友即可。群成员添加为联系人过程如图 4-28 所示。





图 4-28 群成员添加为联系人过程

4.4.6 即时通信

用户在群列表中选择群名片点击进入聊天室。用户在输入栏中输入聊天信息，然后发送即可。群聊的操作如图 4-29 所示。



图 4-29 群聊的操作

5 系统测试

5.1 系统测试分析

一般情况下，软件测试包含白盒测试和黑盒测试两种测试方式：

(1) 白盒测试

测试人员通过对程序员所编代码的访问，利用检查代码来达到测试目的。白盒测试多在系统单元测试中使用，用于测试系统功能模块中所有可能走的途径，然后执行所有的循环路径并测试其中所有的逻辑表达式。

(2) 黑盒测试

侧重于系统软件的整体功能，基于系统功能，而不是程序的内部结构，它是从用户的角度出发的。

5.2 功能测试

在本次功能测试中重点测试内容为网络数据包的丢包问题和服务端的并发连接问题。

5.2.1 UDP 丢包问题

(1) 本系统中使用两种传输协议 TCP 和 UDP，TCP 用于消息传输，UDP 用于文件的收发。由于 TCP 的是一种可靠的连接方式，所以丢包问题在本次系统测试没有出现。但是 UDP 是一种无连接不可靠的传输协议，所以丢包问题会出现。如图 5-1 是文件收发的功能测试结果。

图表 5-1 文件收发丢包测试结果表

原文件大小	发送端发送数据大小	接收端接收数据大小	丢包率
33227bytes	33227bytes	33227bytes	0%
60866bytes	60866bytes	24000bytes	60%
15548bytes	15548bytes	15548bytes	0%
4698112bytes	4698112bytes	168000bytes	96%

(2) 根据丢包测试结果分析，一旦要发送端要发送的原文件越大，丢包率越大。当然除此之外，也有以下几种可能导致丢包。

① 接收端处理时间过长导致丢包；

② 发送的包较大，超过接受者缓存导致丢包；

③ 发送的包频率太快导致丢包，UDP 的发送没有像 TCP 那样阻塞问题，只要知道 IP 地址和端口号，就直接将数据发送，并不关心数据是否到达对端。

(3) 通过丢包的原因分析，文件传输丢包问题是无法避免的，但是可以通过以下几点来降低丢包率。

① 降低发送包的频率，发送过快设计 sleep。

② 设置 socket 接收缓冲区增大。

③ 至于报文过大的问题，可以通过控制报文大小来解决，使得每个报文的长度小于 MTU。以太网的 MTU 通常是 1500 bytes，其他一些诸如拨号连接的网络 MTU 值为 1280 bytes，如果使用 speaking 这样很难得到 MTU 的网络，那么最好将报文长度控制在 1280 bytes 以下。

5.2.2 服务端并发连接问题

(1) 服务端为保持其服务性能，能够服务更多的客户端连接，使用 **epoll** 来做事件触发，具有以下优点：

① 支持一个进程打开大数目的 **socket** 描述符。

② **IO** 效率不随 **FD** 数目增加而线性下降。

③ 使用 **mmap** 加速内核与用户空间的消息传递。

(2) 本系统服务端由于使用 Qt 平台开发，所以并没有使用 linux 系统自带的 **epoll** 库。而是使用 GitHub 开源项目 **qt_eventdispatcher_epoll**，该开源项目具有以下优点：

① 非常快速。

② 支持 **Qt4** 和 **Qt5**。

③ 不需要 **Qt** 的私有文件头。

④ 通过了 **Qt4** 和 **Qt5** 的事件调度，事件循环，定时器和 **socket** 通知测试。

(3) 使用方法：**Qt** 默认的事件转发方式为 **select**，所以直接设置 **Qt** 的事件转发方式即可。

(4) 使用 **ab** (apache bench) 压力测试工具对服务端性能进行测试，通过次测试明显提高服务端的并发连接数。

5.3 测试总结

在对本系统代码进行系统测试之后，系统的总体功能模块实现还是比较顺利，基本上已经达到了系统设计初期的功能需求。从用户登录注册，到个人管理功能实现，再到联系人管理和群管理的各个功能模块的操作，最后退出系统，整个流程可以正常运转。但是鉴于这只是本人自己进行的系统测试，很多地方的 bug 可能会被自己开发时的视角所掩盖。

结论

本次毕业设计到这里也就结束了。在自己的刻苦努力和指导老师的帮助下，终于达到了设计初期的设计目标，实现了诸多功能模块。在整个设计过程中，每一个设计阶段都是去认真思考、完善，每一个模块都是去仔细检查、修正。完成所有的设计功能后，进行软件功能的测试阶段出现了许多的问题与阻碍，但是，经过自己请教老师、查找资料，或者与熟知的同学进行讨论，最终问题都得到了解决，修复了代码中的 bug，使系统健壮性、实用性增强。以下是本系统所实现的主要功能以及一些不足之处。

- 1) 支持用户注册、登录验证功能。
- 2) 支持客户端之间的消息传输和文件传输。
- 3) 支持多用户之间的群聊。
- 4) 支持用户添加好友、查看好友信息、好友分组管理、备注更改。
- 5) 支持用户创建群、解散群、添加群和退出群，同样也支持添加群成员，备注更改等功能。
- 6) 支持用户查看和更改自己信息，更改密码，密保答案找回密码，搜索好友等功能。
- 7) 支持用户密码加密存储和登录时的记住密码功能。
- 8) 支持新消息接收提示功能。

虽然本次设计的即时通信系统软件基本上实现了设计初期应有的功能，但是由于自己动手实践能力的薄弱以及设计时间的局限性，该系统还是存在着诸多不足之处。下面是一些完善的改进方案。

(1) 对用户聊天方式的丰富性应该改善，比如聊天字体的样式、表情图片传输、语音聊天、视频聊天等，只有丰富聊天方式，才能使用户得到更好的体验，满足用户需求。

(2) 对用户离线消息的管理应该改善，一旦用户离线，服务器应该将该用户接收到的新消息存储到服务端数据库中，一旦用户上线便接收服务端发来的离线消息，服务端也因此将该用户的离线消息从数据库中删除。

(3) 对注册用户信息的内容应该丰富，比如增加签名、生日日期、地址信息等。

致谢

在西安科技大学度过了四年的学习和生活，系统的学习了电子信息方面的专业知识，在此也深深的感谢各位专业老师的培养，使我不仅仅学到了专业知识，而且也教会我为人处世的道理。本次即时通信系统的设计与实现是在黄健老师的细心指导下圆满完成的。自己之前也开发过与该设计题目相关的项目，因此也有些这方面的经验，自己在拿到设计题目后开始寻找以前做过的项目，搜集文献资料，分析功能需求以及运用的技术手段，设计系统方案，由于自己对即时通信系统的具体功能需求不是很了解，再加上对 Qt 开发平台的操作不够熟练，遇到了各种各样的问题，所以在设计具体方案以及功能测试方面，自己遇到不懂的地方，就会及时请教老师，老师也能够悉心分析讲解，提供解决方案及思路，并且给出了一些相关的资料供我去参考、学习。

首先，我要感谢我的毕业设计导师黄健老师，此次系统设计和论文的顺利完成，凝结着老师的智慧与心血，在这期间，每次遇到自己解决不了的问题，老师都能够细心指导，总是不要其烦的为我讲解所有的难题。老师渊博的专业知识、丰富的指导经验、对待工作的认真态度以及对科学严谨的作风都是我受益匪浅。如果没有老师的帮助与鼓励，我想我是不会这么顺利完成设计与论文，同时，我也感谢大学四年里所有给我带过课的专业老师，是他们让我拥有丰富的专业知识与广阔的视野，是他们的谆谆教导，才使我有着更大的进步。

其次，我要感谢大学四年中一直陪伴着我的所有同学，不论是在学习还是日常生活中，都有着他们的鼓励与帮助，让我认识到同学与朋友的珍贵，也明白团队与集体的力量是多么的强大。

最后，我要感谢母校，能够给我提供良好的学习与成长环境，使我自身更加的进步与提升。

参考文献

- [1] 张红艳. Qt5开发实战[M]. 北京: 人民邮电出版社, 2015年9月
- [2] 陆文周. Qt5开发及实例[M]. 北京: 电子工业出版社
- [3] 侯捷. More Effective C++中文版[M]. 北京: 电子工业出版社, 2011年1月
- [4] 戚正伟 张亚英 尤晋元译, UNIX 环境高级编程[M]. 北京: 人民邮电出版社, 2014年6月
- [5] 侯捷. STL 源码剖析[M]. 武汉: 华中科技大学出版社, 2002年6月
- [6] 孟庆昌 牛欣源. Linux 教程[M]. 北京: 电子工业出版社, 2007年3月
- [7] (美) 史蒂文斯 (美) 芬纳 (美) 鲁道夫. UNIX 网络编程[M]. 北京: 人民邮电出版社, 2015年8月
- [8] 程杰. 大话设计模式[M]. 北京: 清华大学出版社, 2007年12月
- [9] 吴英 张玉 许昱伟. TCP/IP 详解[M]. 北京: 机械工业出版社, 2016年6月
- [10] 张海龙 袁国忠. C++ Primer Plus 中文版[M]. 北京: 人民邮电出版社, 2012年7月
- [11] 黄晓春. C++沉思录[M]. 北京: 人民邮电出版社
- [12] 徐波. C 和指针[M]. 北京: 人民邮电出版社, 2008 年 4 月
- [13] 宁海元 周振兴 彭立勋 瞿伟祥. 高性能 MYSQL[M]. 北京: 电子工业出版社
- [14] 刘增杰 .mysql 从入门到精通[M]. 北京: 清华大学出版社, 2016 年 9 月