

**课程设计**

**课程名称： 数据库课程设计**

**学 院：计算机科学与技术专 业： 信息安全**

**姓 名： 王成昱 学 号： 1500170108**

**年 级： 2015 任课教师： 武彤**

**2018年 8 月 1日**

**贵州大学本科课程设计任务书**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学生信息 | 姓名 | | | 李振18656886561  董正禹  马利超  赵光森  王成昱 | 学院 | 计算机科学与技术 | | | | 班级 | |
| 专业 | 信息安全 | | | | 152 | |
| 教师信息 | 姓名 | | | 武彤 | 职称 | 教授 | | 学历 | | 博士 | |
| 任务书 发出时间 | | 2018.7.9 | | | 论文（设计）题目 | | **即时通信系统** | | | | |
| 论文（设计）起止时间 | | | 2018.7.9-2018.8.2 | | | | | | 共需周数 | | 4 |
| **设计内容：**设计一个类似qq的即时通信系统的服务端与客户端的数据库，分工完成需求分析，概要设计、逻辑结构设计、数据模型优化、物理结构设计、数据库实施，最后完成整个系统的编写与调试改进。 | | | | | | | | | | | |
| **主要要求:**数据库符合系统需求，并做相应的规范化。服务器采用sqlserver 2016作为数据库管理系统，客户端采用文件作为本地数据库，密码采用库外md5加密，系统能正常提供通信服务。  小组成员主要分工如下：  马利超：需求分析，概要设计  李振：逻辑结构设计、数据模型优化  赵光森：数据库实施，物理结构设计  董正禹、王成昱：程序编写，调试。  程序编写完成后调试阶段对数据库的调整和改动需全员参与讨论，设 计的各个阶段均可合作完成任务。系统具有安全性（数据库安全、密码、权限、防SQL注入攻击等等）。 | | | | | | | | | | | |
| **预期目标：**程序实现与qq的聊天类似的功能，能够实现客户与管理员的注册与注册信息的管理，实现客户间一对一，多对多的通信，服务器可以保存离线消息，相应客户上线时发送消息并发出提醒，客户之间可以添加与删除好友。服务器保存客户相应的的信息，客户端在本地也保存自身一定的信息。两方数据库均做一定程度的优化，避免使用时的各种异常。程序界面友好，易于操作。 | | | | | | | | | | | |
| **计划进程:**需求分析：7.11-7.13  概要设计：7.13-7.15  逻辑结构设计：7.15-7.17  数据模型优化：7.17-7.19  物理结构设计：7.19-7.21  数据库实施：7.21-7.23  程序编写：7.23-7.29  调试、改进与最终完成：7.29-8.02  答辩：8.03 | | | | | | | | | | | |
| 主要参考文献:  【1】 王珊，萨师煊. 数据库系统概论（第五版）. 北京：高等教育出版社,2014. | | | | | | | | | | | |
| 系审查意见：  系主任签字： 年 月 日 | | | | | | | | | | | |

目录

[第1章 概述 1](#_Toc520906647)

[第2章 需求分析 1](#_Toc520906648)

[2.1功能需求 1](#_Toc520906649)

[2.2性能要求 1](#_Toc520906650)

[2.3 DBMS管理员组成 1](#_Toc520906651)

[2.4 其他约束 1](#_Toc520906652)

[2.5数据流图 2](#_Toc520906653)

[2.6数据字典 3](#_Toc520906654)

[第3章 概要设计 5](#_Toc520906655)

[3.1设计简述 5](#_Toc520906656)

[3.2总体设计 5](#_Toc520906657)

[3.3模块说明 5](#_Toc520906658)

[3.4接口设计 7](#_Toc520906659)

[3.5数据结构 7](#_Toc520906660)

[第4章 数据库设计 9](#_Toc520906661)

[4.1 E-R图 9](#_Toc520906662)

[4.2数据库建表 9](#_Toc520906663)

[4.2.1 sqlserver库表 9](#_Toc520906664)

[4.2.2 sqlite库表 10](#_Toc520906665)

[4.3范式优化 10](#_Toc520906666)

[4.4物理设计 10](#_Toc520906667)

[4.5数据库安全 11](#_Toc520906668)

[4.6 数据库用户及权限 11](#_Toc520906669)

[4.6.1软件使用 11](#_Toc520906670)

[4.6.2数据库管理使用 11](#_Toc520906671)

[4.7 数据同步 11](#_Toc520906672)

[4.8 建表SQL语句 11](#_Toc520906673)

[5.1 程序模块流程图 12](#_Toc520906674)

[5.2主要数据结构 15](#_Toc520906675)

[5.3主要函数接口 16](#_Toc520906676)

[第6章 用户手册 17](#_Toc520906677)

[6.1如何注册 17](#_Toc520906678)

[6.2如何加好友 18](#_Toc520906679)

[6.3如何聊天 18](#_Toc520906680)

[第7章 总结 18](#_Toc520906681)

[参考文献: 19](#_Toc520906682)

即时通讯系统

摘要

随着计算机互联网技术的不断发展,聊天软件层出不群,即时通讯系统的应用极为普遍。本文实现一个运行在.net平台的即时通讯系统。该软件采用c/s模型，包括客户端和服务端两个模块，客户端向服务端发送请求，服务端向客户端提供服务。客户端主要实现注册，登录，搜索，加好友的功能，服务端主要实现消息的缓存，存储与转发。该软件采用tcp通信技术，实现了即时通讯的基本功能，但还有很多不足之处有待改进。

关键词：即时通讯，c/s，数据库，安全

Abstract

With the continuous development of computer Internet technology, chat software is out of the ordinary, and the application of instant messaging systems is extremely common. This article implements an instant messaging system running on the .net platform. The software adopts the c/s model, including two modules: client and server. The client sends a request to the server, and the server provides the service to the client. The client mainly implements the functions of registration, login, search, and adding friends. The server mainly implements message caching, storage and forwarding. The software uses tcp communication technology to realize the basic functions of instant messaging, but there are still many shortcomings to be improved.

Keywords: instant messaging, c/s, database, security

# 第1章 概述

互联网技术经过多年的发展，其应用已经渗透到了生活的方方面面，即时通讯就是其中之一，从qq的巨大规模就可以看出即时通讯系统对人们有多么重要。本项目为一个即时通讯的小系统，实现通讯需求的最基本功能，如注册，登录，加好友和聊天。本软件基于.net平台与sqlserver数据库，使用tcp实现网络通信。本文档是软件后续开发的指导。

# 第2章 需求分析

## 2.1功能需求

1.注册

2.登录

3.添加、删除好友

4.发送在线、离线消息

5.加入、退出群组

6.好友聊天

7.组内聊天

## 2.2性能要求

所有用户同时上线时保证不掉线，不丢失数据，网络环境良好的情况下，数据转发迅速准确，不漏传、错传或丢失数据。保证用户本地与服务器数据库数据一致，所有用户可同时通信，保证数据库长久的可用性，数据库要有一定的安全性。

## 2.3 DBMS管理员组成

1.数据库维护管理的管理员。

2.软件运行使用的管理员。

## 2.4 其他约束

（1）添加好友是双向的，要经过被添加者同意，删除好友不需要被删除者同意，可直接删除，删除好友也是双向的。

（2）账号由服务器分配，每次加一。

（3）本地不存储账号密码，数据库存储密码的md5。

（4）个性签名信息写入要先提交至服务器检查数据，以保证安全性。

（5）本系统为小型即时通讯系统，成员数量在10000以内，个人可创建群组数量在5个以内，好友数量在50以内，可加入群组数量30以内。

（6）群组可由任意成员创建，由创建者解散，群组成员上限为100人。

## 2.5数据流图

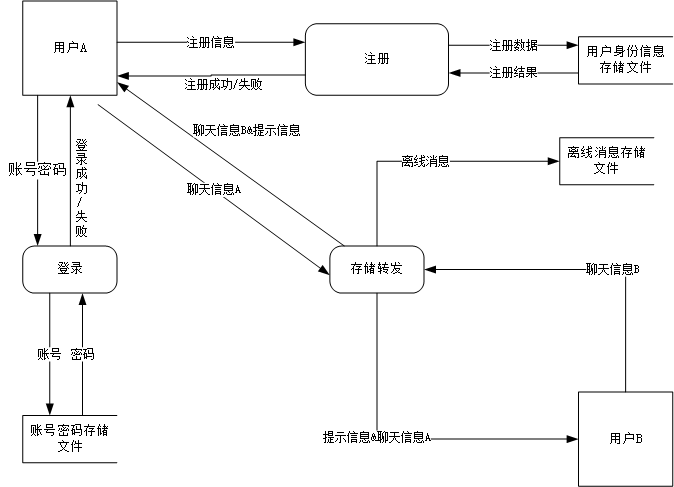


图2.1用户注册与聊天数据流图

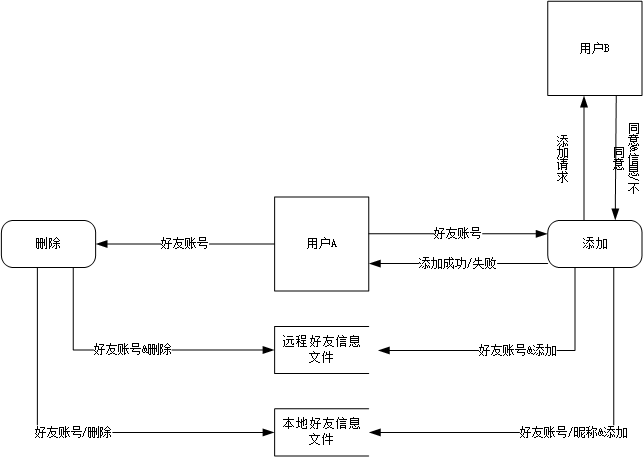


图2.2 增删好友数据流图

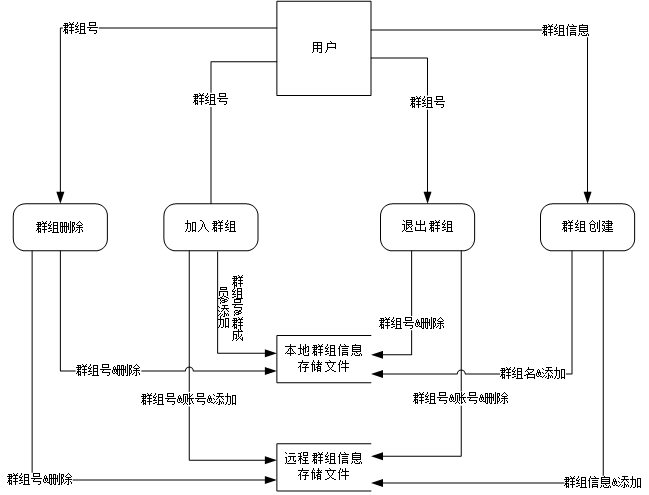


图2.3群组操作数据流图

## 2.6数据字典

(1)用户身份信息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 账号 | varchar | 用户凭证，用户申请账号时由系统分配 |
| 密码 | varchar | 用户凭证，存储前要进行加密处理 |
| 年龄 | Int | 用户年龄 |
| 性别 | Char | 用户性别 |
| 昵称 | Varchar | 用户自行设定的昵称 |
| 生日 | Datetime | 用户生日 |
| 个性签名 | Varchar | 一串文字，展现用户个性，用户可随意填写 |

（2）好友关系表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 主账号 | nchar | 用户本人账号 |
| 好友账号串 | Varchar | 好友账号拼成的字符串（使用&符号连接账号） |

（3）账号密码表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 账号 | Nchar | 用户凭证，用户申请账号时由系统分配 |
| 密码 | nchar | 用户凭证，存储前要进行加密处理 |
| 昵称 | nchar | 用户昵称 |

（4）群组表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 群组编号 | Int | 唯一编号 |
| 群组名 | Nchar |  |
| 群组成员 | nchar | 成员账号拼成的字符串 |
| 群组管理员账号 | nchar | 群组创建者，仅创建者可解散群组 |

（5）离线消息表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 源账号 | Nchar | 发送者账号 |
| 目的账号 | Nchar | 接受者账号 |
| 时间 | datetime | 发送者发出消息的时间 |
| 消息内容 | text | 消息的内容 |

（6）在线成员表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 账号 | nchar | 在线成员账号 |

(7)消息类型表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 消息类型 | Nchar | 消息的种类 |
| 类型编码 | nchar | 消息类型的编码 |

（1）本地好友表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 好友账号 | Nchar |  |
| 好友昵称 | nchar |  |

（2）本地群组表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 群组名 | Nchar | 加入的群组名称 |
| 群组编号 | nchar | 群组的本地编号 |

（3）本地群组细表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| 群组号 | Nchar | 群组本地编号 |
| 成员账号 | Nchar | 群组成员账号 |
| 成员昵称 | Nchar | 群组成员昵称 |

# 第3章 概要设计

## 3.1设计简述

需求概要

用户的使用方式要简单方便，软件界面大方得体，具有良好的视觉效果，基础功能完整，后期可对功能进一步完善。

限制描述

模块化设计，软件从逻辑上划分为多个部件，分别实现软件的各个功能与子功能。

运行环境

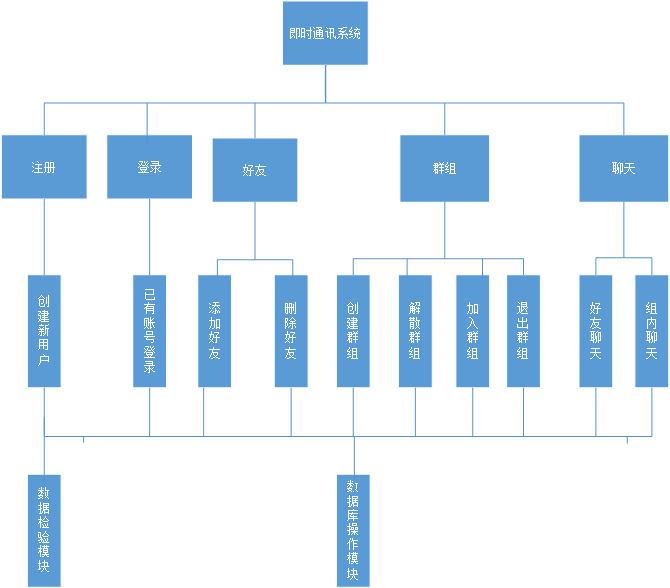
主机：pc兼容机

操作系统：Win 7/8/10

命名规则

函数名使用帕斯卡命名法。

## 3.2总体设计



## 3.3模块说明

（1）创建新用户

调用关系：被注册模块调用，调用数据库操作模块和数据校验模块。

逻辑功能：向服务器注册一个新用户。

输入：用户昵称，签名等信息

输出：注册成功&账号/失败。

（2）已有账号登录

调用关系：被登录模块调用，调用数据库操作模块。

逻辑功能：用户登录

输入：账号密码。

输出：登录成功/失败。

（3）添加好友

调用关系：被好友模块调用，调用数据库操作模块。

逻辑功能：添加好友。

输入：好友账号。

输出：添加成功/失败。

（4）删除好友

调用关系：被好友模块调用，调用数据库操作模块。

逻辑功能：删除好友。

输入：好友账号。

输出：删除成功/失败。

（5）创建群组

调用关系：被群组模块调用，调用数据库操作模块。

逻辑功能：创建群组。

输入：群组名等群组信息。

输出：创建成功/失败。

（6）解散群组

调用关系：被群组模块调用，调用数据库操作模块。

逻辑功能：解散群组。

输入：群组号。

输出：解散成功/失败。

（7）加入群组

调用关系：被群组模块调用，调用数据库操作模块。

逻辑功能：加入群组。

输入：群组号。

输出：加入成功/失败。

（8）退出群组

调用关系：被群组模块调用，调用数据库操作模块。

逻辑功能：退出群组。

输入：群组号。

输出：退出成功/失败。

（9）好友聊天

调用关系：被聊天模块调用，发送离线数据时调用数据库操作模块。

逻辑功能：好友聊天。

输入：好友账号和需要发送的消息。

输出：发送成功/失败。

（10）群组聊天

调用关系：被聊天模块调用，调用数据库操作模块。

逻辑功能：群组内聊天。

输入：群组号和要发送的消息。

输出：发送成功/失败。

（11）数据校验

调用关系：被用户创建模块调用。

逻辑功能：校验数据安全性。

输入：要校验的数据。

输出：通过/未通过。

（12）数据库操作

调用关系：被所有模块调用。

逻辑功能：实现所有模块的数据库操作。

输入：增删查改的信息。

输出：操作结果与操作状态。

## 3.4接口设计

用户接口

1.用户注册界面

2.用户登录界面

3.好友操作界面

4.群组操作界面

5.聊天界面

内部接口

各功能模块皆可调用数据库操作模块与远程数据库的接口。

## 3.5数据结构

(1)用户身份信息表user\_info

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 说明 |
| ID | Int | 否 | 用户凭证，系统分配，主键，外键 |
| PWD | Char（32） | 否 | 存储前要进行md5 |
| AGE | Int | 能 | 用户年龄 |
| SEX | int | 能 | 用户性别 |
| NICKNAME | Char(24) | 否 | 用户自行设定的昵称 |
| BIRTHDYA | Datetime | 能 | 用户生日 |
| SIGN | Varchar | 能 | 一串文字，展现用户个性，用户可随意填写 |
| G\_JOINED | varchar | 能 | 群组号串，用&连接，记录加入的群组 |

（2）好友关系表friend\_net

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| M\_ID | int | 否 | 用户本人账号，主键，外键 |
| F\_ID | varchar | 能 | 好友账号拼成的字符串（使用&符号连接账号） |

（3）账号密码表id\_pwd

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| ID | int | 否 | 主键 |
| PWD | Char(32) | 能 | 存储前md5 |
| NICKNAME | Char(24) | 否 | 用户昵称 |

（4）群组表group

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| G\_ID | Int | 否 | 唯一编号 |
| G\_NAME | Char(20) | 否 |  |
| G\_MEMBER | varchar | 否 | 成员账号拼成的字符串 |
| G\_ADMIN\_ID | int | 否 | 群组创建者，仅创建者可解散群组 |

（5）离线消息表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| SENDER\_ID | int | 否 | 发送者账号，主键，外键 |
| RECEIVEDER\_ID | int | 否 | 接受者账号，主键，外键 |
| TIME | datetime | 否 | 发送者发出消息的时间 |
| CONTENT | text | 否 | 消息的内容 |

（6）在线成员表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| ONLINE\_ID | int | 否 | 在线成员账号，主键，外键 |

(7)消息类型表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| TYPE | Nchar | 否 | 消息的种类，不能重复 |
| CODE | Char(3) | 否 | 消息类型的编码，如001，主键 |

（1）本地好友表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| F\_ID | int | 否 | 主键 |
| F\_NICKNAME | Char(24) | 否 |  |

（2）本地群组表

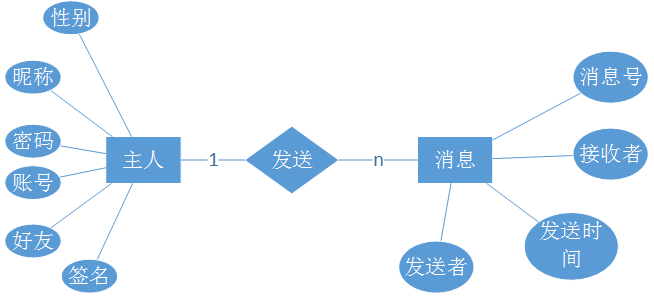
|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| G\_NAME | Char(20) | 否 | 加入的群组名称 |
| G\_ID | int | 否 | 群组编号，主键 |

（3）本地群组细表

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 能否为空 | 描述 |
| G\_ID | int | 否 | 群组本地编号，主键，外键 |
| G\_MEMBER\_ID | int | 否 | 群组成员账号 |
| G\_MEMBER\_NICKNAME | Char(24) | 否 | 群组成员昵称 |

# 第4章 数据库设计

## 4.1 E-R图



用户（账号，密码，好友，所属组）

消息（消息号，发送者，接收者，发送时间）

## 4.2数据库建表

### 4.2.1 sqlserver库表

Users(用户信息表)

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 说明 |
| U\_Account | Varchar（20） | 账号，系统分配，主键，设索引 |
| U\_Signature | Nvarchar（50） | 签名 |
| U\_NickName | Nvarchar（20） | 昵称 |
| U\_Gender | Nchar（1） | 性别 |
| U\_BirthDay | Varchar（20） | 生日 |
| U\_Status | Nvarchar（10） | 在线状态 |
| U\_Email | Varchar（20） | 邮箱 |
| U\_Pwd | Varchar（32） | 密码，md5值 |

Friends（好友关系表）

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 说明 |
| F\_SrcAccount | Varchar（20） | 主账号，主键 |
| F\_DestAccount | Varchar（20） | 好友账号，主键 |
| F\_Remark | Nvarchar（20） | 源对目的的备注 |

Msg（离线消息表）

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 说明 |
| M\_SrcAccount | Varchar（20） | 源账号 |
| M\_DestAccount | Varchar（20） | 目的账号 |
| M\_Time | Varchar（20） | 发出时间 |
| M\_ID | int | 消息号，主键 |
| M\_Msg | Nvarchar（max） | 消息内容 |

### 4.2.2 sqlite库表

FriendMsg（消息记录表）

|  |  |  |
| --- | --- | --- |
| 列名 | 数据类型 | 说明 |
| M\_SrcAccount | Varchar（20） | 源账号 |
| M\_DestAccount | Varchar（20） | 目的账号 |
| M\_Time | Varchar（20） | 发出时间 |
| M\_ID | int | 消息号，主键 |
| M\_Msg | Nvarchar（max） | 消息内容 |

## 4.3范式优化

第3范式定义：第三范式(Third Normal Form,3rd NF)就是指表中的所有数据元素不但要能惟一地被主关键字所标识,而且它们之间还必须相互独立,不存在其他的函数关系。也就是说，对于一个满足2nd NF 的数据结构来说，表中有可能存在某些数据元素依赖于其他非关键字数据元素的现象,必须消除。

将第1，2范式化为第3范式的步骤：

（1）求出R的最小函数依赖集Fmin

（2）找出不在Fmin中出现的属性，并将这些属性从R中去掉，构成一个关系模式

（3）若Fmin中有一个函数依赖涉及R的全部属性，则R不能分解

（4）否则，若Fmin中有X->A,则分解应包含{XA};若有X->A1,X->A2....X->An均属于Fmin，则分解应包含{XA1A2...An}

本项目优化：表中数据已经满足第3范式的要求，无需再改。

## 4.4物理设计

DBMS选择：sqlserver 2016，SQLite

命名规范：表名全部使用次小写，字段名全部采用大写，单词间用“\_”连接，组相关字段使用“G”作为前缀，好友相关字段使用“F”作为前缀，字段名尽量使用单词表明其意义。

数据类型选择:可以使用int型的尽量使用int型，如账号应使用int，以提高效率。密码应采用处理速度较快的char类型，因为md5的结果长度是一样的，字符数较少的字段也应使用char类型并选择合适的长度，较长的可使用varchar，大量文本的存储应使用text类型，时间存储使用直观方便的datetime类型。

索引建立：在所有表的主键上建立索引，以加快查询速度。

加密：采用库外加密的方式。

## 4.5数据库安全

数据库管理与数据库接口使用两个不同的数据库账户，用户只能在登陆后通过程序中给定的相关操作来影响数据库，无其他方法，用户密码要加密存储。

## 4.6 数据库用户及权限

### 4.6.1软件使用

Role:pro

User:prou

Login:prouse

Prou拥有3张表的所有权限

### 4.6.2数据库管理使用

Login:sa拥有3张表的所有权限

## 4.7 数据同步

数据同步和数据绑定都可以使用ADO.Net自身的特性来解决，二者也是相互配合的，当我们将数据绑定到控件时，本地用户更改控件的值后，会自动更新绑定源(也就是内存中的DataSet)的值，本地数据源的改变项会更新到服务器的数据项中，其它在线用户是通过定时访问的方法来获得服务器数据库的改变项。

1. 数据绑定
2. 对于UserWindow中的记录本地用户的TextBox，使用简单绑定的方式。

e.g：。

1. 对于UserWindow中的ListBox，使用复杂绑定。

ListBox中的每个Item实际上都对应着DataTable中的一条记录，所以当我们绑定到数据源后，就能够访问记录，具体流程是：用户双击某条Item并获得下标，通过该下标获取对应的DataRow对象，将该DataRow对象传入新的DialogWindow。

1. 数据同步

数据同步分为三种

（1）一是从控件到本地内存的同步，完成数据绑定后此步自动完成，但是不明白为什么不是双向绑定。

（2）二是从本地内存到服务器数据库的同步，可以利用DataSet和DataAdapter来完成，向服务器请求更新事务的操作可以是即时的，因为更新项很少并且从时间上来看不会太频繁，更新项都是本地用户自身的一些属性

（3）是从服务器数据库到本地内存的同步，此步需要本地客户端主动访问服务器，频率在2秒1次。

## 4.8 建表SQL语句

--create database SQLTest

--go

--use SQLTest

--go

--CREATE TABLE [dbo].[Users](

-- [U\_Account] [varchar](20) primary key,

-- [U\_Signature] [nvarchar](50) NULL,

-- [U\_NickName] [nvarchar](20) NOT NULL,

-- [U\_Gender] [nchar](1) NULL,

-- [U\_BirthDay] [varchar](20) NULL,

-- [U\_Status] [nvarchar](10) NOT NULL,

-- [U\_Email] [varchar](20) NULL,

-- [U\_Pwd] [varchar](32) NOT NULL,

--)

--GO

--CREATE TABLE [dbo].[Friends](

-- [F\_SrcAccount] [varchar](20) NOT NULL,

-- [F\_DestAccount] [varchar](20) NOT NULL,

-- [F\_Remark] [nvarchar](20) NULL,

-- primary key(F\_SrcAccount,F\_DestAccount),

-- foreign key(F\_SrcAccount) references Users(U\_Account),

-- foreign key(F\_DestAccount) references Users(U\_Account)

--)

--GO

--CREATE TABLE [dbo].[Msg](

-- [M\_SrcAccount] [varchar](20) NOT NULL,

-- [M\_DestAccount] [varchar](20) NOT NULL,

-- [M\_Time] [varchar](20) NOT NULL,

-- [M\_ID] [int] NOT NULL,

-- [M\_Msg] [nvarchar](max) NOT NULL,

-- primary key(M\_SrcAccount,M\_DestAccount),

-- foreign key(M\_SrcAccount) references Users(U\_Account),

-- foreign key(M\_DestAccount) references Users(U\_Account)

--)

--GO

--alter table Users add U\_Account int IDENTITY(12356,1) NOT NULL;

--go

--EXEC sp\_addlogin 'prouse','prouse','master'

--go

--create role pro;

--EXEC sp\_adduser 'prou';

--EXEC sp\_adduser 'prouse','prou','pro'

--go

--grant all privileges on Users to pro;

--grant all privileges on Friends to pro;

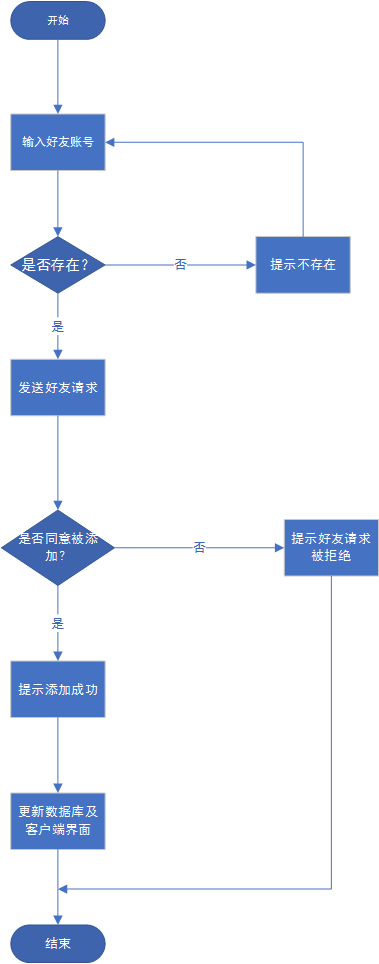
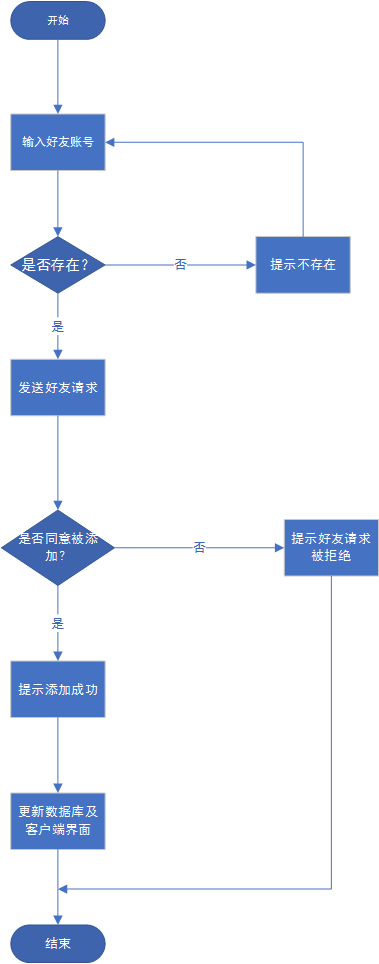
--grant all privileges on Msg to pro;

--go

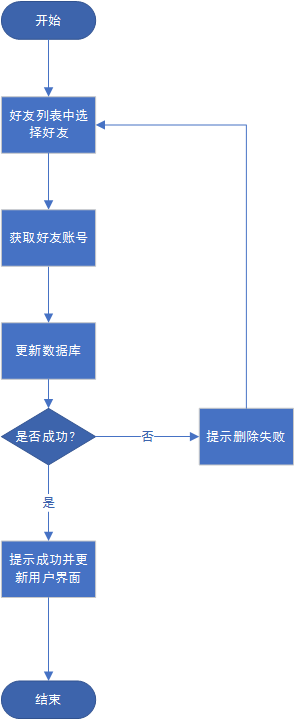
第5章 详细设计

## 5.1 程序模块流程图

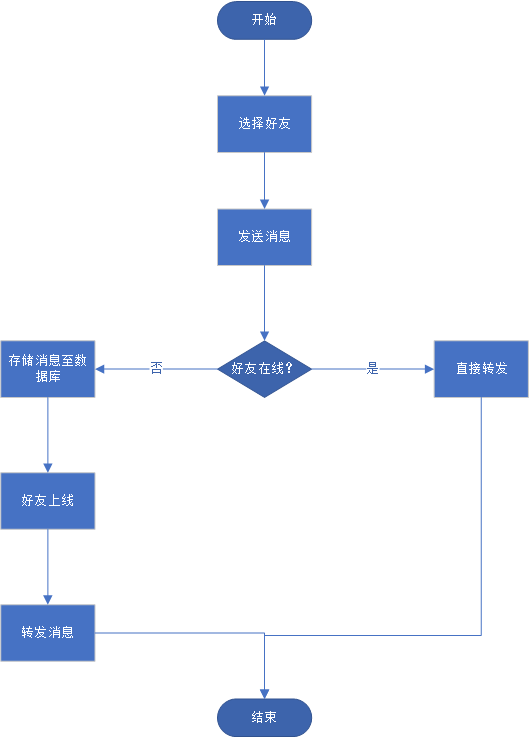
（1）登录与注册 （2）添加好友

（3）删除好友



（4）好友间聊天



## 5.2主要数据结构

public class SocketInfo

{

public static Socket socketConn;

public readonly static IPAddress serverIP;

public readonly static int serverPort;

}

public class Requester

{

public Socket Client { get; set; }

public string Account { get; set; }

public IPAddress IP { get; set; }

public int Port { get; set; }

public byte[] Msg { get; set; }

public int ID;

public string srcAccount;

public string destAccount;

public int Len;

public int acceptLen;

public List<byte> Packet;

}

static SocketInfo()

{

socketConn = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp); //use IPv4, socket type, Tcp

serverIP = IPAddress.Parse("127.0.0.1");

serverPort = 8990;

}

DataSet Ds;

SqlDataAdapter Adapter\_User;

SqlDataAdapter Adapter\_Friends;

## 5.3主要函数接口

（1）构造数据包

byte[] GetPackage(int id, string src, string dest, string data, string name = "", string extension = "")

（2）消息回调

void AcceptMsgCallback(IAsyncResult result)

（3）消息处理

void ProcessData()

（4）好友查找

SearchWindow(string account, string s)

（5）身份验证

bool Authenticate(string account, string pwd\_md5)

（6）连接接受回调

void AcceptClientCallback(IAsyncResult result)

（7）消息接收回调

void AcceptMsgCallback(IAsyncResult result)

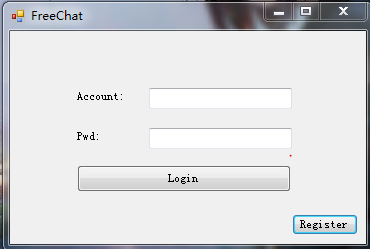
（8）消息处理

void ProcessData(Requester requester)

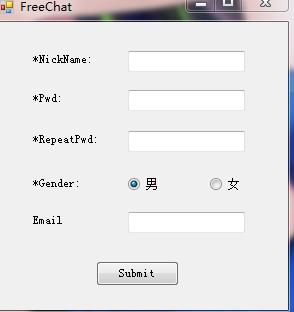
# 第6章 用户手册

## 6.1如何注册

第1步，打开程序，界面如下



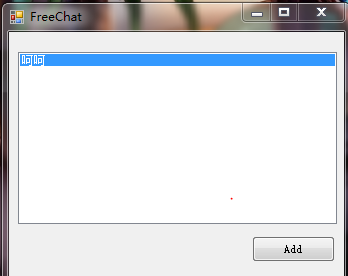
第2步 点击注册并填写信息，界面如下



第3步，点击提交即可得到账号

## 6.2如何加好友

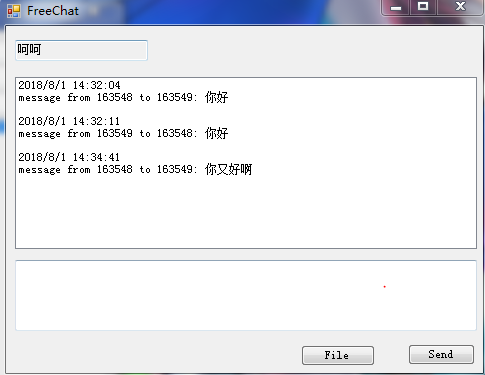
登陆后，在搜索框中输入好友完整账号或部分账号并回车



点击添加，等待对方同意即可。

## 6.3如何聊天

双击好友昵称即可打开聊天对话框开始通讯，界面如下



# 第7章 总结

通过此次对数据库课程设计，小组成员对数据库的应用都有了更为深入的了解，积累了不少经验，对SQL语句的使用也更为熟练，认识到了数据库安全的重要性。经过近1月的课设工作，软件完成了即时通讯的最基本功能，但任然有不少问题尚待改进，如数据库数据读写时代码的安全性，软件缺少群组功能等。

# 

# 参考文献:

[张海藩](http://search.dangdang.com/book/search_pub.php?category=01&key2=%D5%C5%BA%A3%B7%AA&order=sort_xtime_desc). 软件工程导论（第5版）[M].清华大学出版社,2008.

王珊, 萨师煊等. 数据库系统概论（第4版) [M].高等教育出版社.2006

欧立奇Visual C#.NET案例开发集锦[M]，电子工业出版社，2005.10.

Karli Watson等著，齐立波译.C#入门经典[M].清华大学出版社,2006.5