**概要设计文档**

**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**

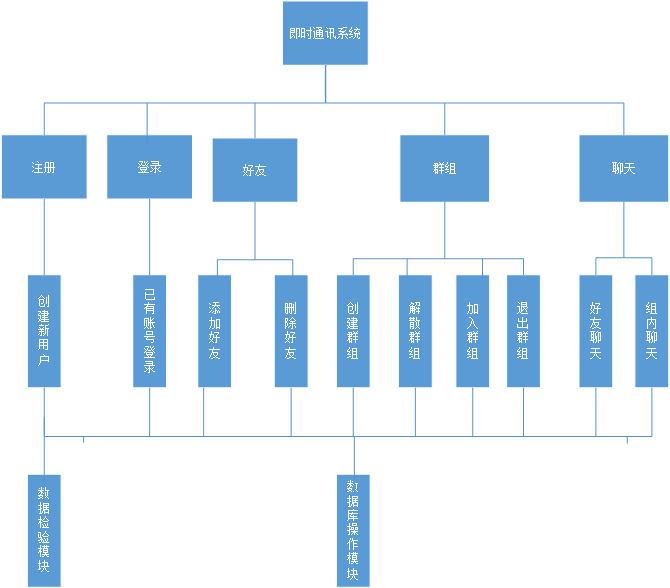
主机：pc兼容机

操作系统：Win 7/8/10

**命名规则**

函数名使用帕斯卡命名法。

**总体设计**



**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。

**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。

**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 |

**客户端**

（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 |  |

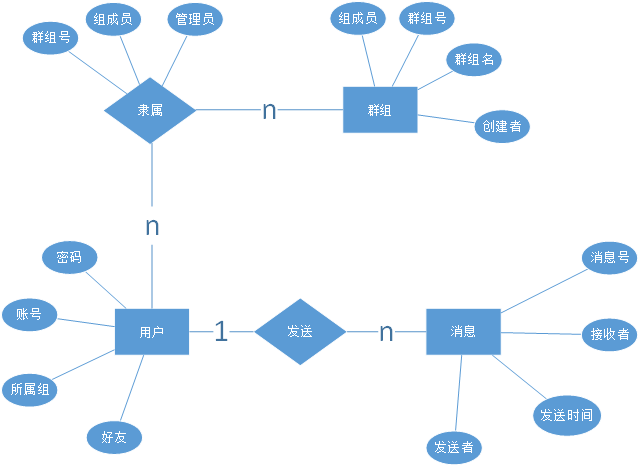
（2）本地群组表

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |

**数据库概念结构**



**数据模型**

用户（账号，密码，好友，所属组）

群组（群组号，群组名，创建者，组成员）

隶属（群组号，管理员，组成员）

消息（消息号，发送者，接收者，发送时间）

**物理设计**

DBMS选择：sqlserver 2016

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。

**安全保密设计**

数据库管理与数据库接口使用两个不同的数据库账户，分配不同的权限，都没有全表删除的权限，密码加密存储。

**测试计划**

数据库的测试先于软件其他部分测试，软件中不依赖数据库的模块可与数据库同时测试，待两边都完成时将两部分组合测试软件功能与性能是否满足要求。