# Chat for reading软件

# 需求分析与概要设计

## 项目说明

## 项目目标：

基于即时通讯工具实现为方便读书交流的一个交流软件

## 软硬件环境需求

本软件的服务器和客户端运行于Windows操作系统，使用mysql数据库进行各种数据的管理和存储。

## 使用的关键技术：

1. 服务器和客户端通过TCP/IP协议进行通信，通过调用Socket接口来实现
2. 交互界面使用C#语言通过其自带的Form类进行设计
3. 通过mysql将用户数据存储在关系型数据库中，并通过封装接口与服务器端进行链接。
4. 相关文献资料的检索
5. Chatgtp接口的嵌入

## 需求分析

## 系统用例

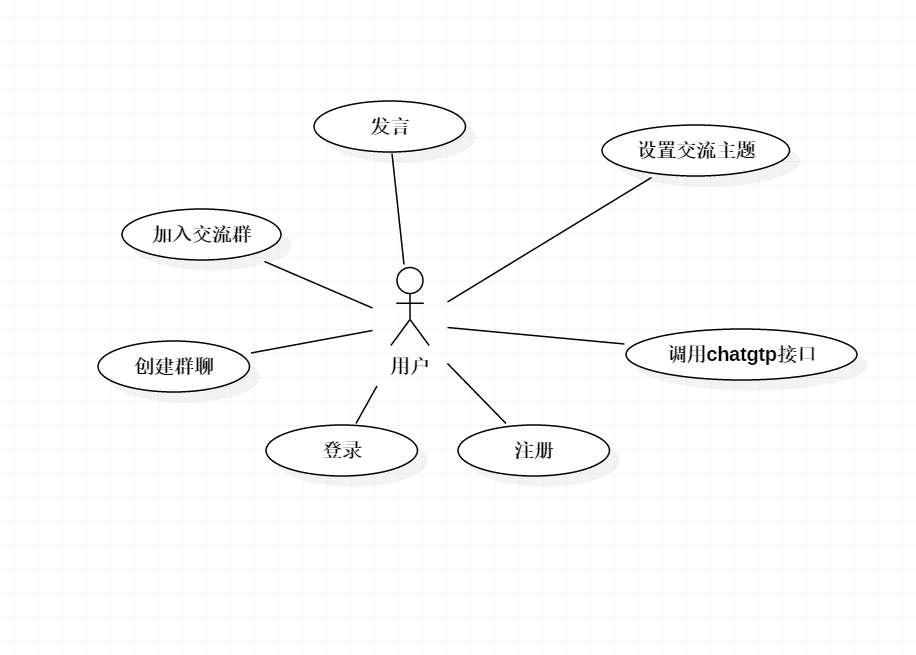


图 1 系统用例图

（

(对用例图中的各个用例进行详细说明，从使用者的角度说明如何使用系统，不用描述软件系统内部如何实现该功能)

1. **注册**

**参与者**：初始用户

**基本事件流**： 使用注册功能设置账号和密码，注册账号。

1. **登录**

**参与者**：已注册用户

**基本事件流**： 使用注册的账号进行登录

1. **创建交流群**

**参与者**：已经登录的用户

**基本事件流**： 使用创建群聊功能创建群聊

1. **加入群聊**

**参与者**：已经登录的用户

**基本事件流**： 加入已经创建的群聊

1. **发言**

**参与者**：加入群聊的用户

**基本事件流**： 在群聊里面发言进行讨论。

1. **设置讨论主题**

**参与者**：加入群聊的用户

**基本事件流**： 在群聊里面设置交流讨论的主题

1. **调用chatgpt接口**

**参与者**：加入群聊的用户

**基本事件流**： 访问内置的chatgpt接口实现答疑解惑。

## 业务流程

(使用UML活动图（使用StarUML）画出系统的使用和操作过程。不用画出所有的使用场景，画出一两个关键用例的使用过程即可)

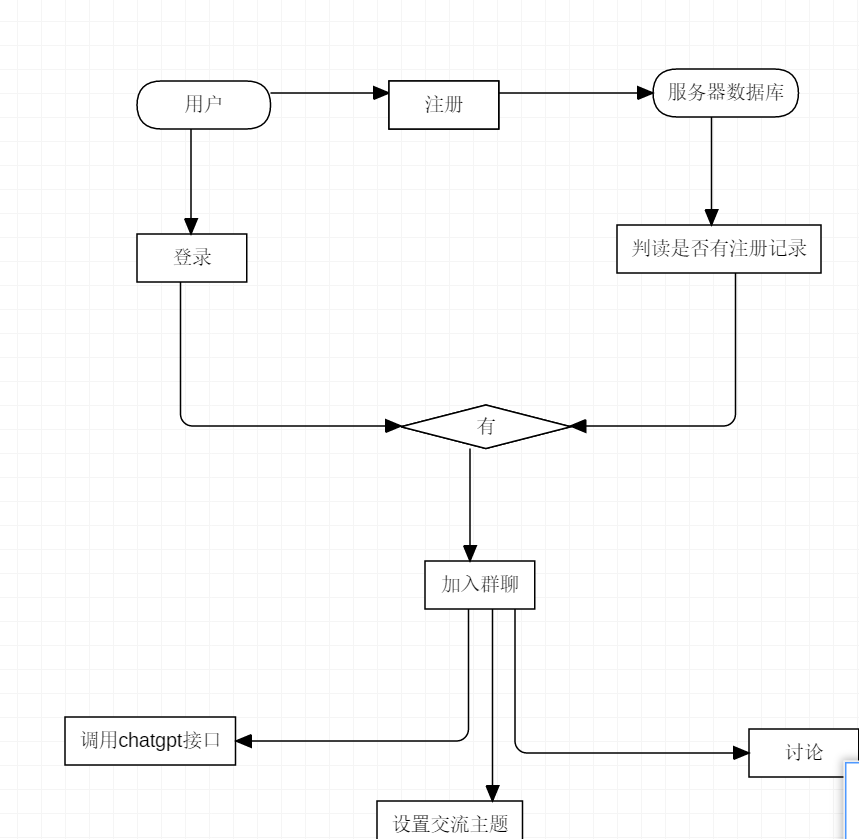


图 2用户交流流程图

## 概要设计

## 功能模块设计



图 3功能模块图

（对各个功能模块需要完成的事情进行简要介绍。注意和用例图不同，这里偏重于软件内部需要做的处理过程

**服务器端模块设计**

(1) 建立链接模块设计

该模块用于建立和客户端的链接，在服务器端启动时建立一个监听用套接字，新上线的客户端通过向该套接字发出请求来实现与服务器的链接。当有新的客户端发起链接请求时，创建一个新的套接字用于和该客户端的套接字组成套接字对，并创建一个网络流，服务器端可以通过对该网络流进行读和写来实现与客户端的数据传输。

(2) 指令解析模块设计

该模块将客户端发送来的信息转化为字符串并进行解析，并根据解析出的不同结果进行不同操作，如对从数据库中获取数据、更新数据库中的数据、向其他在线客户端发送消息、向该客户端返回消息等。

(3) 数据库交互模块设计

该模块可以在关系型数据库和服务器端应用程序之间建立链接，使服务器端能够对数据库中储存的数据进行查询和更新，从而满足整个软件系统运行的数据需求。

**客户端模块设计**

(1) 登录与注册模块设计

用户的注册：用户输入合法的账号和昵称，并输入两次相同的密码，当想注册的账号已经存在时会返回账号已存在信息，当两次密码输入不一致时会提示两次密码不一致并注册失败。输入全部合法时注册成功。

用户的登录：用户输入账号和密码进行系统登录，账号和密码输入正确时进入系统主界面。

(2) 主界面后台功能

建立链接功能：客户端登录成功后创建一个套接字，并向服务器的监听端口发出链接请求，得到服务器的建立链接的回复后，与服务器提供的套接字形成一个套接字对以创建链接，并创建一个网络流用于与服务器进行数据交互。

指令解析功能：客户端将服务器端发送来的信息转化为字符串并进行解析，并根据解析的结果做出操作，如修改客户端运行内存中的数据、将处理后的结果显示在界面中等。

子界面的调度功能：该系统登录后会打开一个主界面，当主界面被关闭时客户端下线。客户端的功能被分别实现在各个子窗口中，子窗口可以通过操作主窗口的控件来打开，客户端应保证子窗口的打开和关闭能够正确进行，每个打开的子窗口能够准确获取到其功能所需的数据，并在其关闭后予以释放，每个子窗口的操作不应影响到其他窗口。

(3) 系统消息模块

系统消息收到的入群申请。在此模块可以列表查看以上申请消息，并可以进行一些操作。收到的请求可以选择同意和拒绝，发出的申请消息可以撤销。群组申请，同意入群申请后，对方会成为该群聊成员，并可通过其客户端进入群组进行多人聊天；拒绝时，对方的已发出入群申请列表中会删除本条申请消息，并且不会加入该群聊。

(4) 发送申请模块

通过该模块可以发送入群申请，申请时，需要输入加入的群的编号，并选择想要群组分组。当该群组不存在的消息，当输入的编号合法，则会在自己系统消息中的入群申请列表中添加一条申请条目

(5) 群组管理模块

通过该模块可以管理自己创建和管理的群。用户可以创建自己的群组，并在该群组拥有群主权限，群组权限分为群主、管理员和普通群员。群主可以选择讨论的主题，并将文档设置显示在群聊中，群主和管理员可以在系统消息模块对用户的入群申请进行批准和拒绝，也可以将群员踢出群聊。群主还可以设定和取消群聊用户的管理员权限，或者将群转让给其他用户，群主拥有解散群组的权限。

(7) 多人聊天模块

用户可以在加入的群组中进行多人聊天，如果有群员发送消息，则所有在线群员会收到提示消息，并且离线的群员也可以在上线后对未读聊天记录进行接受和查看。

(8) 调用chatgpt模块

对每个群聊中有需要的用户提供访问chatgpt的接口。

## 核心类

1. 服务器端使用的类

用户实体类User

服务器中需要能够查看所有在线用户的信息，因此需要一个用户类来将在线用户实体化，用户实体类的定义如下：

private string Account; //用户的账号

private string Password; //用户的密码

private string Username; //用户的昵称

private string UID; //用户的UID

private string Account; //用户的账号

private TcpClient client; //用户的TCP链接类

private BinaryReader br; //该用户网络数据流的读端

private BinaryWriter bw; //该用户网络数据流的写端

数据库链接类SQL\_u

服务器需要能和数据库建立链接，本系统中使用MySQL对数据库进行管理，MySQL中提供了对.Net开发平台的动态链接库，在MySQL官方网站可以获取文件MySQL.Data.dll，将其链接到服务器项目中，便可以使用MySQL.Data命名空间中的方法来对数据库进行操作。数据库链接类的定义如下：

rivate MySQLConnection MySQL\_Con; //MySQL链接类

private string server; //MySQL服务器地址

private string database; //需要操作的数据库

private string uid; //本次进行操作的MySQL用户

private string password; //MySQL用户的密码

1. 客户端使用的类

用户实体类User

客户端中的用户类仅仅是用于储存用户信息，不需要与其建立链接，因此客户端中的User类仅包含用户基本信息。客户端用户实体类定义如下：

private string UID; //用户的UID

private string Username; //用户的昵称

群组实体类Group

对客户端中需要使用的群组进行了定义，群组是一组人进行多人聊天的单位。Group类的定义如下：

public string GID; //群组的GID

public string Groupname; //群组名

群组未读消息类Groupnoread

客户端将群聊中发送的聊天信息保存通过noread类来保存，客户端创建noread实体的List，新收到的信息都会被加入到该List中，当读取了信息后便将已读的信息从noread类中移除。noread类的定义如下：

：

public string GID; //群组的GID

public string time; //消息发送的时间

public string words; //消息内容本体

群组读写类Grouplock

客户端在读取noread List时，往往是打开一个聊天子窗口，子窗口打开后不仅是读取目前未读的消息，并且还需显示在打开窗口后收到的所有信息，此时主界面将新消息加入List，子界面需要读取List，为了使数据能够正确被操作，需要设计多个线程的同步互斥，因此在此创建该用户读写锁类来进行协调。grouplock类的定义如下：

public string GID; //群组的GID

public object newmsglock; //读取该群组信息的锁

public int msgcount; //未读消息的数量

## 界面设计

（画出系统主要操作界面，草图示意图即可）

图 5 系统界面