目 录

1 网络协议分析 1

1.1 系统概述 1

1.2 开发环境 1

1.3 系统设计 1

1.4 程序流程 1

1.5 主要数据结构 1

1.6 主要函数说明 1

1.7 系统使用说明 1

1.8 项目分析总结 2

2 网络应用编程（注明具体题目） 2

2.1 系统概述 2

2.2 开发环境 2

2.3 系统设计 2

2.4 程序流程 2

2.5 主要数据结构 2

2.6 主要函数说明 2

2.7 系统使用说明 2

2.8 项目分析总结 3

3 课程设计总结 3

参考文献 3

格式要求：正文采用小四号字体，行距为固定值20磅，首行缩进2字符。

# 网络应用编程（基于SSL的安全聊天软件）

第二模块的编程是实现一个简单的聊天软件，软件主要实现了客户端用户的登陆注册（使用数据库来进行存储用户的登录信息，并实现用户在下列表的自动更新。还要求能够处理各种不合法的请求并返回正确的错误提示）发文件（100M以上的大文件）和发消息（私聊，公聊，SSL消息加密）的功能，所有的功能按键触发相应的事件，将按键对应的功能的要处理的消息转发给服务器，服务器做相应的消息处理之后再回应客户端，服务器除了做上述的操作之外还可以进行用户的管理（删除相应用户的信息，展示所有用户），踢人功能。

## 系统概述

基于SSL的安全聊天软件，本软件实现了图形化界面，支持注册新用户，自动刷新在线用户列表，实现了公聊与私聊，客户端与客户端点对点传输文件，服务器动态管理在线用户的功能。

## 开发环境

Eclipse Mars

## 系统设计

**系统架构：**

该系统要求实现基于SSL安全通信，首先建立基于SSL的客户端与服务器，建立SSL通信的serversocket（服务器建立）和socket（客户端建立）以及安全证书和密钥来连接验证服务器和客户端，验证并连接成功后，就可以实现服务器与多个客户端进行各种操作。

其中对于客户端发来的操作消息来说，服务器相当于处理者和中转站。客户端可以通过界面进行登录注册发消息等操作，控件触发的事件传给服务器去做相应的处理，处理完成后产生的对应消息再返回给客户端去识别处理。

需要注意的是登录实现与数据库的关联，使得用户的注册登录信息存入数据库中，当用户登录时，检索数据库，匹配数据库中用户信息表中的用户名与密码，匹配成功则可以登录，匹配失败则说明用户没有注册。

另外一点是发送文件控件只是实现了服务器和客户端之间表面消息的传递，真正发文件是从发送客户端直接连接被当做服务器的接收客户端（不经过服务器因为转发速度慢）实现点对点传输。

**功能模块：**

客户端实现的基本功能有：注册，登录，发送消息，发送文件，退出。

（1）注册

注册实现了客户端手动输入需要注册的信息，将此信息发给与数据库相连的服务器去实现插入用户信息功能。

（2）登录

登录实现了客户端手动输入用户信息，将此信息发给与数据库相连的服务器去实现检索用户信息并插入到上线列表功能。

（3）发送消息

发送消息实现了客户端输入需要发送的消息并选择是公聊还是私聊，将按键消息转发服务器处理，服务器将对应的消息转发客户端。

（4）发送文件

发送文件实现了客户端输入需要发送的文件，将按键消息转发服务器处理，服务器讲对应的消息转发客户端。双方商量好接收文件后实现点对点通信。

（5）退出

用户退出时服务器端和客户端在线列表中将此用户剔除。

服务器实现的基本功能有：处理用户状态，处理用户消息，转发消息，管理用户，踢人。

（1）处理用户状态

处理客户端的状态消息（上线，下线），或者是客户端用户重复登录的错误消息提醒。

（2）处理用户消息

处理客户端发过来的聊天消息，（公聊，私聊，发送文件）根据不同的消息类型，进行不同的消息处理。

（3）转发消息

根据发来的消息中某些参数的判定，将一个客户端发过来的消息，转发给另外一个客户端，或者是另外多个客户端。

（4）管理用户

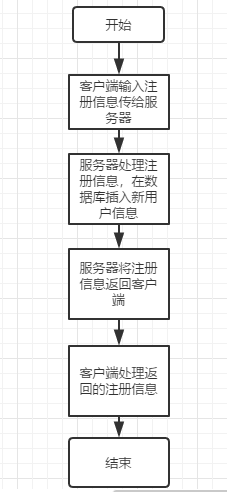
可以查看当前所有已经注册的用户，并且可以从数据库删掉某用户。

（5）踢人

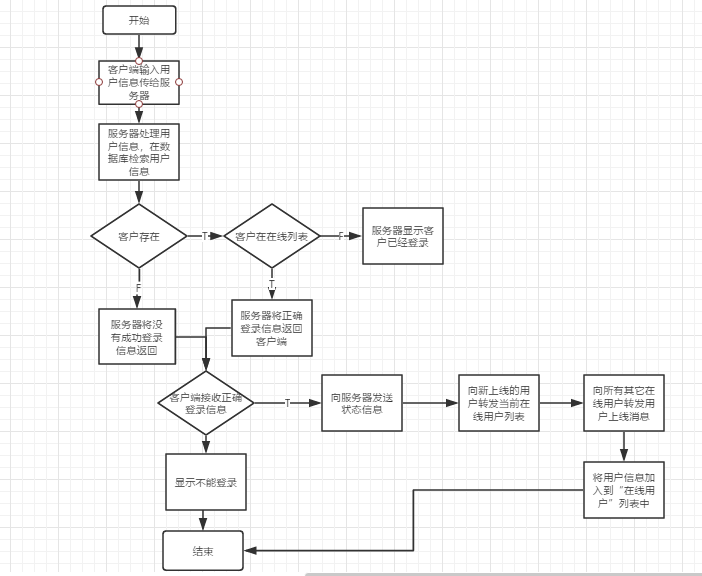
将某个用户踢出在线列表（用户强制下线）。

## 程序流程

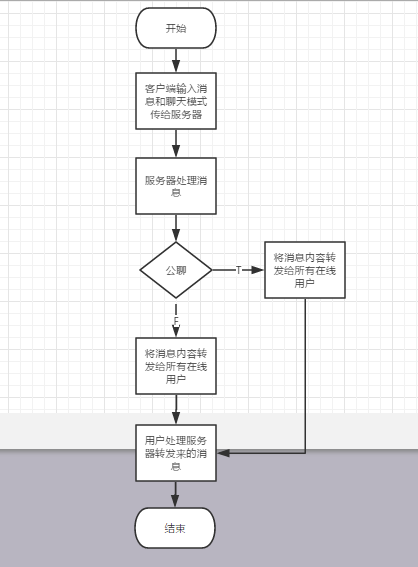
注册



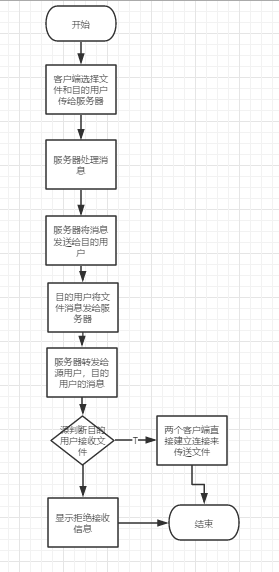
客户端登录以及用户状态



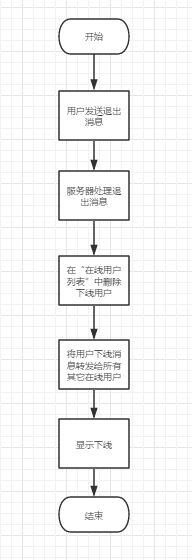
客户端发送消息（服务器转发消息，处理用户消息）



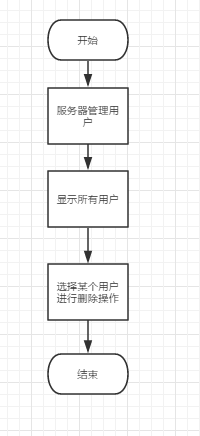
客户端发送文件（服务器处理用户消息）



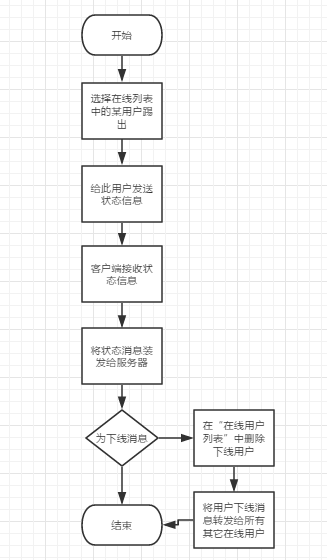
客户端退出（服务器处理用户状态）



服务器管理用户



服务器踢人



## 主要数据结构

主要使用了类，定义一个父类和多个子类，通过super函数让子类继承父类并进行不同的扩展。

class Message

消息类，在该消息类中只定义了发送端的用户名，接收端的用户名。

class ChatMessage

聊天消息类，定义了客户端的聊天内容，与继承自父类的源用户和目的用户，以及判断公聊和私聊的标志。

class UserStateMessage

状态消息类，定义了用户的状态消息，与继承自父类的源用户和目的用户，以及是否需要广播状态消息的函数。

class FileSendMessage

文件发送消息类，定义了文件名，文件大小，文件路径与继承自父类的源用户和目的用户以及发送成功的标志。

class FileResponseMessage

文件回复消息类，定义了文件大小，文件路径与继承自父类的源用户和目的用户以及是否接收的标志。

class RegisterMessage

注册消息类，定义了性别和密码与继承自父类的源用户和目的用户。

class LoginMessage

登录消息类，定义了密码与继承自父类的源用户和目的用户。

## 主要函数说明

**（1）客户端：**

处理用户注册消息processRegisterMessage((RegisterMessage)msg);

接收服务器发来的注册消息，成功则显示成功。

处理用户状态消息processUserStateMessage((UserStateMessage) msg);

接收服务器发来的状态消息，如果是新用户登录，则将此用户添加到自己的在线列表，如果用户下线，则将此用户删除出在线列表，并将下线消息转发给服务器。

处理聊天消息processChatMessage((ChatMessage) msg);

接收服务器发来的聊天消息，判断是公聊还是私聊，在对话框显示不同的情况。

处理登录消息processLoginMessage((LoginMessage) msg);

接收服务器发来的状态信息，显示登录信息并将登录按钮设为退出按钮。

处理被请求用户回应请求用户消息processFileSendMessage((FileSendMessage) msg);

发送方发送文件消息给服务器，服务器询问接收方是否接收，接收的话服务器将接收的消息发给发送方，发送方返回服务器文件大小的消息，服务器转发给接收方。

处理请求用户发送文件processFileResponseMessage((FileResponseMessage) msg);

接收方同意接收文件时，设置滚动条，建立点对点通信，发送文件。

**（2）服务器：**

处理用户登录消息processLoginMessage((LoginMessage) msg);

接收客户端发来的登录消息，检索数据库，如果用户存在于数据库且不在当前在线列表，则将正确的登录消息转发回客户端，如果用户存在于数据库且在当前在线列表，提示用户已经登录，不然提示用户错误信息。

处理用户注册消息processRegisterMessage((RegisterMessage) msg);

接收客户端发来的注册消息，检索数据库，如果是新用户，则将用户信息插入数据库，如果不是，则显示重复登录，并将对应的消息转发回客户端。

处理用户状态消息processUserStateMessage((UserStateMessage) msg);

接收客户端发来的状态消息，当是上线消息时向新上线用户转发当前在线用户列表，将新上线用户的消息转发给当前的每一个用户，将用户信息加入到服务器端的在线用户列表中，如果是下线消息，将此用户从在线用户列表移除，并将此用户的下线消息发给其他在线用户做客户端的相应处理。

处理聊天消息processChatMessage((ChatMessage) msg);

接收客户端发来的聊天消息，如果是公聊消息，则把这条消息转发给每一个（除了发送者）在线用户，如果是私聊消息，则把这条消息转发给指定（除了发送者）在线用户。

处理被请求用户回应请求用户消息processFileSendMessage((FileSendMessage) msg);

接收接收方的回应是否接收文件的消息，将此消息发送给源用户。

处理请求用户发送文件消息processFileResponseMessage((FileResponseMessage) msg);

接收发送方的发送文件请求消息，将此消息转发给目的用户。

向其他用户转发消息private void transferMsgToOtherUsers(Message msg)

遍历在线用户列表，将指定的消息通过获取每一个用户的输出流并封装成对象输出流的方法转发给其他用户。

向指定用户转发消息private void transferMsgToDstUser(Message msg)

遍历在线用户列表，得到指定用户，将指定的消息通过获取每一个用户的输出流并封装成对象输出流的方法转发给目的用户。

## 系统使用说明

该系统能够实现用户间的公聊和私聊，以及发送文件。先启动服务器，再连接客户端。

1. 未注册过的用户先进行注册（注册信息有姓名，性别，密码），注册成功后可以进行登录（登录信息有姓名，密码）。
2. 用户可以通过选择来决定是要发公聊消息还是私聊消息（文本框输入公聊消息后直接点击发送消息按钮，私聊消息需要选择用户并且勾选私聊按钮）。
3. 用户也可以通过在线用户列表选择一个其他在线用户来点击发送文件按钮发送指定文件。

服务器用来转发处理客户发来的各种请求消息，并且服务器和数据库连接用来管理所有用户。

1. 服务器可以选中当前在线用户列表中的用户点击下线按钮来进行踢人操作（注：踢人操作只是将用户强制踢下线，数据库中此用户信息还在）。
2. 服务器可以点击用户管理按钮进入用户管理界面，先选中用户再点击删除按钮可以删除此信息（数据库中不再存在此用户信息）。

## 项目分析总结

遇到的问题：

接收文件和发送文件不会写。滚动条有问题。

解决方法：

询问同学，文件的传输不经过服务器，直接把接收方当成服务器，两个客户端进行socket的连接，用文件流来进行文件的传输。滚动条的改正通过改变收文件和发文件的线程中的while循环来改动。

亮点：

可以实现私聊，以及服务器踢人功能。

不足：

发送消息和发送文件不是加密传输，时间原因没有实现用户头像。

# 课程设计总结

在经过本次课设之后，熟悉了Java语言，巩固了上学期学的知识，也熟悉了Eclipse中图形化界面的使用和编辑，windowbuilder也用的比上学期熟练了，编程能力有所提升，了解到通讯加密的重要性，知道了SSL的加密算法，对实践课程的重要性有了更加深刻的理解。

建议：希望学院能给实验课多一些课时。

# 参考文献

示例《chatExample2》，《SSLExample》