# Java聊天室需求分析文档

## 1 需求分析

1.1 业务需求

1. 与聊天室成员一起聊天。 完成

2. 可以与聊天室成员私聊。 完成

3. 可以改变聊天内容风格。

4. 用户注册（含头像）、登录。 完成 //头像未显示

5. 服务器监控聊天内容。 完成

6. 服务器过滤非法内容。 完成

7. 服务器发送通知。完成

8. 服务器踢人。 完成

9. 保存服务器日志。 完成

10.保存用户聊天信息。 完成

1.2 系统功能模块

1.2.1 服务器端

1.处理用户注册 完成

2.处理用户登录 完成

3.处理用户发送信息 完成

4.处理用户得到信息 完成

5.处理用户退出 完成

1.2.2 客户端

1.用户注册界面及结果 完成

2.用户登录界面及结果 完成

3.用户发送信息界面及结果 完成

4.用户得到信息界面及结果 完成

5.用户退出界面及结果 完成

1.2.3 客户端结构

ChatClient.java 为客户端程序启动类，负责客户端的启动和退出。

Login.java 为客户端程序登录界面，负责用户帐号信息的验证与反馈。

Register.java 为客户端程序注册界面，负责用户帐号信息的注册验证与反馈。

ChatRoom.java 为客户端程序聊天室主界面，负责接收、发送聊天内容与服务器端的Connection.java 亲密合作。

Windowclose 为ChatRoom.java的内部类，负责监听聊天室界面的操作，当用户退出时返回给服务器信息。

Clock.java 为客户端程序的一个小程序，实现的一个石英钟功能。

## 2 .系统实现原理

当用户聊天时，将当前用户名、聊天对象、聊天内容、聊天语气和是否私聊进行封装，然后与服务器建立Socket连接，再用对象输出流包装Socket的输出流将聊天信息对象发送给服务器端.

当用户发送聊天信息时，服务端将会收到客户端用Socket传输过来的聊天信息对象，然后将其强制转换为Chat对象，并将本次用户的聊天信息对象添加到聊天对象集Message中，以供所有聊天用户访问。

接收用户的聊天信息是由多线程技术实现的，因为客户端必须时时关注更新服务器上是否有最新消息，在本程序中设定的是3秒刷新服务器一次，如果间隔时间太短将会增加客户端与服务器端的通信负担，而间隔时间长就会让人感觉没有时效性，所以经过权衡后认为3秒最佳，因为每个用户都不可能在3秒内连续发送信息。

当每次用户接收到聊天信息后将会开始分析聊天信息然后将适合自己的信息人性化地显示在聊天信息界面上。

## 3. 实现功能

* 接受用户注册信息并保存在一个基于文件的对象型数据库。
* 能够允许注册过的用户登陆聊天界面并可以聊天。 完成
* 能够接受私聊信息并发送给特定的用户。完成
* 服务器运行在自定义的端口上。
* 服务器监控用户列表和用户聊天信息（私聊除外）。
* 服务器踢人，发送通知。
* 服务器保存日志。