

# 武汉理工大学华夏学院

## 课程设计报告书

课程名称: Java 课程设计

题    目: 基于 Java 平台的飞鸽传书聊天室系统的设  
计与开发

系    名: 信息工程系

专业班级: 软件 1121

姓    名: \_\_\_\_\_

学    号: \_\_\_\_\_

指导教师: 邱珊

2014 年 9 月 12 日



# 课程设计任务书

学生姓名：\_\_\_\_\_ 专业班级：\_\_\_\_\_ 软件 1121

指导教师：\_\_\_\_\_ 邱珊 \_\_\_\_\_ 工作单位：\_\_\_\_\_ 信息工程系 \_\_\_\_\_

设计题目：\_\_\_\_\_ 基于 Java 平台的飞鸽传书聊天室系统的设计与开发 \_\_\_\_\_

初始条件：

Java SE，Socket 网络编程，聊天室系统。

要求完成的主要任务：

（包括课程设计工作量及其技术要求，以及说明书撰写等具体要求）

使用 JAVA SE 开发一个微型聊天室系统，该聊天室系统包括客户端和服务端。本系统使用 JAVA SWING 技术进行前台用户界面的设计和开发，使用 Socket 与 ServerSocket 网络编程进行客户端与服务器的连接与通信，使用输入输出流进行客户端与服务器端之间的数据传输。该系统将能够实现用户登录功能，登录的用户将可以进行群发和私聊，并能够发送文件。

系统总体功能模块图如下：

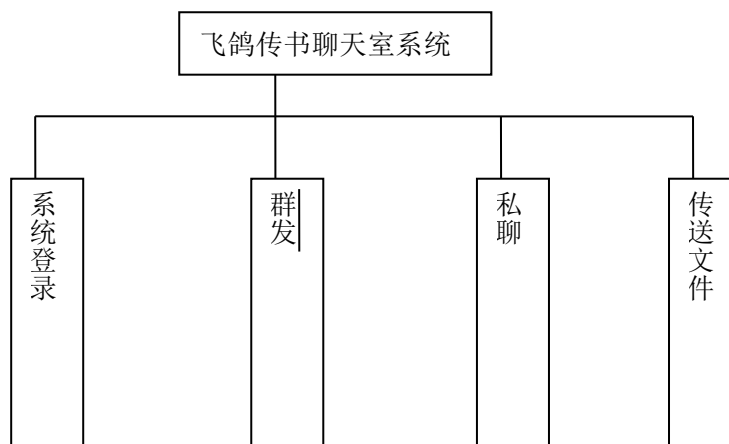


图 1 系统功能模块图

## 主要任务：

| 编号 | 功能名  | 功能描述                             |
|----|------|----------------------------------|
| 1  | 登录系统 | 运行系统，输入用户名和密码后，进入系统              |
| 2  | 群发   | 客户端发送的信息被服务器发送并显示到其他所有登录的用户的客户端。 |
| 3  | 私聊   | 可以选择跟登录的某个用户进行私下聊天。              |
| 4  | 传送文件 | 可以选择向登录的某个用户发送文件。                |

## 时间安排：

| 日期   | 时间段 | 内容                                                                            |
|------|-----|-------------------------------------------------------------------------------|
| 9.9  | 上午  | 项目的分析, 研究现状                                                                   |
|      | 下午  | 本系统要实现的基本目标和研究内容                                                              |
| 9.10 | 上午  | 飞鸽传书的界面布局及组件的使用                                                               |
|      | 下午  | 对界面上的组件的事件响应和处理                                                               |
| 9.11 | 上午  | 采用 UDP 协议进行数据流的发送信息                                                           |
|      | 下午  | 利用数据报包的信息接收及信息的展示                                                             |
| 9.12 | 上午  | 利用 UDP 协议将文件转换成二进制对象进行网络中的数据传递及群发功能<br>利用 UDP 协议将数据接收后转换成相应的文件类型保存到本地磁盘实现文件接收 |
|      | 下午  | 项目测试与验收                                                                       |

指导教师签字：



2014 年 9 月 5 日

系主任签字：

马德宝

2014 年 9 月 6 日

# 1 目标

设计并实现一个基于 TCP 协议或者 UDP 协议的文字聊天软件，可以是点对点的，通信双方都可以主动发起连接；或者是基于服务器的，客户端之间通过服务器通信。要求：采用 Windows 应用程序风格，功能完整，显示直观，色彩协调，界面友好，操作简单。

## 2 软件开发技术与环境简介

### 2.1 Java 技术和 Socket 网络编程简介

本聊天室系统是基于桌面的 J2SE 应用系统，使用 JAVA Swing 技术进行前台界面的设计，使用 Socket/ServerSocket 技术进行网络编程。

### 2.2 Java 技术和 Java Swing 技术介绍

Swing 是一个用于开发 Java 应用程序用户界面的开发工具包。它以抽象窗口工具包(AWT)为基础使跨平台应用程序可以使用任何可插拔的外观风格。Swing 开发人员只用很少的代码就可以利用 Swing 丰富、灵活的功能和模块化组件来创建优雅的用户界面。

### 2.3 Java Socket 编程技术介绍

首先，应利用 JFrame 建立一个基本窗口框架，其包含有输入信息的文本条 JTextField 类和输出信息即显示外来的信息的文本域 JTextArea 类，在文本条上添加一个事件监听器，利用监听器可是对服务器端用户即管理员进行交互。并且，在文本域上添加一个滑动条，这样就使其文本域可以接受比当前所能显示的最多还多。

服务器端是循环等待来检查指定端口是否有用户申请的，所以，用到了多线程，在多线程中是利用一个死循环 while(true)来实现的 当没有用户申请时，则在 new ServerSocket 这条语句上阻塞，不会执行循环，而当有用户申请时，则会被唤醒，这样就利用 Server Demo.list.add(s);这条语句将用户当前的 Socket 插入链表，而这条链表，则记录了当前所有的申请用户。对于服务器向所有的用户通信，则可以直接在输入的文本条中的 监听器中实现，即获取当前所要输出的信息后，利用链表，遍历存储了所有用户的链表逐一发送信息。对于服务器对某个用户的信息进行处理即传递给其他用户的实现，也是通过线程来实现的。首先在其内部存在一个死循环，这个死循环的上方一定要判断是否有用户申请，必须由用户申请才能进入，否则将会出错。当进入死循环后，将会建立第一个线程，也就是第一个申请用户的，让后判断是否还有用户，直到没有用户申请时，即所有的用户都已经被创建完线程，则利用 while(true)if(i==list.size()-1) continue;语句死循环，当又有一个用户申请，则推出这层死循环，重复上一次外层循环的内容，这样只是实现了对每个用户的线程的创建，对于每个线程的实现，则只需要利用输入输出流就行一层一层的包装，最后包装成所希望的输出类型就可以输出。

### 2.4 硬件环境

1. 操作系统：Microsoft Windows XP 或 Microsoft Windows NT Service/Workstation4.0

或更高版本。

2. 内存: 64MB 内存以上。

## 2.5 软件环境

1) 开发工具: MyEclipse 8.6

2) 开发环境:JDK 1.7

# 3 系统需求分析

本系统主要通过客户端, 服务器端来体现聊天系统的基本功能。

## 3.1 服务器端

1. 处理用户注册
2. 处理用户登录
3. 处理用户发送信息, 服务器监控聊天内容, 服务器过滤非法内容
4. 处理用户得到信息
5. 处理用户退出
6. 服务器踢人。
7. 保存服务器日志。
8. 保存用户聊天信息。

## 3.2 客户端

1. 用户注册界面及结果
2. 用户登录界面及结果
3. 用户发送信息界面及结果
4. 用户得到信息界面及结果
5. 用户退出界面及结果

# 4 系统总体设计

本系统针对局域网进行联机聊天。聊天室分为服务器端和客户端两部分, 服务器端程序主要负责侦听客户端发来的信息, 客户端需要登陆到服务器端才可以实现正常的聊天功能。本系统主要实现如下功能:

服务器:

在特定端口上进行侦听, 等待客户端连接。

用户可以配置服务器端的侦听端口, 默认端口为 8888。

向已经连接到服务器端的用户发送系统消息。

统计在线人数。

当停止服务时, 断开所有的用户连接。

系统运行稳定、安全可靠。

一台主机只能启动一个服务器。

客户端：

连接到已经开启聊天服务的服务器端。

用户可以配置要连接的服务器端的 IP 地址和端口号。

用户可以配置连接后显示的用户名。

用户可以更改聊天时的表情。

当服务器端开启后，用户可以随时登录和注销。

用户可以向所有人或某一个人发送消息。

系统运行稳定、安全可靠。

可以默认连接到自己主机启动的服务器。

5 系统详细设计

5.1 服务器模块

服务器模块主要完成接收并处理用户登录申请，转发登录用户发送的消息、启动聊天服务、关闭聊天服务和踢开指定用户功能。该模块应包括三个子功能模块：处理用户登录申请功能、转发用户消息功能、聊天室管理功能。模块功能树如图 2-2 所示。

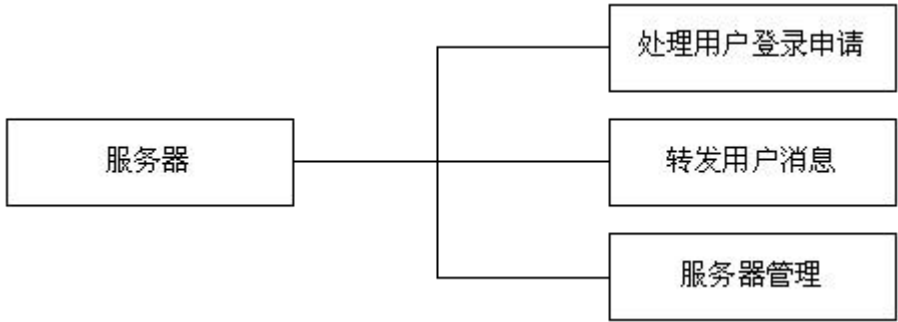


图 1 服务器模块功能树图

5.2 客户端模块

客户端模块为用户提供聊天室操作界面，用户通过客户端可以发送消息给全部或指定的其他用户，可以看到聊天内容，可以看到聊天室内有哪些用户在线。客户端模块功能树如图 2-3 所示。

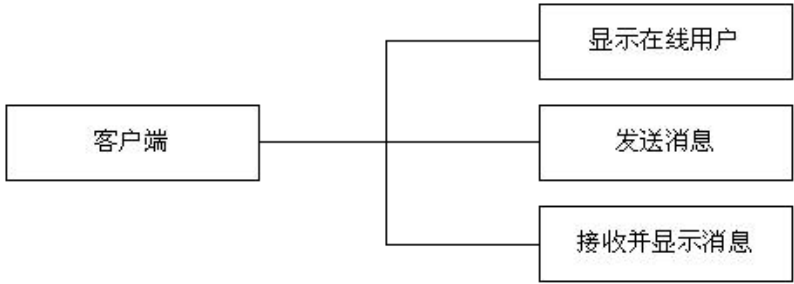


图 2 客户端模块功能树图

## 6 系统测试与运行

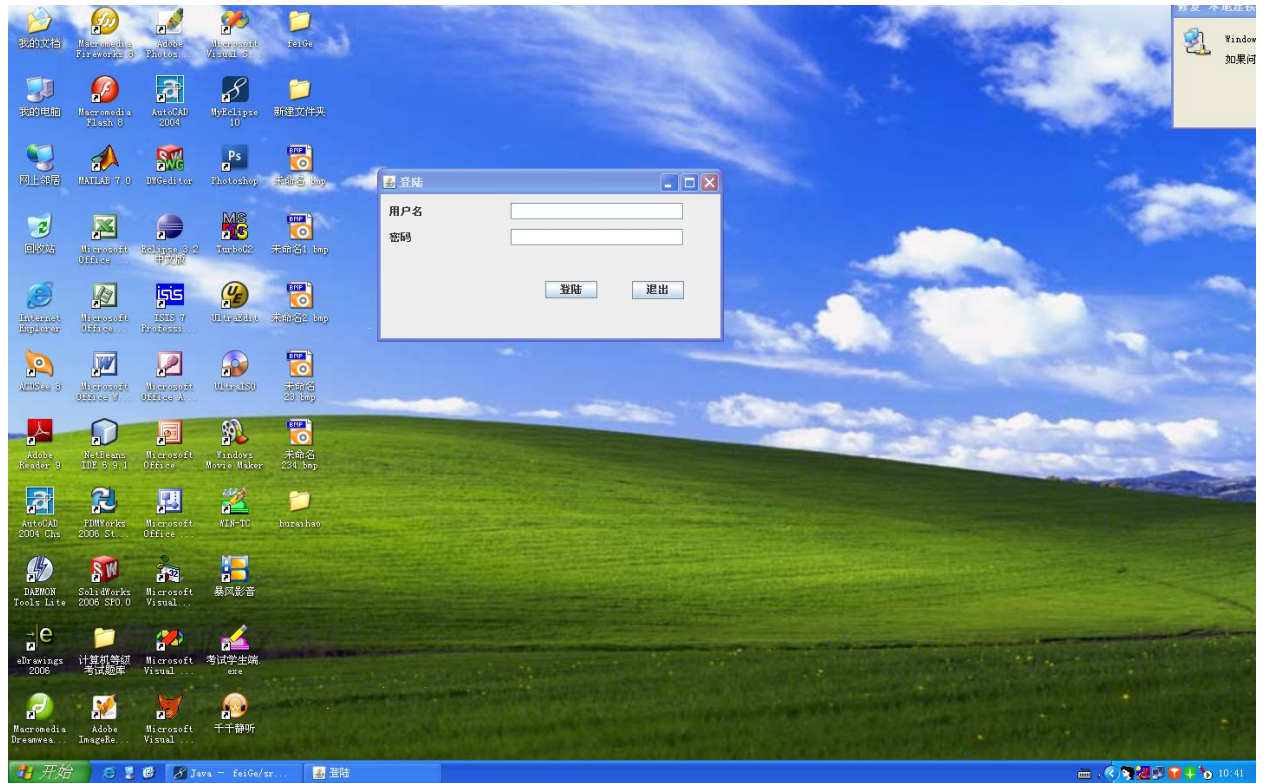


图 1.登陆界面

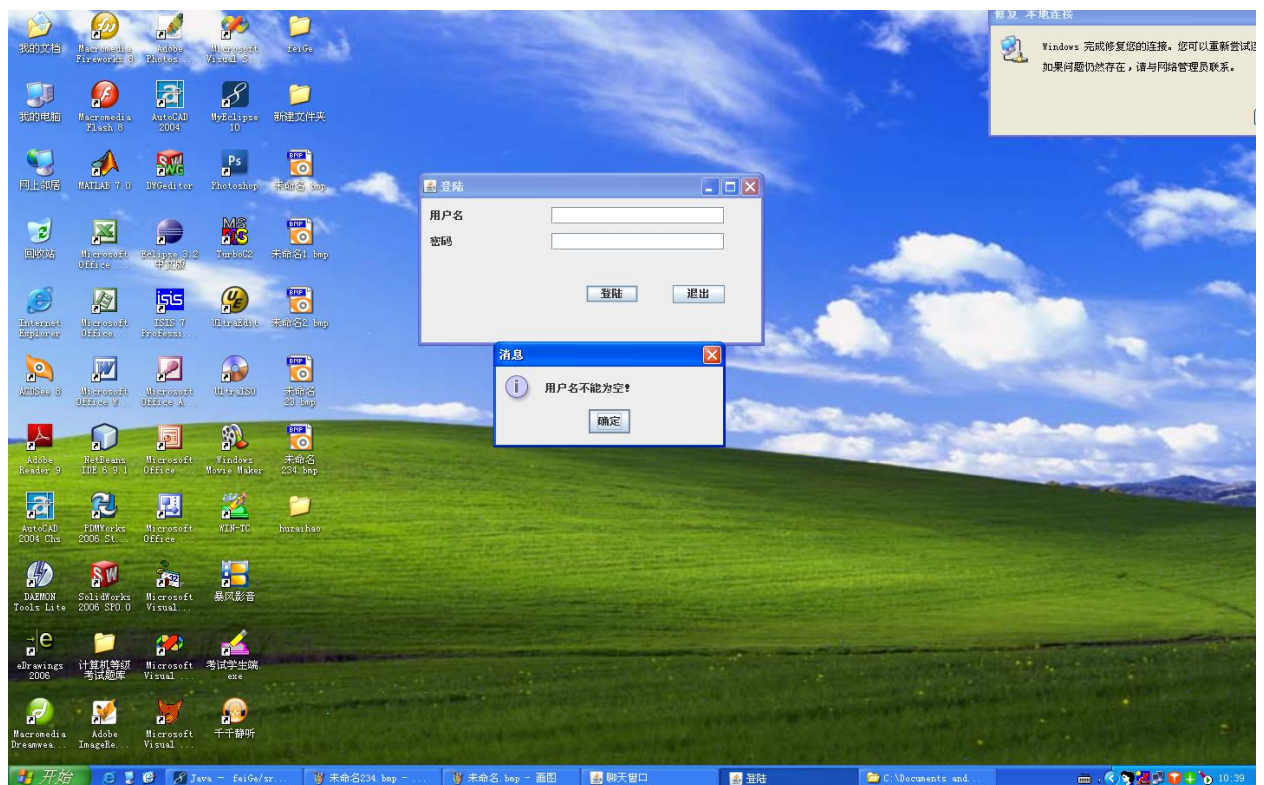


图 2.若输入用户名为空后提示

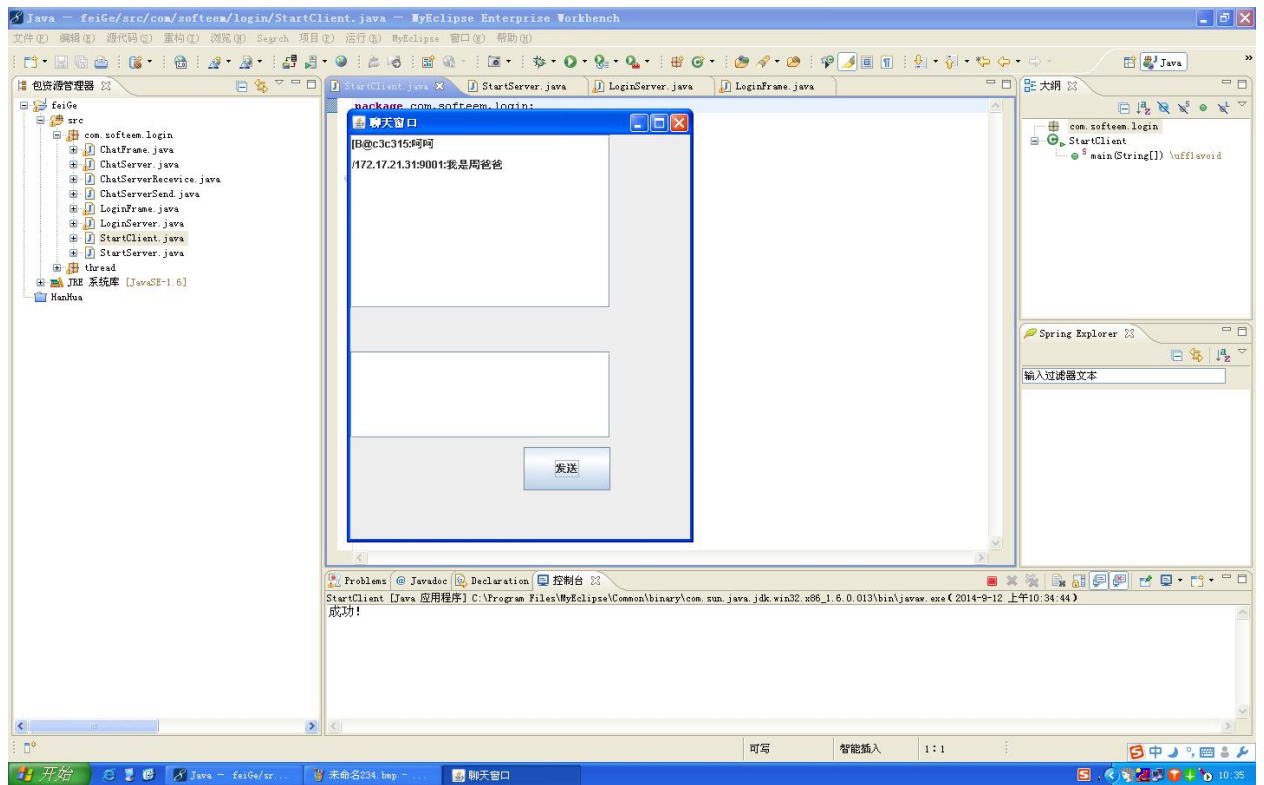


图 3.聊天界面

## 7 总结

一周的课程设计结束了，我的收获很大。通过课程设计的编程和对编程过程中遇到的问题的思考，有效的提高了我对数据结构的认识和理解，切实的提高了我的编程能力，补全了我的不足。通过这次课程设计，我巩固了所学过的知识，能用 JAVA 熟练地运用其中的一些，进一步掌握了各个指令的功能是转向。课程设计是对我们所学的这些东西一个综合的运用。在整个程序的编写中，调试占了很大部分的时间。要想写好程序，必须认真对待代码的每一个细节，还必须熟练的掌握各种命令，这对程序的调试是非常重要的。通过这次实际动手能力的培养，我进一步熟悉了使用方法，基本达到了能独立阅读、编制和调试一定规模的水平。课程设计是对自己所学知识掌握程度最好的检验，特别是对于一些细节问题。考试还并不能完全的说明什么，因为考题都是一些单个的问题，大多只是知识的某一方面。而课程设计不同，我们要用到很多的单个知识点，把他们综合起来运用到自己的课题当中。学了一个学期的数据结构，现在终于迎来了课程设计，当然是一个很好的锻炼自己的机会。

在本次的课程设计也发现了自己很多的缺点不足之处，对于一些细节性的问题老犯错，特别是一个标点的错误，很急人。对于有些知识掌握的不行，有的是模模糊糊的不牢靠，虽然有的东西考试考过，但当它用到实际问题的时候，往往不知道用哪，放哪。实践出真知。课程设计是个实践，虽然实际不长，但给我们对知识的运用，利用都有了很好的发挥。人多力量大这句话是真理啊，不懂得地方问同学，不会的地方也问同学，自己努力的同时跟别人共同进步，这就是合作，学会合作是我们以后必备的重要的一项人际技能。一个程序的完成需

要不断地修改才能够得到完善，人生正如那程序一样，只有不断地努力、奋斗、勇往直前，才能够完善自己，做最好的自己。

最后感谢同学的帮助，感谢老师的指导！

## 8 参 考 文 献

- 1、《JAVA 编程技巧》，机械工业出版社
- 2、《JAVA 语言面向对象程序设计》，机械工业出版社
- 3、《JAVA 项目案例开发》，电子工业出版社

## 9 附录 系统核心代码

```
package com.softem.login;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class ChatFrame extends JFrame{
// 声明成员变量 方便在各个方法中使用
    // 定义一个 jlist
    private JList jlist = null;
    // 定义一个 jtextarea
    private JTextArea jtextarea= null;
    // 定义 jlist 滚动条
    private JScrollPane jls = null;
    // 定义 jtextarea 滚动条
    private JScrollPane jta = null;
    // jpanel
    private JPanel panel = null;
    // 定义一个按钮 “发送”
    private JButton send = null;

    //定义按钮的方法
    public JButton getJbtu(){
        if(send==null){
            send = new JButton("发送");
            send.setBounds(200, 360, 100, 50);
        }
        send.addActionListener(
            new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent e) {
                    String msg = jtextarea.getText();
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/325010012203011303>