哈爾濱工業大學（威海）
Harbin Institute of Technology, Weihai

《软件设计与开发实践 I》课程报告

任务名称： 志愿者管理系统

班　　级： 1604104　组号： 34

学　　号： 160820321

姓　　名： 胡　起

学年学期： 第二学年第一学期

任课教师： 李 春 山

哈尔滨工业大学（威海）计算机科学与技术学院

前　言

软件设计与开发实践 I 是高级语言程序设计 I、II 和 Java 程序设计语言的配套实践课。计算机科学与技术、信息安全等专业的本科生在完成高级语言程序设计 I、II 和 Java 程序设计课程学习后，通过本课程进入专业实践训练环节，其主要目的是使学生深入理解并实践高级语言程序设计 I、II 和 Java 程序设计语言课程中所学的有关知识。该课程对增强学生对基础知识的掌握和综合运用是非常必要的，对后继许多专业课程的学习和实践训练都具有十分重要的意义。通过课程设计培养学生将理论知识应用于实践的能力，独立思考解决问题的能力，综合分析与设计的能力，主动、自学习的能力，团队合作的能力，归纳、总结、写作、评价的能力，以及互相交流的能力。

本课题实践环节要求学生分组进行（每组 1~2 人），自行选题，选题的思想是根据实际需求进行调研，每人提交课程选题任务书（同组人员需各自独立撰写完成任务书，并明确任务分工），给出所选项目的背景和意义、所要解决的应用问题及软件系统的核心功能等，并由任课教师进行开题检查认可；之后进入程序设计、编码和调试运行阶段，主要是以实用性和系统性为主，完成具有一定实际使用价值的软件系统项目。经过中期检查、软件结题验收等环节后，撰写完整的课程报告并提交。

教师评语：   报告成绩：

1. 选题背景与应用意义

1)选题背景

校内很多组织和社团都有一些志愿活动，志愿时间统计都是由各组织填写写实表然后提交给志愿者服务中心，志愿者服务中心累加时长，将志愿时长添加到数据库中。但是这些数据是通过手工登记，既浪费了时间，又可能发生漏登错登等问题，所以通过编写程序，提高志愿时长的统计效率和准确率。

2)应用意义

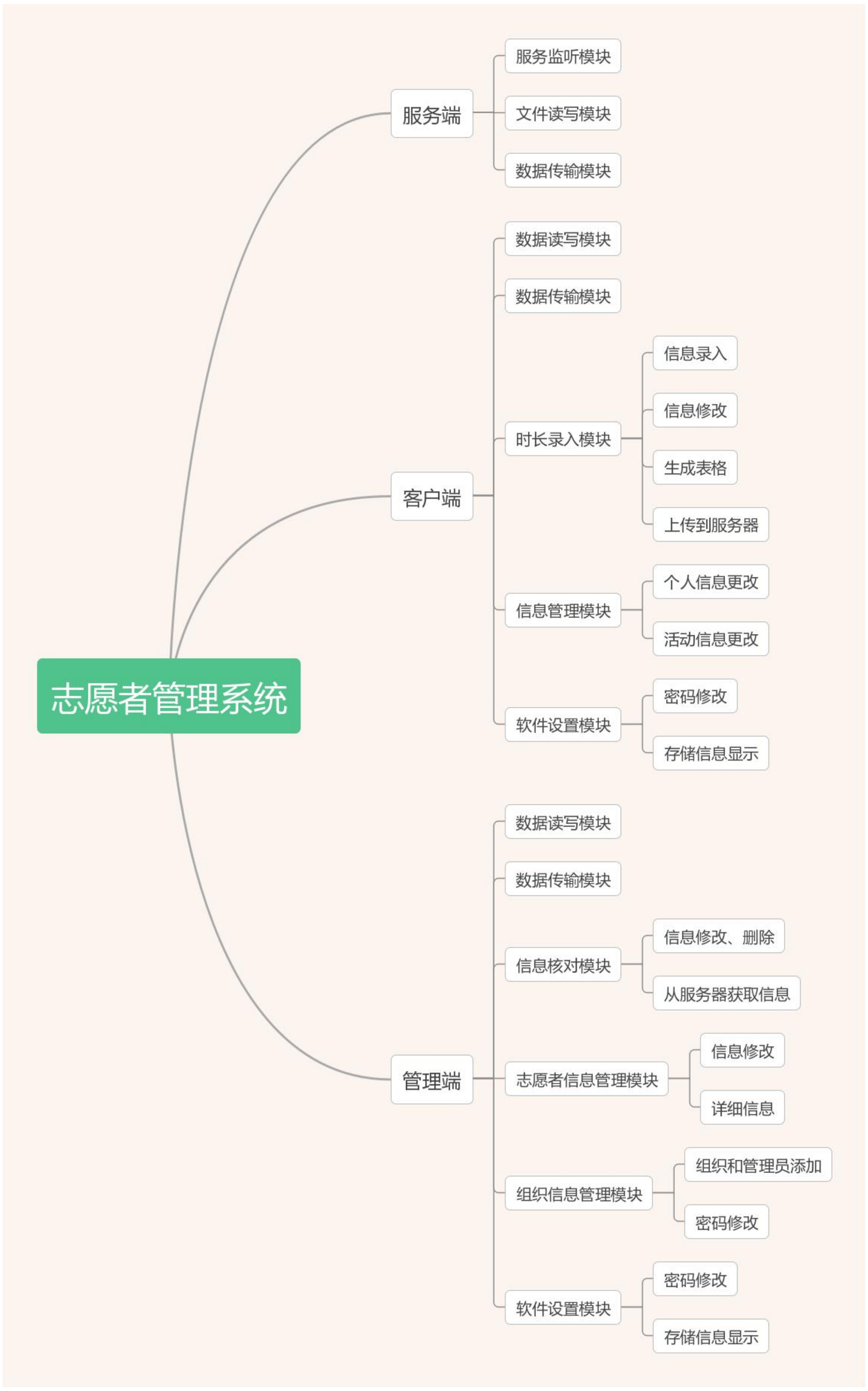
志愿时长统计过程比较复杂而且容易出错，将统计时长的工作下放给各大组织，减少了很多中间环节，降低了数据出错的可能，同时减轻了干事们平时统计志愿时长的工作强度。

2. 系统功能需求分析

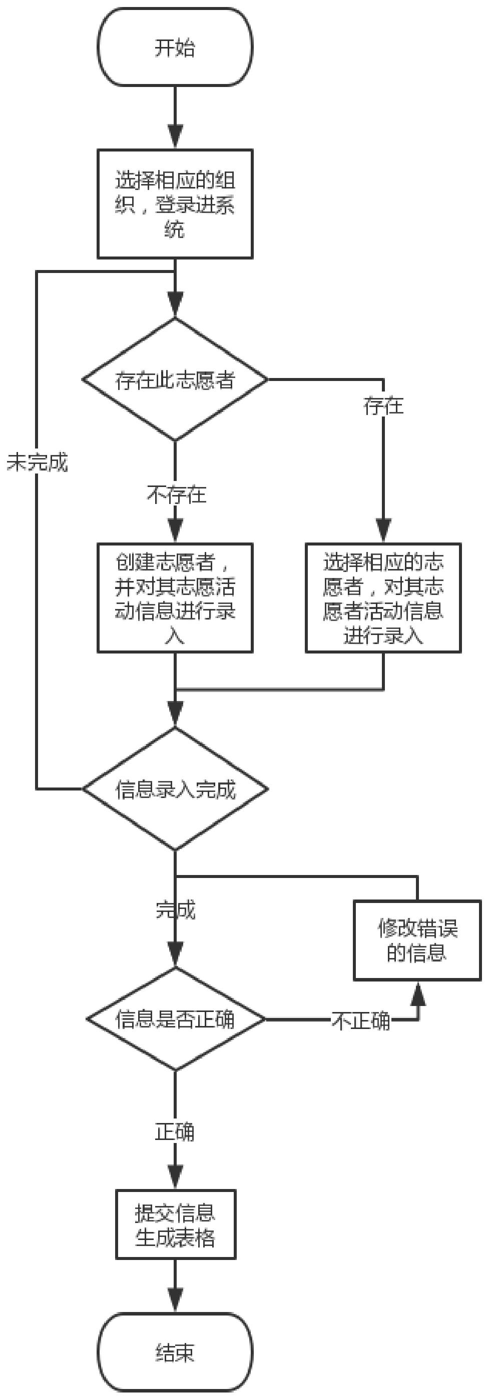
服务端主要负责连接客户端和管理端的数据请求，同时对数据文件进行存储、读写以及向客户端和管理端发送请求的数据。

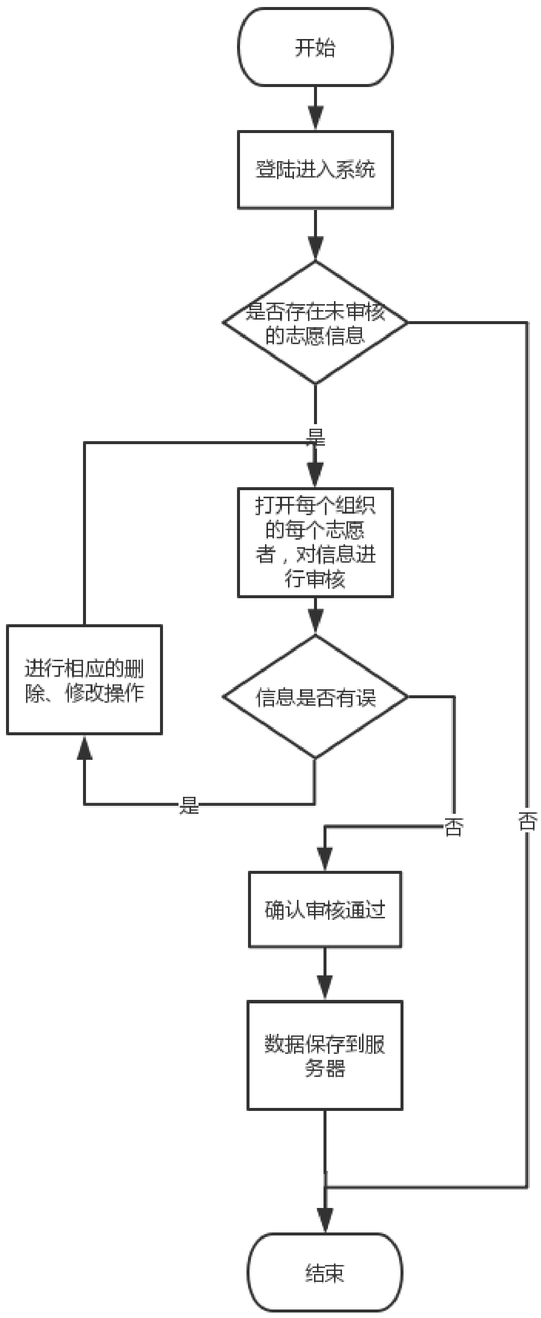
客户端主要用于信息录入。**时长录入模块**用于录入时长，录入时长后可以对已经录入的信息进行修改、删除，也可以生成相应的表格，同时可以将数据上传到服务器，表明提交志愿活动信息。**信息管理模块**里可以修改个人信息，也可以查看以往的活动信息。**软件设置模块**用于修改当前用户的密码，也可以查看配置文件和志愿者信息文件占用的系统空间大小。

管理端主要用于管理客户端发送的信息以及各大组织的信息。**信息核对模块**用于核对客户端发送来的信息，对于有误的信息可以有修改和删除的操作。核对完成的数据会被打上标签，系统会对这些信息进行计算。**志愿者信息管理模块**用于查看所有组织的志愿者信息，可以对以往的信息追溯检查，对于错误的信息可以修改、删除。**组织信息管理模块**用于对各组织的信息和管理员进行管理，可以添加组织以及各组织的管理员。**软件设置模块**用于修改管理端的管理员和密码，同时查看文件占用的空间。


志愿者管理系统
服务端
服务监听模块
文件读写模块
数据传输模块
客户端
数据读写模块
数据传输模块
时长录入模块
信息录入
信息修改
生成表格
上传到服务器
信息管理模块
个人信息更改
活动信息更改
软件设置模块
密码修改
存储信息显示
管理端
数据读写模块
数据传输模块
信息核对模块
信息修改、删除
从服务器获取信息
志愿者信息管理模块
信息修改
详细信息
组织信息管理模块
组织和管理员添加
密码修改
软件设置模块
密码修改
存储信息显示

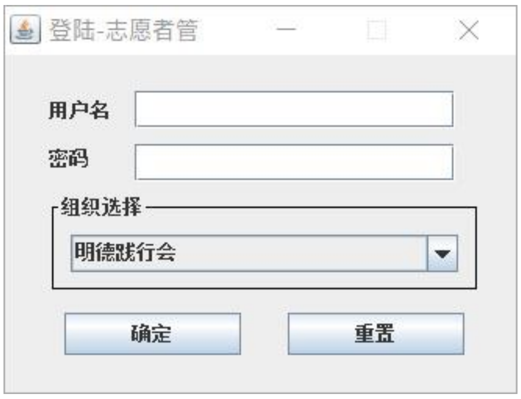
系统流程图


开始
选择相应的组织，登录进系统
存在此志愿者
存在
不存在
未完成
创建志愿者，并对其志愿活动信息进行录入
选择相应的志愿者，对其志愿者活动信息进行录入
信息录入完成
完成
修改错误的信息
信息是否正确
不正确
正确
提交信息生成表格
结束



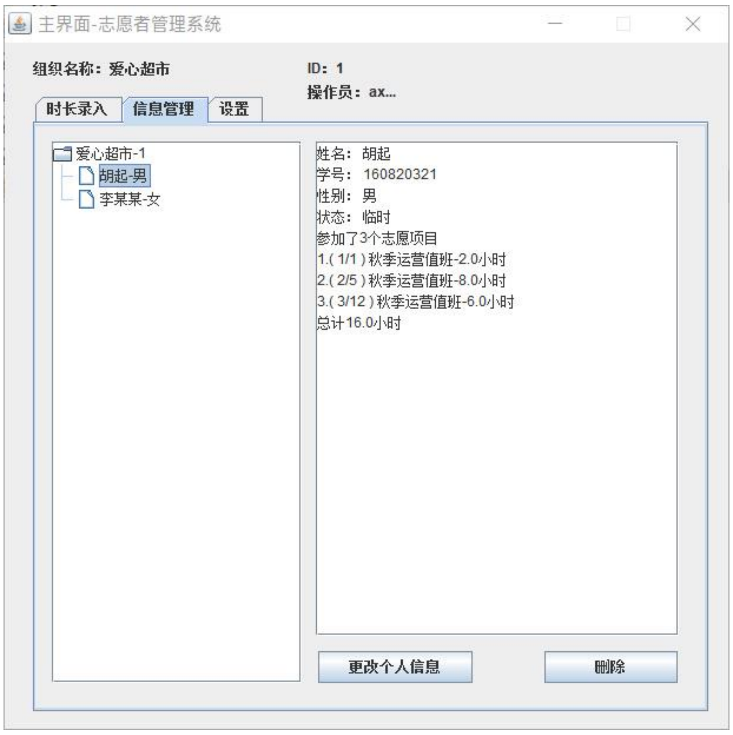
客户端录入志愿者信息流程　　　　管理端志愿者信息审核流程

系统核心功能展示



明德践行会

客户端登陆界面

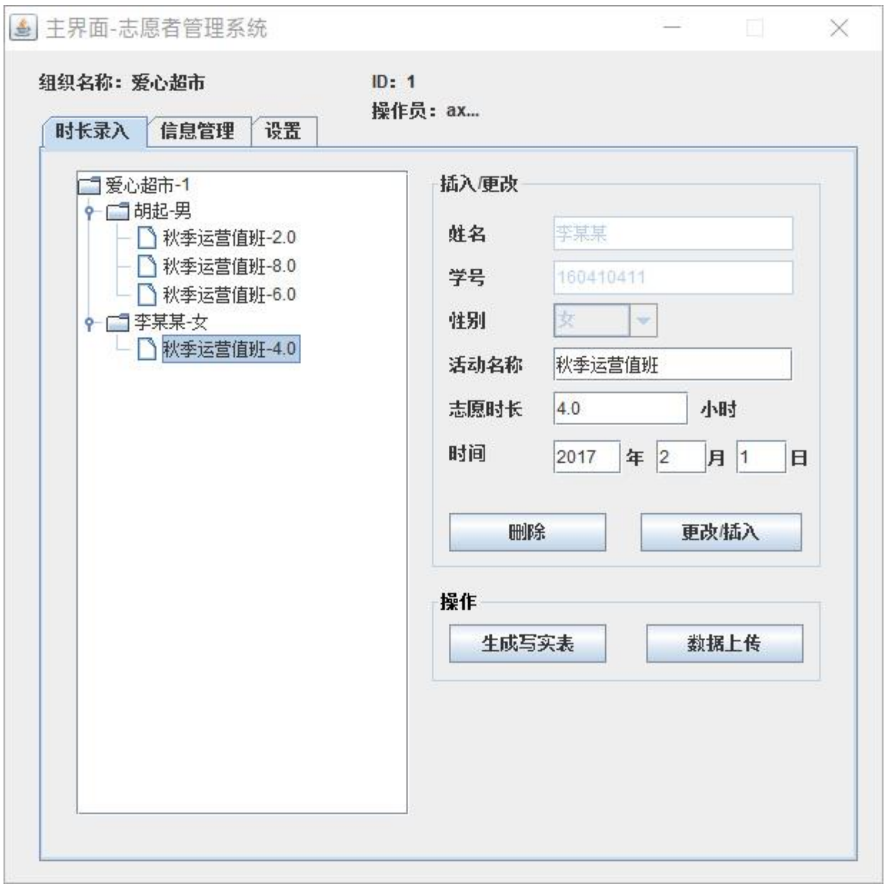


组织名称：爱心超市　ID：1
操作员：ax...

爱心超市-1
- 胡起-男
- 李某某-女

姓名：胡起
学号：160820321
性别：男
状态：临时
参加了3个志愿项目
1.(1/1) 秋季运营值班-2.0小时
2.(2/5) 秋季运营值班-8.0小时
3.(3/12) 秋季运营值班-6.0小时
总计16.0小时

客户端信息管理界面

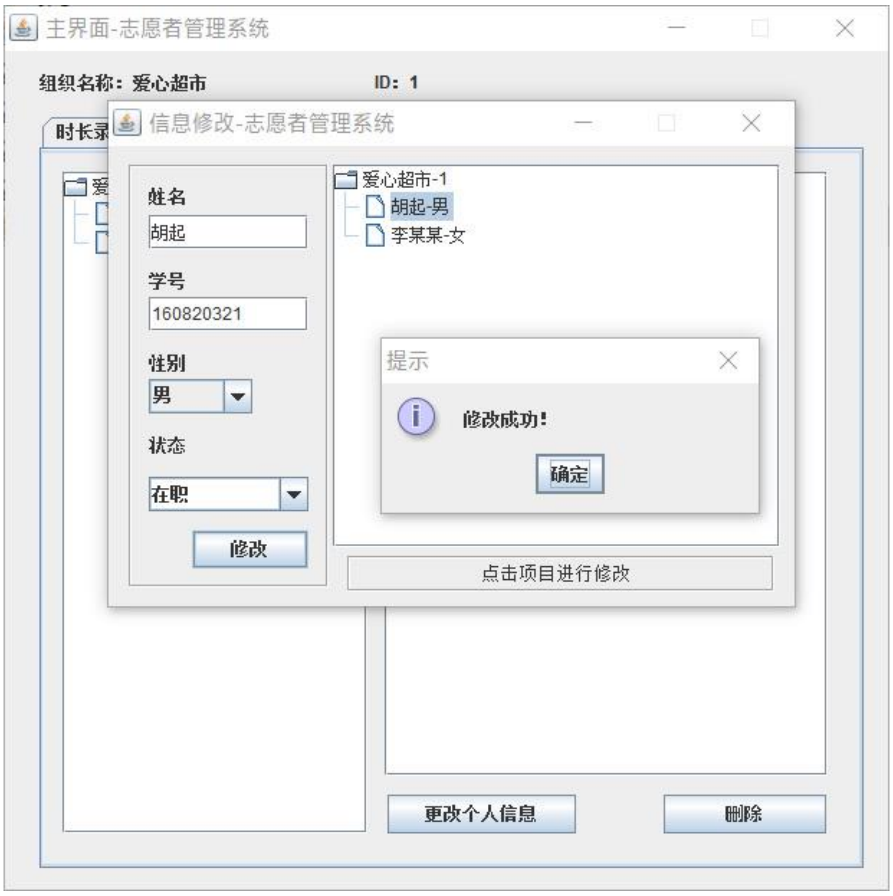


组织名称：爱心超市　ID：1
操作员：ax...

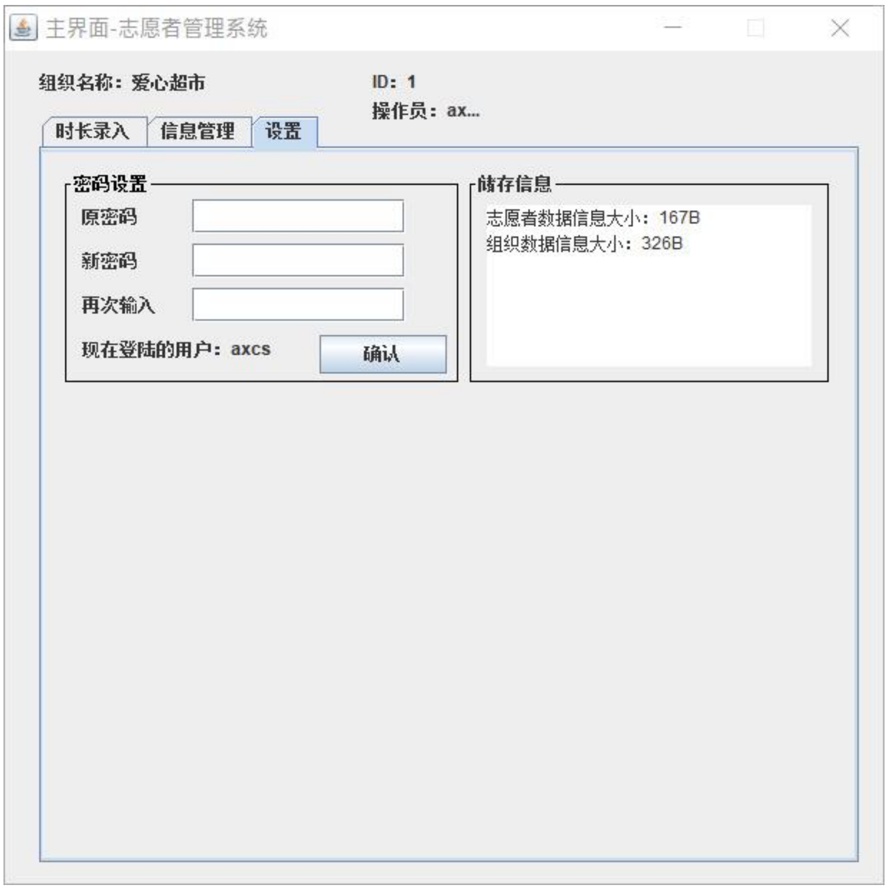
爱心超市-1
- 胡起-男
  - 秋季运营值班-2.0
  - 秋季运营值班-8.0
  - 秋季运营值班-6.0
- 李某某-女
  - 秋季运营值班-4.0

姓名：李某某
学号：160410411
性别：女
活动名称：秋季运营值班
志愿时长：4.0 小时
时间：2017 年 2 月 1 日

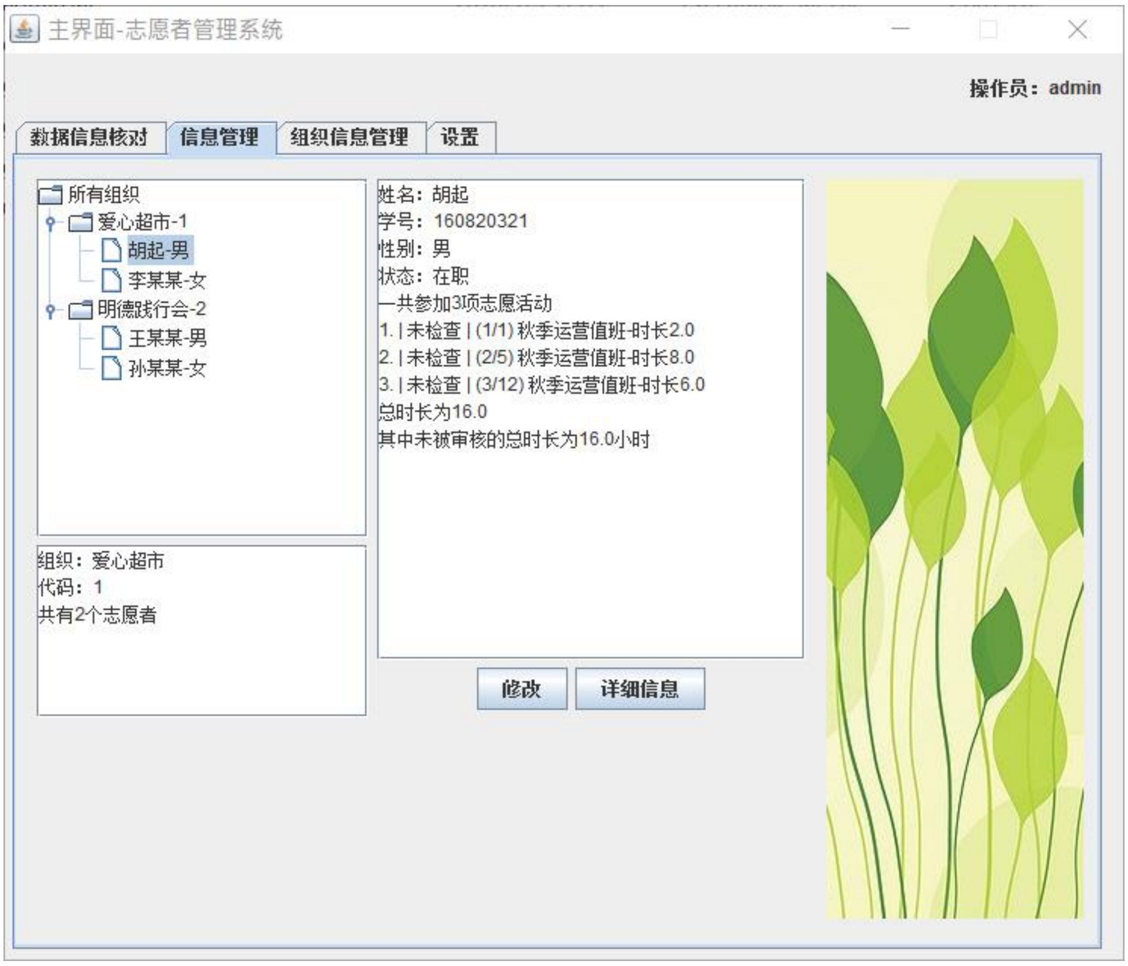
客户端时长录入界面



修改志愿者信息



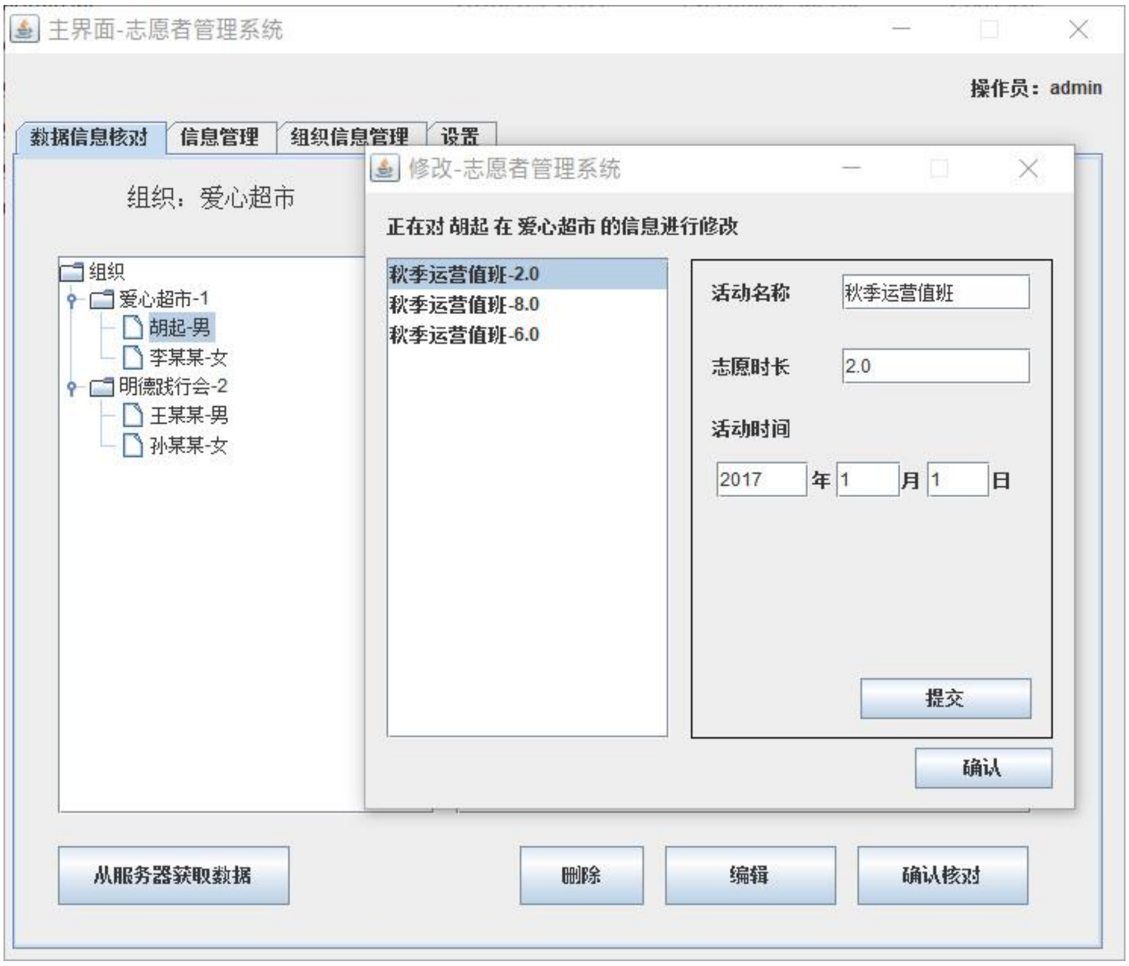
客户端设置界面



姓名：胡起
学号：160820321
性别：男
状态：在职
一共参加3项志愿活动
1.|未检查|(1/1)秋季运营值班-时长2.0
2.|未检查|(2/5)秋季运营值班-时长8.0
3.|未检查|(3/12)秋季运营值班-时长6.0
总时长为16.0
其中未被审核的总时长为16.0小时

组织：爱心超市
代码：1
共有2个志愿者

管理端信息管理界面



正在对 胡起 在 爱心超市 的信息进行修改

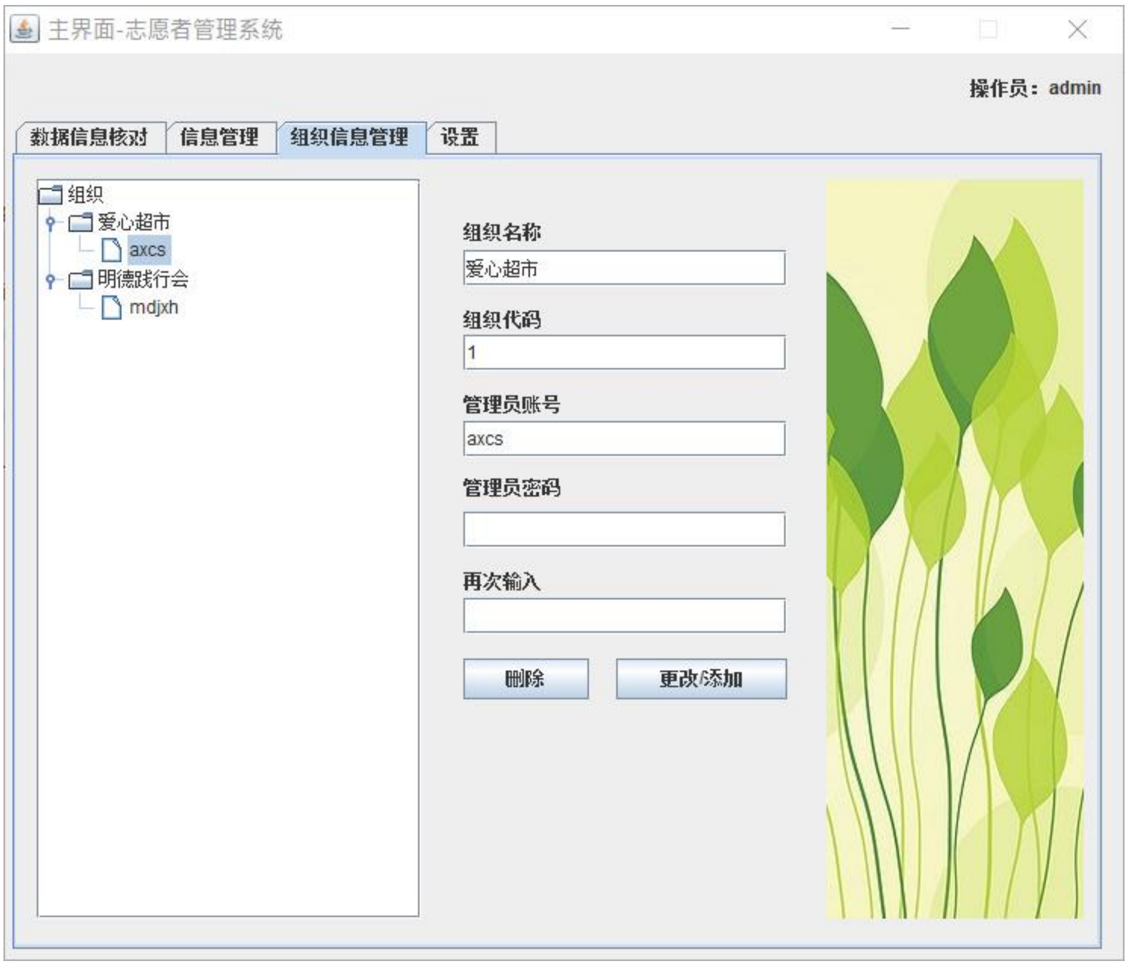
活动名称 秋季运营值班

志愿时长 2.0

活动时间

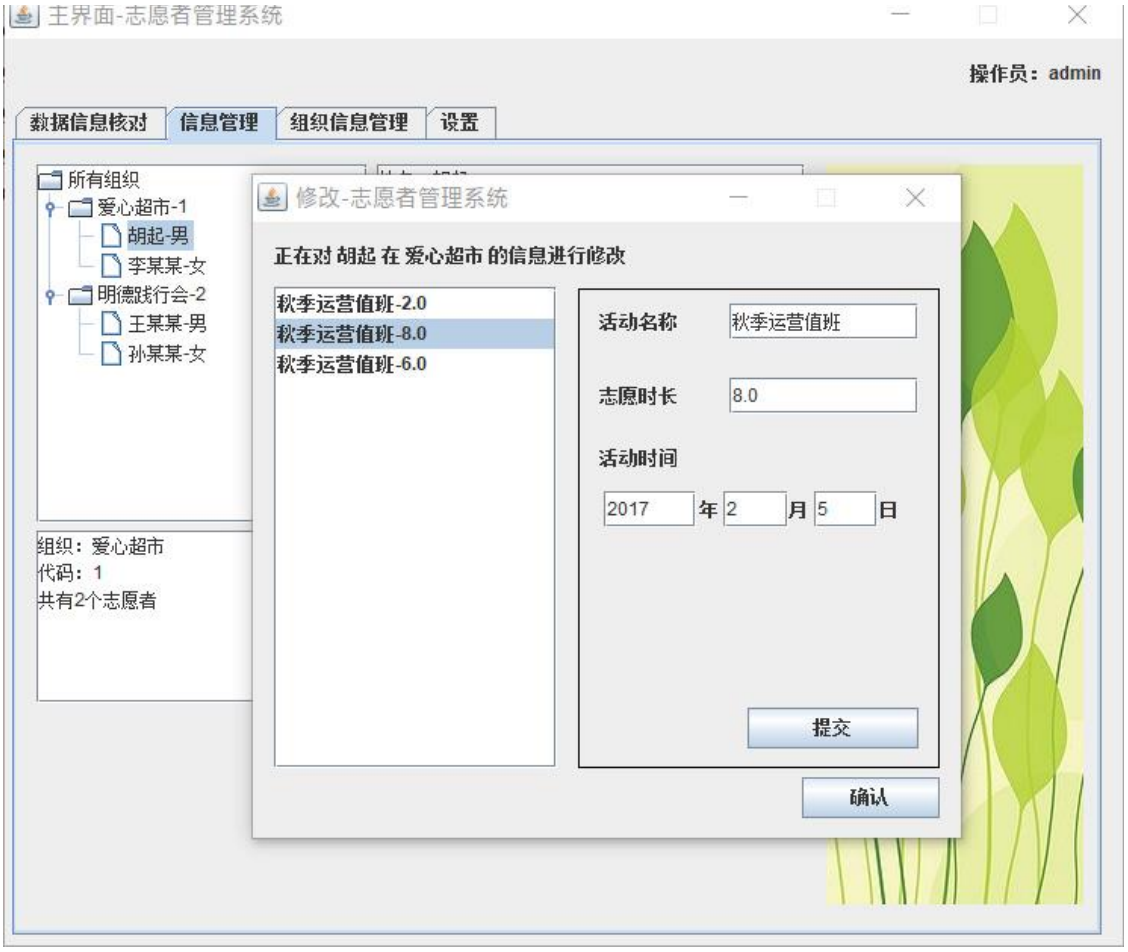
2017 年 1 月 1 日

信息修改界面



组织名称: 爱心超市
组织代码: 1
管理员账号: axcs
管理员密码:
再次输入:

管理端组织信息管理界面



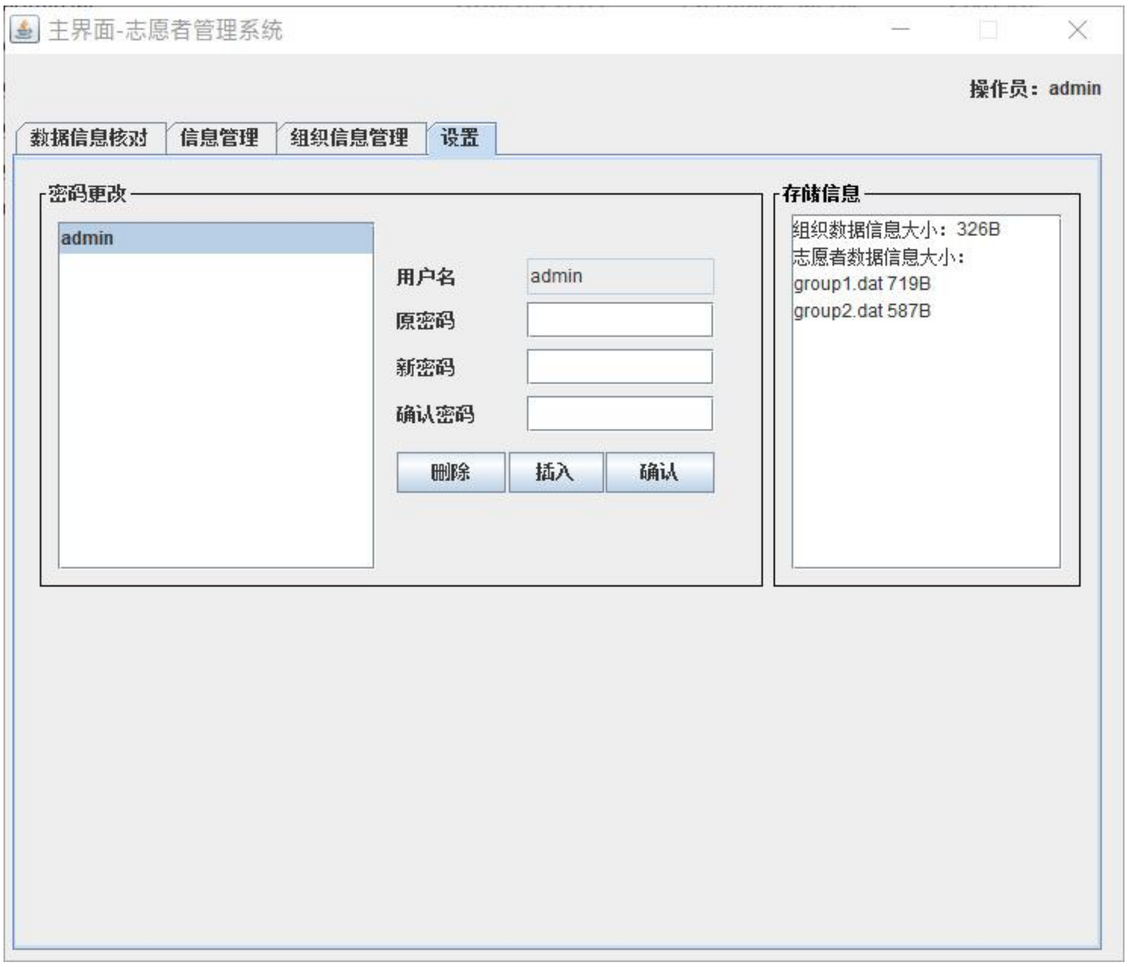
组织：爱心超市
代码：1
共有2个志愿者

正在对 胡起 在 爱心超市 的信息进行修改

秋季运营值班-2.0
秋季运营值班-8.0
秋季运营值班-6.0

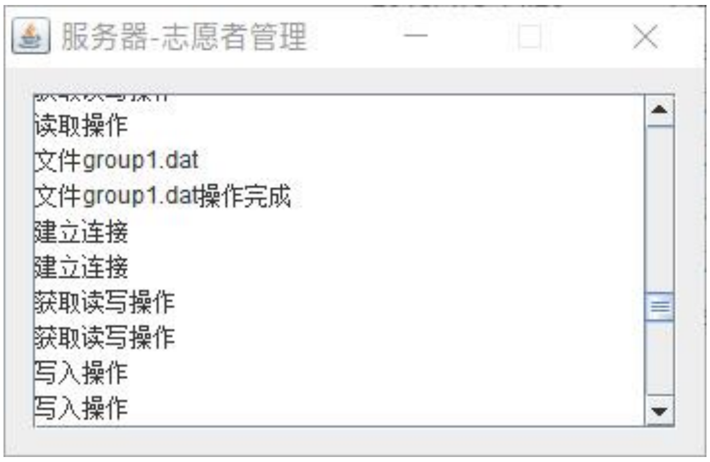
活动名称: 秋季运营值班
志愿时长: 8.0
活动时间: 2017 年 2 月 5 日

信息修改界面



组织数据信息大小：326B
志愿者数据信息大小：
group1.dat 719B
group2.dat 587B

管理端设置界面



```
读取操作
文件group1.dat
文件group1.dat操作完成
建立连接
建立连接
获取读写操作
获取读写操作
写入操作
写入操作
```

服务端界面

3. 系统实现及调试分析

编程开发环境：Java、Eclipse

由于要实现系统功能，就需要存储大量的信息，自然想到使用文件存储信息。但信息量过大以及内容复杂，不可能一条一条的向文件中存入，所以采用对象流，把同一类的信息封装在容器中，统一读写。但是过多的数据可能会导致程序启动速度慢，所以我把不同类别的信息分类存储在不同的文件中，程序启动时只会读取配置文件，其他的部分根据需要在进行读取。

在调试过程中文件读写经常出现问题。对出现的问题，我采用分块调试的方法，先判断内容是否能够读取出，然后判断读取的文件是否正确。往往文件读取出错是由于原文件损坏，但写入的内容是正常的。用记事本打开写入前的文件和读取后的文件进行比较，发现读取后的文件明显比读取前的文件短，于是认识到读取文件有一定问题。经过排查，分段调试，发现文件读写不能同时进行。打开输入流的同时不能打开输出流，否则文件会出错。找出这个问题后修改代码，解决了文件读写的问题。

另一个技术问题是实现网络化。开始准备实时和服务器交互，后来发现这样数据读写并不可靠，决定采用先储存在本地然后传输到服务端。保存就采用两步法，第一步将数据存储在本地文件内，第二部是将本地文件传输到服务端，读取文件也是先访问服务器取出文件，然后进行读写。但是从服务端获取文件会有文件不存在的情况，这种情况必须服务器告知客户端没有文件，然后由客户端向用户交互。这一部分我采用对象流实现。每次与服务器进行通讯的时候，客户端发送接受文件请求，此时服务端并不是直接发送数据，而是先返回文件是否存在等等文件信息，然后才开始进行传输。这样解决了文件不存在的问题。

总之，通过这次软件开发，让我知道数据正确的重要性，每次数据的读写都要考虑意外情况，文件不存在、文件读取出错等等方面程序都需要有一套自身的自动解决方法，以此来保证程序运行的稳定性。

问题及难点挑战

难点集中在文件读写、多线程开发和代码模块化调用上。

文件读写方面，所有数据全部存在文件中，而且程序在打开和关闭的时候要对文件进行大量的读写操作。读取文件数量多会导致程序启动速度慢，过多的写入会导致再次读取失败。

解决方法：将文件拆分，数据分别存放于不同的文件内。每个组织的志愿者数据文件放在单独的数据文件内，组织信息放在一个文件中，程序的配置文件放在一个文件中。每次程序启动只需要读取配置文件和组织信息文件即可，加快启动速度。

多线程开发方面最大的问题就是数据准备。当数据未准备完成就运行程序就会导致程序错误运行。

解决办法：采用线程锁方式，防止出现此问题

代码模块化调用主要问题集中在值传递和调用。编写过程中发生变量无法调用和权限分配等问题。

解决办法：常用的数据采用静态存储，密码等敏感数据采用私有的变量存储。

4. 总结（收获与体会）

通过软件开发实践让我对软件开发设计有了更深入的了解和认识。要想设计出一套比较优秀的程序，各方面都要努力。软件开发不仅仅重在开发，海中在开发前的分析与设计。首先需要我们了解市场，了解大家需要什么，再分析我们做的软件要达到什么的程度，要对各种情况以及需求作分析，还需要考虑以后软件的拓展，最后进行设计。将程序模块化，将界面和相应的操作分离，同时还要设计一些意外情况，比如没有所要的文件，文件数据量过大等等，最后才开始开发。

开发期初我并没有考虑这些，后来遇到了一些问题，发现如果要修改代码需要大面积地变化，甚至相当于重写代码。于是从这时起认识到进行模块化和分离化的重要性。如果软件开发前没有将这一部分弄清楚很有可能事倍功半甚至得重头再来。

在软件开发的时候一定要写好注释，包括函数调用的参数说明和函数作用。在程序编写过程中经常会发生调用函数发现参数值不知道填写什么的情况。但做好注释，IDE会将注释展示给自己，方便自己的使用，减少返工时间。

在调试上很多时候要分块调试，遇到调试问题也要分块处理。不仅要看错误和运行错误指示，还要考虑值传递是是否有类型不一致和数据是否被定义。对于文件流的错误要通过多次调试才能排查出。文件流错误大都出现在打开关闭文件流顺序有一定要求，打开输入流的同时不能打开输出流，不然会导致文件无法读取和写入。

通过实践也提升了自己对问题的解决的能力。对于程序遇到的很多问题，比如函数使用，错误解决等等，都可以查看文档以及上网查找答案。同时查阅一些书籍，模仿着写一写代码，可以更快地实现相应的功能。

另外，自己独立完成一个项目也是对自己的一个锻炼。从开题到结题到完成报告书，通过这个过程，让我知道做好一个程序，从规划到编写到调试，每一环都要尽心尽力，才能做得更好，程序才能被用户接受。

5. 附录：主要源代码

Server：

```
package server;

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;

import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;

public class Main extends JFrame {
    public JTextArea textArea;

    public Main() {
        setTitle("\u670D\u52A1\u5668-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 362, 232);
        setResizable(false);
        getContentPane().setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(14, 13, 328, 171);
        getContentPane().add(scrollPane);

        textArea = new JTextArea();
        scrollPane.setViewportView(textArea);
        setVisible(true);
        textArea.append("开始运行\n");
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int port = 9999;
        // 定义一个ServerSocket监听在端口9999上
        Main m = new Main();
        try {
            ServerSocket server = new ServerSocket(port);
            while (true) {
                // server尝试接收其他Socket的连接请求，server的accept方法是阻塞式的
                Socket socket = server.accept();
                // 每接收到一个Socket就建立一个新的线程来处理它
                m.textArea.append("建立连接\n");
```

```
                    new Thread(new Task(socket, m.textArea)).start();
                }
            } catch (Exception e) {
                m.textArea.append("发生错误\n");
                m.textArea.append(e.toString());
                e.printStackTrace();
            }
        }
}

/**
 * using to exchange the information
 *
 * @author huqi1
 *
 */
class Task implements Runnable {
    private Socket socket;
    private JTextArea textArea;

    public Task(Socket socket, JTextArea textArea) {
        this.socket = socket;
        this.textArea = textArea;
    }

    @Override
    public void run() {
        // TODO Auto-generated method stub
        DataPack data = new DataPack();
        textArea.append("获取读写操作\n");
        try {// first read info about file
            ObjectInputStream obin = new ObjectInputStream(socket.getInputStream());
            data = (DataPack) obin.readObject();
            System.out.println(data.fileName);
            System.out.println(data.type);
            if (data.type == 1) {
                textArea.append("读取操作\n");
                readFile(data.fileName);
            } else {
                textArea.append("写入操作\n");
                writeFile(data.fileName);
            }
            obin.close();
            socket.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (Exception e) {
            e.printStackTrace();
        }

    }

    public void writeFile(String name) throws Exception {
        // client to server
        // 创建网络接受流接受服务器文件数据
        InputStream netIn = socket.getInputStream();
        ObjectInputStream in = new ObjectInputStream(new BufferedInputStream(netIn));
        textArea.append("文件" + name + "\n");
        File file = new File(name);
        if (!file.exists())
            file.createNewFile();// 文件不存在 创建
        ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream(file));
        out.writeObject(in.readObject());
        out.flush();

        //close all stream
        netIn.close();
        in.close();
        out.close();
        textArea.append("文件" + name + "操作完成\n");
    }

    public void readFile(String name) throws Exception {
        // server to client
        // maybe cannot find file
        OutputStream netOut = socket.getOutputStream();
        ObjectOutputStream out = new ObjectOutputStream(new
BufferedOutputStream(netOut));
        textArea.append("文件" + name + "\n");
        File file = new File(name);
        if (!file.exists()) {// file not exit
            out.writeBoolean(false);// return false that make client use local file
            out.flush();
            textArea.append("文件不存在，返回false\n");
        } else {
            out.writeBoolean(true);
            out.flush();
            FileInputStream fos = new FileInputStream(file);
            ObjectInputStream in = new ObjectInputStream(new BufferedInputStream(fos));
            out.writeObject(in.readObject());
            out.flush();

            in.close();
            fos.close();
        }
        // close all stream
        netOut.close();
        out.close();
        textArea.append("文件" + name + "操作完成\n");
    }
    }
```

Admin：

data 包：

Data.java

```
package admin.data;

import java.io.*;
import java.util.ArrayList;
import java.util.Collection;

import javax.swing.JOptionPane;

import data.*;

public class Data {
    //set login user
    public static Collection<User> LoginUser;
    //set other group user
    public static Collection<Groups> groups;
    //set group collection
    //this is save the group which is read
    public static Collection<Group> readGroup;
    //group recent read
    public static Group recentGroup;
    //group witch has read
    public static ArrayList<Integer> groupReadID;
    //user info
    public static String user;
    //ready?
    public static boolean Userready;//user
    public static boolean Groupsready;//all group info
    public static boolean recentReadReady;//recent reading is ready

    //save?
    public static boolean GroupSaveOK;
    public static boolean GroupsSaveOK;
    public static boolean UsersSaveOK;

    public Data() {
        LoginUser=new ArrayList<User>();
        groups=new ArrayList<Groups>();
        readGroup=new ArrayList<Group>();
        groupReadID=new ArrayList<Integer>();
        user=null;
        Userready=false;
        recentGroup=null;
        GroupSaveOK=false;
        GroupsSaveOK=false;
        recentReadReady=false;
        UsersSaveOK=false;
    }
    /**
     * 将文件配置读入，得到User
     *
     * @param file
     *            文件路径
     */
    public static void getUser(String file) {
        try {
            ObjectInputStream input = new ObjectInputStream(new FileInputStream(file));
            LoginUser = (ArrayList<User>) input.readObject();

            input.close();
            Userready = true;

        } catch(FileNotFoundException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "没有用户配置信息，已重新生成！\n用户名：
admin\n密码：admin\n登陆后请及时修改密码！","提示",JOptionPane.WARNING_MESSAGE);
            User user=new User();
            user.setUser("admin");
            user.setPsw("admin");
            Data.LoginUser.add(user);
        }catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
            System.err.print("Read setting file ERROR!");
            JOptionPane.showMessageDialog(null, "Read Setting File
ERROR","ERROR",JOptionPane.ERROR_MESSAGE);
            System.exit(0);//遇到问题直接退出

        }
    }

    /**
     * 将文件配置读入，得到Groups
     *
     * @param file
     *            文件路径
     */
    public static void getGroups(String file) {
        try {
            ObjectInputStream input = new ObjectInputStream(new FileInputStream(file));
            groups = (ArrayList<Groups>) input.readObject();

            input.close();
            Groupsready = true;
        } catch (FileNotFoundException e1) {
            // TODO: handle exception
            e1.printStackTrace();
            // file not find then we will create it
            // also help to first load
            writeFile(file,groups);
            Groupsready=true;
            JOptionPane.showMessageDialog(null, "似乎没有找到组织信息文件
","ERROR",JOptionPane.ERROR_MESSAGE);

        } catch (Exception e2) {
            e2.printStackTrace();
            System.err.print("Read setting file ERROR!");
            JOptionPane.showMessageDialog(null, "Read Setting File
ERROR","ERROR",JOptionPane.ERROR_MESSAGE);
        }
    }
    /**
     * 将文件配置读入，得到Group
     *
     * @param file
     *            文件路径
     */
    public static void getGroup(String file) {
        try {
            ObjectInputStream input = new ObjectInputStream(new FileInputStream(file));
            recentGroup = (Group)input.readObject();//save recent read and save it to
the collection
            readGroup.add(recentGroup);
            groupReadID.add(recentGroup.ID);//add to read id

            input.close();

        } catch (FileNotFoundException e1) {
            // TODO: handle exception
            e1.printStackTrace();
            JOptionPane.showMessageDialog(null, "似乎没有找到组织信息文件
","ERROR",JOptionPane.ERROR_MESSAGE);

        } catch (Exception e2) {
            e2.printStackTrace();
            System.err.print("Read file ERROR!");
            JOptionPane.showMessageDialog(null, "Read File
ERROR","ERROR",JOptionPane.ERROR_MESSAGE);
        }
    }

    /**
     * 将信息写入文件，所有的信息将会一次性写入
     *
     * @param file
     *            文件目录
     */
    public static void writeFile(String file,Object obj) {
        try {
            ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(file));
            out.writeObject(obj);//存入信息

            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

process 包

GetGroupData.java

```
package admin.process;

import java.util.Iterator;
import javax.swing.JFrame;

import admin.data.Data;
import admin.programUI.ReadUI;
import data.*;

/**
 * this class is using to get group info, if it at group list return it; else we
 * will read it ,save it to the collection then return it
 *
 * @author huqi1
 *
 */
public class GetGroupData {
    public GetGroupData(int ID) {
        JFrame jf=new ReadUI();
        // ID is group ID

        // first find it in the list
        if (Data.groupReadID.contains(ID)) {
            // have it & find it
            Group group = null;
            for (Iterator<Group> it = Data.readGroup.iterator(); it.hasNext();) {
                group = it.next();
                if (group.ID == ID) {// got it
                    Data.recentGroup = group;
                }
            }

        } else {
            // no

            ReadData.readGroup(Integer.valueOf(ID));
            // store recent group
            while(!Data.recentReadReady) {
                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        }
        jf.dispose();
    }


}
```

ReadData.java

```
package admin.process;

import admin.data.Data;
import data.NetFileWork;

public class ReadData {
    /**
     * using to read user info
     */
    public static void readUser() {
        new ReadUser();
    }

    /**
     * using to read group info
     *
     * @param ID
     *            group ID
     */
    public static void readGroup(int ID) {
        new ReadGroup(ID);

    }

    /**
     * using to read all group info
     *
     */
    public static void readGroups() {
        ReadGroups th=new ReadGroups();
        while(th.isAlive()) {
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                // TODO 自动生成的 catch 块
                e.printStackTrace();
            }
        }
    }
```

```
}

class ReadUser extends Thread {
    public ReadUser() {
        this.start();
    }

    public void run() {
        try {
            new NetFileWork("config.dat", 1);
        } catch (Exception e) {
            e.printStackTrace();
        }
        Data.getUser("config.dat");
    }

}

class ReadGroup extends Thread {
    int ID;

    /**
     * Read Group info
     *
     * @param ID
     *            using to identify which group
     */
    public ReadGroup(int ID) {
        this.ID = ID;
        this.start();
    }

    public void run() {
//      try {
//          new NetFileWork("group" + ID + ".dat", 1);
//      } catch (Exception e) {
//          e.printStackTrace();
//      }
        Data.getGroup("group" + ID + ".dat");
        Data.recentReadReady = true;
    }

}

class ReadGroups extends Thread {
    public ReadGroups() {
        this.start();
    }

    public void run() {
        try {
            new NetFileWork("groups.dat", 1);
        } catch (Exception e) {
            e.printStackTrace();
        }
        Data.getGroups("groups.dat");
    }

}
```

WriteData. java

```
package admin.process;

import java.util.ArrayList;
import java.util.Iterator;

import admin.data.Data;
import data.*;

public class WriteData {

    public WriteData(int tag, Object obj) {
        // tag 1:ArryList<Group> 2:ArryList<Groups> 3:
        switch (tag) {
        case 1:
            ArrayList<Group> group = (ArrayList<Group>) obj;
            new WriteGroup(group);
            break;
        case 2:
            ArrayList<Groups> groups = (ArrayList<Groups>) obj;
            new WriteGroups(groups);
            break;
        case 3:
            ArrayList<User> users = (ArrayList<User>) obj;
            new WriteUser(users);
            break;
        }

    }
}
```

```
class WriteGroup extends Thread {
    ArrayList<Group> groups;

    public WriteGroup(ArrayList<Group> group) {
        this.groups = group;
        this.start();
    }

    public void run() {
        // 将打开过的信息保存到各个文件
        Group group = null;
        for (Iterator<Group> it = groups.iterator(); it.hasNext();) {
            group = it.next();
            Data.writeFile("group" + group.ID + ".dat", group);
            try {
                new NetFileWork("group" + group.ID + ".dat", 2);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        Data.GroupSaveOK = true;

    }
}

class WriteGroups extends Thread {
    ArrayList<Groups> groups;

    public WriteGroups(ArrayList<Groups> groups) {
        this.groups = groups;
        this.start();
    }

    public void run() {
        Data.writeFile("groups.dat", groups);
        try {
            new NetFileWork("groups.dat", 2);
        } catch (Exception e) {
            e.printStackTrace();
        }
        Data.GroupsSaveOK = true;
    }
}

class WriteUser extends Thread {
    ArrayList<User> users;
    public WriteUser(ArrayList<User> users) {
        this.users = users;
        this.start();
    }

    public void run() {
        // maybe config.data can save more infomation
        try {
            new NetFileWork("config.dat", 2);
        } catch (Exception e) {
            e.printStackTrace();
        }
        Data.writeFile("config.dat", users);
        Data.UsersSaveOK=true;
    }
}
```

programUI 包

MainUIPanel.java

```
package admin.programUI;

import java.awt.BorderLayout;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;

import admin.data.Data;
import javax.swing.JTabbedPane;

public class MainUIPanel extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;

    /**
     * Create the frame.
     */
    public MainUIPanel() {
        setTitle("\u4E3B\u754C\u9762-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100,100, 745, 634);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JPanel panel = new JPanel();
        panel.setBounds(5, 5, 720, 36);
        contentPane.add(panel);
        panel.setLayout(new BorderLayout(0, 0));

        JLabel lab_userName = new JLabel("{userName}");
        panel.add(lab_userName, BorderLayout.EAST);
        setResizable(false);//不可放大
        lab_userName.setText("操作员: "+Data.user);

        JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
        tabbedPane.setBounds(5, 43, 720, 549);
        contentPane.add(tabbedPane);

        JPanel InsertUI = new InsertUI();
        tabbedPane.addTab("数据信息核对" , InsertUI);

        JPanel ManageUI = new ManageUI();
        tabbedPane.addTab("信息管理", ManageUI);

        JPanel GroupsManageUI = new GroupsManageUI();
        tabbedPane.addTab("组织信息管理", GroupsManageUI);

        JPanel Setting = new SettingUI();
        tabbedPane.addTab("设置", Setting);


        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                new SaveUI();//保存作业
                dispose();
            }

        });
    }
}
```

Login. java

```java
package admin.programUI;

import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Iterator;

import javax.swing.*;

import admin.data.Data;
import data.User;
import admin.programUI.MainUIPanel;
import admin.process.ReadData;

public class LoginUI extends JFrame implements ActionListener {
    /**
     * LoginUI 登陆界面
     */
    private static final long serialVersionUID = 1L;
    private JPanel jp1, jp2, jp3, jp4, jp5;
    private JButton btn1, btn2;
    private JLabel lb1, lb2;
    private JTextField user;
    private JPasswordField password;

    public LoginUI() {
        super("登录系统");
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        jp1 = new JPanel();// 用于存放标题
        jp1.add(new JLabel("欢迎登陆管理系统"));
        jp2 = new JPanel();// 用于存放登录提示
        jp2.setLayout(new GridLayout(2, 1, 15, 15));
        lb1 = new JLabel("用户名");
        lb2 = new JLabel("密    码");
        jp2.add(lb1);
        jp2.add(lb2);
        jp3 = new JPanel();// 用于存放登陆输入
        jp3.setLayout(new GridLayout(2, 1, 15, 15));
        user = new JTextField(20);
        password = new JPasswordField(20);
        jp3.add(user);
        jp3.add(password);
        jp4 = new JPanel(new BorderLayout());// 用于存放组合
        jp4.add(jp2, BorderLayout.WEST);
        jp4.add(jp3, BorderLayout.CENTER);
        jp5 = new JPanel();// 用于存放按钮
        btn1 = new JButton("登陆");
        btn2 = new JButton("重置");
        jp5.add(btn1);
        jp5.add(btn2);

        getContentPane().setLayout(new BorderLayout(10, 10));// 设置样式

        getContentPane().add(jp1, BorderLayout.NORTH);
        getContentPane().add(jp4, BorderLayout.CENTER);
        getContentPane().add(jp5, BorderLayout.SOUTH);

        // 注册按钮
        btn1.addActionListener(this);
        btn2.addActionListener(this);
        pack();
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        Object source = e.getSource();
        if (source == btn1) {
            // 提交按钮
            String users, passwords;
            users=user.getText().toString();
            passwords=new String (password.getPassword());
            System.out.println(users+" "+passwords);
            User loginUser=null;
            for(Iterator<User> it=Data.LoginUser.iterator();it.hasNext();) {
                //user is exist? password is correct?
                loginUser=it.next();
                System.out.println(loginUser.getUser());
                System.out.println(loginUser.getPassword());

    if(loginUser.getUser().equals(users)&&loginUser.getPassword().equals(passwords))
{
                    //correct!
                    Data.user=loginUser.getUser();
                    ReadData.readGroups();
                    new MainUIPanel();
                    dispose();
                }
            }
            //is not correct
            user.setText("");
            password.setText("");

        } else if (source == btn2) {
            // 重置按钮
            user.setText("");
            password.setText("");
        }
    }

}
```

InsertUI. java

```
package admin.programUI;

import java.util.Iterator;

import javax.swing.*;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;

import admin.data.Data;
import data.Groups;
import data.User;

import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.Graphics;

public class GroupsManageUI extends JPanel {
    private JTextField txt_name;
    private JTextField txt_ID;
    private JTextField txt_count;
    private JPasswordField psw_psw;
    private JPasswordField psw_confirm;
    private JButton btn_delete;
    private JButton btn_modify;

    public GroupsManageUI() {
        setBounds(100, 100,720, 515);
        setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(14, 13, 254, 489);
        add(scrollPane);

        DefaultMutableTreeNode top = new DefaultMutableTreeNode("组织");

        JTree tree_group = new JTree(top);
        scrollPane.setViewportView(tree_group);
        new BuildTree(tree_group, top);
        tree_group.addTreeSelectionListener(new TreeSelectionListener() {

            @Override
            public void valueChanged(TreeSelectionEvent arg0) {
                // TODO Auto-generated method stub
                // 获取树的节点
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_group.getLastSelectedPathComponent();
                // 判断是否为空树
                if (node == null) {
                    return;
                }
                DefaultMutableTreeNode parant = (DefaultMutableTreeNode)
node.getParent();
                Object nodeInfo = node.getUserObject();
                if (nodeInfo instanceof User) {
                    btn_delete.setEnabled(true);
                    User user = (User) nodeInfo;

                    txt_name.setText(((Groups) parant.getUserObject()).name);
                    txt_ID.setText(String.valueOf(((Groups)
parant.getUserObject()).ID));
                    txt_count.setText(user.getUser());
                    psw_psw.setText("");
                    psw_confirm.setText("");

                } else if (nodeInfo instanceof Groups) {
                    btn_delete.setEnabled(true);
                    Groups groups = (Groups) nodeInfo;
                    txt_name.setText(groups.name);
                    txt_ID.setText(String.valueOf(groups.ID));
                    txt_count.setText("");
                    psw_psw.setText("");
                    psw_confirm.setText("");

                } else {
                    btn_delete.setEnabled(false);
                    txt_name.setText("");
                    txt_ID.setText("");
                    txt_name.setText("");
                    psw_psw.setText("");
                    psw_confirm.setText("");

                }
            }
        });

        JPanel panel = new JPanel();
        panel.setBounds(282, 26, 254, 476);
        add(panel);
        panel.setLayout(null);

        JLabel label = new JLabel("\u7EC4\u7EC7\u540D\u79F0");
        label.setBounds(14, 13, 72, 18);
        panel.add(label);

        JLabel label_1 = new JLabel("\u7EC4\u7EC7\u4EE3\u7801");
        label_1.setBounds(14, 71, 72, 18);
        panel.add(label_1);

        JLabel label_2 = new JLabel("\u7BA1\u7406\u5458\u8D26\u53F7");
        label_2.setBounds(14, 127, 83, 18);
        panel.add(label_2);

        txt_name = new JTextField();
        txt_name.setBounds(14, 34, 214, 24);
        panel.add(txt_name);
        txt_name.setColumns(10);

        txt_ID = new JTextField();
        txt_ID.setBounds(14, 90, 214, 24);
        panel.add(txt_ID);
        txt_ID.setColumns(10);

        txt_count = new JTextField();
        txt_count.setBounds(14, 147, 214, 24);
        panel.add(txt_count);
        txt_count.setColumns(10);

        JLabel label_3 = new JLabel("\u7BA1\u7406\u5458\u5BC6\u7801");
        label_3.setBounds(14, 182, 83, 18);
        panel.add(label_3);

        psw_psw = new JPasswordField();
```

```
        psw_psw.setBounds(14, 207, 214, 24);
        panel.add(psw_psw);

        JLabel label_4 = new JLabel("\u518D\u6B21\u8F93\u5165");
        label_4.setBounds(14, 244, 72, 18);
        panel.add(label_4);

        psw_confirm = new JPasswordField();
        psw_confirm.setBounds(14, 264, 214, 24);
        panel.add(psw_confirm);

        btn_delete = new JButton("\u5220\u9664");
        btn_delete.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // 获取树的节点
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_group.getLastSelectedPathComponent();
                // 判断是否为空树
                if (node == null) {
                    return;
                }
                DefaultMutableTreeNode parant = (DefaultMutableTreeNode)
node.getParent();
                Object nodeInfo = node.getUserObject();
                if (nodeInfo instanceof User) {
                    ((Groups) parant.getUserObject()).users.remove(nodeInfo);
                    parant.remove(node);
                    tree_group.updateUI();
                    JOptionPane.showMessageDialog(null, "删除成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);

                } else if (nodeInfo instanceof Groups) {
                    Data.groups.remove(nodeInfo);
                    parant.remove(node);
                    tree_group.updateUI();
                    JOptionPane.showMessageDialog(null, "删除成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);

                } else {
                    JOptionPane.showMessageDialog(null, "不允许删除！", "提示",
JOptionPane.WARNING_MESSAGE);

                }
            }
        });
        btn_delete.setBounds(14, 304, 83, 27);
        panel.add(btn_delete);

        btn_modify = new JButton("\u66F4\u6539/\u6DFB\u52A0");
        btn_modify.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                // 获取树的节点
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_group.getLastSelectedPathComponent();
                // 判断是否为空树
                if (node == null) {
                    return;
                }
                DefaultMutableTreeNode parant = (DefaultMutableTreeNode)
node.getParent();
                Object nodeInfo = node.getUserObject();
                if (nodeInfo instanceof User) {
                    if (infocheck(true)) {
                        User user = (User) nodeInfo;
                        user.setUser(txt_name.getText());
                        user.setPsw(new String(psw_psw.getPassword()));
                        tree_group.updateUI();
                        JOptionPane.showMessageDialog(null, "修改成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);
                    }
                } else if (nodeInfo instanceof Groups) {
                    if (txt_count.getText().trim().equals("")) {// 修改
                        if (infocheck(false)) {
                            ((Groups) nodeInfo).name = txt_name.getText();
                            ((Groups) nodeInfo).ID =
Integer.valueOf(txt_ID.getText());
                            tree_group.updateUI();
                            JOptionPane.showMessageDialog(null, "修改成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);

                        }
                    } else {// 添加
                        if (infocheck(true)) {

                            User user = new User();
                            user.setUser(txt_count.getText());
                            user.setPsw(new String(psw_psw.getPassword()));
                            ((Groups) nodeInfo).users.add(user);

                            DefaultMutableTreeNode nodes = new
DefaultMutableTreeNode(user);
```

```
                            node.add(nodes);
                            tree_group.updateUI();
                            JOptionPane.showMessageDialog(null, "添加成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);
                        }
                    }

                } else {// 添加组织
                    if (infocheck(true)) {
                        Groups groups = new Groups();
                        groups.name = txt_name.getText();
                        groups.ID = Integer.valueOf(txt_ID.getText());
                        DefaultMutableTreeNode node1 = new
DefaultMutableTreeNode(groups);
                        User user = new User();
                        user.setUser(txt_count.getText());
                        user.setPsw(new String(psw_psw.getPassword()));
                        groups.users.add(user);
                        DefaultMutableTreeNode node2 = new
DefaultMutableTreeNode(user);
                        node1.add(node2);
                        node.add(node1);
                        Data.groups.add(groups);
                        tree_group.updateUI();

                        JOptionPane.showMessageDialog(null, "添加成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);
                    }
                }

            }
        });
        btn_modify.setBounds(115, 304, 113, 27);
        panel.add(btn_modify);

        JPanel Jpic = new JPanel() {
            public void paint(Graphics g) {
                g.drawImage(new ImageIcon("1.jpg").getImage(), 0,0,this);
            }
        };
        Jpic.setBounds(536, 13, 170, 489);
        add(Jpic);

    }
    public boolean infocheck(boolean checkall) {
        // checkall mark check user and psw
        String psw = new String(psw_psw.getPassword());
        String psw_confirms = new String(psw_confirm.getPassword());
        if (checkall) {
            if (txt_name.getText().trim().equals("")) {
                JOptionPane.showMessageDialog(null, "组织名称错误！", "提示",
JOptionPane.WARNING_MESSAGE);
                return false;
            } else if (txt_ID.getText().trim().equals("")) {
                JOptionPane.showMessageDialog(null, "组织ID错误！", "提示",
JOptionPane.WARNING_MESSAGE);
                return false;
            } else if (txt_count.getText().trim().equals("")) {
                JOptionPane.showMessageDialog(null, "用户名错误！", "提示",
JOptionPane.WARNING_MESSAGE);
                return false;
            } else if (!psw_confirms.equals(psw)) {
                JOptionPane.showMessageDialog(null, "两次密码不一致！", "提示",
JOptionPane.WARNING_MESSAGE);
                return false;
            } else
                return true;
        } else {
            if (txt_name.getText().trim().equals("")) {
                JOptionPane.showMessageDialog(null, "组织名称错误！", "提示",
JOptionPane.WARNING_MESSAGE);
                return false;
            } else if (txt_ID.getText().trim().equals("")) {
                JOptionPane.showMessageDialog(null, "组织ID错误！", "提示",
JOptionPane.WARNING_MESSAGE);
                return false;
            } else
                return true;
        }

    }
}

class BuildTree extends Thread {
    JTree jt;
    DefaultMutableTreeNode node;

    public BuildTree(JTree jt, DefaultMutableTreeNode node) {
        this.jt = jt;
        this.node = node;
        this.start();
    }

    public void run() {
        Groups groups = null;
        for (Iterator<Groups> it = Data.groups.iterator(); it.hasNext();) {
            groups = it.next();
            DefaultMutableTreeNode node1 = new DefaultMutableTreeNode(groups);
            node.add(node1);
            User user = null;
            for (Iterator<User> its = groups.users.iterator(); its.hasNext();) {
                user = its.next();
                DefaultMutableTreeNode node2 = new DefaultMutableTreeNode(user);
                node1.add(node2);
            }
        }
        jt.updateUI();
    }
}
```

Setting. java

```
package admin.programUI;

import javax.swing.border.TitledBorder;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

import admin.data.Data;
import data.Groups;
import data.User;

import javax.swing.border.LineBorder;
import java.awt.Color;
import java.util.Iterator;

import javax.swing.*;

import java.awt.event.ActionListener;
import java.io.File;
import java.awt.event.ActionEvent;

public class SettingUI extends JPanel {
    private int pswMode;// 1-insert 2-modify
    private JTextField txt_userName;
    private JPasswordField txt_oldPW;
    private JPasswordField txt_newPW;
    private JPasswordField txt_confirmPW;
    private JList<User> list_user;
    private DefaultListModel<User> line;

    public SettingUI() {

        setBounds(100, 100, 720, 388);
        setLayout(null);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new TitledBorder(new LineBorder(new Color(0, 0, 0)),
"\u5BC6\u7801\u66F4\u6539",
               TitledBorder.LEADING, TitledBorder.TOP, null, null));
        panel_1.setBounds(14, 13, 482, 271);
        add(panel_1);
        panel_1.setLayout(null);

        JPanel panel = new JPanel();
        panel.setBounds(224, 29, 244, 188);
        panel_1.add(panel);
        panel.setLayout(null);

        JLabel label = new JLabel("\u539F\u5BC6\u7801");
        label.setBounds(14, 55, 72, 18);
        panel.add(label);

        JLabel lblNewLabel = new JLabel("\u7528\u6237\u540D");
        lblNewLabel.setBounds(14, 26, 72, 18);
        panel.add(lblNewLabel);

        JLabel label_1 = new JLabel("\u65B0\u5BC6\u7801");
        label_1.setBounds(14, 86, 72, 18);
        panel.add(label_1);

        JLabel label_2 = new JLabel("\u786E\u8BA4\u5BC6\u7801");
        label_2.setBounds(14, 117, 72, 18);
        panel.add(label_2);

        txt_userName = new JTextField();
        txt_userName.setEditable(false);
        txt_userName.setBounds(100, 23, 124, 24);
        panel.add(txt_userName);
        txt_userName.setColumns(10);

        txt_oldPW = new JPasswordField();
        txt_oldPW.setBounds(100, 52, 124, 24);
        panel.add(txt_oldPW);

        txt_newPW = new JPasswordField();
        txt_newPW.setBounds(100, 83, 124, 24);
        panel.add(txt_newPW);

        txt_confirmPW = new JPasswordField();
        txt_confirmPW.setBounds(100, 114, 124, 24);
        panel.add(txt_confirmPW);

        JButton button = new JButton("\u786E\u8BA4");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                User select = list_user.getSelectedValue();

                // first identify password is right
                boolean tag = false;// user and psw is right
                if (pswMode == 2) {
                    User user = null;
                    for (Iterator<User> it = Data.loginUser.iterator(); it.hasNext();)
{
                        user = it.next();
                        if (user.getUser().equals(select.getUser())
                                && user.getPassword().equals(new
String(txt_oldPW.getPassword()))) {
                            // correct
                            tag = true;
                        }
                    }
                } else
                    tag = true;
                // second password is right in two box
                if (tag) {
                    if ((new String(txt_newPW.getPassword())).equals(new
String(txt_confirmPW.getPassword()))) {
                        if (pswMode == 2) {
                            // modify
                            select.setPsw(new String(txt_newPW.getPassword()));
                        } else {
                            // final write in
                            User newUser = new User();
                            newUser.setUser(txt_userName.getText());
                            newUser.setPsw(new String(txt_newPW.getPassword()));
                            Data.loginUser.add(newUser);
                            line.addElement(newUser);
                        }

                    } else
                        JOptionPane.showMessageDialog(null, "原密码错误！", "提示",
JOptionPane.INFORMATION_MESSAGE);

                } else
                    JOptionPane.showMessageDialog(null, "原密码错误！", "提示",
JOptionPane.INFORMATION_MESSAGE);

            }
        });
        button.setBounds(152, 151, 72, 27);
        panel.add(button);

        JButton button_1 = new JButton("\u5220\u9664");
        button_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                if (JOptionPane.showConfirmDialog(null, "是否删除？", "警告",
JOptionPane.WARNING_MESSAGE,
                        JOptionPane.OK_CANCEL_OPTION) == JOptionPane.OK_OPTION) {
                    User select = list_user.getSelectedValue();
                    Data.LoginUser.remove(select);// 系统去除
                    line.removeElement(select);// 列表去除
                }

            }
        });
        button_1.setBounds(14, 151, 72, 27);
        panel.add(button_1);

        JButton button_2 = new JButton("\u63D2\u5165");
        button_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                pswMode = 1;
                txt_userName.setEditable(true);
                txt_userName.setText("");
                txt_newPW.setText("");
                txt_confirmPW.setText("");
                txt_oldPW.setEditable(false);
                button_2.setEnabled(false);
                list_user.clearSelection();
            }
        });
        button_2.setBounds(88, 151, 63, 27);
        panel.add(button_2);
```

```
        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(14, 28, 210, 230);
        panel_1.add(scrollPane);

        list_user = new JList<User>();
        scrollPane.setViewportView(list_user);
        line = new DefaultListModel<User>();
        showList(list_user, line);// show list

        JPanel panel_2 = new JPanel();
        panel_2.setBorder(new TitledBorder(new LineBorder(new Color(0, 0, 0)),
"\u5B58\u50A8\u4FE1\u606F", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0,
0, 0)));
        panel_2.setBounds(499, 13, 207, 271);
        add(panel_2);
        panel_2.setLayout(null);

        JScrollPane scrollPane_1 = new JScrollPane();
        scrollPane_1.setBounds(14, 23, 179, 235);
        panel_2.add(scrollPane_1);

        JTextArea txt_info = new JTextArea();
        txt_info.setEditable(false);
        scrollPane_1.setViewportView(txt_info);
        setVisible(true);
        list_user.addListSelectionListener(new ListSelectionListener() {

            @Override
            public void valueChanged(ListSelectionEvent e) {
                // TODO 自动生成的方法存根
                if (!e.getValueIsAdjusting() && list_user.getSelectedValue() != null)
{
                    User user = list_user.getSelectedValue();
                    txt_userName.setText(user.getUser());
                    button_2.setEnabled(true);
                    txt_userName.setEditable(false);// 不能修改
                    txt_oldPW.setEditable(true);
                    pswMode = 2;// modify
                    txt_newPW.setText("");
                    txt_confirmPW.setText("");
                }

            }
        });
        new InfoThread(txt_info);
```

```
    }

    public void showList(JList<User> list, DefaultListModel<User> line) {
        list.setModel(line);
        for (Iterator<User> it = Data.loginUser.iterator(); it.hasNext();) {
            line.addElement(it.next());
        }
        list.updateUI();

    }
}
class InfoThread extends Thread{
    JTextArea txt_info;
    public InfoThread(JTextArea txt_info) {
        this.txt_info = txt_info;
        this.start();
    }
    public void run() {
        File file;
        txt_info.setText("");
        txt_info.append("组织数据信息大小: ");
        file=new File("groups.dat");
        if(file.length()>5*1024&&file.length()<5*1024*1024)
txt_info.append((file.length()/1024)+"KB");
        else if(file.length()>5*1024*1024)
txt_info.append((file.length()/1024/1024)+"MB");
        else txt_info.append(file.length()+"B");
        txt_info.append("\n");
        //志愿者信息
        Groups groups=null;
        txt_info.append("志愿者数据信息大小: \n");
        for(Iterator<Groups>it=Data.groups.iterator();it.hasNext();) {
            groups=it.next();
            file=new File("group"+groups.ID+".dat");
            if(file.exists()) {
                //存在
                txt_info.append(file.getName()+" ");
                if(file.length()>5*1024&&file.length()<5*1024*1024)
txt_info.append((file.length()/1024)+"KB");
                else if(file.length()>5*1024*1024)
txt_info.append((file.length()/1024/1024)+"MB");
                else txt_info.append(file.length()+"B");
                txt_info.append("\n");
            }
        }
    }
}
```

SaveUI. java

```
package admin.programUI;

import javax.swing.JFrame;
import javax.swing.JProgressBar;

import admin.data.Data;
import admin.process.WriteData;

import javax.swing.JLabel;
import java.awt.Color;

public class SaveUI extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JProgressBar progressBar;

    public SaveUI() {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 440, 127);
        setResizable(false);

    setTitle("\u4FE1\u606F\u6B63\u5728\u4FDD\u5B58-\u5FD7\u613F\u8005\u7BA1\u7406\u
7CFB\u7EDF");
        getContentPane().setLayout(null);

        progressBar = new JProgressBar();
        progressBar.setForeground(Color.GREEN);
        progressBar.setBounds(14, 48, 404, 23);
        getContentPane().add(progressBar);

        JLabel label = new
JLabel("\u914D\u7F6E\u6587\u4EF6\u4EE5\u53CA\u4FE1\u606F\u6B63\u5728\u4FDD\u5B58...
");
        label.setBounds(14, 13, 212, 18);
        getContentPane().add(label);

        setVisible(true);
        new Saving(progressBar);
    }
```

```
}

class Saving extends Thread {
    JProgressBar progressBar;

    public Saving(JProgressBar progressBar) {
        this.progressBar = progressBar;
        this.start();
    }

    public void run() {
        new WriteData(1, Data.readGroup);
        new WriteData(2, Data.groups);
        new WriteData(3, Data.loginUser);
        new ProgressBar(progressBar);
        while (true) {
            try {
                Thread.sleep(10);
            } catch (InterruptedException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
            if (Data.GroupSaveOK && Data.GroupsSaveOK && Data.UsersSaveOK) {// 完成写
入
                progressBar.setValue(progressBar.getMaximum());
                try {
                    Thread.sleep(10);
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                break;
            }
        }

        // Saving is OK
        System.exit(0);// 结束所有
    }
}

class ProgressBar {
    public ProgressBar(JProgressBar bar) {
        for (int i = bar.getMinimum(); i <= bar.getMaximum() - 5; i++) {
            try {
                Thread.sleep(10);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            bar.setValue(i);
        }
    }
}
```

ReadUI. java

```
package admin.programUI;

import javax.swing.JFrame;
import javax.swing.JLabel;

import admin.data.Data;

import java.awt.Font;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.awt.event.WindowListener;

public class ReadUI extends JFrame{
    public ReadUI() {
        setTitle("\u4FE1\u606F\u8BFB\u53D6");
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 265, 98);
        setResizable(false);
        getContentPane().setLayout(null);

        JLabel label = new JLabel("\u6B63\u5728\u8BFB\u53D6\u4FE1\u606F...");
        label.setFont(new Font("宋体", Font.PLAIN, 17));
        label.setBounds(14, 25, 159, 18);
        getContentPane().add(label);
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                Data.recentReadReady=false;
                dispose();
            }
        });
    }

}
```

ManageUI. jaav

```java
package admin.programUI;

import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;

import data.*;
import admin.data.Data;
import admin.process.GetGroupData;

import javax.swing.JScrollPane;
import java.awt.BorderLayout;
import java.util.Iterator;

import javax.swing.*;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTree;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;
import java.awt.Graphics;

public class ManageUI extends JPanel {
    /**
     * 
     */
    private static final long serialVersionUID = 1L;
    private DefaultListModel<Groups> list1;
    private DefaultListModel<Volenteer> list2;
    private JTextArea txt_volSimpleInfo;
    private JButton btn_infoDetail;

    public ManageUI() {
        setBounds(100, 100, 720, 515);
        setLayout(null);

        JPanel panel = new JPanel();
        panel.setBounds(14, 13, 222, 356);
        add(panel);
        panel.setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(0, 242, 219, 114);
        panel.add(scrollPane);
        JTextArea txt_groupInfo = new JTextArea();
        txt_groupInfo.setEditable(false);
        scrollPane.setViewportView(txt_groupInfo);

        JScrollPane scrollPane_2 = new JScrollPane();
        scrollPane_2.setBounds(0, 0, 219, 237);
        panel.add(scrollPane_2);

        DefaultMutableTreeNode top = new DefaultMutableTreeNode("所有组织");
        JTree tree = new JTree(top);
        scrollPane_2.setViewportView(tree);

        tree.addTreeSelectionListener(new TreeSelectionListener() {

            @Override
            public void valueChanged(TreeSelectionEvent e) {
                // TODO Auto-generated method stub
                btn_infoDetail.setEnabled(true);
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();
                if (node == null)
                    return;
                else {
                    Object nodeInfo = node.getUserObject();
                    if (nodeInfo instanceof Volenteer) {
                        // 叶节点,vol
                        // 文档部分更新
                        txt_volSimpleInfo.setText("");
                        txt_volSimpleInfo.append("姓名: " + ((Volenteer) nodeInfo).name
+ "\n");
                        txt_volSimpleInfo.append("学号: " + ((Volenteer)
nodeInfo).IDnum + "\n");
                        txt_volSimpleInfo.append("性别: " + ((Volenteer) nodeInfo).sex
+ "\n");
                        txt_volSimpleInfo.append("状态: " + ((Volenteer)
nodeInfo).state + "\n");
                        txt_volSimpleInfo.append("一共参加" + ((Volenteer)
nodeInfo).activities.size() + "项志愿活动" + "\n");
                        Activity act = null;
                        float timecheck = 0f, timeuncheck = 0f;
                        int num = 0;
                        for (Iterator<Activity> it = ((Volenteer)
nodeInfo).activities.iterator(); it.hasNext();) {
                            act = it.next();
                            if (!act.checked) {// 未被审核的显示
                                timeuncheck += act.hour;
```

```
                                        txt_volSimpleInfo.append((++num) + ". | 未检查 | (" + 
act.getMonth() + "/" + act.getDay()
                                                + ") " + act.name + "-时长" + act.hour + "\n");
                                    } else {
                                        timecheck += act.hour;
                                        txt_volSimpleInfo.append((++num) + ". (" + 
act.getMonth() + "/" + act.getDay() + ") "
                                                + act.name + "-时长" + act.hour + "\n");
                                    }
                                }
                                txt_volSimpleInfo
                                        .append("总时长为" + (timecheck + timeuncheck) + "\n其
中未被审核的总时长为" + timeuncheck + "小时");
                            } else if (nodeInfo instanceof Groups) {
                                // 根节点，未读取信息
                                new BuildTreeVol(tree, node, ((Groups) nodeInfo).ID);
                                // 文档部分更新
                                txt_groupInfo.setText("");
                                txt_groupInfo.append("组织:" + ((Groups) nodeInfo).name + "\n");
                                txt_groupInfo.append("代码: " + ((Groups) nodeInfo).ID + "\n");
                                // txt_groupInfo.append("共有" + Data.recentGroup.vol.size() + 
"个志愿者" + "\n");

                            } else if (nodeInfo instanceof Group) {
                                // 根节点，已读取信息
                                // 文档部分更新
                                txt_groupInfo.setText("");
                                txt_groupInfo.append("组织: " + ((Group) nodeInfo).name + "\n");
                                txt_groupInfo.append("代码: " + ((Group) nodeInfo).ID + "\n");
                                txt_groupInfo.append("共有" + ((Group) nodeInfo).vol.size() + "
个志愿者" + "\n");
                            }
                        }
                    }
                });

        new ShowGroupTree(top, tree);// 建树

        JPanel panel_1 = new JPanel();
        panel_1.setBounds(239, 13, 283, 356);
        add(panel_1);
        panel_1.setLayout(new BorderLayout(0, 0));

        JPanel panel_2 = new JPanel();
        panel_1.add(panel_2, BorderLayout.SOUTH);
        JButton btn_infoModify = new JButton("\u4FEE\u6539");
        btn_infoModify.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent arg0) {
                        DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();
                        if (node == null)
                            return;
                        else {
                            Object nodeInfo = node.getUserObject();
                            if (nodeInfo instanceof Volenteer) {
                                // 叶节点,vol
                                new IndividuActionChangeUI((Group) ((DefaultMutableTreeNode)
node.getParent()).getUserObject(),
                                        (Volenteer) nodeInfo, false);

                            } else
                                JOptionPane.showMessageDialog(null, "不允许修改此结点！", "警告
", JOptionPane.WARNING_MESSAGE);
                        }
                    }
        });
        panel_2.add(btn_infoModify);

        btn_infoDetail = new JButton("\u8BE6\u7EC6\u4FE1\u606F");
        btn_infoDetail.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();
                        btn_infoDetail.setEnabled(false);
                        if (node == null)
                            return;
                        else {
                            Object nodeInfo = node.getUserObject();
                            if (nodeInfo instanceof Volenteer) {
                                // 叶节点,vol
                                // 文档部分更新
                                txt_volSimpleInfo.append("\n以下是详细信息：\n");
                                Activity act = null;
                                int num = 0;
                                for (Iterator<Activity> it = ((Volenteer)
nodeInfo).activities.iterator(); it.hasNext();) {
                                    act = it.next();
                                    if (!act.checked) {// 未被审核的显示
                                        txt_volSimpleInfo.append("----------" + (++num) +
"----------" + "\n");
                                        txt_volSimpleInfo.append("活动名称 " + act.name + "\n");
                                        txt_volSimpleInfo.append("项目未被审核" + "\n");
                                        txt_volSimpleInfo.append("志愿时长：" + act.hour + "\n");
                                        txt_volSimpleInfo.append("活动日期： " + act.getYear() +
"年" + act.getMonth() + "月"
                                        + act.getDay() + "日" + "\n");
                                        txt_volSimpleInfo.append("负责组织： " + act.group.name +
"\n");
                                    } else {
                                        txt_volSimpleInfo.append("----------" + (++num) +
"----------" + "\n");
                                        txt_volSimpleInfo.append("活动名称 " + act.name + "\n");
                                        txt_volSimpleInfo.append("志愿时长：" + act.hour + "\n");
                                        txt_volSimpleInfo.append("活动日期： " + act.getYear() +
"年" + act.getMonth() + "月"
                                        + act.getDay() + "日" + "\n");
                                        txt_volSimpleInfo.append("负责组织： " + act.group.name +
"\n");
                                    }
                                }
                            }
                        }

                    }
        });
        panel_2.add(btn_infoDetail);

        JScrollPane scrollPane_1 = new JScrollPane();
        panel_1.add(scrollPane_1, BorderLayout.CENTER);

        txt_volSimpleInfo = new JTextArea();
        txt_volSimpleInfo.setEditable(false);
        scrollPane_1.setViewportView(txt_volSimpleInfo);

        JPanel panel_3 = new JPanel() {
                    public void paint(Graphics g) {
                        g.drawImage(new ImageIcon("1.jpg").getImage(), 0, 0, this);
                    }
        };
        panel_3.setBounds(536, 13, 170, 489);
        add(panel_3);

    }
}

class ShowGroupTree extends Thread {
    DefaultMutableTreeNode node;
    JTree jt;

    public ShowGroupTree(DefaultMutableTreeNode node, JTree jt) {
        this.node = node;
        this.jt = jt;
        this.start();
    }

    public void run() {
        for (Iterator<Groups> it = Data.groups.iterator(); it.hasNext();) {
            DefaultMutableTreeNode nodes = new DefaultMutableTreeNode(it.next());
            node.add(nodes);
        }
        jt.updateUI();
    }

}

class BuildTreeVol extends Thread {
    JTree jt;
    DefaultMutableTreeNode parant;
    int ID;

    public BuildTreeVol(JTree jt, DefaultMutableTreeNode parant, int ID) {
        this.jt = jt;
        this.parant = parant;
        this.ID = ID;
        this.start();
    }

    public void run() {
        new GetGroupData(ID);
        Volenteer vol = null;
        for (Iterator<Volenteer> it = Data.recentGroup.vol.iterator(); it.hasNext();)
{
            vol = it.next();
            DefaultMutableTreeNode node = new DefaultMutableTreeNode(vol);
            parant.add(node);
        }
        // insert is OK
        parant.setUserObject(Data.recentGroup);
    }
}
```

IndividuActionChangeUI.java

```
package admin.programUI;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JList;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.border.LineBorder;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

import data.*;

import java.awt.Color;
import javax.swing.JTextField;
import java.awt.event.ActionListener;
import java.util.Iterator;
import java.awt.event.ActionEvent;

public class IndividuActionChangeUI extends JFrame {
    /**
     * 
     */
    private static final long serialVersionUID = 1L;
    private JTextField txt_actionName;
    private JTextField txt_actionTime;
    private JTextField txt_year;
    private JTextField txt_month;
    private JTextField txt_day;

    Group group;
    Volenteer vol;
    boolean showUnChecked;// 模式 只显示未被检查的

    public IndividuActionChangeUI(Group group, Volenteer vol, boolean showUnChecked)
{
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        // try {
        // UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        // } catch (Exception e) {
        // // TODO Auto-generated catch block
        // e.printStackTrace();
        // }
        this.group = group;
        this.vol = vol;
        this.showUnChecked = showUnChecked;
        setTitle("\u4FEE\u6539-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF");
        setResizable(false);
        setBounds(100, 100, 474, 442);
        getContentPane().setLayout(null);

        JLabel txt_infomation = new JLabel(

    "\u6B63\u5728\u5BF9{name}\u5728{groupName}\u7684\u4FE1\u606F\u8FDB\u884C\u4FEE\
u6539");
        txt_infomation.setBounds(14, 13, 360, 18);
        getContentPane().add(txt_infomation);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(14, 43, 187, 318);
        getContentPane().add(scrollPane);

        DefaultListModel<Activity> l = new DefaultListModel<Activity>();

        JList<Activity> list = new JList<Activity>(l);
        scrollPane.setViewportView(list);
        new CreatListThread(vol, l, showUnChecked);

        list.addListSelectionListener(new ListSelectionListener() {

            @Override
            public void valueChanged(ListSelectionEvent e) {
                // TODO Auto-generated method stub
                if (!e.getValueIsAdjusting()) {
                    Activity act = list.getSelectedValue();
                    txt_actionName.setText(act.name);
                    txt_actionTime.setText(String.valueOf(act.hour));
                    txt_year.setText(String.valueOf(act.getYear()));
                    txt_month.setText(String.valueOf(act.getMonth()));
                    txt_day.setText(String.valueOf(act.getDay()));
                }

            }
        });

        JPanel panel = new JPanel();
        panel.setBorder(new LineBorder(new Color(0, 0, 0)));
        panel.setBounds(215, 44, 239, 317);
        getContentPane().add(panel);
```

```
panel.setLayout(null);

JLabel label = new JLabel("\u6D3B\u52A8\u540D\u79F0");
label.setBounds(14, 13, 72, 18);
panel.add(label);

JLabel label_1 = new JLabel("\u5FD7\u613F\u65F6\u957F");
label_1.setBounds(14, 62, 72, 18);
panel.add(label_1);

JButton btn_submit = new JButton("\u63D0\u4EA4");
btn_submit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // submit
        if (insertInfoCheck()) {
            Activity act = list.getSelectedValue();
            act.name = txt_actionName.getText();
            act.hour = Float.valueOf(txt_actionTime.getText());
            act.setYear(Integer.valueOf(txt_year.getText()));
            act.setMonth(Integer.valueOf(txt_month.getText()));
            act.setDay(Integer.valueOf(txt_day.getText()));
        }
    }
});
btn_submit.setBounds(112, 277, 113, 27);
panel.add(btn_submit);

txt_actionName = new JTextField();
txt_actionName.setBounds(100, 10, 125, 24);
panel.add(txt_actionName);
txt_actionName.setColumns(10);

txt_actionTime = new JTextField();
txt_actionTime.setBounds(100, 59, 125, 24);
panel.add(txt_actionTime);
txt_actionTime.setColumns(10);

JPanel panel_1 = new JPanel();
panel_1.setBounds(14, 128, 211, 35);
panel.add(panel_1);
panel_1.setLayout(null);

txt_year = new JTextField();
txt_year.setBounds(3, 6, 61, 24);
txt_year.setColumns(10);
panel_1.add(txt_year);
```

```
        JLabel label_2 = new JLabel("\u5E74");
        label_2.setBounds(66, 9, 15, 18);
        panel_1.add(label_2);

        txt_month = new JTextField();
        txt_month.setBounds(82, 6, 43, 24);
        panel_1.add(txt_month);
        txt_month.setColumns(10);

        txt_day = new JTextField();
        txt_day.setBounds(142, 6, 42, 24);
        panel_1.add(txt_day);
        txt_day.setColumns(10);

        JLabel label_3 = new JLabel("\u6708");
        label_3.setBounds(125, 9, 15, 18);
        panel_1.add(label_3);

        JLabel label_4 = new JLabel("\u65E5");
        label_4.setBounds(185, 9, 15, 18);
        panel_1.add(label_4);

        JLabel label_5 = new JLabel("\u6D3B\u52A8\u65F6\u95F4");
        label_5.setBounds(14, 103, 72, 18);
        panel.add(label_5);

        JButton btn_confirm = new JButton("\u786E\u8BA4");
        btn_confirm.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                dispose();
            }
        });
        btn_confirm.setBounds(363, 367, 91, 27);
        getContentPane().add(btn_confirm);

        txt_infomation.setText("正在对 " + vol.name + " 在 " + group.name + " 的信息进
行修改");
        setVisible(true);

    }

    public boolean insertInfoCheck() {
        // action name
        try {
            if (txt_actionName.getText().trim().equals("")) {
                JOptionPane.showMessageDialog(null, "活动名称不能为空！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_actionTime.getText().trim().equals("") ||
Float.valueOf(txt_actionTime.getText()) < 0f
                    || Float.valueOf(txt_actionTime.getText()) > 24f) {
                // time is incorrect
                JOptionPane.showMessageDialog(null, "志愿时长不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_year.getText().trim().equals("") ||
Integer.valueOf(txt_year.getText()) < 1000
                    || Integer.valueOf(txt_year.getText()) >= 10000) {
                // year is incorrect
                JOptionPane.showMessageDialog(null, "年不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_month.getText().trim().equals("") ||
Integer.valueOf(txt_month.getText()) < 0
                    || Integer.valueOf(txt_month.getText()) > 12) {
                // month is incorrect
                JOptionPane.showMessageDialog(null, "月不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_day.getText().trim().equals("") ||
Integer.valueOf(txt_day.getText()) < 0
                    || Integer.valueOf(txt_day.getText()) > 31) {
                // day is incorrect
                JOptionPane.showMessageDialog(null, "日不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else
                return true;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "输入有误！", "提示",
JOptionPane.ERROR_MESSAGE);
            return false;
        }
    }
}

class CreatListThread extends Thread {
    Volenteer vol;
    DefaultListModel<Activity> list;
    boolean showUnChecked;
    public CreatListThread(Volenteer vol, DefaultListModel<Activity> list, boolean
showUnChecked) {
        this.vol = vol;
        this.list = list;
        this.start();
    }

    public void run() {
        Activity act = null;
        for (Iterator<Activity> it = vol.activities.iterator(); it.hasNext();) {
            act = it.next();
            if (!act.checked || !showUnChecked) {
                list.addElement(act);
            }

        }
    }
}
```

User:

data 包

Data.java

```
package user.data;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.ArrayList;

import javax.swing.JOptionPane;

import data.*;

public class Data {
    // in this section will define all global set
    // user set which to login
    public static String groupName;// 组织信息
    public static int groupID;// 组织ID
    public static String loginUser;// 登陆进去的用户
    // user collection
    public static ArrayList<Groups> groups;
    // group
    public static Group group;
    // ready?using to identify reading is OK
    public static boolean Userready;
    public static boolean Groupready;

    //Need Save or Save is OK?
    public static boolean saveGroup;
    public static boolean saveUser;
    public static boolean saveGroups;

    public static User user;//user login


    public Data() {
        // TODO Auto-generated constructor stub
        // set null
        groupName = null;
        groupID = -1;
        loginUser = null;
        groups = null;
        group = null;
        Userready = false;
        Groupready = false;
        saveGroup=false;
        saveUser=false;
        saveGroups=false;
    }

    /**
     * 将文件配置读入，得到User
     *
     * @param file
     *            文件路径
     */
    public static void getGroupUser(String file) {
        try {
            ObjectInputStream input = new ObjectInputStream(new FileInputStream(file));
            groups = (ArrayList<Groups>) input.readObject();

            input.close();
            Userready = true;

        } catch (Exception e) {
            // TODO: handle exception
            e.printStackTrace();
            System.err.print("Read setting file ERROR!");
            JOptionPane.showMessageDialog(null, "Read Setting File
ERROR","ERROR",JOptionPane.ERROR_MESSAGE);
            System.exit(0);

        }
    }

    /**
     * 将文件配置读入，得到Groups
     *
     * @param file
     *            文件路径
     */
    public static void getGroup(String file) {
        try {
            ObjectInputStream input = new ObjectInputStream(new FileInputStream(file));
            group = (Group) input.readObject();

            input.close();
            Groupready = true;
        } catch (FileNotFoundException e1) {
            // TODO: handle exception
            e1.printStackTrace();
            // file not find then we will create it
            // also help to first load
            Data.group=new Group();
            Data.group.ID=Data.groupID;
            Data.group.name=Data.groupName;
            writeGroup(file);
            Groupready=true;
            JOptionPane.showMessageDialog(null, "似乎没有找到志愿者信息文件
","ERROR",JOptionPane.ERROR_MESSAGE);

        } catch (Exception e2) {
            e2.printStackTrace();
            System.err.print("Read setting file ERROR!");
            JOptionPane.showMessageDialog(null, "Read Setting File
ERROR","ERROR",JOptionPane.ERROR_MESSAGE);
        }
    }

    /**
     * 将信息写入文件，所有的信息将会一次性写入
     *
     * @param file
     *            文件目录
     */
    public static void writeGroup(String file) {
        try {
            ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(file));
            out.writeObject(group);//存入志愿者等等信息

            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    /**
     * 将信息写入文件，所有的信息将会一次性写入
     *
     * @param file
     *            文件目录
     */
    public static void writeUser(String file) {
        try {
            ObjectOutputStream out = new ObjectOutputStream(new
FileOutputStream(file));
            out.writeObject(groups);//存入组织等等信息

            out.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

}
```

process 包

ReadData. java

```
package user.process;

import data.NetFileWork;
import user.data.Data;

public class ReadData {
    /**
     * using to read user info
     */
    public static void readGroupUser() {
        new ReadGroupUser();
    }
    /**
     * using to read groups info
     * @param ID group ID
     */
    public static void readGroup(int ID) {
        new ReadGroup(ID);
    }

}

class ReadGroupUser extends Thread {
    public ReadGroupUser() {
        this.start();
    }

    public void run() {
        try {
            new NetFileWork("groups.dat", 1);
            Thread.sleep(100);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Data.getGroupUser("groups.dat");
    }

}

class ReadGroup extends Thread {
    int ID;

    /**
     * Read Group info
     *
     * @param ID
     *           using to identify which group
     */
    public ReadGroup(int ID) {
        this.ID = ID;
        this.start();
    }

    public void run() {
        Data.getGroup("group" + ID + ".dat");
    }
```

```
}
```

WriteData.java

```
package user.process;

import data.NetFileWork;
import user.data.Data;

public class WriteData {
    /**
     * 
     * @param ID
     *            groupID
     * @param type
     *            1-write group 2-write groups
     */
    public WriteData(int ID, int type) {
        if (type == 1)
            new WriteGroup(ID);
        else
            new WriteGroups();
    }
}

class WriteGroup extends Thread {
    int ID;

    public WriteGroup(int ID) {
        this.ID = ID;
        this.start();
    }

    public void run() {
        Data.writeGroup("group" + ID + ".dat");
        try {
            Thread.sleep(100);
            new NetFileWork("group" + ID + ".dat", 2);
            Thread.sleep(100);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Data.saveGroup = true;
    }
}

class WriteGroups extends Thread {

    public WriteGroups() {
        this.start();
    }

    public void run() {
        Data.writeUser("groups.dat");
        try {
            Thread.sleep(100);
            new NetFileWork("groups.dat", 2);
            Thread.sleep(100);
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        Data.saveGroups = true;
    }

}
```

programUI 包

LoginUI. java

```
package user.programUI;

import javax.swing.*;
import javax.swing.border.TitledBorder;

import user.data.Data;
import data.*;
import user.process.ReadData;

import javax.swing.border.LineBorder;
import java.awt.Color;
import java.awt.event.ActionListener;
import java.util.Collection;
import java.util.Iterator;
import java.awt.event.ActionEvent;

public class LoginUI extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    JTextField txt_userName;
    JPasswordField txt_passWord;
    JComboBox<String> comboBox;

    public LoginUI() {
        // try {
        // UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        // } catch (Exception e) {
        // // TODO Auto-generated catch block
        // e.printStackTrace();
        // }
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("\u767B\u9646-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF");
        setResizable(false);
        setBounds(100, 100, 333, 251);
        getContentPane().setLayout(null);

        JPanel panel = new JPanel();
        panel.setBounds(14, 13, 302, 188);
        getContentPane().add(panel);
        panel.setLayout(null);

        JLabel label = new JLabel("\u7528\u6237\u540D");
        label.setBounds(14, 13, 72, 18);
        panel.add(label);

        JLabel label_1 = new JLabel("\u5BC6\u7801");
        label_1.setBounds(14, 44, 72, 18);
        panel.add(label_1);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new TitledBorder(new LineBorder(new Color(0, 0, 0)),
"\u7EC4\u7EC7\u9009\u62E9",
               TitledBorder.LEADING, TitledBorder.TOP, null, null));
        panel_1.setBounds(14, 75, 276, 64);
        panel.add(panel_1);
        panel_1.setLayout(null);

        comboBox = new JComboBox();
        comboBox.setBounds(14, 27, 248, 24);
        panel_1.add(comboBox);

        JButton btn_confirm = new JButton("\u786E\u5B9A");
        btn_confirm.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                // login
                String pwd = new String(txt_passWord.getPassword());
                if (loginCheck(txt_userName.getText(), pwd,
comboBox.getSelectedItem().toString(),
                        Data.groups) == true) {
                    // 成功登陆
                    // System.out.println(txt_userName.getText()+ ","+pwd);
                    try {
                        new NetFileWork("group" + Data.groupID + ".dat", 1);
                        Thread.sleep(100);
                    } catch (Exception e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                    }
                    ReadData.readGroup(Data.groupID);// 读取信息
                    new ReadUI();
                    while (true) {
                        if (Data.Groupready) {
                            new MainUIPanel();
                            //new MainUI();// 打开MainUI
                            break;
                        }
                        //System.out.println(Data.Groupready);
                        try {
                            Thread.sleep(100);
                        } catch (InterruptedException e) {
                            // TODO Auto-generated catch block
                            e.printStackTrace();
                        }
                    }
                    dispose();

                } else {
                    // 登录失败
                    setNull();// 恢复初始
                }

            }
        });
        btn_confirm.setBounds(24, 152, 113, 27);
        panel.add(btn_confirm);

        JButton btn_reset = new JButton("\u91CD\u7F6E");
        btn_reset.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                setNull();// 恢复初始
            }
        });
        btn_reset.setBounds(167, 152, 113, 27);
        panel.add(btn_reset);

        txt_userName = new JTextField();
        txt_userName.setBounds(69, 10, 205, 24);
        panel.add(txt_userName);
        txt_userName.setColumns(10);

        txt_passWord = new JPasswordField();
        txt_passWord.setBounds(69, 44, 205, 24);
        panel.add(txt_passWord);
        setNull();// 初始化
        setVisible(true);
    }
    public static boolean loginCheck(String userName,String password,String
groupName,Collection<Groups> groups) {//判断是否登陆成功
        Groups group=null;
        boolean flag=false;//is find?
        for(Iterator<Groups> it=groups.iterator();it.hasNext();) {
            group=it.next();
            if(group.name.equals(groupName)) {
                //找到组织
                flag=true;
                break;
            }
        }
        if(flag==true) {
            User user =null;flag=false;
            for(Iterator<User> it=group.users.iterator();it.hasNext();) {
                //查找用户
                user=it.next();

    if(user.getUser().equals(userName)&&user.getPassword().equals(password)) {
                    //确认正确，变量存入
                    Data.user=user;
                    Data.groupName=groupName;
                    Data.groupID=group.ID;
                    Data.LoginUser=userName;
                    flag=true;
                }
            }
            if(flag==false) return false;
            else return true;
        }else return false;

    }

    private void setNull() {
        // 设置成初始值
        txt_passWord.setText("");
        txt_userName.setText("");
        Groups gp = null;// 暂存group
        comboBox.removeAllItems();
        for (Iterator<Groups> it = Data.groups.iterator(); it.hasNext();) {
            gp = it.next();
            comboBox.addItem(gp.name);
        }
        comboBox.setSelectedIndex(0);

    }

}
```

```
MainUIPanel.java
package user.programUI;


import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;


import javax.swing.JFrame;
import javax.swing.JLabel;

import user.data.Data;
import javax.swing.JTabbedPane;

import user.programUI.SettingUI;

import javax.swing.JPanel;

public class MainUIPanel extends JFrame {
    JLabel txt_groupName;
    JLabel txt_code;
    JLabel txt_user;
    private JPanel volManageSet;
    private JPanel setting;
    public MainUIPanel() {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    setTitle("\u4E3B\u754C\u9762-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF")
;
        setResizable(false);
        setBounds(100, 100, 620, 618);
        getContentPane().setLayout(null);

        txt_groupName = new JLabel("{groupName}");
        txt_groupName.setBounds(24, 13, 150, 18);
        getContentPane().add(txt_groupName);

        txt_code = new JLabel("{Code}");
        txt_code.setBounds(257, 13, 79, 18);
        getContentPane().add(txt_code);

        txt_user = new JLabel("{user}");
        txt_user.setBounds(257, 33, 79, 18);
        getContentPane().add(txt_user);

        JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
        tabbedPane.setBounds(24, 44, 573, 524);
        getContentPane().add(tabbedPane);

        JPanel insertRecordSet = new InsertRecordUI();
        tabbedPane.addTab("\u65F6\u957F\u5F55\u5165",  insertRecordSet);

        volManageSet = new VolManageUI();
        tabbedPane.addTab("\u4FE1\u606F\u7BA1\u7406",  volManageSet);

        setting = new SettingUI();
        tabbedPane.addTab("设置", setting);


        firstShow();
        setVisible(true);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                new SaveUI(1);//保存作业
            }

        });
    }
    /**
     * show frame setting
     */
    public void firstShow() {
        txt_groupName.setText("组织名称："+Data.groupName);
        txt_code.setText("ID："+String.valueOf(Data.groupID));
        txt_user.setText("操作员："+Data.loginUser);
    }
}
```

ModifyInfoUI.java

```
package user.programUI;

import javax.swing.*;
import javax.swing.border.EtchedBorder;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;

import data.*;
import user.data.Data;

import java.awt.Font;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.util.Iterator;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class ModifyInfoUI extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JTextField txt_name;
    private JTextField txt_IDNumber;
    private JTree tree_info;
    private JComboBox cb_state;

    public ModifyInfoUI() {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    setTitle("\u4FE1\u606F\u4FEE\u6539-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF")
;
        setResizable(false);
        setBounds(100, 100, 483, 355);
        getContentPane().setLayout(null);

        JPanel panel = new JPanel();
        panel.setBorder(new EtchedBorder(EtchedBorder.LOWERED, null, null));
        panel.setBounds(14, 13, 139, 294);
        getContentPane().add(panel);
        panel.setLayout(null);

        JLabel label = new JLabel("\u59D3\u540D");
        label.setBounds(14, 13, 72, 18);
        panel.add(label);

        JLabel label_1 = new JLabel("\u5B66\u53F7");
        label_1.setBounds(14, 72, 72, 18);
        panel.add(label_1);

        JLabel label_2 = new JLabel("\u6027\u522B");
        label_2.setBounds(14, 129, 72, 18);
        panel.add(label_2);

        JLabel label_3 = new JLabel("\u72B6\u6001");
        label_3.setBounds(14, 186, 72, 18);
        panel.add(label_3);

        txt_name = new JTextField();
        txt_name.setBounds(14, 35, 111, 24);
        panel.add(txt_name);
        txt_name.setColumns(10);

        txt_IDNumber = new JTextField();
        txt_IDNumber.setBounds(14, 92, 111, 24);
        panel.add(txt_IDNumber);
        txt_IDNumber.setColumns(10);

        JComboBox cb_sex = new JComboBox();
        cb_sex.setModel(new DefaultComboBoxModel(new String[] { "\u7537", "\u5973" }));
        cb_sex.setBounds(14, 149, 72, 24);
        panel.add(cb_sex);

        cb_state = new JComboBox();
        cb_state.setEditable(true);
        cb_state.setModel(new DefaultComboBoxModel(
                new String[] { "\u5728\u804C", "\u8BD5\u7528", "\u4E34\u65F6",
"\u975E\u5728\u804C" }));
        cb_state.setBounds(14, 217, 111, 24);
        panel.add(cb_state);

        JButton btn_modify = new JButton("\u4FEE\u6539");
        btn_modify.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                // check!
                if (checkInfo()) {
                    DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_info.getLastSelectedPathComponent();

                    // 判断是否为空树
                    if (node == null) {
                        return;
                    }
                    Object nodeInfo = node.getUserObject();
                    if (nodeInfo instanceof Volenteer) {
                        // 叶节点,vol
                        Volenteer vol = (Volenteer) nodeInfo;
                        vol.name = txt_name.getText();
                        vol.IDnum = Integer.valueOf(txt_IDNumber.getText());
                        vol.sex = cb_sex.getSelectedItem().toString();
                        vol.state = cb_state.getSelectedItem().toString();

                        JOptionPane.showMessageDialog(null, "修改成功！", "提示",
JOptionPane.INFORMATION_MESSAGE);

                    } else
                        return;
                }
            }
        });
        btn_modify.setBounds(44, 254, 81, 27);
        panel.add(btn_modify);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(156, 13, 311, 266);
        getContentPane().add(scrollPane);

        DefaultMutableTreeNode top = new DefaultMutableTreeNode(Data.group);
        tree_info = new JTree(top);

        new TreeShowThread(tree_info, top);
        scrollPane.setViewportView(tree_info);

        tree_info.addTreeSelectionListener(new TreeSelectionListener() {

            @Override
            public void valueChanged(TreeSelectionEvent e) {
                // TODO Auto-generated method stub
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_info.getLastSelectedPathComponent();

                // 判断是否为空树
                if (node == null) {
                    return;
                }
                Object nodeInfo = node.getUserObject();
                if (nodeInfo instanceof Volenteer) {
                    // 叶节点,vol
                    btn_modify.setEnabled(true);
                    Volenteer vol = (Volenteer) nodeInfo;
                    txt_name.setText(vol.name);
                    txt_IDNumber.setText(String.valueOf(vol.IDnum));
                    cb_sex.setSelectedItem(vol.sex);
                    cb_state.setSelectedItem(vol.state);
                } else if (nodeInfo instanceof Group) {
                    // 根节点,group
                    btn_modify.setEnabled(false);
                    txt_name.setText("");
                    txt_IDNumber.setText("");
                    cb_sex.setSelectedItem("男");
                    cb_state.setSelectedIndex(0);
                }
            }
        });

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new EtchedBorder(EtchedBorder.LOWERED, null, null));
        panel_1.setBounds(166, 285, 297, 25);
        getContentPane().add(panel_1);
        panel_1.setLayout(null);
        JLabel label_4 = new
JLabel("\u70B9\u51FB\u9879\u76EE\u8FDB\u884C\u4FEE\u6539");
        label_4.setBounds(93, 4, 104, 16);
        label_4.setFont(new Font("新宋体", Font.PLAIN, 13));
        panel_1.add(label_4);
        // group展开
        tree_info.expandRow(0);
        // first make btn disable
        btn_modify.setEnabled(false);

        setVisible(true);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }

        });
    }

    public ModifyInfoUI(Object object) {
        this();
        // find object

    }

    public boolean checkInfo() {
        try {
            if (txt_name.getText().trim().equals("")) {
                // name is incorrect
                JOptionPane.showMessageDialog(null, "姓名不能为空！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_IDNumber.getText().trim().equals("") ||
Integer.valueOf(txt_IDNumber.getText()) <= 10000000
                    || Integer.valueOf(txt_IDNumber.getText()) >= 1000000000) {
                // id is incorrect
                JOptionPane.showMessageDialog(null, "学号格式不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (cb_state.getSelectedItem().toString().trim().equals("")) {
                // state is incorrect
                JOptionPane.showMessageDialog(null, "状态不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else
```

```
                return true;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "输入有误，请检查！", "提示",
JOptionPane.ERROR_MESSAGE);
            return false;

        }

    }
}

class TreeThread extends Thread {
    DefaultMutableTreeNode jt;
    JTree tree;

    public TreeThread(DefaultMutableTreeNode jt, JTree tree) {
        this.jt = jt;
        this.tree = tree;
        this.start();
    }

    public void run() {
        // 更新信息
        Volenteer vols;
        for (Iterator<Volenteer> vol = Data.group.vol.iterator(); vol.hasNext();) {
            vols = vol.next();
            // 加入志愿者节点
            jt.add(new DefaultMutableTreeNode(vols));
        }
        // tree.setSelectionPath(new TreePath(jt.getPath()));
        tree.updateUI();

    }

}
```

InsertRecordUI.java

```
package user.programUI;

import javax.swing.*;
import javax.swing.border.TitledBorder;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;
import data.*;
import user.data.Data;
import user.process.WriteData;

import java.awt.Color;
import java.awt.event.ActionListener;
import java.util.Iterator;
import java.awt.event.ActionEvent;

public class InsertRecordUI extends JPanel {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JTextField txt_name;
    private JTextField txt_id;
    private JTextField txt_time;
    private JTextField txt_activity;
    private JTextField txt_year;
    private JTextField txt_month;
    private JTextField txt_day;
    private DefaultMutableTreeNode top;
    private JComboBox box_sex;
    private JTree tree;

    private JButton btn_delete;

    public InsertRecordUI() {
        setLayout(null);
        setBounds(0, 0, 575, 480);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(24, 13, 233, 452);
        add(scrollPane);

        top = new DefaultMutableTreeNode(Data.groupName);
        top.setUserObject(Data.group);

        tree = new JTree(top);
        scrollPane.setViewportView(tree);
        new CreateTree(top, tree);// 开始建树

        // 加入监听
        tree.addTreeSelectionListener(new TreeSelectionListener() {

            @Override
            public void valueChanged(TreeSelectionEvent e) {
                // TODO Auto-generated method stub
                // 获取树的节点
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();
                // 判断是否为空树
                if (node == null) {
                    return;
                }
                DefaultMutableTreeNode parant = (DefaultMutableTreeNode)
node.getParent();
                Object nodeInfo = node.getUserObject();
                if (nodeInfo instanceof Volenteer) {
                    // 根节点
                    txt_name.setEnabled(false);
                    txt_id.setEnabled(false);
                    box_sex.setEnabled(false);// 不允许修改
                    Volenteer vol = (Volenteer) nodeInfo;
                    txt_name.setText(vol.name);
                    txt_id.setText(String.valueOf(vol.IDnum));
                    box_sex.setSelectedItem(vol.sex);
                    txt_activity.setText("");
                    txt_time.setText("");
                    txt_year.setText("");
                    txt_month.setText("");
                    txt_day.setText("");
                } else if (nodeInfo instanceof Activity) {
                    // 叶子
                    txt_name.setEnabled(false);
                    txt_id.setEnabled(false);
                    box_sex.setEnabled(false);// 不允许修改
                    Activity act = (Activity) nodeInfo;
                    Volenteer vol = (Volenteer) parant.getUserObject();
                    txt_name.setText(vol.name);
                    txt_id.setText(String.valueOf(vol.IDnum));
                    box_sex.setSelectedItem(vol.sex);
                    txt_activity.setText(act.name);
                    txt_time.setText(String.valueOf(act.hour));
                    txt_year.setText(String.valueOf(act.getYear()));
                    txt_month.setText(String.valueOf(act.getMonth()));
                    txt_day.setText(String.valueOf(act.getDay()));

                } else if (nodeInfo instanceof Group) {
                    // 最根节点
                    txt_name.setEnabled(true);
                    txt_id.setEnabled(true);
                    box_sex.setEnabled(true);// 允许
                    txt_name.setText("");
                    txt_id.setText("");
                    box_sex.setSelectedIndex(0);
                    txt_activity.setText("");
                    txt_time.setText("");
                    txt_year.setText("");
                    txt_month.setText("");
                    txt_day.setText("");

                }

            }
        });

        JPanel panel = new JPanel();
        panel.setBorder(new TitledBorder(UIManager.getBorder("TitledBorder.border"),
 "\u64CD\u4F5C", TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
        panel.setBounds(271, 306, 276, 67);
        add(panel);
        panel.setLayout(null);

        JButton btn_createTable = new JButton("\u751F\u6210\u5199\u5B9E\u8868");
        btn_createTable.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                new DateSelectUI();

            }
        });
        btn_createTable.setBounds(14, 25, 110, 27);
        panel.add(btn_createTable);

        JButton btn_createData = new JButton("\u6570\u636E\u4E0A\u4F20");
        btn_createData.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                btn_createData.setText("请等待...");
                new WriteData(Data.groupID,1);
                while(!Data.saveGroup) {
                    try {
                        Thread.sleep(100);
                    } catch (InterruptedException e) {
                        // TODO 自动生成的 catch 块
                        e.printStackTrace();
                    }
                }
                Data.saveGroup=false;
```

```
                JOptionPane.showMessageDialog(null, "成功上传！");
                btn_createData.setText("\u6570\u636E\u4E0A\u4F20");
            }
        });
        btn_createData.setBounds(152, 25, 110, 27);
        panel.add(btn_createData);

        JPanel panel_2 = new JPanel();
        panel_2.setBorder(new TitledBorder(null, "\u63D2\u5165/\u66F4\u6539",
TitledBorder.LEADING, TitledBorder.TOP,
                null, null));
        panel_2.setBounds(271, 13, 276, 280);
        add(panel_2);
        panel_2.setLayout(null);

        JLabel label = new JLabel("\u59D3\u540D");
        label.setBounds(14, 35, 72, 18);
        panel_2.add(label);

        JLabel label_1 = new JLabel("\u5B66\u53F7");
        label_1.setBounds(14, 66, 72, 18);
        panel_2.add(label_1);

        JLabel label_2 = new JLabel("\u6D3B\u52A8\u540D\u79F0");
        label_2.setBounds(14, 127, 72, 18);
        panel_2.add(label_2);

        JLabel label_3 = new JLabel("\u5FD7\u613F\u65F6\u957F");
        label_3.setBounds(14, 158, 95, 18);
        panel_2.add(label_3);

        txt_name = new JTextField();
        txt_name.setBounds(87, 32, 168, 24);
        panel_2.add(txt_name);
        txt_name.setColumns(10);

        txt_id = new JTextField();
        txt_id.setBounds(87, 63, 168, 24);
        panel_2.add(txt_id);
        txt_id.setColumns(10);

        txt_time = new JTextField();
        txt_time.setBounds(87, 155, 95, 24);
        panel_2.add(txt_time);
        txt_time.setColumns(10);
        txt_activity = new JTextField();
        txt_activity.setBounds(87, 124, 168, 24);
        panel_2.add(txt_activity);
        txt_activity.setColumns(10);

        JLabel label_5 = new JLabel("\u5C0F\u65F6");
        label_5.setBounds(190, 158, 72, 18);
        panel_2.add(label_5);

        JButton btn_confirm = new JButton("\u66F4\u6539/\u63D2\u5165");
        btn_confirm.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                // 首先检查是否正确输入
                if (infoCheck()) {

                    // 修改删除，若为根节点则插入，若为叶子节点则修改
                    // 获取树的节点
                    DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree.getLastSelectedPathComponent();
                    // 判断是否为空树
                    if (node == null) {

                        return;
                    }
                    Object nodeInfo = node.getUserObject();
                    if (nodeInfo instanceof Volenteer) {
                        // 在根节点位置,vol
                        Volenteer vol = (Volenteer) nodeInfo;
                        Activity act = new Activity();

                        act.group = Data.group;
                        act.name = txt_activity.getText();
                        act.hour = Integer.valueOf(txt_time.getText());

                        act.setYear(Integer.valueOf(txt_year.getText()));
                        act.setMonth(Integer.valueOf(txt_month.getText()));
                        act.setDay(Integer.valueOf(txt_day.getText()));
                        act.checked=false;

                        vol.activities.add(act);
                        // 由于新加入 需要生成节点

                        node.add(new DefaultMutableTreeNode(act));// act

                        tree.updateUI();
                    } else if (nodeInfo instanceof Activity) {
                        // 在叶子节点,act
                        Activity act = (Activity) nodeInfo;
                        act.group = Data.group;
                        act.name = txt_activity.getText();
                        act.hour = Float.valueOf(txt_time.getText());

                        act.setYear(Integer.valueOf(txt_year.getText()));
                        act.setMonth(Integer.valueOf(txt_month.getText()));
                        act.setDay(Integer.valueOf(txt_day.getText()));
                        act.checked=false;

                        tree.updateUI();
                    } else if (nodeInfo instanceof Group) {
                        // 最根节点,root
                        Group group = (Group) nodeInfo;

                        Volenteer vol = new Volenteer();
                        vol.name = txt_name.getText();
                        vol.IDnum = Integer.valueOf(txt_id.getText());
                        vol.sex = box_sex.getSelectedItem().toString();

                        Activity act = new Activity();

                        act.group = Data.group;
                        act.name = txt_activity.getText();
                        act.hour = Integer.valueOf(txt_time.getText());

                        act.setYear(Integer.valueOf(txt_year.getText()));
                        act.setMonth(Integer.valueOf(txt_month.getText()));
                        act.setDay(Integer.valueOf(txt_day.getText()));
                        act.checked=false;

                        vol.activities.add(act);

                        group.vol.add(vol);

                        // 由于新加入 需要生成节点
                        DefaultMutableTreeNode nodes = new
 DefaultMutableTreeNode(vol);// 根节点,vol
                        DefaultMutableTreeNode leaf = new
 DefaultMutableTreeNode(act);// 叶子节点,act

                        nodes.add(leaf);
                        node.add(nodes);
                        tree.updateUI();
                    }
                }
            }
        });
        btn_confirm.setBounds(148, 240, 113, 27);
        panel_2.add(btn_confirm);

        JLabel label_4 = new JLabel("\u65F6\u95F4");
        label_4.setBounds(14, 189, 95, 18);
        panel_2.add(label_4);

        JLabel label_6 = new JLabel("\u5E74");
        label_6.setBounds(138, 192, 22, 18);
        panel_2.add(label_6);

        txt_year = new JTextField();
        txt_year.setBounds(87, 189, 48, 24);
        panel_2.add(txt_year);
        txt_year.setColumns(10);

        txt_month = new JTextField();
        txt_month.setBounds(159, 189, 36, 24);
        panel_2.add(txt_month);
        txt_month.setColumns(10);

        txt_day = new JTextField();
        txt_day.setBounds(215, 189, 36, 24);
        panel_2.add(txt_day);
        txt_day.setColumns(10);

        JLabel label_7 = new JLabel("\u6708");
        label_7.setBounds(195, 192, 15, 18);
        panel_2.add(label_7);

        JLabel label_8 = new JLabel("\u65E5");
        label_8.setBounds(253, 192, 15, 18);
        panel_2.add(label_8);

        btn_delete = new JButton("\u5220\u9664");
        btn_delete.setBounds(14, 240, 110, 27);
        panel_2.add(btn_delete);

        JLabel label_9 = new JLabel("\u6027\u522B");
        label_9.setBounds(14, 96, 72, 18);
        panel_2.add(label_9);
```

```
        box_sex = new JComboBox();
        box_sex.setModel(new DefaultComboBoxModel(new String[] { "\u7537", 
"\u5973" }));
        box_sex.setSelectedIndex(0);
        box_sex.setBounds(87, 93, 73, 24);
        panel_2.add(box_sex);

        btn_delete.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if (JOptionPane.showConfirmDialog(null, "确认删除？", "警告", 
JOptionPane.WARNING_MESSAGE,
                        JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {

                    // 修改删除，若为根节点则插入，若为叶子节点则修改
                    // 获取树的节点
                    DefaultMutableTreeNode node = (DefaultMutableTreeNode) 
tree.getLastSelectedPathComponent();
                    // 判断是否为空树
                    if (node == null) {
                        return;
                    }
                    Object nodeInfo = node.getUserObject();
                    if (nodeInfo instanceof Volenteer) {
                        // 在根节点位置,vol
                        Volenteer vol = (Volenteer) nodeInfo;
                        Data.group.vol.remove(vol);

                        top.remove(node);
                        tree.updateUI();
                        JOptionPane.showMessageDialog(null, "已删除");

                    } else if (nodeInfo instanceof Activity) {
                        // 在叶子节点,act
                        Activity act = (Activity) nodeInfo;
                        // 父节点
                        DefaultMutableTreeNode parent = (DefaultMutableTreeNode) 
node.getParent();

                        Volenteer vol = (Volenteer) parent.getUserObject();
                        vol.activities.remove(act);
                        parent.remove(node);
                        tree.updateUI();
                        JOptionPane.showMessageDialog(null, "已删除");

                    } else if (nodeInfo instanceof Group) {
                        // 最根节点,root
                        JOptionPane.showMessageDialog(null, "根节点不能删除");
                    }
                }
            }
        });
        tree.expandRow(0);// 首个展开

     }

     /**
      * check information
      *
      * @return true marked information is checked;false marked information is not
      *         correct
      */
     public boolean infoCheck() {
        // 不符合条件的
        try {
            if (txt_name.getText().trim().equals("")) {
                // name is incorrect
                JOptionPane.showMessageDialog(null, "姓名不能为空！", "提示",
 JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_id.getText().trim().equals("") ||
 Integer.valueOf(txt_id.getText()) <= 10000000
                    || Integer.valueOf(txt_id.getText()) >= 1000000000) {
                // id is incorrect
                JOptionPane.showMessageDialog(null, "学号格式不正确！", "提示",
 JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_activity.getText().trim().equals("")) {
                // activity is incorrect
                JOptionPane.showMessageDialog(null, "活动名称不能为空！", "提示",
 JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_time.getText().trim().equals("") ||
 Float.valueOf(txt_time.getText()) < 0f
                    || Float.valueOf(txt_time.getText()) > 24f) {
                // time is incorrect
                JOptionPane.showMessageDialog(null, "志愿时长不正确！", "提示",
 JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_year.getText().trim().equals("") ||
 Integer.valueOf(txt_year.getText()) < 1000
                    || Integer.valueOf(txt_year.getText()) >= 10000) {
```

```
                // year is incorrect
                JOptionPane.showMessageDialog(null, "年不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_month.getText().trim().equals("") ||
Integer.valueOf(txt_month.getText()) < 0
                    || Integer.valueOf(txt_month.getText()) > 12) {
                // month is incorrect
                JOptionPane.showMessageDialog(null, "月不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else if (txt_day.getText().trim().equals("") ||
Integer.valueOf(txt_day.getText()) < 0
                    || Integer.valueOf(txt_day.getText()) > 31) {
                // day is incorrect
                JOptionPane.showMessageDialog(null, "日不正确！", "提示",
JOptionPane.ERROR_MESSAGE);
                return false;
            } else
                return true;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "输入有误，请检查！", "提示",
JOptionPane.ERROR_MESSAGE);
            return false;

        }
    }
}

class CreateTree extends Thread {
    DefaultMutableTreeNode jt;
    JTree tree;

    public CreateTree(DefaultMutableTreeNode jt, JTree tree) {
        this.jt = jt;
        this.tree = tree;
        this.start();
    }

    public void run() {
        // 更新信息
        Volenteer vols;
        for (Iterator<Volenteer> vol = Data.group.vol.iterator(); vol.hasNext();) {
            vols = vol.next();
            // 加入学生节点
            DefaultMutableTreeNode node1 = new DefaultMutableTreeNode(vols);
            jt.add(node1);
            Activity act;
            for (Iterator<Activity> activity = vols.activities.iterator();
activity.hasNext();) {
                act = activity.next();
                DefaultMutableTreeNode node2 = new DefaultMutableTreeNode(act);
                node1.add(node2);
            }

        }
        // tree.setSelectionPath(new TreePath(jt.getPath()));
        tree.updateUI();

    }

}
```

DataSelectUI.java

```
package user.programUI;

import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.filechooser.FileFilter;

import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.util.CellRangeAddress;

import user.data.Data;
import data.Activity;
import data.Volenteer;

import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JButton;
import javax.swing.JFileChooser;

import java.awt.event.ActionListener;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStream;
```

```java
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Iterator;
import java.awt.event.ActionEvent;

public class DateSelectUI extends JFrame {
    private JTextField year1;
    private JTextField month1;
    private JTextField day1;
    private JTextField year2;
    private JTextField month2;
    private JTextField day2;

    public DateSelectUI() {
        setResizable(false);// 不可放大
        setBounds(100, 100, 277, 236);
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        getContentPane().setLayout(null);

        year1 = new JTextField();
        year1.setBounds(14, 38, 64, 24);
        getContentPane().add(year1);
        year1.setColumns(10);

        JLabel label = new JLabel("\u5E74");
        label.setBounds(80, 41, 15, 18);
        getContentPane().add(label);

        month1 = new JTextField();
        month1.setBounds(100, 38, 48, 24);
        getContentPane().add(month1);
        month1.setColumns(10);

        JLabel label_1 = new JLabel("\u6708");
        label_1.setBounds(148, 41, 15, 18);
        getContentPane().add(label_1);

        day1 = new JTextField();
        day1.setBounds(170, 38, 48, 24);
        getContentPane().add(day1);
        day1.setColumns(10);

        JLabel label_2 = new JLabel("\u65E5");
        label_2.setBounds(221, 41, 15, 18);
        getContentPane().add(label_2);
```

```
year2 = new JTextField();
year2.setColumns(10);
year2.setBounds(14, 99, 64, 24);
getContentPane().add(year2);

JLabel label_3 = new JLabel("\u5E74");
label_3.setBounds(80, 102, 15, 18);
getContentPane().add(label_3);

month2 = new JTextField();
month2.setColumns(10);
month2.setBounds(100, 99, 48, 24);
getContentPane().add(month2);

JLabel label_4 = new JLabel("\u6708");
label_4.setBounds(148, 102, 15, 18);
getContentPane().add(label_4);

day2 = new JTextField();
day2.setColumns(10);
day2.setBounds(170, 99, 48, 24);
getContentPane().add(day2);

JLabel label_5 = new JLabel("\u65E5");
label_5.setBounds(221, 102, 15, 18);
getContentPane().add(label_5);

JLabel label_6 = new JLabel("\u4ECE");
label_6.setBounds(23, 13, 72, 18);
getContentPane().add(label_6);

JLabel label_7 = new JLabel("\u5230");
label_7.setBounds(24, 75, 72, 18);
getContentPane().add(label_7);

JButton button = new JButton("\u786E\u8BA4");
button.addActionListener(new ActionListener() {
    @SuppressWarnings("deprecation")
    public void actionPerformed(ActionEvent arg0) {
        // 判断是否正常
        int yearint1, yearint2, monthint1, monthint2, dayint1, dayint2;
        Date date1, date2;
        try {
            yearint1 = Integer.valueOf(year1.getText());
            yearint2 = Integer.valueOf(year2.getText());
```

```
                    monthint1 = Integer.valueOf(month1.getText());
                    monthint2 = Integer.valueOf(month2.getText());
                    dayint1 = Integer.valueOf(day1.getText());
                    dayint2 = Integer.valueOf(day2.getText());
                    date1 = new ~~Date~~(yearint1, monthint1, dayint1);
                    date2 = new ~~Date~~(yearint2, monthint2, dayint2);

                    if (date1.before(date2)) {
                        JFileChooser jfc = new JFileChooser();
                        jfc.setFileFilter(new MyFileFilter(".xls", "Excel格式"));
                        jfc.setFileSelectionMode(JFileChooser.FILES_ONLY);
                        int result = jfc.showSaveDialog(null);

                        if (result == JFileChooser.APPROVE_OPTION) {
                            File file = jfc.getSelectedFile(); // 获得文件名 // 获得被
选中的过滤器
                            MyFileFilter filter = (MyFileFilter) jfc.getFileFilter();
// 获得过滤器的扩展名
                            String ends = filter.getEnds();
                            File newFile = null;
                            if
(file.getAbsolutePath().toUpperCase().endsWith(ends.toUpperCase())) { // 如果文件是以
选定扩展名结束的，则使用原名
                                newFile = file;
                            } else { // 否则加上选定的扩展名
                                newFile = new File(file.getAbsolutePath() + ends);
                            } // 以下用 newFile 完成保存文件的操作
                            if (!newFile.exists()) {
                                new OutExcel(newFile, date1, date2, button);
                                button.setText("正在处理");
                                button.setEnabled(false);
                            } else {
                                if (JOptionPane.showConfirmDialog(null, "文件已存在，是
否覆盖？", "提示",
                                        JOptionPane.WARNING_MESSAGE,
                                        JOptionPane.OK_CANCEL_OPTION) ==
JOptionPane.OK_OPTION) {
                                    new OutExcel(newFile, date1, date2, button);
                                    button.setText("正在处理");
                                    button.setEnabled(false);
                                }
                            }

                        }

                    } else
```

```
                         JOptionPane.showMessageDialog(null, "输入有误！", "提示",
JOptionPane.WARNING_MESSAGE);
                    } catch (Exception e) {
                        e.printStackTrace();
                        JOptionPane.showMessageDialog(null, "输入有误！", "提示",
JOptionPane.WARNING_MESSAGE);
                    }

                }
            });
            button.setBounds(148, 161, 109, 27);
            getContentPane().add(button);

            setVisible(true);
        }
}

class OutExcel extends Thread {
    File file;// Excel文件生成后存储的位置。
    Date date1, date2;
    JButton btn;

    public OutExcel(File file, Date date1, Date date2, JButton btn) {
        this.file = file;
        this.date1 = date1;
        this.date2 = date2;
        this.btn = btn;
        this.start();
    }

    @SuppressWarnings("deprecation")
    public void run() {
        HSSFWorkbook wb = new HSSFWorkbook();
        // 生成一个sheet1
        HSSFSheet sheet = wb.createSheet("志愿者信息");
        // 为sheet1生成第一行，用于放表头信息
        sheet.setColumnWidth(2, 4000);// 设置列宽
        sheet.setColumnWidth(4, 6000);// 设置列宽

        CellRangeAddress cra = new CellRangeAddress(0, 0, 0, 6);
        sheet.addMergedRegion(cra);
        HSSFRow row = sheet.createRow(0);
        HSSFCell cell = row.createCell(0);
        SimpleDateFormat dateFormat = new SimpleDateFormat("MM月dd日");
        System.out.println(date1);
        System.out.println(date2);
        cell.setCellValue(date1.~~getYear~~()+"年
"+dateFormat.format(date1)+"--"+date2.~~getYear~~()+"年"
        +dateFormat.format(date2)+" | 志愿活动统计表");

        row = sheet.createRow(1);
        // 第二行的第一个单元格的值为 '序号'
        cell = row.createCell(0);
        cell.setCellValue("序号");

        cell = row.createCell(1);
        cell.setCellValue("姓名");

        cell = row.createCell(2);
        cell.setCellValue("学号");

        cell = row.createCell(3);
        cell.setCellValue("性别");

        cell = row.createCell(4);
        cell.setCellValue("参加活动");

        cell = row.createCell(5);
        cell.setCellValue("志愿时长");

        cell = row.createCell(6);
        cell.setCellValue("参加时间");

        Volenteer vol = null;
        Date date;
        int i = 1;
        // 获得List中的数据，并将数据放到Excel中
        for (Iterator<Volenteer> it = Data.group.vol.iterator(); it.hasNext();) {
            vol = it.next();
            Activity acts = null;
            // 判断是否是在时间范围内的
            date = new Date();
            for (Iterator<Activity> act = vol.activities.iterator(); act.hasNext();) {
                acts = act.next();
                date.~~setYear~~(acts.getYear());
                date.~~setMonth~~(acts.getMonth());
                date.~~setDate~~(acts.getDay());
                //System.out.println(date);
                if (date.before(date2) && date.after(date1)) {
                    // 数据每增加一行，表格就再生成一行
                    row = sheet.createRow(++i);
                    // 第一个单元格，放序号随着i的增加而增加
                    cell = row.createCell(0);
                    cell.setCellValue(i-1);
                    // 第二个单元格放name
                    cell = row.createCell(1);
                    cell.setCellValue(vol.name);
                    // 第三个单元格放id
                    cell = row.createCell(2);
                    cell.setCellValue(vol.IDnum);
                    // 第四个单元格放sex
                    cell = row.createCell(3);
                    cell.setCellValue(vol.sex);
                    // 第五个单元格放act
                    cell = row.createCell(4);
                    cell.setCellValue(acts.name);
                    // 第六个单元格放time
                    cell = row.createCell(5);
                    cell.setCellValue(acts.hour);
                    // 第七个单元格放活动时间
                    cell = row.createCell(6);
                    cell.setCellValue(acts.getYear() + "-" + acts.getMonth() + "-" +
acts.getDay());
                }

            }

            ByteArrayOutputStream os = new ByteArrayOutputStream();
            try {
                wb.write(os);
            } catch (IOException e) {
                e.printStackTrace();
            }

            byte[] content = os.toByteArray();

            OutputStream fos = null;

            try {
                fos = new FileOutputStream(file);

                fos.write(content);
                os.close();
                fos.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        btn.setText("确认");
        btn.setEnabled(true);
    }
}

class MyFileFilter extends FileFilter {
    String ends; // 文件后缀
    String description; // 文件描述文字

    public MyFileFilter(String ends, String description) { // 构造函数
        this.ends = ends; // 设置文件后缀
        this.description = description; // 设置文件描述文字
    }

    @Override // 只显示符合扩展名的文件，目录全部显示
    public boolean accept(File file) {
        if (file.isDirectory())
            return true;
        String fileName = file.getName();
        if (fileName.toUpperCase().endsWith(this.ends.toUpperCase()))
            return true;
        return false;
    }

    @Override // 返回这个扩展名过滤器的描述
    public String getDescription() {
        return this.description;
    }

    // 返回这个扩展名过滤器的扩展名
    public String getEnds() {
        return this.ends;
    }
}
```

ReadUI. java

```
package user.programUI;

import javax.swing.JFrame;
import javax.swing.JLabel;

import user.data.Data;

public class ReadUI extends JFrame {
    public ReadUI() {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    setTitle("\u8BFB\u53D6\u6587\u4EF6-\u5FD7\u613F\u8005\u7BA1\u7406\u7CFB\u7EDF")
;
        setBounds(100, 100, 308, 105);
        setResizable(false);
        getContentPane().setLayout(null);

        JLabel label = new JLabel("\u8BFB\u53D6\u6587\u4EF6\u4E2D...");
        label.setBounds(14, 23, 118, 18);
        getContentPane().add(label);
        setVisible(true);

        new ReadTread(this);
    }
}
class ReadTread extends Thread{
    JFrame jf;
    public ReadTread(JFrame jf) {
        this.jf=jf;
        this.start();
    }
    public void run() {
        while(true) {
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            if(Data.Groupready) {
                //group read ready
                jf.dispose();
            }
        }
    }
}
```

VolMannageUI.java

```
package user.programUI;

import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;
import data.*;
import user.data.Data;

import javax.swing.*;

import java.util.Iterator;

import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class VolManageUI extends JPanel {
    /**
     * 
     */
    private static final long serialVersionUID = 1L;

    private JTextArea txt_info;
    private JButton btn_modify;
    private JButton btn_delete;

    public VolManageUI() {

         setLayout(null);
         setBounds(0, 0, 575, 498);

        setLayout(null);

        JScrollPane scrollPane = new JScrollPane();
        scrollPane.setBounds(14, 13, 212, 460);
         add(scrollPane);

        DefaultMutableTreeNode top = new DefaultMutableTreeNode(Data.group);// 根节点
加入

        JTree tree_vol = new JTree(top);
        scrollPane.setViewportView(tree_vol);

        // 开始读取树结构
        new TreeShowThread(tree_vol, top);

        tree_vol.addTreeSelectionListener(new TreeSelectionListener() {

            @Override
            public void valueChanged(TreeSelectionEvent e) {
                // TODO Auto-generated method stub
                // 获取树的节点
                DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_vol.getLastSelectedPathComponent();
                // 判断是否为空树
                if (node == null) {
                    return;
                }
                DefaultMutableTreeNode parant = (DefaultMutableTreeNode)
node.getParent();
                Object nodeInfo = node.getUserObject();
                if (nodeInfo instanceof Volenteer) {
                    // 叶子节点,vol
                    // enable btn
                    btn_delete.setEnabled(true);
                    btn_modify.setEnabled(true);

                    Volenteer vol = (Volenteer) nodeInfo;
                    // 显示个人信息
                    showVolInfo(vol, txt_info);
                } else if (nodeInfo instanceof Group) {
                    // 根节点,group
                    // disable btn
                    btn_delete.setEnabled(false);
                    btn_modify.setEnabled(false);
                    Group group = (Group) nodeInfo;
                    // 根节点，显示组织信息
                    showGroupInfo(group, txt_info);
                }

            }
        });

        JScrollPane scrollPane_1 = new JScrollPane();
        scrollPane_1.setBounds(238, 13, 308, 420);
         add(scrollPane_1);

        txt_info = new JTextArea();
        txt_info.setEditable(false);
        scrollPane_1.setViewportView(txt_info);

        btn_delete = new JButton("\u5220\u9664");
        btn_delete.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // delete
                if (JOptionPane.showConfirmDialog(null, "确认删除？删除不可恢复！", "警
告", JOptionPane.WARNING_MESSAGE,
                        JOptionPane.YES_NO_OPTION) == JOptionPane.YES_OPTION) {
                    // 获取树的节点
                    DefaultMutableTreeNode node = (DefaultMutableTreeNode)
tree_vol.getLastSelectedPathComponent();
                    // 判断是否为空树
                    if (node == null) {
                        return;
                    }
                    Object nodeInfo = node.getUserObject();
                    if (nodeInfo instanceof Volenteer) {
                        // 叶子节点,vol
                        Volenteer vol = (Volenteer) nodeInfo;
                        Data.group.vol.remove(vol);
                        top.remove(node);

                        tree_vol.updateUI();
                        JOptionPane.showMessageDialog(null, "成功删除", "提示",
JOptionPane.INFORMATION_MESSAGE);

                    } else if (nodeInfo instanceof Group) {
                        // 根节点,group
                        JOptionPane.showMessageDialog(null, "根节点不允许删除！", "提示
", JOptionPane.ERROR_MESSAGE);

                    }
                }
            }
        });
        btn_delete.setBounds(432, 446, 113, 27);
         add(btn_delete);

        btn_modify = new JButton("\u66F4\u6539\u4E2A\u4EBA\u4FE1\u606F");
        btn_modify.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                // modify
                new ModifyInfoUI();


            }
        });
        btn_modify.setBounds(240, 446, 131, 27);
         add(btn_modify);

        //first disable btn
        btn_delete.setEnabled(false);
        btn_modify.setEnabled(false);

        tree_vol.expandRow(0);// 首个展开

    }

    private void showGroupInfo(Group group, JTextArea infoArea) {
        infoArea.setText("");// 初始化
        infoArea.append("组织名：  " + group.name + "\n");
        infoArea.append("组织ID:  " + group.ID + "\n");
        infoArea.append("成员数：  " + group.vol.size() + "\n");

    }

    private void showVolInfo(Volenteer vol, JTextArea infoArea) {
        infoArea.setText("");// 初始化
        infoArea.append("姓名：  " + vol.name + "\n");
        infoArea.append("学号：  " + vol.IDnum + "\n");
        infoArea.append("性别：  " + vol.sex + "\n");
        infoArea.append("状态：  " + vol.state + "\n");
        infoArea.append("参加了" + vol.activities.size() + "个志愿项目\n");
        Activity act = null;
        float time = 0f;// 志愿时长
        int no = 0;
        for (Iterator<Activity> it = vol.activities.iterator(); it.hasNext();) {
            act = it.next();
            time += act.hour;
            infoArea.append(
                    (++no) + ".( " + act.getMonth() + "/" + act.getDay() + " ) " + act.name
+ "-" + act.hour + "小时\n");
        }
        infoArea.append("总计" + time + "小时");

    }
}

class TreeShowThread extends Thread {
    JTree tree;
    DefaultMutableTreeNode node;

    public TreeShowThread(JTree tree, DefaultMutableTreeNode node) {
        this.tree = tree;
        this.node = node;
        this.start();
    }
    public void run() {
        // create tree
        Volenteer vol = null;
        for (Iterator<Volenteer> it = Data.group.vol.iterator(); it.hasNext();) {
            vol = it.next();
            node.add(new DefaultMutableTreeNode(vol));
        }
        tree.updateUI();

    }
}
```

SaveUI.java

```
package user.programUI;

import javax.swing.JFrame;
import javax.swing.JProgressBar;

import user.data.Data;
import user.process.WriteData;

import javax.swing.JLabel;
import java.awt.Color;

public class SaveUI extends JFrame {
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private JProgressBar progressBar;

    /**
     *
     * @param statue 1-save and exit 2-save not exit
     *
     */
    public SaveUI(int statue) {
        setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        setBounds(100, 100, 440, 127);
        setResizable(false);

    setTitle("\u4FE1\u606F\u6B63\u5728\u4FDD\u5B58-\u5FD7\u613F\u8005\u7BA1\u7406\u
7CFB\u7EDF");
        getContentPane().setLayout(null);
        progressBar = new JProgressBar();
        progressBar.setForeground(Color.GREEN);
        progressBar.setBounds(14, 48, 404, 23);
        getContentPane().add(progressBar);

        JLabel label = new
JLabel("\u914D\u7F6E\u6587\u4EF6\u4EE5\u53CA\u4FE1\u606F\u6B63\u5728\u4FDD\u5B58...
");
        label.setBounds(14, 13, 212, 18);
        getContentPane().add(label);

        setVisible(true);
        new Saving(progressBar, statue,this);

    }

}

class Saving extends Thread{
    private JProgressBar progressBar;
    private int statue;//1-save and exit else-save not exit
    private JFrame jf;
    public Saving(JProgressBar progressBar,int statue,JFrame jf) {
        this.progressBar=progressBar;
        this.statue=statue;
        this.jf=jf;
        this.start();
    }
    public void run() {
        new WriteData(Data.groupID,1);
        new WriteData(Data.groupID,2);//save group
        new ProgressBar(progressBar);
        while (true) {
            try {
                Thread.sleep(10);
            } catch (InterruptedException e1) {
                // TODO Auto-generated catch block
                e1.printStackTrace();
            }
            if (Data.saveGroup&&Data.saveGroups) {// 完成写入
                progressBar.setValue(progressBar.getMaximum());
                try {
                    Thread.sleep(10);
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                e.printStackTrace();
            }
            break;
        }
    }

    // Saving is OK
    if(statue==1) System.exit(0);
    else jf.dispose();
  }
}

class ProgressBar {
    public ProgressBar(JProgressBar bar) {
        for (int i = bar.getMinimum(); i <= bar.getMaximum() - 5; i++) {
            try {
                Thread.sleep(10);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            bar.setValue(i);
        }
    }
}
```

SettingUI.java

```
package user.programUI;

import java.awt.Color;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.UIManager;
import javax.swing.border.LineBorder;
import javax.swing.border.TitledBorder;

import user.data.Data;
import javax.swing.JPasswordField;
import java.awt.event.ActionListener;
import java.io.File;
import java.util.Iterator;
import java.awt.event.ActionEvent;
import javax.swing.JTextArea;

public class SettingUI extends JPanel {
    private static final long serialVersionUID = 1L;
    JLabel txt_user;
    private JPasswordField pswOld;
    private JPasswordField pswNew;
    private JPasswordField pswConfirm;

    public SettingUI() {
        setBounds(0, 0, 575, 190);
        setLayout(null);

        JPanel panel = new JPanel();
        panel.setBorder(new TitledBorder(new LineBorder(new Color(0, 0, 0)),
"\u5BC6\u7801\u8BBE\u7F6E",
                TitledBorder.LEADING, TitledBorder.TOP, null, new Color(0, 0, 0)));
        panel.setBounds(14, 13, 279, 150);
        add(panel);
        panel.setLayout(null);

        JLabel label = new JLabel("\u539F\u5BC6\u7801");
        label.setBounds(14, 23, 72, 18);
        panel.add(label);

        JLabel label_1 = new JLabel("\u65B0\u5BC6\u7801");
        label_1.setBounds(14, 54, 72, 18);
        panel.add(label_1);

        JLabel label_2 = new JLabel("\u518D\u6B21\u8F93\u5165");
        label_2.setBounds(14, 85, 72, 18);
        panel.add(label_2);

        pswOld = new JPasswordField();
        pswOld.setBounds(91, 20, 149, 24);
        panel.add(pswOld);

        pswNew = new JPasswordField();
        pswNew.setBounds(91, 51, 149, 24);
        panel.add(pswNew);

        pswConfirm = new JPasswordField();
        pswConfirm.setBounds(91, 82, 149, 24);
        panel.add(pswConfirm);
        JButton button = new JButton("\u786E\u8BA4");
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                if (new String(pswOld.getPassword()).equals(Data.user.getPassword()))
                // correct
                {
                    if (new String(pswNew.getPassword()).equals(new
String(pswConfirm.getPassword())))
                    // setting
                    {
                        Data.user.setPsw(new String(pswNew.getPassword()));
                        JOptionPane.showMessageDialog(null, "设置成功");
                        pswOld.setText("");
                        pswNew.setText("");
                        pswConfirm.setText("");
                    } else {
                        pswNew.setText("");
                        pswConfirm.setText("");
                        JOptionPane.showMessageDialog(null, "两次密码不一致");
                    }
                } else {
                    pswOld.setText("");
                    pswNew.setText("");
                    pswConfirm.setText("");
                    JOptionPane.showMessageDialog(null, "原密码错误");
                }
            }
        });
        button.setBounds(180, 115, 89, 27);
        panel.add(button);

        txt_user = new JLabel("{user}");
        txt_user.setBounds(14, 116, 152, 18);
        panel.add(txt_user);

        JPanel panel_1 = new JPanel();
        panel_1.setBorder(new TitledBorder(new LineBorder(new Color(0, 0, 0)),
"\u50A8\u5B58\u4FE1\u606F", TitledBorder.LEADING, TitledBorder.TOP, null, null));
        panel_1.setBounds(297, 13, 255, 150);
        add(panel_1);
        panel_1.setLayout(null);

        JTextArea txt_info = new JTextArea();
        txt_info.setEditable(false);
        txt_info.setBounds(14, 24, 227, 113);
```

```java
        panel_1.add(txt_info);

        firstShow();
        new InfoThread(txt_info);
        setVisible(true);
    }

    /**
     * show frame setting
     */
    public void firstShow() {
        txt_user.setText("现在登陆的用户： " + Data.user.getUser());
    }
}
class InfoThread extends Thread{
    JTextArea txt_info;
    public InfoThread(JTextArea txt_info) {
        this.txt_info = txt_info;
        this.start();
    }
    public void run() {
        txt_info.setText("");
        txt_info.append("志愿者数据信息大小： ");
        File file=new File("group"+Data.groupID+".dat");
        if(file.length()>5*1024&&file.length()<5*1024*1024)
txt_info.append((file.length()/1024)+"KB");
        else if(file.length()>5*1024*1024)
txt_info.append((file.length()/1024/1024)+"MB");
        else txt_info.append(file.length()+"B");
        txt_info.append("\n");
        txt_info.append("组织数据信息大小： ");
        file=new File("groups.dat");
        if(file.length()>5*1024&&file.length()<5*1024*1024)
txt_info.append((file.length()/1024)+"KB");
        else if(file.length()>5*1024*1024)
txt_info.append((file.length()/1024/1024)+"MB");
        else txt_info.append(file.length()+"B");
        txt_info.append("\n");
    }
}
```

公用数据 data 块

Activity.java

```java
package data;
```

```
import java.io.Serializable;

public class Activity implements Serializable {// 志愿活动信息
    public String name;// 活动名字
    public float hour;// 志愿时长
    public Group group;// 组织活动组织
    public boolean checked;//是否审核
    Dates date;// 活动时间
    public Activity(){
        date=new Dates();
    }

    public int getYear() {
        return date.year;
    }

    public int getMonth() {
        return date.mouth;
    }

    public int getDay() {
        return date.day;
    }

    public void setYear(int i) {
        date.year = i;
    }

    public void setMonth(int i) {
        date.mouth = i;
    }

    public void setDay(int i) {
        date.day = i;
    }

    public String toString() {
        return name + "-" + hour;
    }

}

class Dates implements Serializable{// 日期
    int year;
    int mouth;
    int day;
}
```

Volunteer. java

```
package data;

import java.io.Serializable;
import java.util.ArrayList;

public class Volenteer implements Serializable {//志愿者信息，存放志愿者信息以及志愿者参加的各类志愿活动
    public String name;//姓名
    public ArrayList<Activity> activities;//志愿活动集合
    public int IDnum;//学号
    public String sex;//性别
    public String state;//志愿者在组织的状态

    public Volenteer() {
        activities=new ArrayList<Activity>();
        state="临时";
    }

    public String toString() {
        return name+"-"+sex;
    }

}
```

User. java

```
package data;

import java.io.Serializable;


public class User implements Serializable{//用户信息
    String userName;//用户名
    private String password;//密码

    public User() {

    }
    public User(String str) {
        if(str.equals("Test")) {
            userName="admin";
            password="admin";
        }
    }



    public String getPassword() {
        return password;
    }
    public String getUser() {
        return userName;
    }
    public void setUser(String userName) {
        this.userName=userName;
    }
    public void setPsw(String password) {
        this.password=password;
    }
    public String toString() {
        return userName;
    }

}
```

Groups. java

```
package data;

import java.io.Serializable;
import java.util.ArrayList;

public class Groups implements Serializable{//组织用户
    public ArrayList<User> users;//用户
    public String name;//组织名称
    public int ID;//ID

    public Groups() {
        users=new ArrayList<User>();
    }
    public String toString() {
        return name;
    }
}
```

Group. java

```java
package data;

import java.io.Serializable;
import java.util.ArrayList;

public class Group implements Serializable{//组织，用于存放志愿者信息以及组织的信息
    public String name;
    public int ID;
    public ArrayList<Volenteer> vol;

    public Group() {
        vol=new ArrayList<Volenteer>();
    }

    public String toString() {
        return name+"-"+ID;
    }


}
```

NetFileWork.java

```java
package data;

import java.io.*;
import java.net.InetAddress;
import java.net.Socket;

import javax.swing.JOptionPane;

import server.DataPack;

public class NetFileWork {
    /**
     *
     * @param name
     *            file name
     * @param type
     *            1-server to client 2-client to server
     */
    public NetFileWork(String name, int type) throws Exception {
        int port = 9999;
        InetAddress address=InetAddress.getLocalHost();
        Socket socket = new Socket(address, port);
        ObjectOutputStream obout = new ObjectOutputStream(socket.getOutputStream());
        // first write info
        DataPack pack = new DataPack();
        pack.fileName = name;
        pack.type = type;
        obout.writeObject(pack);
        obout.flush();

        // then read or write
        if (type == 1) {
            // server to client
            // maybe server has not this file
            InputStream netIn = socket.getInputStream();
            ObjectInputStream in = new ObjectInputStream(new
BufferedInputStream(netIn));

            if (!in.readBoolean()) {
                System.err.println("Server has not the file");
            } else {
                File file = new File(name);
                if (!file.exists())// file not exit create it
                    file.createNewFile();
                FileOutputStream fos = new FileOutputStream(file);
                ObjectOutputStream out = new ObjectOutputStream(new
BufferedOutputStream(fos));
                out.writeObject(in.readObject());
                out.flush();

                // close all stream
                out.close();
                fos.close();
            }
            // close all stream
            in.close();
            netIn.close();
        } else {
            // client to server
            OutputStream netOut = socket.getOutputStream();
            ObjectOutputStream out = new ObjectOutputStream(new
BufferedOutputStream(netOut));

            FileInputStream fos = new FileInputStream(name);
            ObjectInputStream in = new ObjectInputStream(new BufferedInputStream(fos));

            out.writeObject(in.readObject());
            out.flush();
            //close all stream
            fos.close();
            in.close();
            out.close();
            netOut.close();
        }
        //close all stream
        obout.close();
        socket.close();
    }

}
```