# 最终代码及演示

## ——王海娟20161856123 + 王涵20161856124

1. PC端
2. **功能实现**
3. 管理员方面：

实现了管理员登录，及对数据库（目前还只有一个表）的数据管理（增删改查操作）。

1. 用户方面：

实现了用户登录、用户界面，及用户的私聊、群聊基本代码。实现了和服务器一对一对话。

1. 服务器方面：

实现了服务器与客户端的连接和传输消息。

1. **界面**

将管理和用户登录界面分开设计。

1. 欢迎界面

如图1所示，启动客户端，会出现选择身份的界面。

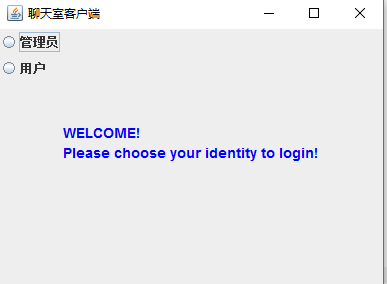


图 1

1. 用户界面

在欢迎界面选择用户，界面跳转到用户界面。如图3。其中，使用setResizable可设置界面无法改变大小；使用setUndecorated，设置是否含有 Java程序界面原有的标题栏 及 界面外框；使用setContentAreaFilled使得按钮是否呈现透明状态，此处不可使用setOpaque，此处若使用该方法无法使按钮透明化；setFocusPainted使得按钮不会有聚焦后的情况，显示在界面里，就是点击或焦点聚焦在透明按钮上时，其在按钮文字外框有一个外框。另外setForeground设置组件内文字等的颜色，而setBackground则设置组件的背景色。

另外，由于用户界面不含有java程序界面原有的标题栏等，其无法拖动，此处给整个界面加入了MouseMotionListener和MouseListener接口，使得界面可被拖动。

如图二所示，为设置界面出现的位置的代码，如此使得界面看似是界面中心启动的。

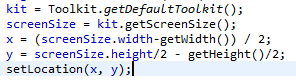


图 2

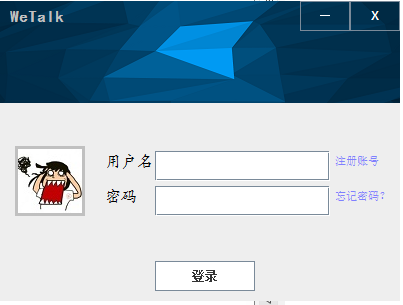


图 3

1. 管理员界面

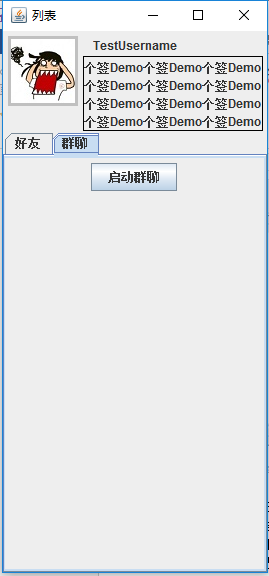
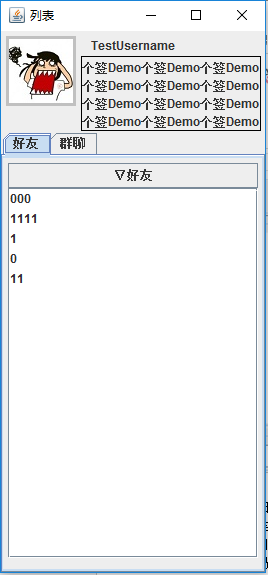
欢迎界面选择管理员，则界面跳转至管理员界面。如图4所示。管理员界面 较用户界面相对简单。



图 4

1. 用户界面

如下图所示，好友选项卡中显示的是数据库中的所有用户，双击可打开与好友的聊天界面。群聊方面则是点击按钮启动。此界面文件为UserMainFrame.java，而好友选项卡内则使用了FriendPanel.java文件。



1. 管理员界面

如图5所示，是管理员登录后的界面，目前数据库中只含有UserInfo表，所以界面只显示此表。双击即可进入对表的管理界面（如图6）。

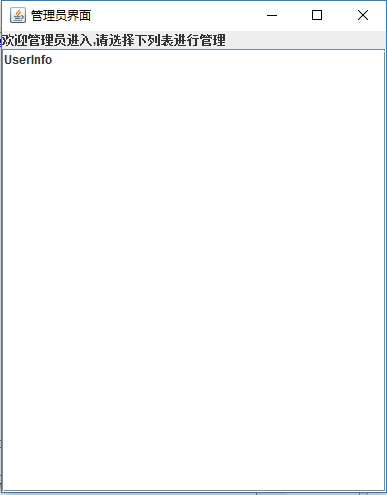


图 5

图6为管理界面，右侧文本域显示表的所有记录信息。由标题和文本域上的标签文字可知，此表为UserInfo。



图 6

1. 客户端界面

在用户界面内，点击 启动群聊 按钮，可启动如下图7界面（此时用户界面不消失）。双击好友选项卡中的好友，即可启动如下图8的好友私聊界面（此时用户界面不消失），可以看到消息区的文字，文字为服务器和客户端连接后发送的信息（暂时还未能实现两个用户间的聊天）。

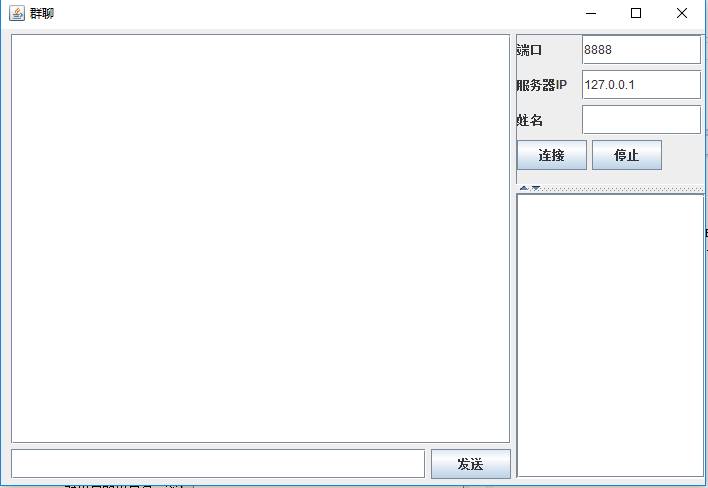


图 7

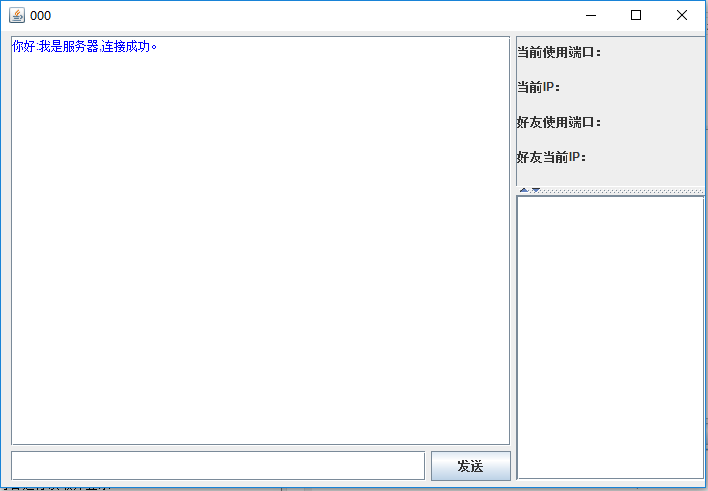


图 8

1. 服务器界面，可设置连入服务器的最大用户数。超过，则连接的客户端会提示连接失败

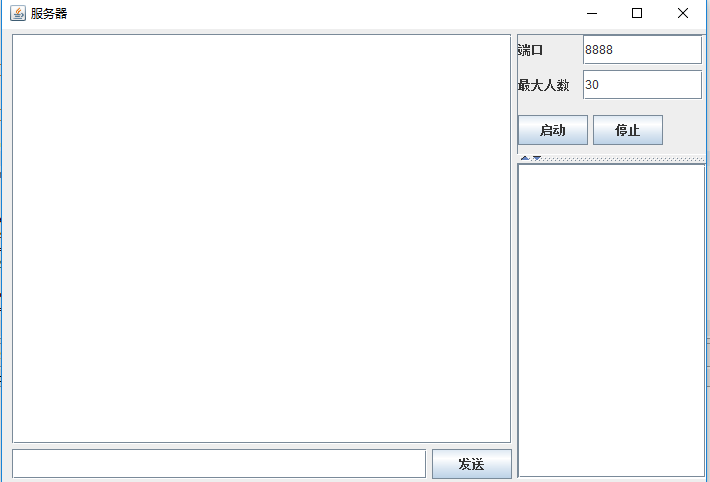


图 9

服务器界面如上图9所示，另外，服务器界面与上面客户端聊天界面使用了同一个父类界面（ChatFrame）。

1. **使用技术**
2. 数据库连接方式：JDBC，使用Access
3. 目前传送信息等阶段使用了多线程
4. 消息格式化：目前设定传输信息的固定格式，收发消息都会有对应的解封和封装消息。

代码如下图所示：

服务器：启动时通过serverStart方法启动，其中创建ClientThread对象，用作代表服务的每一位用户，并为其收发消息。如图所示。



图 10



图 11

客户端：



图 12

此外消息的封装形式如下图13所示：

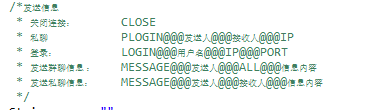


图 13

最前面是命令参数，各内容间用@@@做分隔符。

1. Web端
2. **功能实现**
3. 用户方面：

实现了用户登录，及群聊基本代码。实现了和服务器一对一会话。客户端网页只负责连接后，把收到的消息显示在聊天页面上。

1. 服务器方面：

实现了服务器与客户端的连接和传输消息。服务器负责连接到客户端后，转发给所有客户端页面。

1. **界面**
2. 登入界面

如图14所示，启动Tomcat后，输入网址<http://localhost:8080/ChatroomWeb/login.jsp>，进入登录界面。

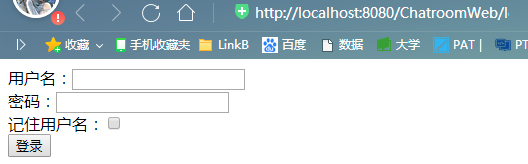


图 14

1. 群聊界面

如图15所示，登录两个用户sa和test，并聊天

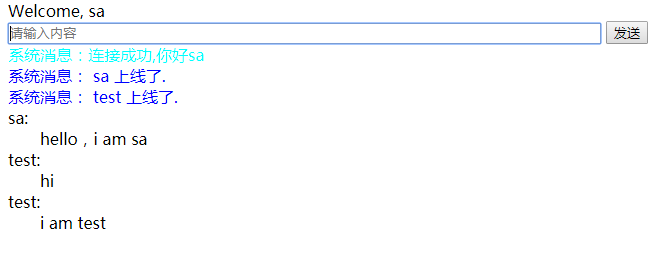


图 15

如果登录失败，则会提示游客身份登录，或重新登录如图16

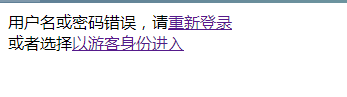


图 16

游客登录如图17所示，其后编号为随机生成



图 17

1. **使用技术**
2. 未使用数据库，通过UserManager文件，设定了三个默认用户
3. 通过servlet、jsp、js等实现页面及功能
4. 使用Websocket实现服务器和客户端连接
5. 曾尝试的扩展平台
6. PC端：具体介绍如上。
7. 微信小程序：暂时只做出了登录界面，聊天等功能还未完成。（因服务器设置等，暂时搁浅）
8. Web端：简易的网页群聊，使用了Tomcat做服务器，可以游客登入。
9. 展望
10. 微信小程序的Websocket实现
11. 解决PC端的管理员操作不人性化的特点。
12. 考虑增加udp连接（发送图片）
13. 还有余力的话，进一步完善界面和功能细节、修复隐藏Bug。
14. 同4，也考虑添加在线用户列表等。
15. 代码

具体代码见GitHub：https://github.com/Newbie-W/ChatRoomProject

PC端文件在ChatroomPC中，Web的在ChatroomWeb中。

其中，PC端，数据库管理相关操作在DBCon.java中；ChatroomPC.java为客户端的启动起点。剩余含Frame和Panel的为相关界面及界面中的操作。