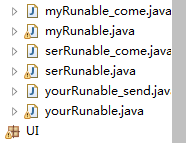
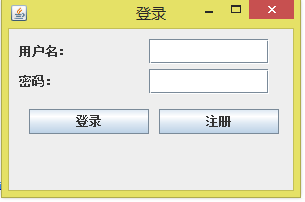
代码采用了分层的思想,Modal包下放的java文件是关于各个线程要执行的run方法所在的类以及有关数据库的操作的类(本来数据库操作的类应该放在另一个包中，但是给我忘了，感觉这样也行，就没有改)；UI包下放的类是关于UI界面的java文件和相应的事件监听器所在的类；

聊天线程部分分为我方发送线程myRunable.java，我方接收线程myRunable\_come.java，服务器监听我方收到信息后发送给他放的线程serRunable.java，以及服务器收到他方发送的信息并发送给我方的线程serRunable\_come.java，他方监听服务器收服务器发送的信息的线程yourRunable.java以及他方发送信息给服务器的线程yourRunable\_send.java；

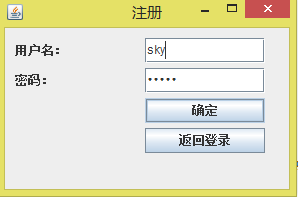


1.

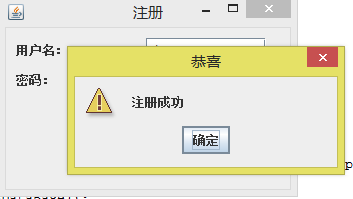
首先打开UI包下的LoginFrame.java文件，启动运行，利用java图形界面的有关代码设置，出现这样的界面，在出现界面前，调用相应的方法，已经创建了数据库和相应的表结构，



登录注册按钮，都已经设置了相应的监听器，点击 注册按钮，触发事件，到如下的注册页面，在文本框中输入相应的信息，



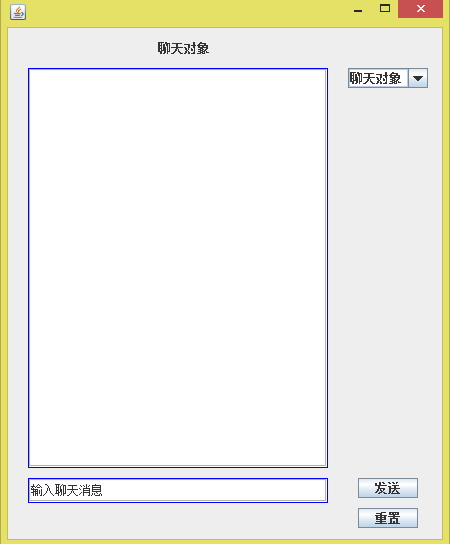
点击确定，触发事件，往数据库中插入文本框中书写的信息，如果插入成功，出现 如下的界面



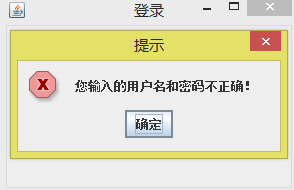
此时在数据库的user表中出现了一条关于此用户的用户名和密码的记录，



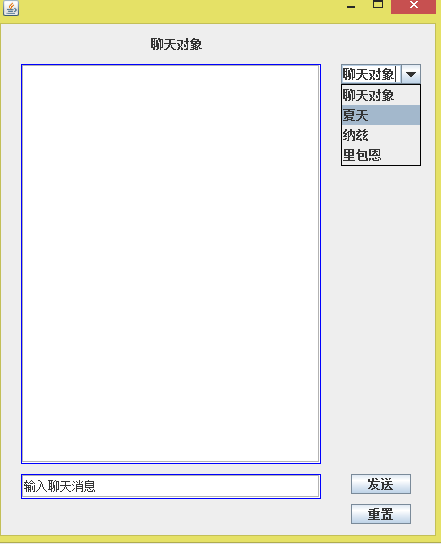
利用此用户名和密码进行登录，点击确定按钮，再点击 返回登录 按钮，重新触发事件new出登录窗口，在窗口中写相应的信息，点击登录按钮，触发事件，如果数据库user表中有相应的用户名和密码记录，进入触发事件，进入如下的界面，

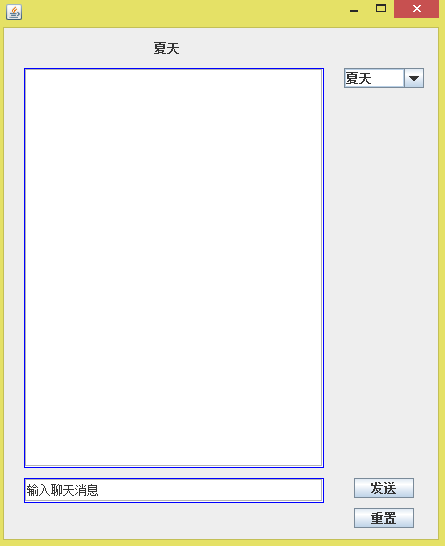


如果没有相应的信息记录，会报错，

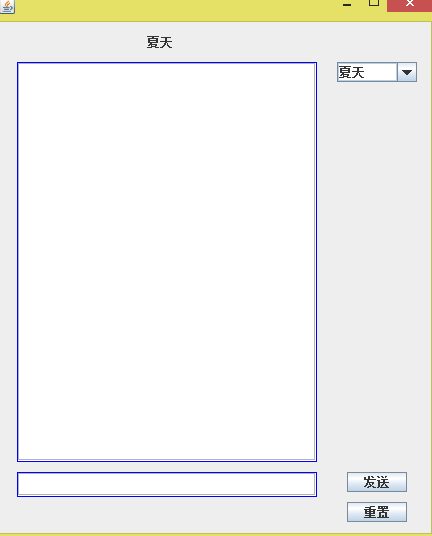


正确进入如下的界面后，点击相应的人员，不同的人员绑定了不同的IP，只要某个人员被选中，就会绑定不同的IP，以及聊天对象label的内容也会和下拉框中选择的信息一样，最重要的是，在选择人员完成后，触发的事件中包括，启动服务器监听我方的线程，以及他方监听服务器的线程，

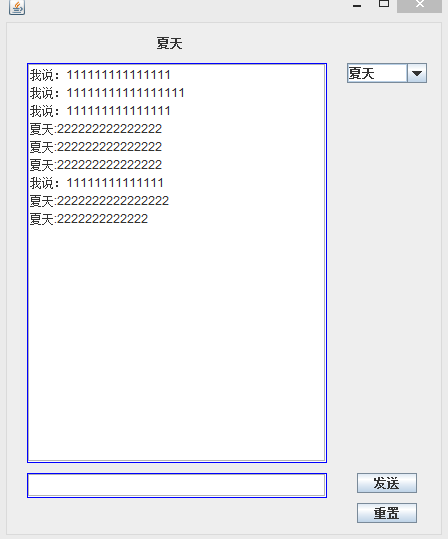


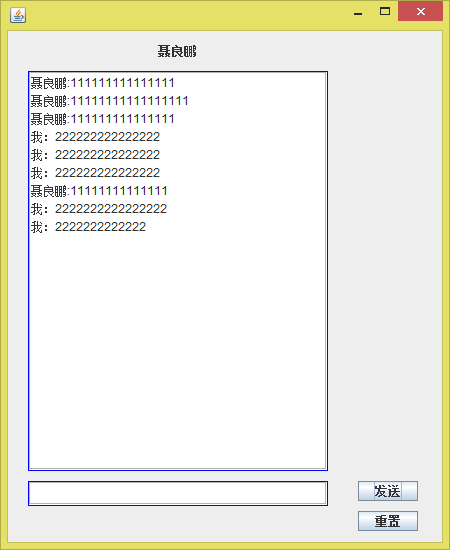


当鼠标定位到聊天输入框后，框中的信息就会清空，



在框中输入信息，点击发送按钮，触发事件，开启线程，将信息发送出去给服务器，服务器监听收到，再转发给他方的监听线程去收信息，展示信息在聊天框中，他方输入好信息后，点击发送按钮，并发启动三个线程，分别是他方发送线程，服务器接收发送线程我方接收线程，让线程并发执行，将信息返回，利用这样的启动线程的方法，可以实现我方在他方没有发的时候发多条信息，他方在我方没有发的时候也可以发多条信息，两方都可以不用等对方发送就能发送下一条信息，更符合类似QQ的功能；





大概就是这种情况，说明的不清楚的地方还请多多包涵；^\_^