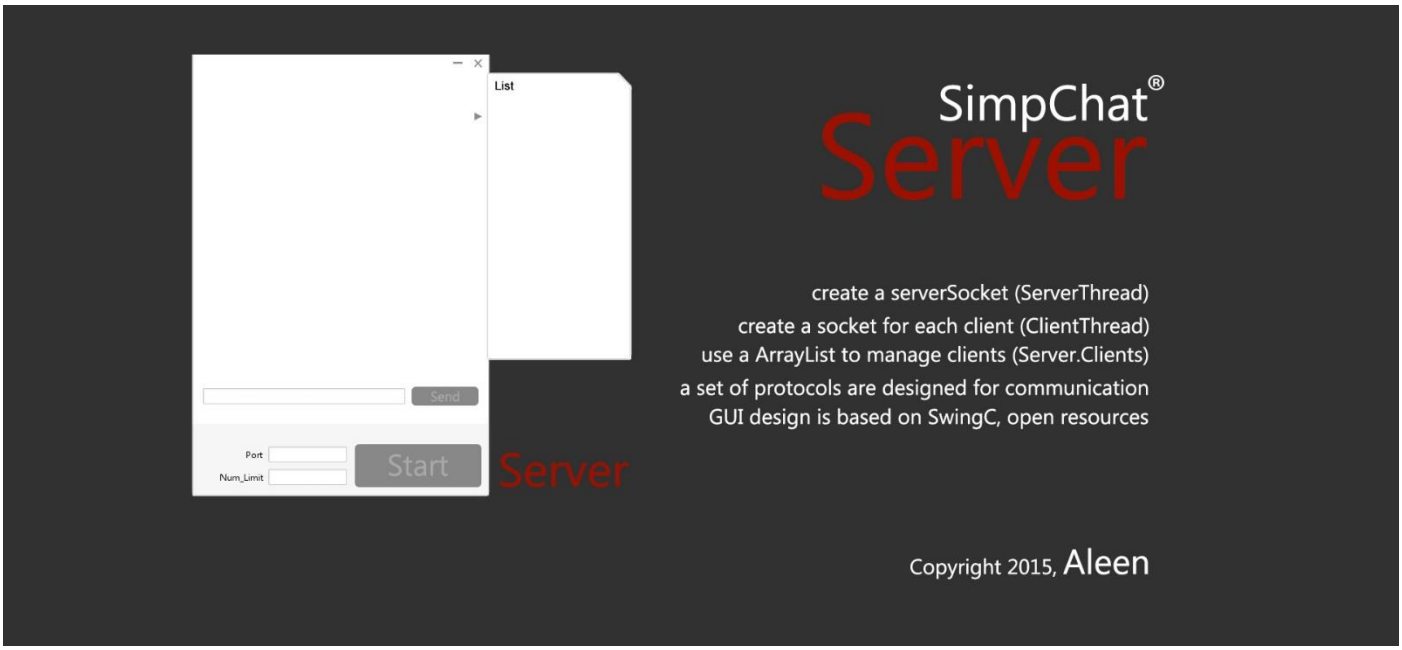
SimpChat

实验报告

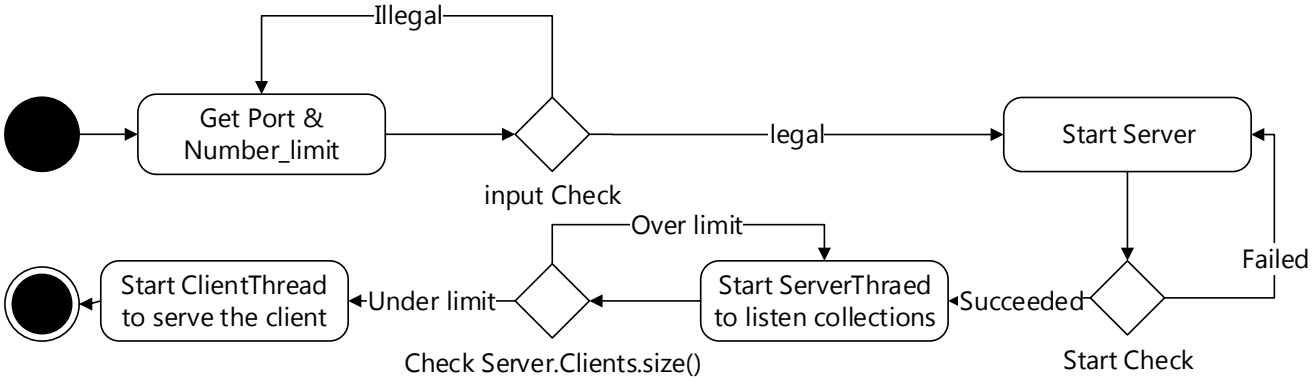
姓名：徐沛文
学号：13121215

SimpChat，是一款定制的聊天软件，它完成了私人聊天以及公共聊天等功能。它结构非常简单，总体被分为服务器端和客户端两部分。

1. Server



1.1 Start the server



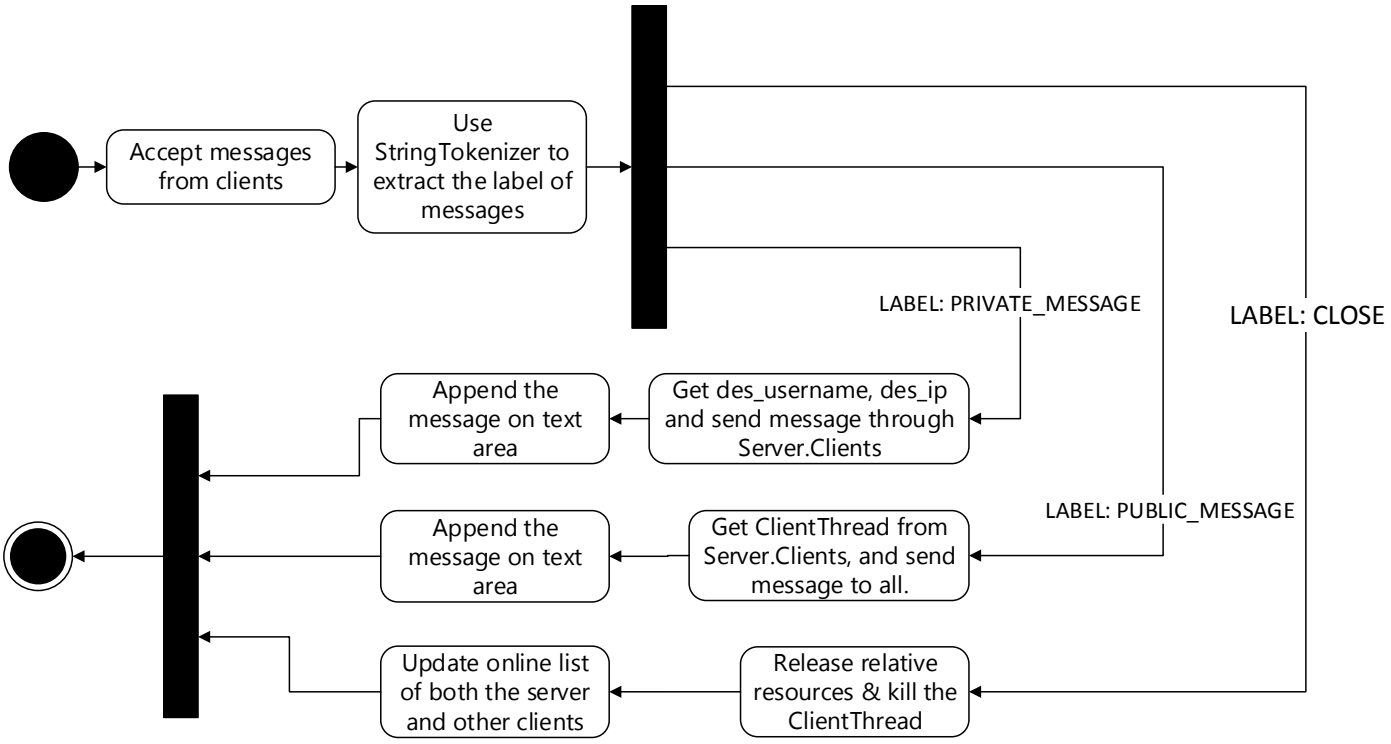
The activity diagram of starting a server

关于服务器端的启动，这里呈现了一个简单的界面和一个活动图。我们从活动图可以看到，在启动服务器之前，我们需要输入注册服务器端的端口号以及该服务器所限制的在线人数。当我们输入合法的端口号以及限制人数后，就可以其一个 ServerThread 来监听客户端的连接。

当有客户端连接该 port 的时候，服务器端首先会判断当前的一个 ArrayList, Server.Clients （用于管理在线的客户端线程 ClientThread）它的 size 大小是否达到人数限制的值。若超过则发送一个 CLOSE 标签的消息给客户端，让它被动关闭连接，而此时的服务器端会继续监听客户端的连接；若没有超过人数限制，则为该客户建立一个 ClientThread 并把它加入到 Server.Clients，更新在线用户列表。当一个 ClientThread 成功构建并运行后，它会给该客户端反馈当前在线人数信息，以及对 Server.Clients 中

的其他在线线程发送 ADD 标签的消息来通知其他客户端该用户上线的信息。ClientThread 运行过程不断监听客户发来的消息（CLOSE, PRIVATE_MESSAGE, PUBLIC_MESSAGE）， 服务器端则根据不同的标签做出相应的动作。

1.2 Handle messages from clients

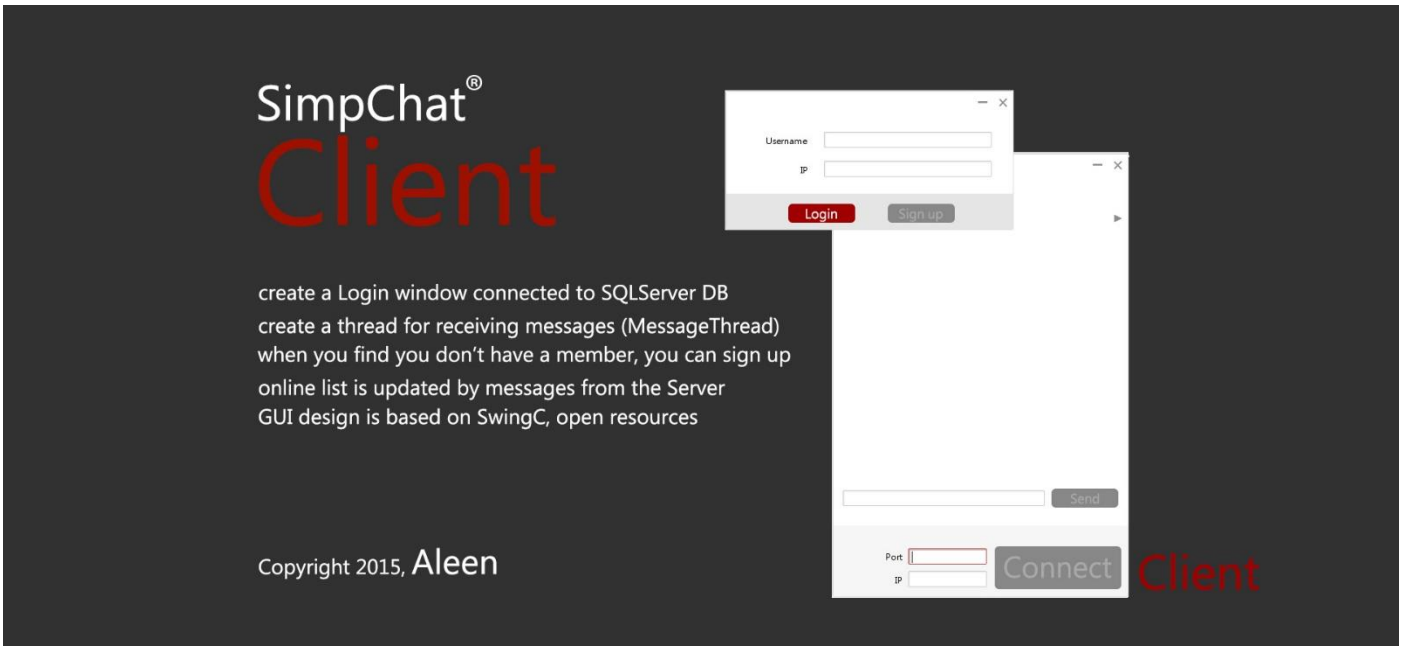


The activity diagram of handling messages

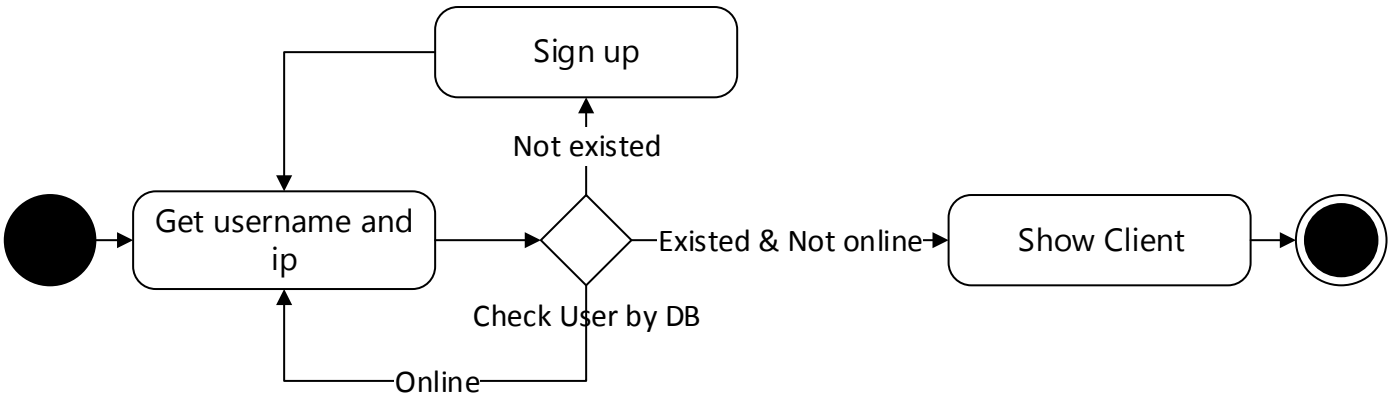
关于服务端的消息处理机制，我主要定义了三种 CLOSE、PRIVATE_MESSAGE 以及 PUBLIC_MESSAGE。

i. CLOSE: 当某客户端要断开连接的时候会发送 CLOSE 标签给服务器端。当服务器端接收到该标签时将会释放该客户端对应的流资源，以及服务该客户端的 ClientServer。然后，更新服务器端的在线人数列表以及发送 DELETE 标签给客户端去更新客户端的在线人数列表。
ii. PUBLIC_MESSAGE: 当客户端在没有选中某一用户的情况下，发送了某条消息。这时候客户端的 MessageThread 将会在消息前加上 PUBLIC_MESSAGE 标签，然后发送给服务器端。当服务器端接收到该标签时将会遍历 Server.Clients 列表，取出 ClientThread 中的输出流，把收到的消息分发给其他客户端（除了发送消息的该客户端）。
iii. PRIVATE_MESSAGE: 当客户端在选中某一用户的情况下，发送了某条消息。这时候客户端的 MessageThread 将会在消息前加上 PRIVATE_MESSAGE 标签，以及 des_username 和 des_ip ，然后发送给服务器端。当服务器端接收到该标签时将会遍历 Server.Clients 列表，找到对应 des_username 和 des_ip 的 ClientThread，把消息发送给改客户端。

2.Client



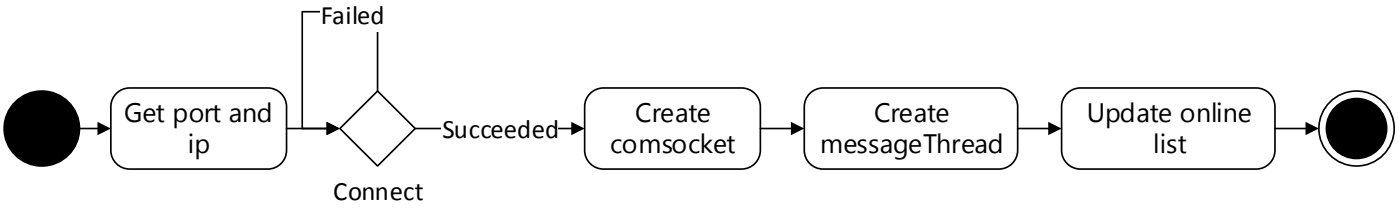
2.1 Login



The activity diagram of login

关于登陆窗口，我只设置了 username 和 IP 来识别用户。当用户登录输入后，点击 login。这时候，客户端将连接后台 DB 去查询对应的用户及其 IP 是否存在，若不存在则提示用户去 sign up；若存在则检查其是否在线，若在线则提示用户已在线；若不在线则成功登陆，并显示 Client 窗体。

2.2 Connect to the Server



The activity diagram of connecting to the server

当客户端输入好 port 以及 ip，连接的时候客户端将会创建 Socket (comsocket)，然后创建 messageThread 用于接收消息，最后则是更新在线的用户列表。

2.3 Handle messages

与服务器一样，客户端设计了四种相似的标签 CLOSE、ADD、DELETE 以及 MESSAGE。

i. CLOSE: 当客户端收到 CLOSE 标签，表示服务器要客户端关闭连接，这时候客户端将被动关闭连接，释放相应资源并关闭 messageThread 线程。
ii. ADD: 当客户端收到 ADD 标签，表示有新用户上线，这时候将根据紧接 ADD 标签的 username 以及 ip 更新在线列表。
iii. DELETE: 当客户端收到 DELETE 标签，表示有用户下线，这时候将根据紧接 DELETE 标签的 username 以及 ip 更新在线列表。
iv. MESSAGE: 当客户端收到 MESSAGE 标签，表示服务器发来了消息，这时候将把收到的消息输出到 Text Area 上。

More details: http://aleen42.github.io/SimpChat/