前端（后端的功能中还有些没有写在前端中的功能，最好是在看下后端的功能要求）：

1. 登陆界面
   1. 判空
   2. 判断输入的ID的长度（＜＝15位）
   3. 判断输入密码长度（＜＝２０位）
   4. 对不存在的ID的错误的处理

２.注册界面

ａ）判空

ｂ）判断各种输入数值的长度的正确性

ｃ）该给出下拉框的地方要有（地址，出生年月）

３.聊天界面：

ａ）使用web Socket与后端实时沟通，使用ｊｓｏｎ封装数据

ｂ）获取后端发送的数据并及时处理

ｃ）好友发送消息过来要有提示

ｄ）其他要求的功能

４.查询界面

ａ）对获取的数据进行分页（如果数据量较大）

ｂ）查询的其他基本功能

５.下载聊天记录界面

ａ）下载时最好是有提示选择下载的位置，最好是有进度条，下载完成后有提示

６.页面尽可能好看，用户体验性尽可能好

7.前端在用户更改聊天对象时要将聊天对象的id通过websocket发送给后端（消息种类为“5”）

8.当用户查看消息时，前端使用ajax向后端获取消息，后端返回json字符串，前端将其对象化后取出数据，再显示再对应界面上

后端：

1. 使用spring boot＋mybatis，实时聊天使用web Socket协议，与前端的数据传输使用json
2. 注册，登录，注销功能
   1. 注册时将用户信息分别加入usertable表和userInfo表，并将userTable表中的userstatus置为1，之后将需要的信息传递给登录有关的servlet
   2. 登录时先获取该用户的全部好友以及他们的状态封装成一个类的list<friend>(其中friend含有id,name,status三个属性)，之后查看是否有未处理的消息，如果有则获取发送人的id与消息数量，将这些信息封装成一个类，之后在查看是否有好友请求，如果有则获取数量，最后将所有的数据封装成json存在session中，跳转到登陆后的页面，由前端获取session中的值并输出（前端可以尝试使用window.onload+ajax获取session中的值并做处理）
   3. 注销时，将有关的session值置移除，将用户当前状态置为0，关闭所有的流。
3. 消息实时发送与接受功能
   1. 登录时前端将调用websocket有关方法，后端接受到开启请求之后，将与该用户链接的session存入map中，key为该用户的id
   2. Websocket将用户当前正在聊天的对象的id发送到后端
   3. 当接受到用户发送的消息的json时，从中分析出目标对象的id、发送方的id与信息，若在map中存在该用户（即该用户在线），则查看该用户当前聊天是否为发送者，如果是则将消息直接发送给客户端，如果不是，则将消息存入未处理消息表，将用户的id发送给目标客户端
4. 查询功能
   1. 获取接收到的ID，如果不存在则返回空，如果存在则返回结果
5. 消息的下载功能
   1. 在聊天框中添加一个可伸缩的侧边栏，当用户点开后，在每一个聊天记录中都存在一个复选框，当用户选择了并点击了侧边栏下方的下载按钮后，弹出选择存储位置的消息，选择存储位置后，将时间戳的集合与用户id，好友id发送给后端，后端查询数据库后将文件下载到指定位置（暂定如此）
6. 添加好友并发送验证消息（可多次发送）
   1. 使用websocket，该消息的类型为添加好友，将类型用websocket发送到目标客户端，请求信息放入未处理好友请求中，状态设置为0
   2. 如果对象客户端点击了查看，则将所有好友请求的详细信息发送给目标客户端
   3. 如果对象客户端点击了同意，则客户端刷新当前的好友表，并将同意的用户的id与分组发送给后端，后端获取后将未处理好友请求表中状态改为1，将该好友加入双方的好友表中，如果拒绝，则将状态改为2
   4. 请求方在点击好友申请界面时，可以查看到当前的申请状态