

☐ **进入App软件，MainActivity中的status()方法会校验此设备的登陆状态，若此设备的登陆状态为未登录：（电话号码从未登录过）**

- 点击UserInfoFragment中的登录框，校验登录标志islogin，发现未登录；
- 此时弹出发送验证码界面PhoneDialogFragment，填写手机号，将手机号和phoneid发送至服务端；
- 服务端校验手机号，UserImpl中的sendcode()会查找数据库，发现数据库中无对应手机号的数据，说明该手机号此前从未登录过；
- 此时服务器发送验证码，在app中接收到验证码，填写正确后将验证码、电话号码和phoneid传至服务端；
- 此时服务器UserImpl中的validation()方法进行验证，由于此前未登录过，此时电话号码对应的数据为空，则将此电话号码和当前设备phoneid存入数据库中；
- 登录之后，点击登录框进行个人信息设置，此时校验登录标志islogin，发现已登录，UserInfoDisplayActivity的userinfo()方法会向服务器请求用户信息，由于第一次登陆，个人信息是默认的；
- 服务端的UserImpl中的sendInfo()校验若此时该手机号对应的用户已登录，则返回个人信息；
- 此时可以点击修改按钮，在app中修改个人信息，进入UserInfoSetActivity中，填写好个人信息后，将个人信息传至服务器；
- 服务端的UserImpl中的updateuserinfo()会校验登录状态，若为已登录，则修改个人信息，此时客户端刷新个人信息将会改变；
- 若用户点击用户退出，则将电话号码发送至客户端，服务端的UserImpl中的unregist()将用户的登陆状态更新为2退出登录。

☐ **进入App软件，MainActivity中的status()方法会校验此设备的登陆状态，若此电话号码以前已经注册过：（更换设备未退出或未更换但此前已经退出）（**phone电话号码才是代表用户身份的凭证**）**

☐ 若用户未更换设备，即phoneid未改变：

- 进入App软件，MainActivity中的status()方法会校验此设备的登陆状态，发现已登录过，则通过MainActivity中的getUserInfo()方法将用户信息传至UserInfoFragment中的onAttach()方法中，此时islogin标志位为true，表示已登录，后续步骤与上述一致。

☐ 若用户更换设备，即phoneid改变：

- 进入App软件，MainActivity中的status()方法会校验此设备的登陆状态，将phoneid发送至服务端；
- 服务端UserImpl中的withoutlogin()方法，校验phoneid发现未登录；
- 此时客户端点击登录框，将电话号码和phoneid发送至服务端；
- 服务端校验手机号，UserImpl中的sendcode()会查找数据库，发现数据库中已经存在电话号码对应的数据了，只是phoneid不同，说明此电话号码此前已经登录过了，校验数据的登陆标志为未登录，发送验证码至客户端；
- 客户端将验证码和当前phoneid（新的phoneid）和phone和验证码发送至服务端，服务端UserImpl中的validation()方法进行校验，验证码输入正确后，发现电话号码对应的数据不为空且登录状态为未登录，则更新登录状态为已登录，并且将数据中的phoneid更新为当前设备phoneid，并将订单表中的订单号也更新成当前设备phoneid对应的订单号；
- 后续步骤与上述一致。

逻辑：电话号phone是代表用户身份的象征，一个设备（phoneid）只能有一个用户为登录状态，其他用户必须都为登出状态，当一个手机号在一个设备登录过以后，若想在其他设备登录，则在服务器端直接更新该手机号对应的phoneid，保持phone对应最新的phoneid即可，订单表中的phoneid也要随之更改。

通过将AndroidId 和 Serial Number 组合的方式对手机进行唯一标识 (Token)

\* 将上述标识码加密成为转化成Md5码