**1.** **需求分析（撰写人：）**

即时消息软件是一种在后PC时代兴起的，以Internet网络为基础的，允许交互双方即时地传送文字、语音、视频等信息的网络应用软件。即时消息软件产生有着深刻的社会原因：人们都渴望社交，获得社会尊重、实现自我的需求，这正是即时消息软件风行的原动力，而物质文明的日益发达所带来的副作用，又使得人们习惯与周围的人保持距离，以致人们更愿意对陌生人敞开心扉，在网络中可以跨越年龄、身份、行业、地域的限制，达到人与人、人与信息之间的零距离交流。从这点上讲，即时消息系统的出现改变了人们的沟通方式和交友文化，大大拓展了个人生活交流的空间。

功能实现：

客户端功能：

注册功能：新用户的注册。

登录功能：建立与服务器的连接并登录，能显示登录错误信息。

好友功能：可以根据用户账号添加好友，将已添加的好友显示在用户列表中

聊天功能：与好友聊天，目前仅包括文字的传输。

聊天记录：能保存聊天记录，并能查看和导出聊天记录。

信息提示：显示未读消息

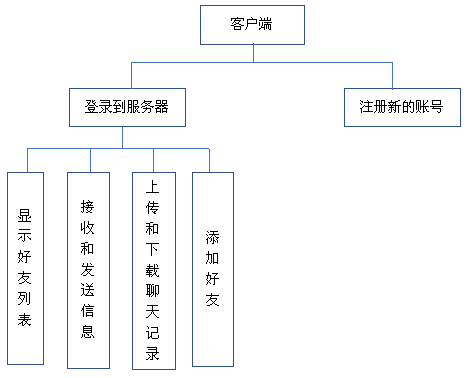


图1-1 需求分析图

服务器功能：

服务器向各个客户端发布系统消息，接受来自客户端的各种信息并分别处理。

登录信息：检查登陆信息是否正确，并向客户端返回登录信息。如果信息正确，就将在线用户发给该用户，同时在服务器端显示出来。

聊天信息：转发给消息指定的用户。

申请信息：分配用户号码，并保存该用户，同时将信息返回给客户端。

**2.** **概要设计（撰写人：）**

# **2.1抽象数据类型的定义**

聊天消息结构：网易云信提供的IMMessage类

联系人结构：联系人账号，联系人昵称，联系人ACCID

添加好友的消息的结构：系统消息的ID，消息的发送者

会话结构：会话对象的ID，最后的一次会话的内容，消息未读数，会话对象的Accid

用户信息结构：用户账号，用户名，密码，网易云信账号，网易云信密码，性别，联系电话

# **2.2主程序的流程**

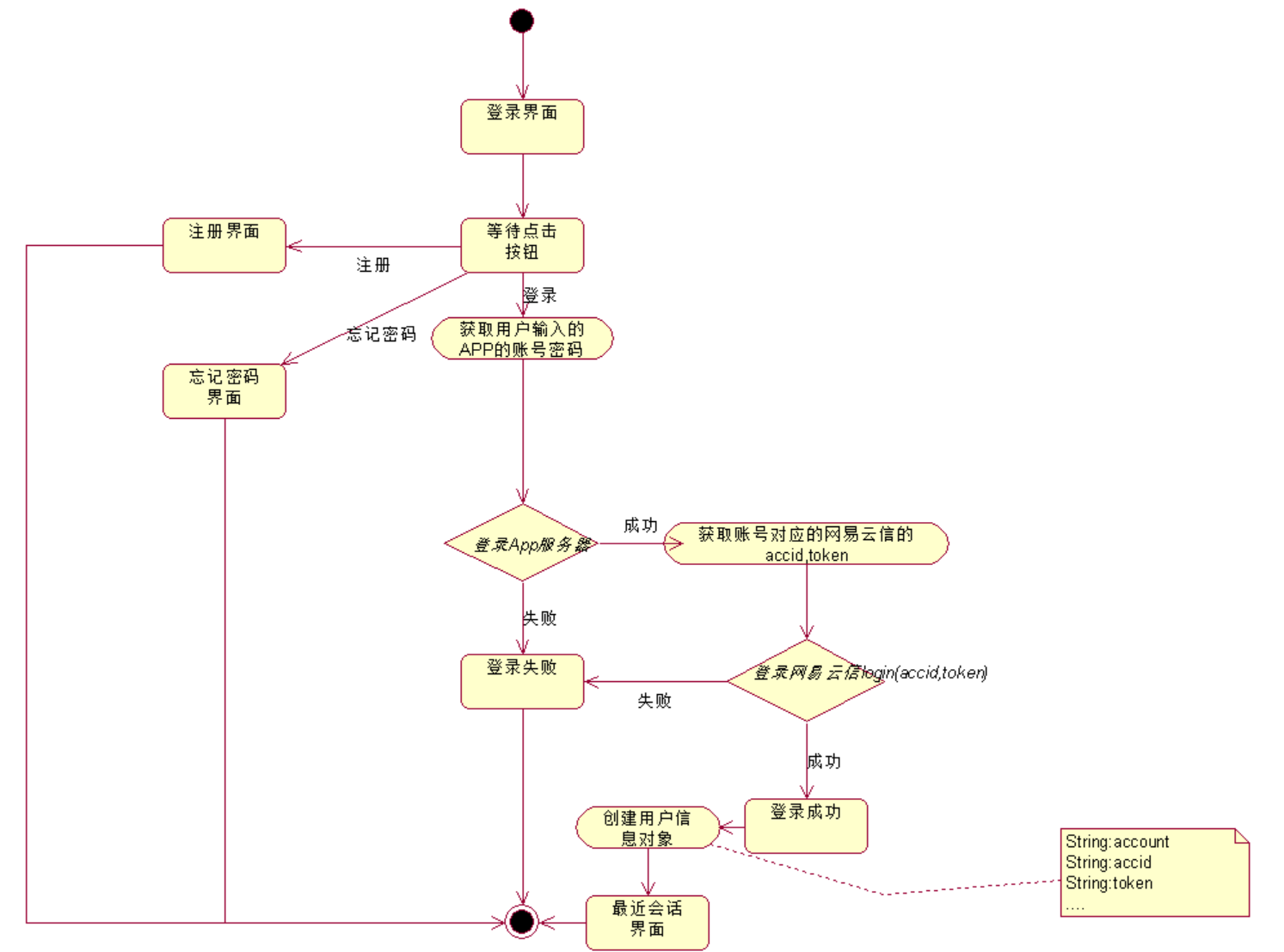


图2-1 登录流程图

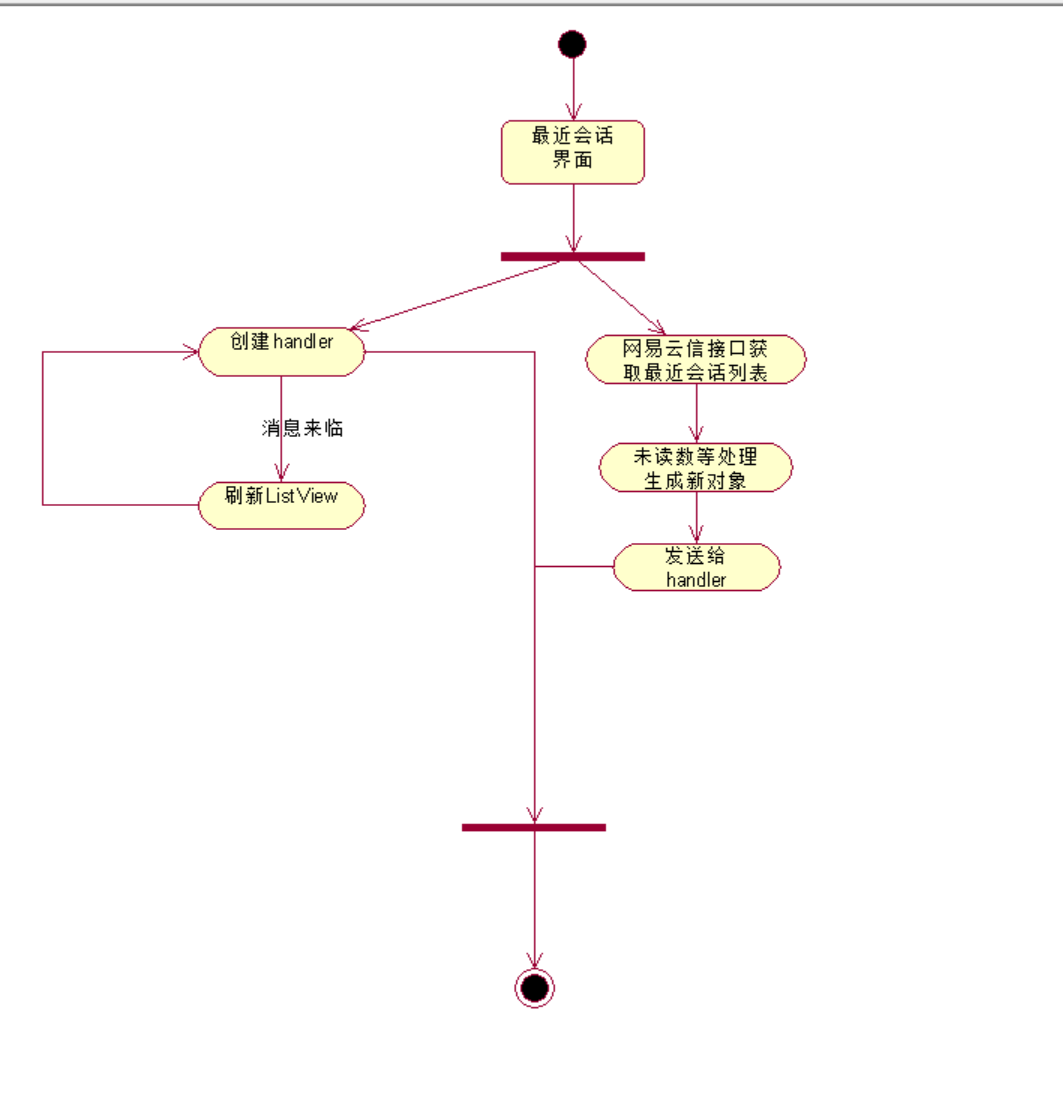


图2-2 获取最近会话列表流程图

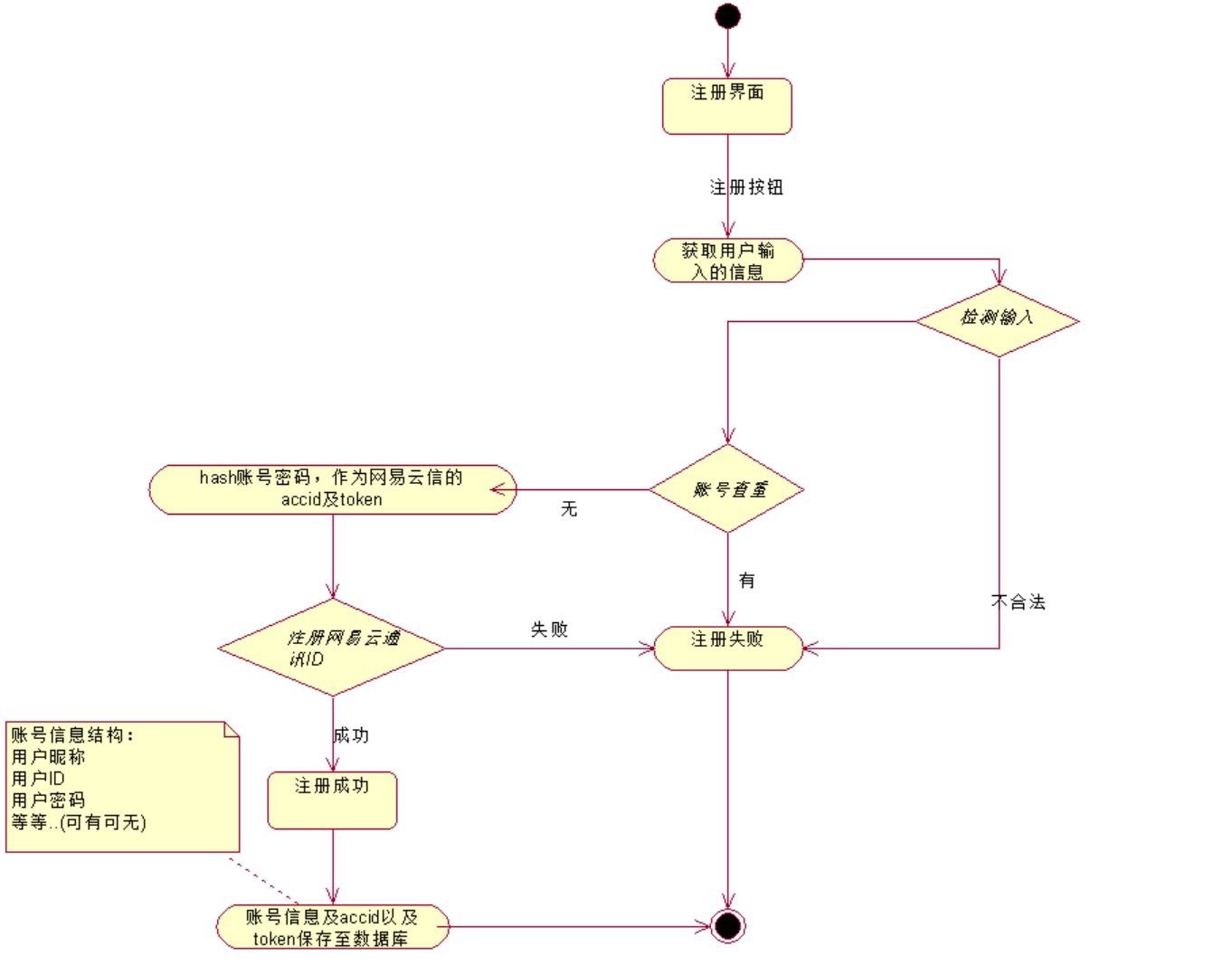


图2-3 注册流程图

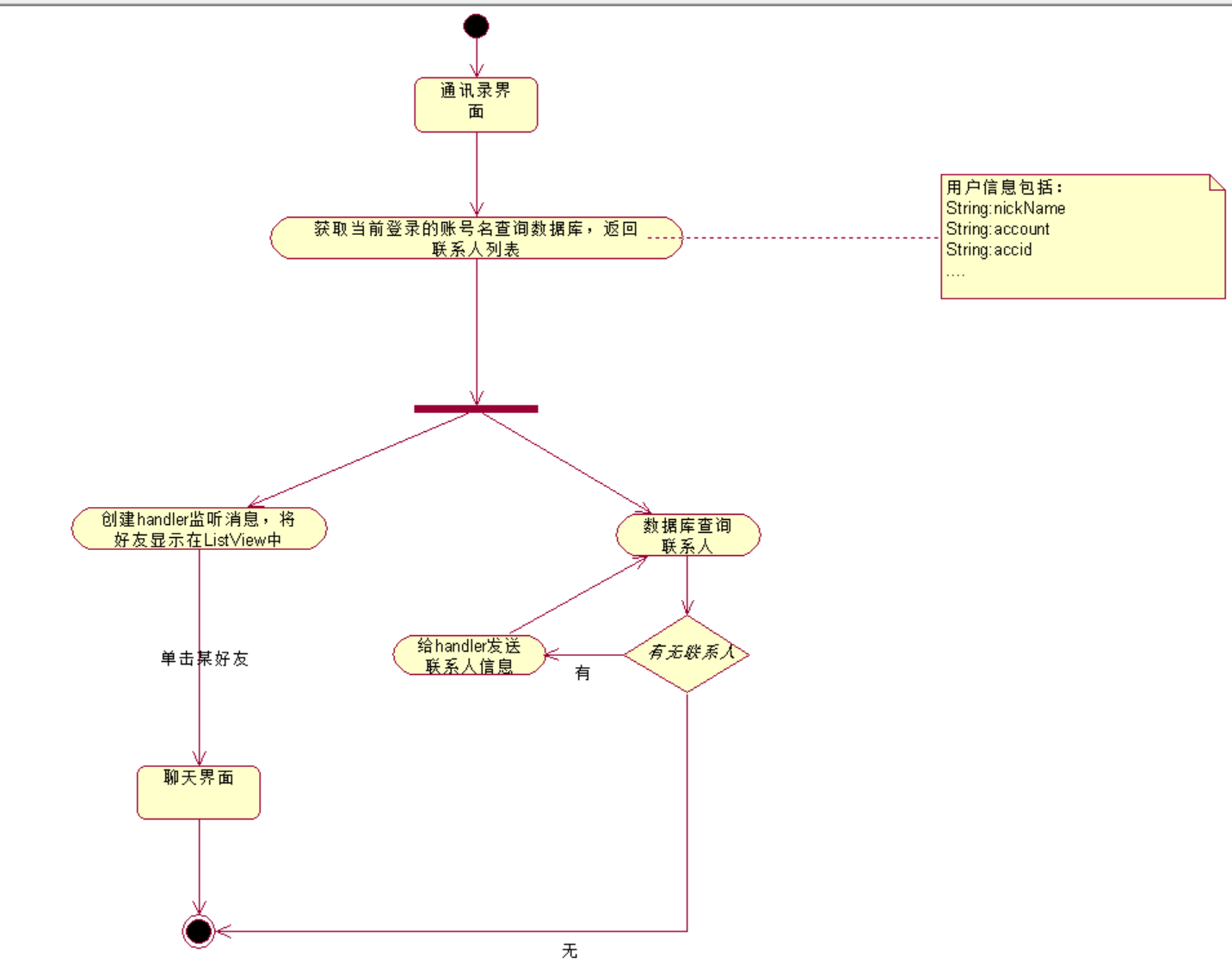


图2-4 获取通讯录流程图

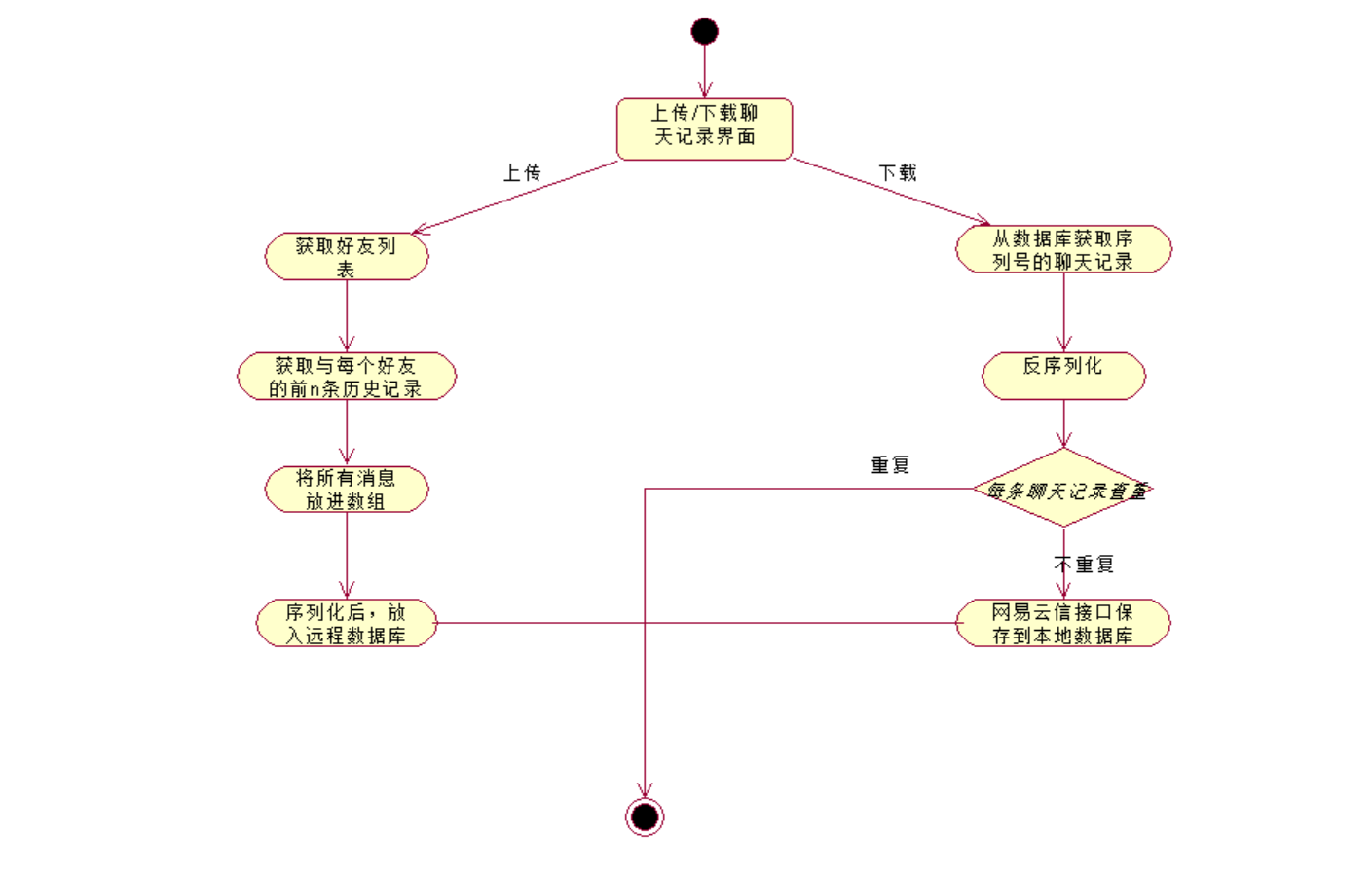


图2-5 上传/下载聊天记录流程图

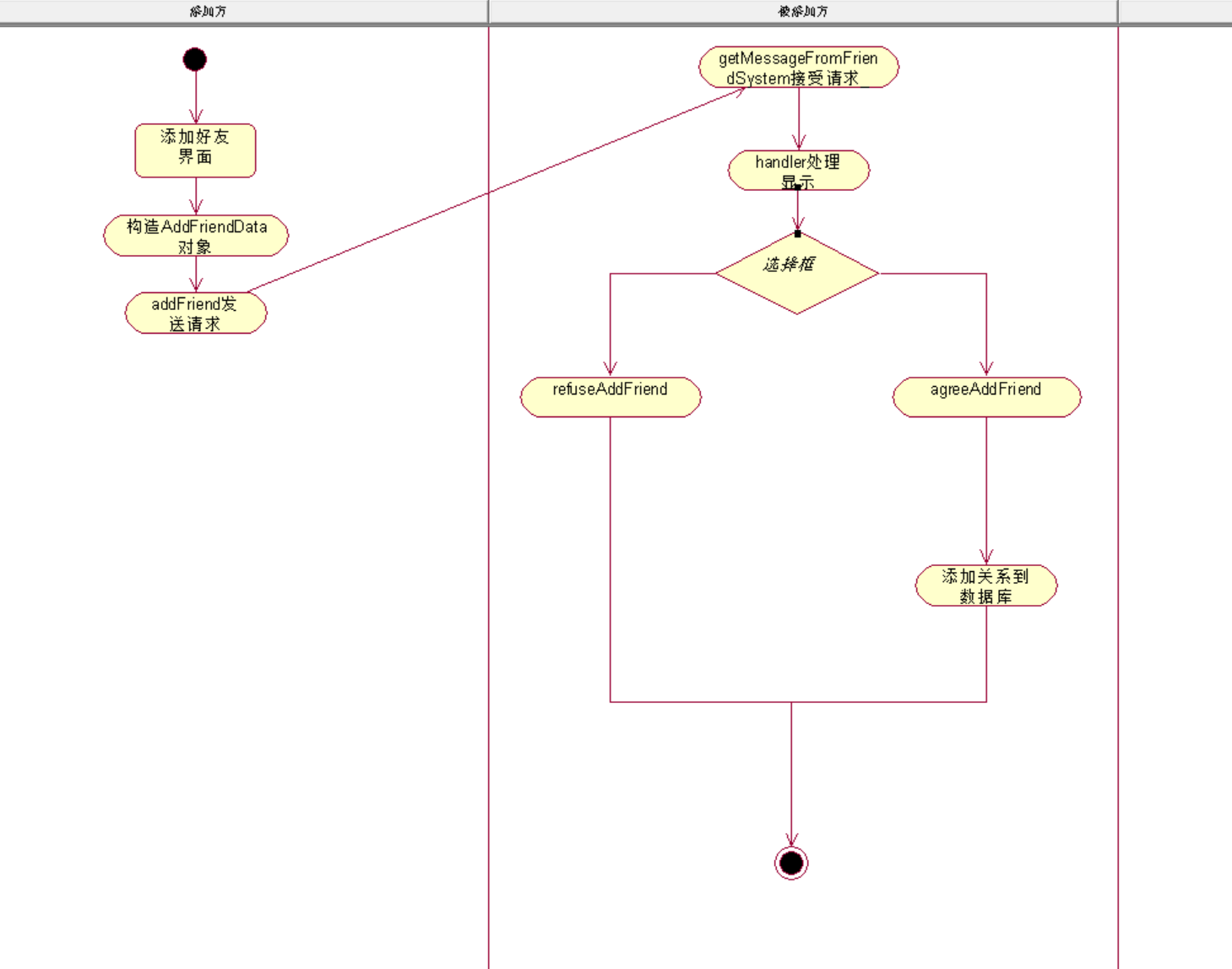


图2-6 添加好友流程图

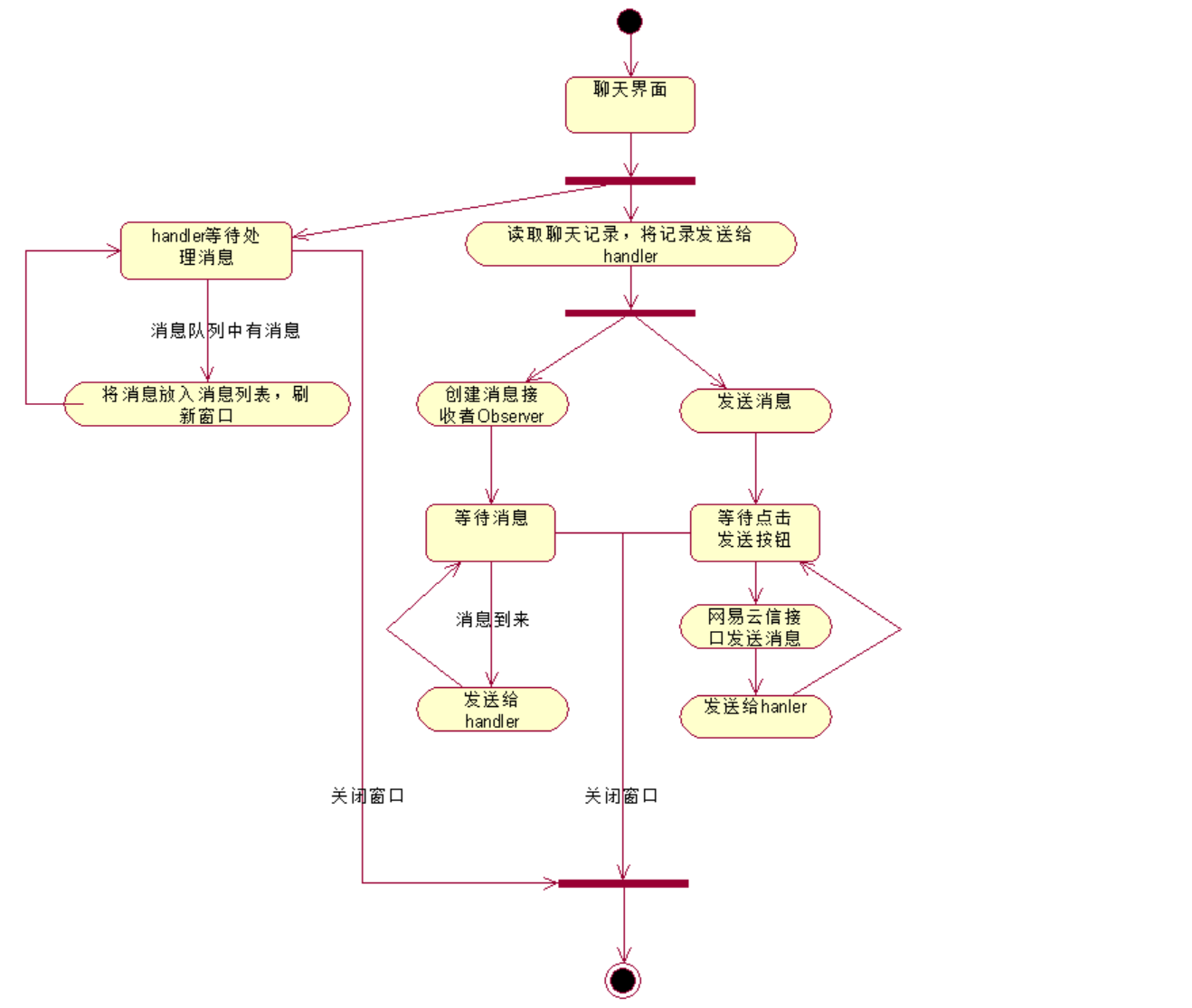


图2-7 聊天流程图

# **2.3各程序模块之间的层次(调用)关系**

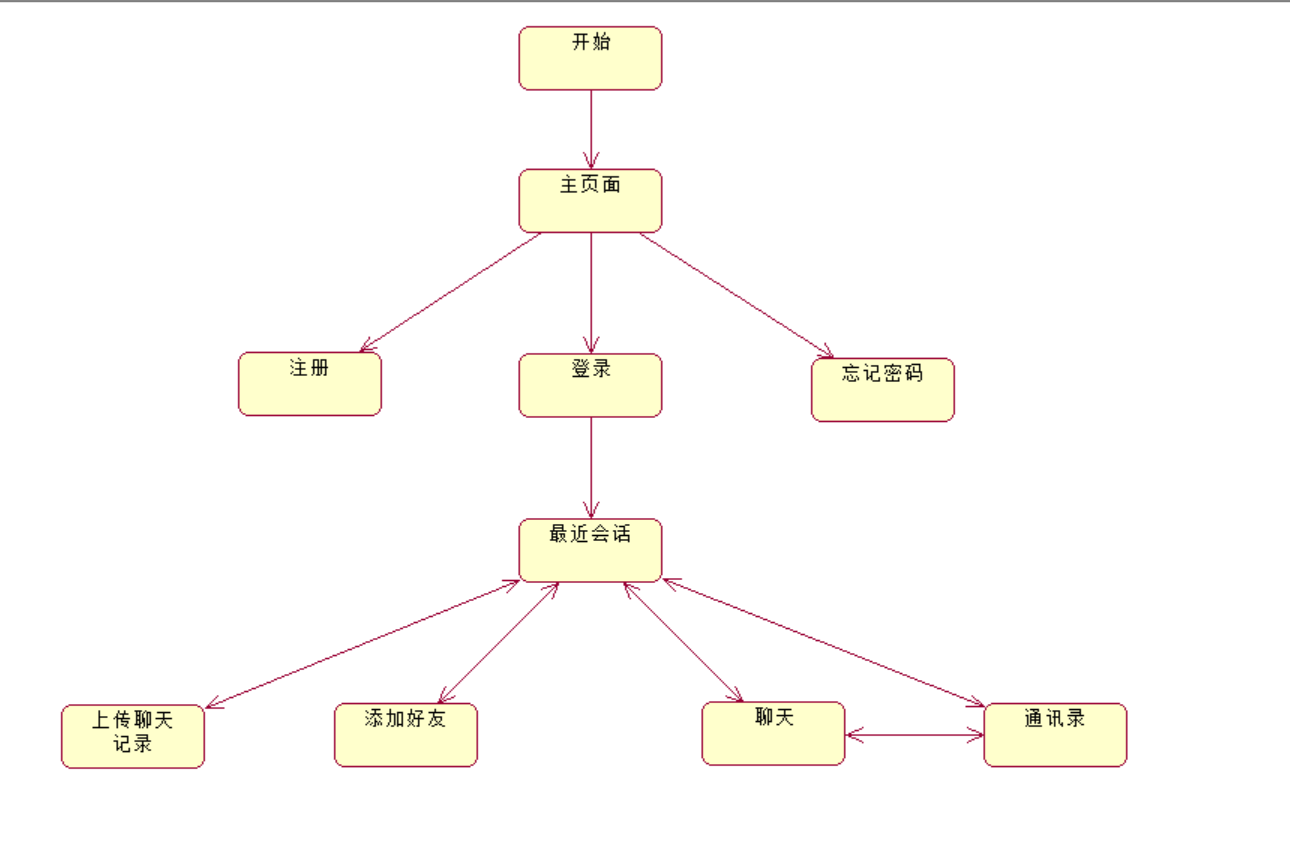


图2-8 模块调用图

# **3. 详细设计（撰写人：）**

# **3.1实现概要设计中的数据结构**

//联系人结构

String contactID; //联系人账号

String contactName; //联系人昵称

String accid; //联系人ACCID

}

//添加好友的系统消息的结构

public class MessageofSystem {

private long messageID; //系统消息的ID

private String sender; //消息的发送者

}

//与好友的会话结构

public class RecentContact {

private String sender; //会话对象的ID

private String text; //最后的一次会话的内容

private String unreadCount; //消息未读数

private String senderAccid; //会话对象的Accid

}

//用户信息结构

public class UserInfo {

private String userId; //用户账号

private String userName; //用户名

private String userPwd; //密码

private String accid; //网易云信账号

private String token; //网易云信密码

private int sex; //性别

private String phone; //联系电话

}

# **3.2数据关系ER图**

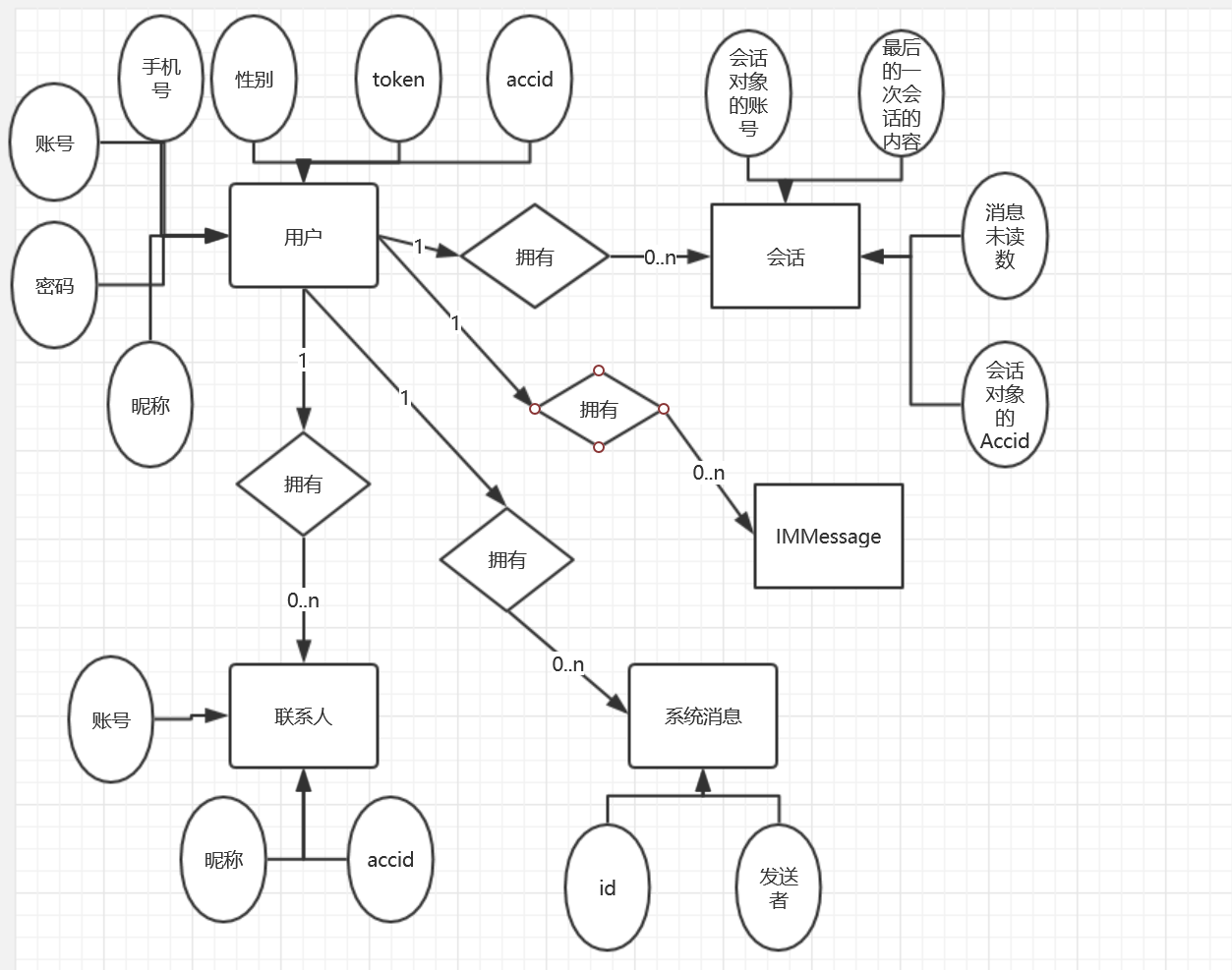


图3-1 ER图

# **3.3功能模块的关系图**

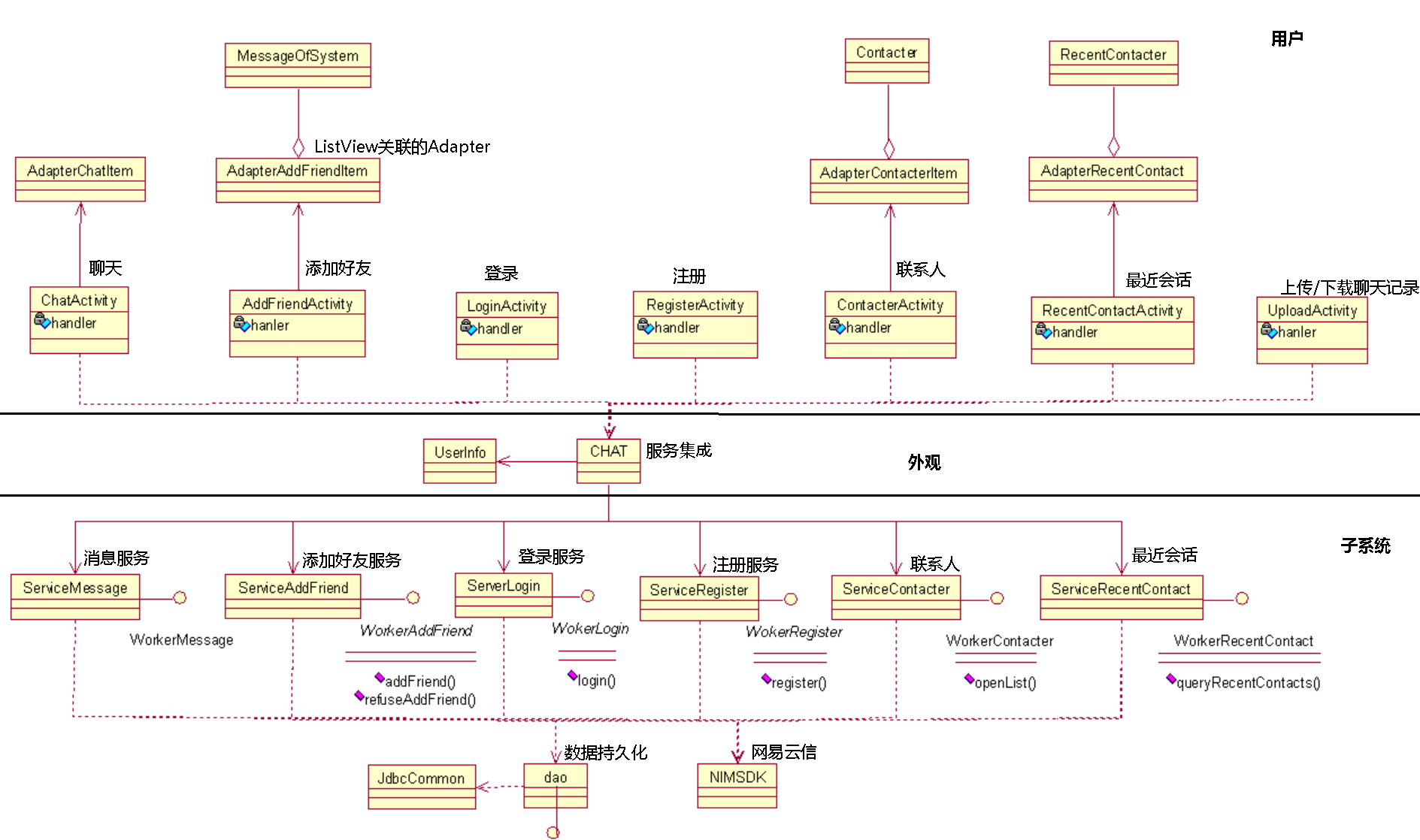


图3-2 类图

# **3.4子系统API表**

表3-1 添加好友系统API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | Handler消息类型 | 功能 |
| addFriend | String contact  String text  Handler handler | void | ISFRIEND:双方以及是好友  ADDFRIEN:消息发送成功  其他:消息发送失败 | 给账号为contact的用户发送一条附言为text的添加好友消息 |
| getMessageFromFriendSystem | Handler handler | void | NEWFRIEND:有新的添加好友请求  其他:获取失败 | 获取添加好友请求 |
| refuseAddFriend | String contactId  long messageId | void |  | 拒绝账号为contactId的用户的好友请求，并将系统消息ID为messageId的消息的状态设置为拒绝 |
| agreeAddFriend | String contactId  long messageId | void |  | 同意账号为contactId的用户的好友请求，并将系统消息ID为messageId的消息的状态设置为同意 |

表3-2 联系人系统API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | Handler消息类型 | 功能 |
| openList | String userID  String listName  Handler handler | Void | CONTACTERITEM: 一个联系人 | 通过userID以及listName获取联系人 |

表3-3 登录系统API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名 | 参数 | 返回值 | Handler消息类型 | 功能 |
| login | String acount  String passwd  Handler handler | void | LOGIN\_SUS:登录成功,无obj  其他:登录失败,无obj | 登录 |

表3-4 注册系统API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名 | 参数 | 返回值 | Handler消息类型 | 功能 |
| register | String acount  String passwd  Handler handler | void | REGISTER\_SUS:注册成功，无obj  REGISTER\_FAIL:注册失败,无obj | 注册账号 |

表3-5 最近会话系统API

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名 | 参数 | 返回值 | Handler消息类型 | 功能 |
| queryRecentContacts | Handler handler | void | 所有：一条会话记录，obj的类型为RecentContact | 查询最近会话 |

表3-6 消息系统API（1）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口名 | 参数 | 返回值 | Handler消息类型 | 功能 |
| sendMessage | String accid  String text  String type  Handler handler | void | SEND\_NOW：发送成功，无obj  其他：发送失败，无obj | 给accid为accid的用户发送一条内容为text的type类型的消息 |
| registerObserverForInstant | String fromAccid  Handler handler | void | RECEIVER\_NOW:接受到一条消息，obj的类型为IMMessage | 接受accid为fromAccid的用户发送的消息 |
| duplicate | IMMessage msg | void |  | 检查本地数据库是否有msg这样一条消息，如果没有则将该消息插入本地数据库；该功能用于从云数据库下载聊天记录时的查重 |
| importMessageFromDatabase | Handler handler | void | DOWN\_FINISH：聊天记录下载完成；  DOWN：每下载完成10条消息则返回该类型消息 | 从云数据库中下载聊天记录到本地 |

表3-7 消息系统API（2）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| exportMessageToDatabase | Handler handler | void | UP: 每上传完成10条消息则返回该类型消息 | 将本地数据库中的聊天记录上传到云数据库，每个会话只上传10条 |
| getMessageListFromLocal | String myAccid  String fromAccid  Handler handler  int limit | void | LOAD\_NUMBER：搜索到得聊天记录的条数，obj类型为int表示条数；  SEND\_OLD：搜索到一条自己发送给formAccid的消息，obj类型为IMMessage；  RECEIVER\_OLD：搜索到一条发送者为fromAccid的消息，obj类型为IMMessage; | 从本地数据库中搜索会话双方为myAccid和fromAccid的limit条聊天记录，如果连续查询则会返回时间更前的聊天记录 |

表3-8 消息系统API（3）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cancelAnchor |  | void |  | 重置锚点；该功能用于在聊天对象切换时，从当前时间向前搜索聊天记录 |
| clearUnread | String fromAccid | void |  | 清零与fromAccid用户的会话的未读数 |
| cancelOberserver |  | void |  | 注销网易云信得消息观察者 |

# **3.5过程调用的流程图与核心代码**

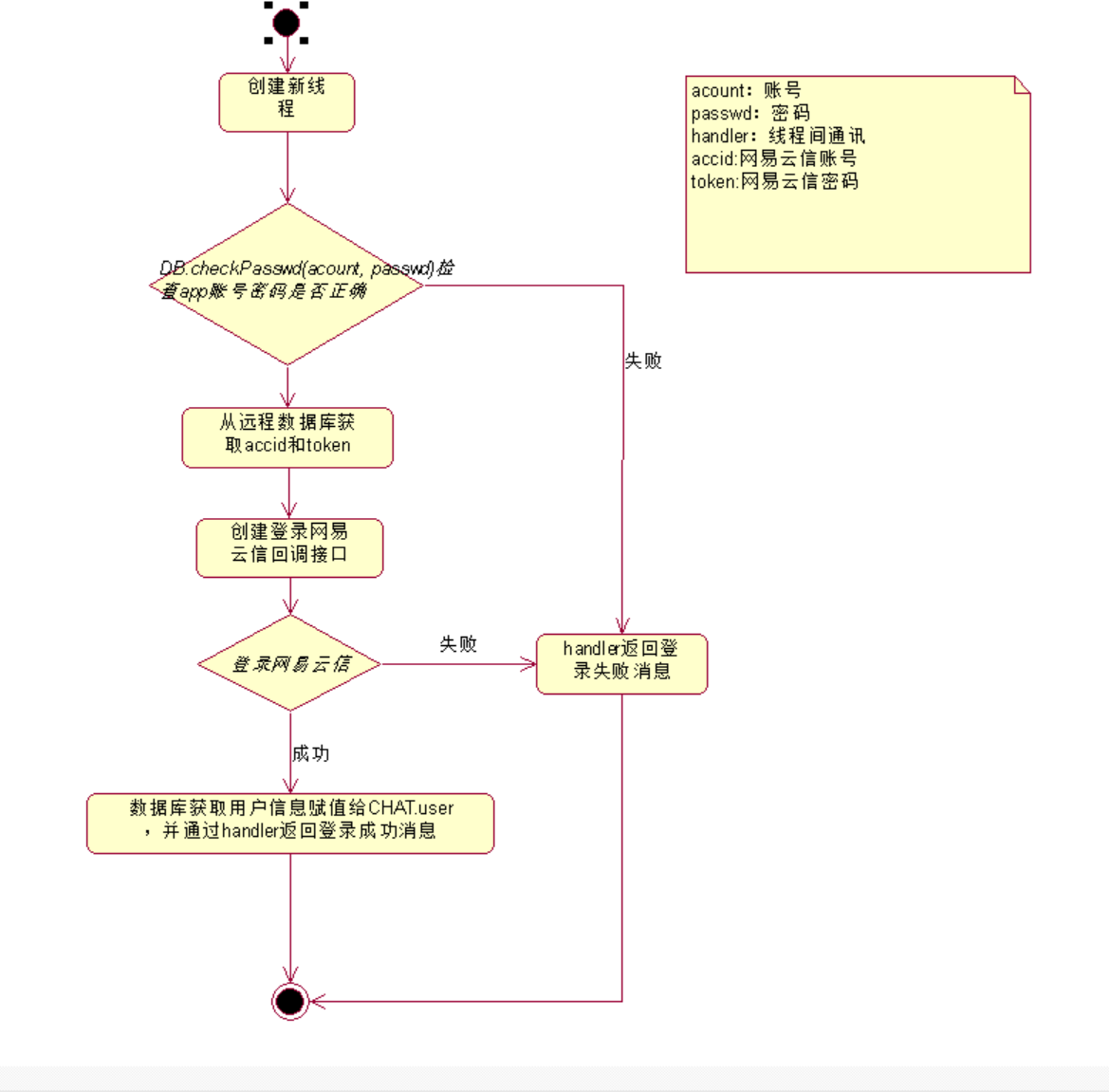


图3-3 login方法过程

login方法核心代码：

accid = DB.*getAccidByAcount*(acount); //从远程数据库获取accid  
token = DB.*getTokenByAcount*(acount); //从远程数据库获取token

msg.what = *LOGIN\_SUC*;  
handler.sendMessage(msg); //验证成功后通过handler给UI线程发送消息

CHAT.*user* = DB.*getUserInfoByAcount*(acount); //从远程数据库获取用户信息赋值给CHAT.user

NIMClient.*getService*(AuthService.class).login(info).setCallback(callback); //网易云信登录接口

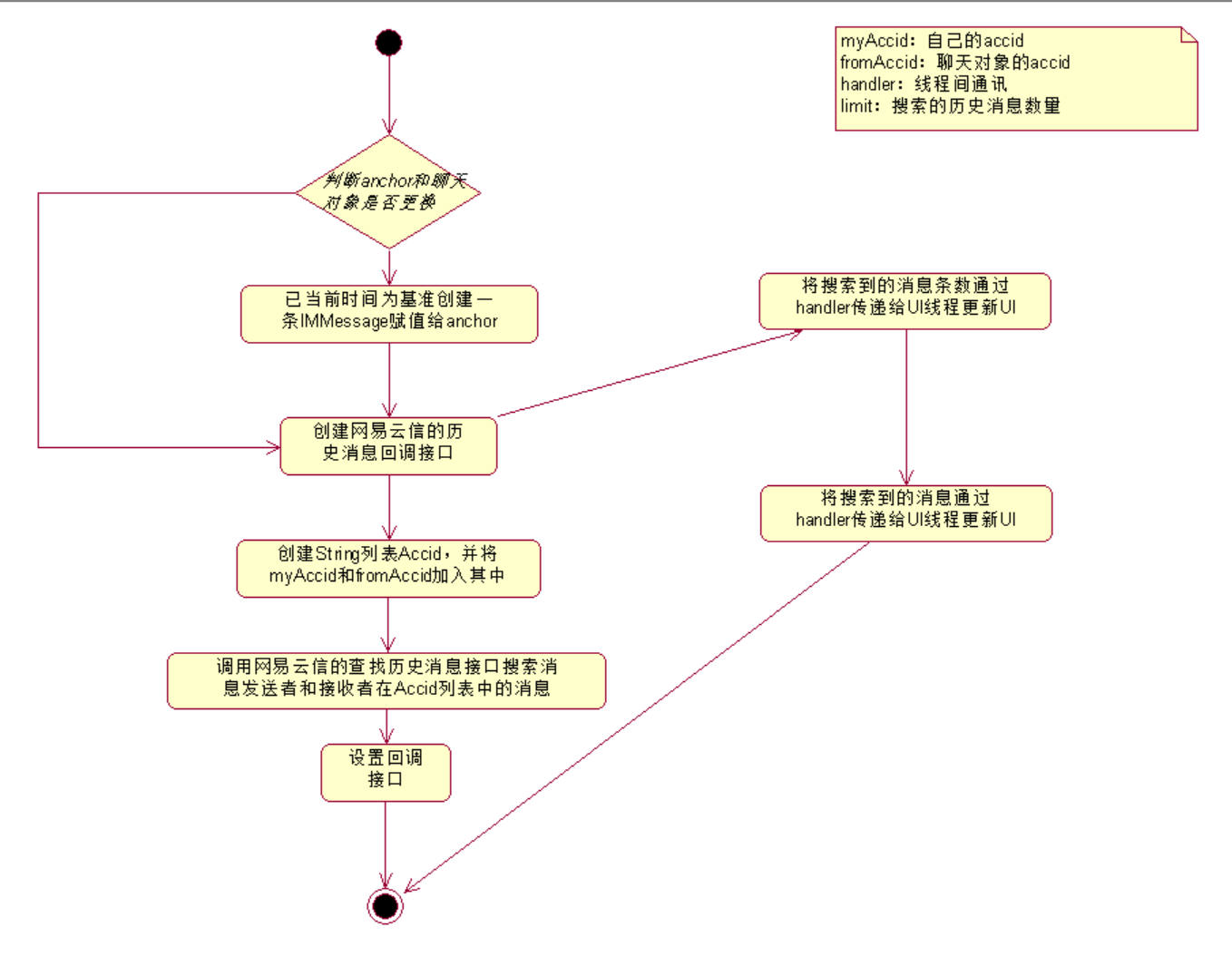


图3-3 getMessageListFromLocal方法过程

getMessageListFromLocal方法核心代码:

// 检测聊天窗口是否切换，切换则定义新的锚点,主要通过检查聊天用户是否更换以及瞄点是否被重置来判断  
if( *anchor* == null || !fromAccid.equals(*fromAccount*) ) {  
 *fromAccount* = fromAccid;  
 *anchor* = MessageBuilder.*createTextMessage*(fromAccid, SessionTypeEnum.*P2P*, "");  
}

//通过判断消息的发送者是否为自己来选择handler的消息类型（自己发送的消息会显示在聊天窗口的右侧）

msgHandler.what = msg.getFromAccount().equals(myAccid) ? *SEND\_OLD* : *RECEIVER\_OLD*;

//更新瞄点：为了在用户下划聊天窗口的时候加载时间更前的聊天记录

*anchor* = msg;

// 网易云信接口：查询anchor往前会话双方为accid的limit条聊天记录  
NIMSDK.*getMsgService*().searchMessageHistory("关键字", accid, *anchor*, limit)  
 .setCallback(callbackWrapper);

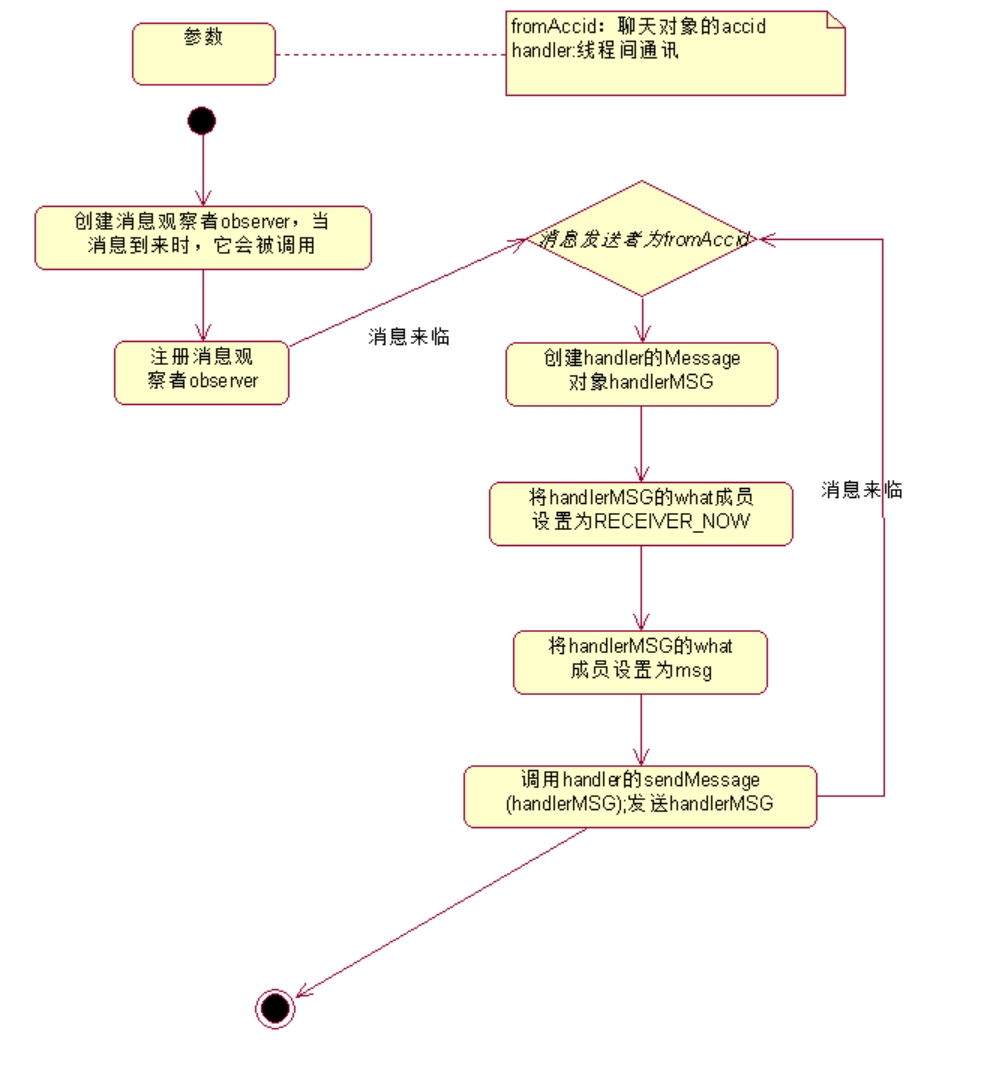


图3-3 registerObserverForInstant方法过程

RegisterObserverForInstant方法核心代码:

//判读消息的发送者是否为当前的聊天对象，如果是则将消息通过handler发送给UI线程显示

if( msg.getFromAccount().equals(fromAccid) ){  
 Message handlerMSG = new Message();  
 handlerMSG.what = *RECEIVER\_NOW*;  
 handlerMSG.obj = msg;  
 handler.sendMessage(handlerMSG);  
}

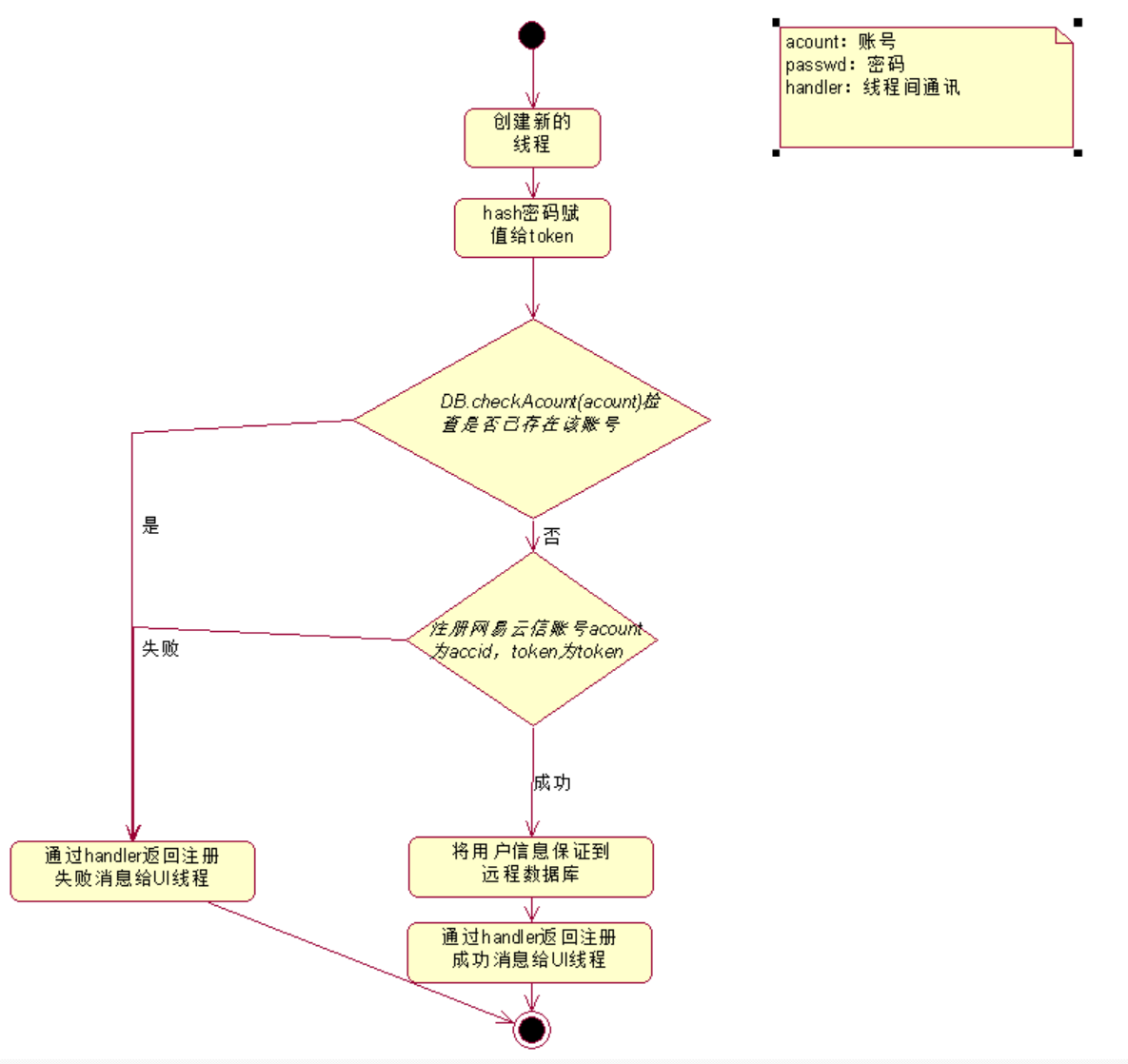


图3-3 register方法过程

Register方法核心代码：

//hash密码作为网易云信的token（密码）

String token = String.*valueOf*( passwd.hashCode() );

//账号查重，确认没有重复的代码

DB.*checkAcount*(acount)

//注册网易云信通讯账号，accid <- aocunt, token <- hash(passwd)

registerNIM( acount, token )

//将用户的信息保存到远程数据库（注册成功）

DB.*register*(acount, passwd, acount, token, acount, 1, "");

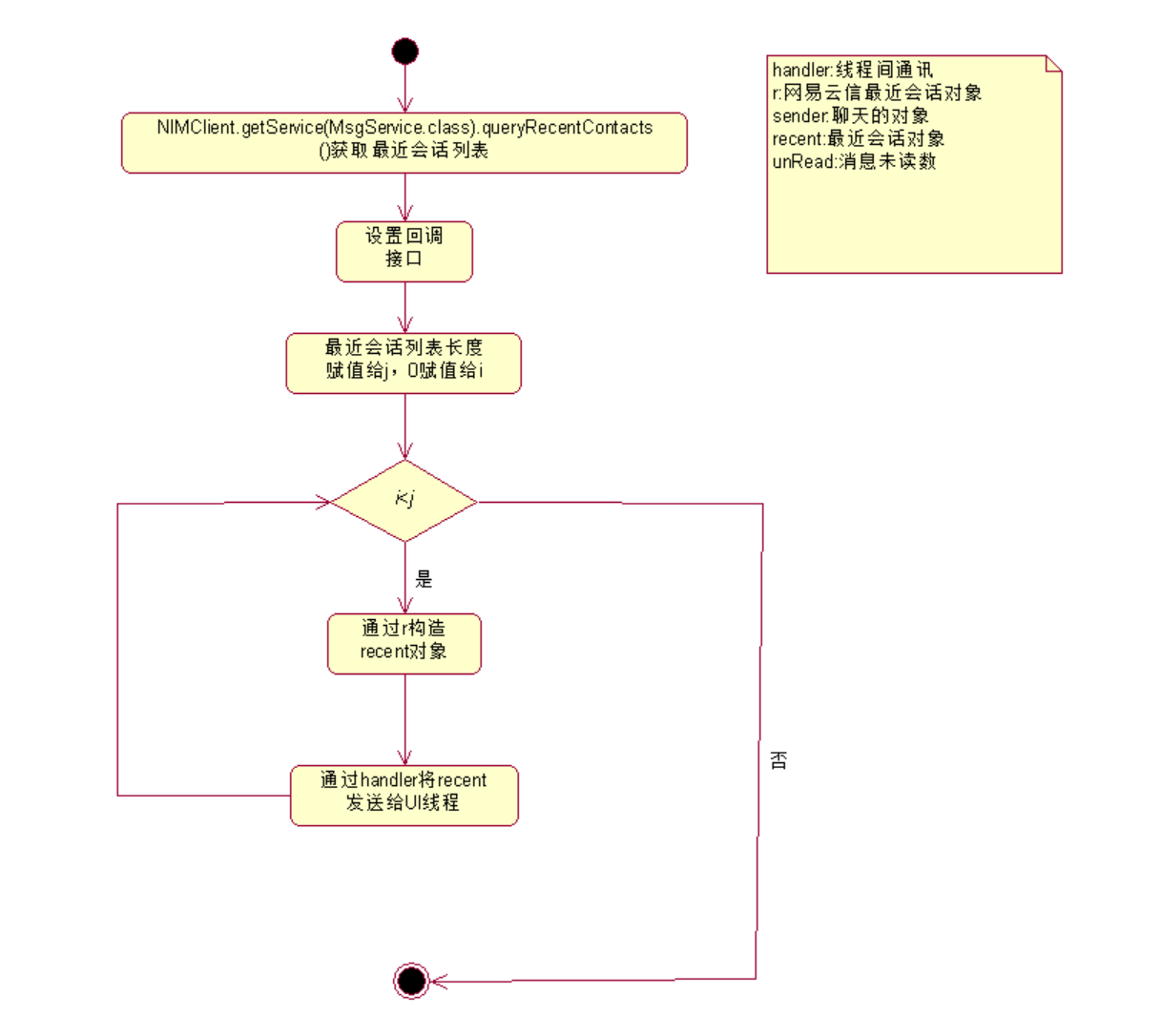


图3-3 queryRecentContacts方法过程

queryRecentContacts方法核心代码:

//因为需要显示消息未读数，主线程睡眠800毫秒，等待网易云信后台成功把消息都保存到数据库，才开始操作

SystemClock.*sleep*(800);

//网易云信接口：获取最近会话列表

NIMClient.*getService*(MsgService.class).queryRecentContacts()

//从最近会话列表中取出第i个会话

RecentContact r = recents.get(i);

//通过handler返回给UI线程的对象，包含了会话的主要内容(聊天对象accid， 会话未读数，最后一次消息的内容)

com.example.k.chat20.datastruct.RecentContact recent = new com.example.k.chat20.datastruct.RecentContact();

## **4. 调试分析（撰写人：）**

**4.1 连接数据库问题**

问题描述：发送HTTP请求以及所有需要做数据库操作的时候无法通过主线程进行，无法连接到数据库。

解决方案：通过创建一个新的线程来进行，因为这个线程没有返回值，所以主线程与新线程之间的通信要通过Handler进行，将子线程的数据传递给主线程，实现任意两个线程的数据传递。

例如：在查询好友时需要到DB类中的getFriendListByAcount()方法查询联系人列表，必须通过创建一个子线程去调用。

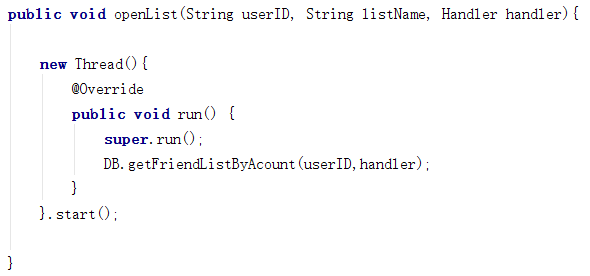


图4-1联系人服务方法

**4.2 网易云信的消息接收问题**

问题描述：在用户登录的时候，网易云信会在后台自动创建一个进程用于接收消息，然后直接将消息保存到数据库，所以在即时通信的时候首先要做的事情是从本地数据库获取聊天对象在用户没有在线的时候发送的消息显示在聊天窗口，也就是说，并不是在用户登录并开始与某个用户聊天的时候消息才会被接收，而是在用户登录成功的时候就自动接收这些消息并保存显示到聊天页面中。

解决方案：用网易云信提供的方法



图4-2接收消息方法

**4.3 Android版本太高无法注册问题**

问题描述：Android版本太高无法注册成功。

问题方案：在AndroidManifest.xml中加上如下所示代码。

<uses-library  
   android:name="org.apache.http.legacy"  
     android:required="false"  
   />

## **5.用户使用说明（撰写人：）**

**5.1 软件登录与注册**

用户进入主页面先点击新用户注册账号再返回主页面点击登录进入登录页面。若不是新用户直接点击登录进入登录页面即可。页面如下图所示：

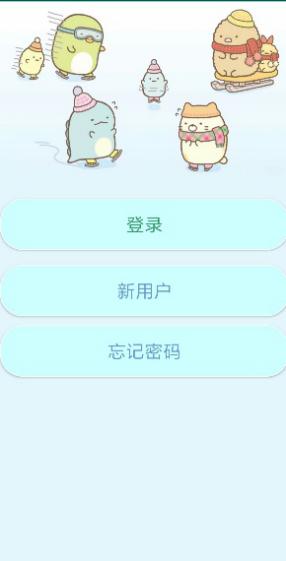
  

图5-1主页面 图5-2注册 图5-3登录

**5.2 消息主页**

登录成功后进入到消息页面，显示与好友的聊天记录以及未读消息。如下图所示：



图5-4消息页面

点击好友即可进入聊天页面通讯，页面会留存聊天的历史记录。如下图所示：

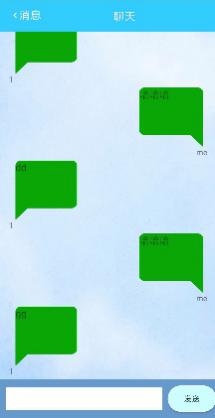


图5-5聊天页面

**5.3 联系人页面**

点击下方导航键中第二个图标可切换至联系人页面，显示所有添加的好友列表，点击好友也可以进入聊天页面通讯，同时会显示出加载了几条消息。

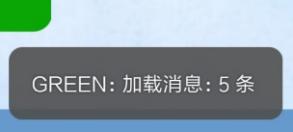
 

图5-6加载消息 图5-7联系人页面

**5.4 添加好友**

点击下方导航键中第三个图标可切换至添加好友页面，输入你要添加的好友账号，点击添加将会给对方发送一条好友申请，对方点击同意即可添加成功，成为好友。

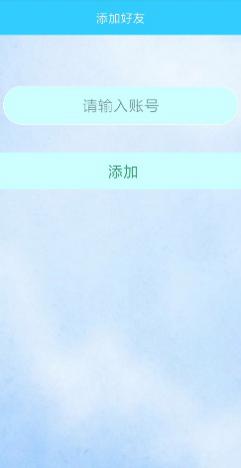


图5-8添加好友页面

**5.5 上传聊天记录**

点击下方导航键中第四个图标可切换至上传和下载聊天记录页面，当点击上传时，会将与好友的聊天记录上传到数据库中存储起来，消息页面也会更新聊天记录。如果卸载软件后再安装回来，点击下载就能从数据库中加载出聊天记录并显示在消息页面。

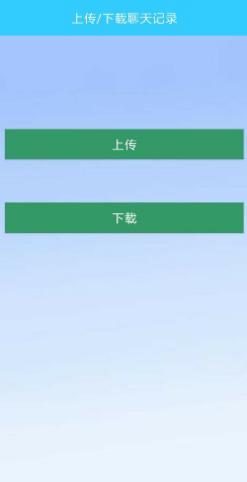


图5-9上传聊天记录

**6.软件测试结果（撰写人：）**

**6.1测试环境与配置**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 硬件配置 | 系统 | 数量 | 备注 |
| 1 | 小米6 | 安卓9  运行内存6G | 1 | 用户使用环境 |
| 2 | 小米9SE | 安卓9  运行内存8G | 1 | 用户使用环境 |

**6.2测试内容**

本次测试主要是对需求进行验收，统计功能完成情况；其次，对于用户使用频率较高的模块进行压力测试，检验其性能是否能满足高并发的要求。

**6.3测试结果与分析**

|  |  |  |
| --- | --- | --- |
| 测试模块 | 测试结果 | 说明 |
| 用户注册功能 | C:\Users\asus\Documents\Tencent Files\1026260455\FileRecv\MobileFile\IMG_20190617_202258.jpg | 注册用户页面可以成功显示。  如果仅输入注册账号或者密码的话，会提示注册失败。  输入已注册过的账号的话，也会提示注册失败。  输入正确的账号密码可成功注册。 |
| 用户登录功能 | C:\Users\asus\Documents\Tencent Files\1026260455\FileRecv\MobileFile\IMG_20190617_202244.jpg | 登录界面可以成功显示。  输入正确的账号密码点击登录可以成功登录。  输入错误的账号或者密码会提示登录失败。 |
| 添加好友功能 | C:\Users\asus\Documents\Tencent Files\1026260455\FileRecv\MobileFile\Screenshot_2019-06-17-21-23-54-096_com.example.k..png | 添加好友页面可成功显示。  若添加的好友不存在，会提示不存在用户。  若添加的好友已在好友列表中，会提示对方已经是好友。  申请好友之后，对方会受到好友申请，可以选择同意或者拒绝。  同意之后，该用户会添加到好友列表之中。 |
| 收发消息功能 | C:\Users\asus\Documents\Tencent Files\1026260455\FileRecv\MobileFile\Screenshot_2019-06-17-21-03-17-515_com.example.k..png | 可以成功发送和接收消息，但是对话框的大小不能根据文字内容的多少进行调整。  只能发送文字信息，无法发送图片音频视频等。  未读的消息会有提示。 |
| 查看好友列表功能 | C:\Users\asus\Documents\Tencent Files\1026260455\FileRecv\MobileFile\Screenshot_2019-06-17-21-00-00-581_com.example.k..png | 好友列表可以成功显示，但仅限于联网的情况下。  每次查看好友列表都要连接服务器重新刷新，网络断开的情况下无法查看好友列表。 |
| 上传和下载聊天记录功能 | C:\Users\asus\Documents\Tencent Files\1026260455\FileRecv\MobileFile\Screenshot_2019-06-17-21-46-36-408_com.example.k..png | 可以成功上传和下载聊天记录。 |