南京邮电大学

毕 业 设 计（论 文）

|  |  |
| --- | --- |
| 题 目 | 基于XMPP协议的Android即时通信软件的设计与实现 |
| 专 业 | 网络工程 |
| 学生姓名 | 徐阳 |
| 班级学号 | B13070134 |
| 指导教师 | 李养群 |
| 指导单位 | 物联网学院 |

日期： 2017 年 1 月 15 日至 2017 年 6 月 17 日

毕业设计（论文）原创性声明

本人郑重声明：所提交的毕业设计（论文），是本人在导师指导下，独立进行研究工作所取得的成果。除文中已注明引用的内容外，本毕业设计（论文）不包含任何其他个人或集体已经发表或撰写过的作品成果。对本研究做出过重要贡献的个人和集体，均已在文中以明确方式标明并表示了谢意。

论文作者签名：

日期： 年 月 日

摘 要

随着互联网的飞速发展，移动互联网正在成为当今社会的主流。得益于移动互联网的方便性和全面性，即时通信软件有了快速发展，比如QQ，微信等，给人们的日常生活带来了极大便利。在如今的移动市场中，存在着两大阵营，苹果和Android，而Android凭借着其开源性，获得了广大的市场和市场前景。而且，XMPP协议是一种基于标准通用标记语言的子集XML的协议，它继承了在XML环境中灵活的发展性，具有超强的可扩展性。因此，基于XMPP的Android即时通信软件具有很好的应用前景。

本文首先讲述了本文的研究背景，对Android和XMPP协议的特征和原理做了详细的说明和介绍。接着，介绍了开源的服务端openfire的原理和使用方法。然后，结合服务端，详细介绍了客户端的详细设计，包括界面设计，以及和服务端openfire的通信方法，来完成用户注册，登录，聊天等功能。

本论文在XMPP协议的基础上，设计与实现Android即时通信软件，任务如下：

1. 开发一个可在Android上运行的XMPP协议客户端。服务器端可任意选择一个开源的都可以。通过该客户端，用户之间可以进行信息、图片、音频等信息的发送。

2. 技术要求：采用XMPP协议，在Android平台上运行。

3. 成果形式：可在模拟器或者手机上运行的即时通信软件。

关键词：即时通信，可扩展通讯和表示协议（XMPP），Android

**ABSTRACT**

With the rapid development of the Internet, mobile Internet is becoming the mainstream of today's society. Thanks to the convenience and comprehensiveness of the mobile Internet, instant messaging software has developed rapidly, such as QQ, WeChat, etc., to people's daily life has brought great convenience. In today's mobile market, there are two camps, Apple and Android, and Android by virtue of its open source, access to the vast market and market prospects. Moreover, the XMPP protocol is a protocol based on a subset of the standard markup language XML, which inherits flexible development in the XML environment, with superior scalability. Therefore, XMPP-based Android instant messaging software has a good application prospects.

This paper first describes the background of this paper, the characteristics of Android and XMPP protocol and the principle of a detailed description and description. Then, introduced the open source server openfire principle and use. Then, combined with the server, detailed description of the client's detailed design, including interface design, and server-side openfire communication methods to complete the user registration, login, chat and other functions.

In this paper, based on the XMPP protocol, the design and implementation of Android instant messaging software, the task is as follows:

   1. Develop an XMPP protocol client that runs on Android. Server-side can choose an open source can be. Through the client, users can be information, pictures, audio and other information sent.

   2. Technical requirements: XMPP protocol, running on the Android platform.

   3. Result form: Instant messaging software that can be run on a simulator or mobile phone.

**Key words：** instant messaging, Extensible communication and presentation protocols（XMPP）, Android

目 录

[第一章 绪论 1](#_Toc482885343)

[1.1 课题研究的背景 1](#_Toc482885344)

[1.2国内外研究现状 1](#_Toc482885345)

[1.3研究内容及意义 2](#_Toc482885346)

[1.4章节安排 2](#_Toc482885347)

[第二章 相关技术简介 4](#_Toc482885348)

[2.1 Java介绍 4](#_Toc482885349)

[2.1.1 java语言特点 4](#_Toc482885350)

[2.1.2 单例模式 4](#_Toc482885351)

[2.2 Android介绍 5](#_Toc482885352)

[2.2.1 Android的架构 5](#_Toc482885353)

[2.2.2 Android之Activity生命周期 7](#_Toc482885354)

[2.2.3 Android之布局 8](#_Toc482885355)

[2.2.4 SQLite数据库 9](#_Toc482885356)

[2.3 XMPP协议解析 9](#_Toc482885357)

[2.3.1 XMPP协议简介 9](#_Toc482885358)

[2.3.2 XMPP协议内容 10](#_Toc482885359)

[2.3.3 XMPP地址格式 11](#_Toc482885360)

[第三章 系统需求分析与设计 13](#_Toc482885361)

[3.1客户端分析 13](#_Toc482885362)

[3.1.1注册和登录模块 13](#_Toc482885363)

[3.1.2好友模块 13](#_Toc482885364)

[3.1.3消息模块 14](#_Toc482885365)

[3.1.4 信息修改模块 14](#_Toc482885366)

[3.2服务端分析 14](#_Toc482885367)

[3.2.1登录功能 14](#_Toc482885368)

[3.2.2界面设置 14](#_Toc482885369)

[3.2.3消息设置 15](#_Toc482885370)

[第四章 系统功能的实现 15](#_Toc482885371)

[4.1服务器实现 15](#_Toc482885372)

[4.1.1系统启动流程 15](#_Toc482885373)

[4.1.2网络处理 15](#_Toc482885374)

[4.1.3数据库处理 16](#_Toc482885375)

[4.1.4页面处理 16](#_Toc482885376)

[4.2客户端实现 17](#_Toc482885377)

[4.2.1连接服务器 17](#_Toc482885378)

[4.2.2用户注册 18](#_Toc482885379)

[4.2.3用户登录 18](#_Toc482885380)

[4. 2. 4好友管理 18](#_Toc482885381)

[4.2.5消息传递 20](#_Toc482885382)

[4.2.6数据库设计 20](#_Toc482885383)

[第五章 功能测试 21](#_Toc482885384)

[5.1用户注册测试 21](#_Toc482885385)

[5.2用户登录测试 22](#_Toc482885386)

[5.3好友测试 23](#_Toc482885387)

[5.4聊天测试 25](#_Toc482885388)

[第六章 总结与展望 26](#_Toc482885389)

[6.1总结 26](#_Toc482885390)

[6.2展望 26](#_Toc482885391)

[结束语 27](#_Toc482885392)

[致 谢 28](#_Toc482885393)

[参考文献 29](#_Toc482885394)

[附录A 基于XMPP的Android即时通信系统使用说明 30](#_Toc482885395)

# 第一章 绪论

## 1.1 课题研究的背景

在我国互联网的发展过程中，PC互联网已日趋饱和，移动互联网却呈现井喷式发展。数据显示，截止2013年底，中国手机网民超过5亿，占比达81%。伴随着移动终端价格的下降及wifi的广泛铺设，移动网民呈现爆发趋势。Google为此于2007年推出了一个专为移动设备设计的软件平台Android。Android是一套真正意义上的开放性的移动设备综合平台，它包括操作系统、中间件和一些关键的平台应用。

Android 是运行于Linux kernel之上，源于它的开放性，开放的平台吸引者越来越多的开发者，开源的代码库、免费的开发软件、社区、第三方开源共享，在带来巨大的竞争的同时也使得android在开放的平台中显得日益成熟。

即时通讯（Instant Messaging）是目前Internet上最为流行的通讯方式，各种各样的即时通讯软件也层出不穷；服务提供商也提供了越来越丰富的通讯服务功能。 不容置疑，Internet已经成为真正的信息高速公路。从实际工程应用角度出发,以计算机网络原理为指导,结合当前网络中的一些常用技术,编程实现基于C/S架构的网络聊天工具是切实可行的。

XMPP（可扩展消息处理现场协议）是基于可扩展标记语言（XML）的协议，它用于即时消息（IM）以及在线现场探测。它在促进服务器之间的准即时操作。这个协议可能最终允许因特网用户向因特网上的其他任何人发送即时消息，即使其操作系统和浏览器不同。

因此，基于XMPP的Android即时通信软件获得了广泛的关注。由于其即时性，直观性，廉价性的特点，人们可以直观的了解到在线情况，节省了打电话，发邮件来找人的麻烦。总之，即时通信软件拥有非常大的市场。

## 1.2国内外研究现状

即时通信(instant message，IM)是指能够即时发送和接收互联网消息等的业务。即时通讯不同于e-mail，在于它的交谈是即时的。大部分的即时通讯服务提供了Presence Awareness的特性──显示联络人名单，联络人是否在线上与能否与联络人交谈。无论是国内，还是国外，都有很多流行的应用。

如今，在中国最火的便是微信。微信是腾讯公司于2011年1月21日推出的一个为智能终端提供即时通信服务的免费应用程序 。微信支持跨通信运行商、跨操作系统平台通过网络快速发送语音短信、视频、图片和文字等。截止到2016年第二季度，微信已经覆盖中国94% 以上的智能手机，月活跃用户达到 8.06亿，用户覆盖 200 多个国家、超过 20 种语言。

其次，便是腾讯的QQ。腾讯QQ（简称“QQ”）是腾讯公司开发的一款基于Internet的即时通信（IM）软件。腾讯QQ支持在线聊天、视频通话、点对点断点续传文件、共享文件、网络硬盘、自定义面板、QQ邮箱等多种功能，并可与多种通讯终端相连。2015年，QQ继续为用户创造良好的通讯体验！其标志是一只戴着红色围巾的小企鹅。

在国外，Skype最受欢迎。其具备IM所需的功能，比如视频聊天、多人语音会议、多人聊天、传送文件、文字聊天等功能。它可以高清晰与其他用户语音对话，也可以拨打国内国际电话，无论固定电话、手机、小灵通均可直接拨打，并且可以实现短信发送等功能。

上面列举了目前主流的几款应用，虽然这些应用都取得了非常大的成绩，但他们用的通信协议都是自主开发，是公司内部的协议，没有统一的标准。这使得不同应用之间不能相互通信。而XMPP协议的使用，则可以实现一个统一的标准，使得这些软件可以相互通信，极大地提高了即时通信软件的扩张性。

## 1.3研究内容及意义

随着智能手机的普及，以及性能的提升，移动网络用户的数量爆炸增长，如今已超过了pc端的网络用户。传统的打电话，发短信，发邮件等通信方式已满足不了日益增长的用户需求，为此，即时通信的概念被提了出来。即时通讯（Instant messaging，简称IM）是一个终端服务，允许两人或多人使用网路即时的传递文字讯息、档案、语音与视频交流。即时通讯按使用用途分为企业即时通讯和网站即时通讯，根据装载的对象又可分为手机即时通讯和PC即时通讯，手机即时通讯代表是短信，网站、视频即时通讯。

如今，即时通信软件比如微信，Skype都掀起了一股热潮。但这些软件的通信协议都用了内部的协议，不能相互通信。而XMPP的出现，提出了一个统一的通信标准，解决了这些问题。

而且，如今的手机系统分为两大阵营：苹果和安卓。由于Android的开源特性，吸引着越来越多的开发者，开源的代码库、免费的开发软件、社区、第三方开源共享，在带来巨大的竞争的同时也使得android在开放的平台中显得日益成熟。

因此，本论文决定研究基于Android的XMPP即时通信软件，主要做了如下工作：

（1）掌握java方面的基础知识。在其基础上，学习android方面的知识。了解android的活动activity，以及各种组件，理解其原理，和如何使用。

（2）根据设计需求，利用Android studio搭建出开发环境，设计出用于注册，登录，好友和聊天的大体页面。

（3）利用基于XMPP的开源库Asmack，结合自己选择的开源服务器，实现用户注册，用户登录，好友管理，实时聊天等功能，并和上一步的设计出的页面相结合。

（4）测试各个功能，验证是否满足需求。可以设计出具体的测试方案，按照步骤来进行测试。

## 1.4章节安排

本文主要讨论了XMPP协议的原理，以及根据XMPP协议，完成一个Android即时通信软件，来满足人们日常的通信需求，改善人们的工作效率。

第一章是绪论，简单描述了课题研究的背景，国内外研究现状，以及研究的内容和意义。

第二章是相关技术简介，主要介绍了Android的架构，活动等重要知识点，以及XMPP协议的相关知识点。

第三章介绍了基于Android的XMPP即时通信的需求分析，从服务端，和客户端这两方面进行阐述，分别介绍了这两部分所需实现的功能。

第四章是基于Android的XMPP即时通信的设计与实现，和需求分析一样，也是从客户端和服务端这两方面进行分析。

第五章则是对自己的成果进行功能测试，主要是针对客户端的用户注册，用户登录，好友管理，消息转发等功能进行测试。

第六章是对自己的总结和展望，讲述自己完成这篇论文的主要步骤，以及还存在哪些问题亟待解决。

# 第二章 相关技术简介

## 2.1 Java介绍

### 2.1.1 java语言特点

Java是一种面向对象的编程语言，不仅吸收了C++的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用这两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。

Java语言的三大特性即是：封装、继承、多态

（1）封装：首先，属性可用来描述同一类事物的特征，方法可描述一类事物可做的操作。封装就是把属于同一类事物的共性（包括属性与方法）归到一个类中，以方便使用。封装也称为信息隐藏，是指利用抽象数据类型将数据和基于数据的操作封装在一起，使其构成一个不可分割的独立实体，数据被保护在抽象数据类型的内部，尽可能地隐藏内部的细节，只保留一些对外接口使之与外部发生联系。系统的其他部分只有通过包裹在数据外面的被授权的操作来与这个抽象数据类型交流与交互。也就是说，用户无需知道对象内部方法的实现细节，但可以根据对象提供的外部接口(对象名和参数)访问该对象。好处：(1)实现了专业的分工。将能实现某一特定功能的代码封装成一个独立的实体后，各程序员可以在需要的时候调用，从而实现了专业的分工。(2)隐藏信息，实现细节。通过控制访问权限可以将可以将不想让客户端程序员看到的信息隐藏起来，如某客户的银行的密码需要保密，只能对该客户开发权限。

（2）继承：就是个性对共性的属性与方法的接受，并加入个性特有的属性与方法一个类继承另一个类，则称继承的类为子类，被继承的类为父类。实现代码的复用。子类与父类的关系并不是日常生活中的父子关系，子类与父类而是一种特殊化与一般化的关系，是is-a的关系，子类是父类更加详细的分类。如class dog extends animal,就可以理解为dog is a animal.注意设计继承的时候，若要让某个类能继承，父类需适当开放访问权限，遵循里氏代换原则，即向修改关闭对扩展开放，也就是开-闭原则。

（3）多态：多态的概念发展出来，是以封装和继承为基础的。多态就是在抽象的层面上实施一个统一的行为，到个体（具体）的层面上时，这个统一的行为会因为个体（具体）的形态特征而实施自己的特征行为。（针对一个抽象的事，对于内部个体又能找到其自身的行为去执行。）相同的事物，调用其相同的方法，参数也相同时，但表现的行为却不同。子类以父类的身份出现，但做事情时还是以自己的方法实现。子类以父类的身份出现需要向上转型(upcast)，其中向上转型是由JVM自动实现的，是安全的，但向下转型(downcast)是不安全的，需要强制转换。子类以父类的身份出现时自己特有的属性和方法将不能使用。

### 2.1.2 单例模式

Java中单例(Singleton)模式是一种广泛使用的设计模式。单例模式的主要作用是保证在Java程序中，某个类只有一个实例存在。一些管理器和控制器常被设计成单例模式。

单例模式有很多好处，它能够避免实例对象的重复创建，不仅可以减少每次创建对象的时间开销，还可以节约内存空间；能够避免由于操作多个实例导致的逻辑错误。如果一个对象有可能贯穿整个应用程序，而且起到了全局统一管理控制的作用，那么单例模式也许是一个值得考虑的选择。

单例模式有很多种写法，大部分写法都或多或少有一些不足。下面将分别对这几种写法进行介绍。

（1）饿汉模式

public class Singleton{

private static Singleton instance = new Singleton();

private Singleton(){}

public static Singleton newInstance(){

return instance;

}

}

（2）懒汉模式

public class Singleton{

private static Singleton instance = null;

private Singleton(){}

public static Singleton newInstance(){

if(null == instance){

instance = new Singleton();

}

return instance;

}

}

（3）双重校验锁

public class Singleton {

private static Singleton instance = null;

private Singleton(){}

public static Singleton getInstance() {

if (instance == null) {

synchronized (Singleton.class) {

if (instance == null) {//2

instance = new Singleton();

}

}

}

return instance;

}

}

## 2.2 Android介绍

### 2.2.1 Android的架构

Android的本质就是在标准的Linux系统上增加了Java虚拟机Dalvik，并在Dalvik虚拟机上搭建了一个JAVA的application framework，所有的应用程序都是基于JAVA的application framework之上。  
 Android分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和linux核心层。

（1）应用程序

所有的应用程序都是使用JAVA语言编写的，每一个应用程序由一个或者多个活动组成，活动必须以Activity类为超类，活动类似于操作系统上的进程，但是活动比操作系统的进程要更为灵活，与进程类似的是，活动在多种状态之间进行切换。利用JAVA的跨平台性质，基于Android框架开发的应用程序可以不用编译运行于任何一台安装有android系统的平台，这点正是Android的精髓所在。

（2） 应用程序框架

架构设计简化了组件的重用；任何一个应用程序都可以发布它的功能块并且任何其它的应用程序都可以使用其所发布的功能块（不过得遵循框架的安全性限制）。帮助程序员快速的开发程序，并且该应用程序重用机制也使用户可以方便的替换程序组件。

（3）系统运行库

a)程序库：Android包含一些C/C++库，这些库能被Android系统中不同的组件使用。它们通过Android应用程序框架为开发者提供服务。  
    b)Android 运行库：Android 包括了一个核心库，该核心库提供了JAVA编程语言核心库的大多数功能。

（4） Linux 内核

Android 的核心系统服务依赖于 Linux 2.6 内核 ，如安全性，内存管理，进程管理， 网络协议栈和驱动模型。 Linux 内核也同时作为硬件和软件栈之间的抽象层。其外还对其做了部分修改，主要涉及两部分修改：Binder（有效的进程间通信）和 电源管理（为手持设备节省能耗）。

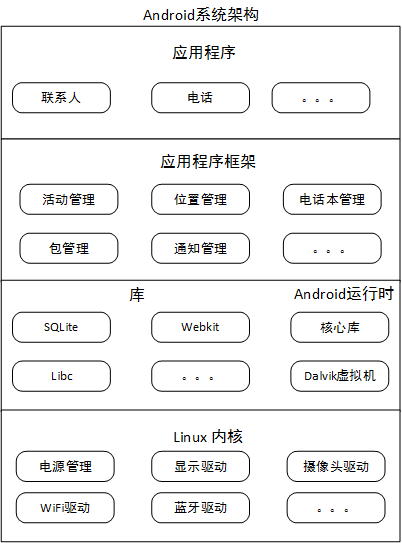


图2.1Android架构

### 2.2.2 Android之Activity生命周期

在日常应用中Activity是与用户交互的接口，它提供了一个用户完成相关操作的窗口。当我们在开发中创建Activity后，通过调用setContentView(View)方法来给该Activity指定一个布局界面，而这个界面就是提供给用户交互的接口。Android系统中是通过Activity栈的方式来管理Activity的，而Activity自身则是通过生命周期的方法来管理的自己的创建与销毁，既然如此，现在我们就来看看Activity生命周期是如何运作的。

1.启动Activity：系统会先调用onCreate方法，然后调用onStart方法，最后调用onResume，Activity进入运行状态。

2.当前Activity被其他Activity覆盖其上或被锁屏：系统会调用onPause方法，暂停当前Activity的执行。

3.当前Activity由被覆盖状态回到前台或解锁屏：系统会调用onResume方法，再次进入运行状态。

4.当前Activity转到新的Activity界面或按Home键回到主屏，自身退居后台：系统会先调用onPause方法，然后调用onStop方法，进入停滞状态。

5.用户后退回到此Activity：系统会先调用onRestart方法，然后调用onStart方法，最后调用onResume方法，再次进入运行状态。

6.当前Activity处于被覆盖状态或者后台不可见状态，即第2步和第4步，系统内存不足，杀死当前Activity，而后用户退回当前Activity：再次调用onCreate方法、onStart方法、onResume方法，进入运行状态。

7.用户退出当前Activity：系统先调用onPause方法，然后调用onStop方法，最后调用onDestory方法，结束当前Activity

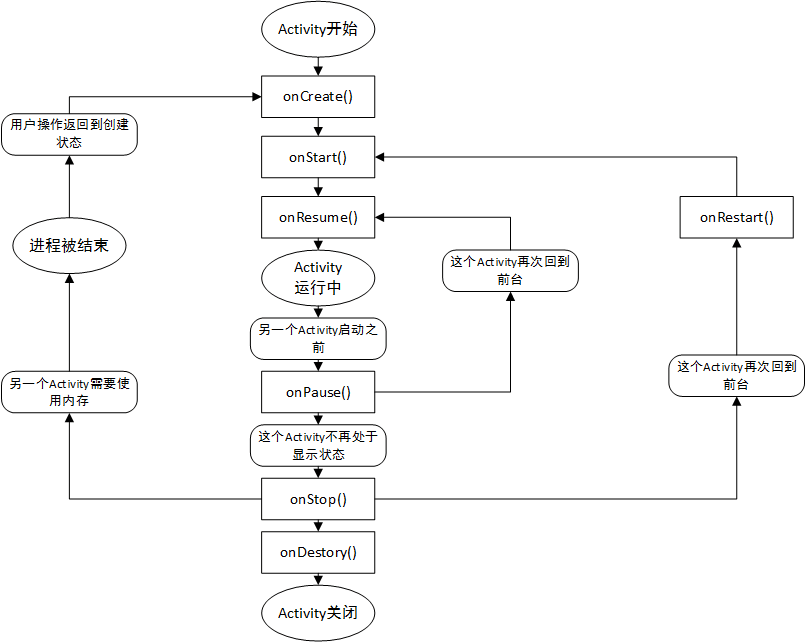


图2. 2 Activity生命周期

### 2.2.3 Android之布局

一个丰富的界面总是要由许多个控件组成的，那我们如何才能让各个控件都有条不紊地摆在界面上，而不是乱糟糟的？这就需要借助布局来实现了。布局是一种可用于放置很多控件的容器，它可以按照一定规则调整内部控件的位置，从而编写出精美的页面。当然，布局的内部除了放置控件外，也可以放置布局，通过多层布局嵌套，可以实现复杂页面。下面详细讲解了四种基本布局。

（1）线性布局（LinearLayout）

线性布局，从外框上可以理解为一个div，他首先是一个一个从上往下罗列在屏幕上。每一个LinearLayout里面又可分为垂直布局和水平布局。当垂直布局时，每一行就只有一个元素，多个元素依次垂直往下；水平布局时，只有一行，每一个元素依次向右排列。linearLayout中还有一个重要的属性 android:layout\_weight="1"，这个weight在垂直布局时，代表行距；水平的时候代表列宽；weight值越大就越大。

（2）帧布局（FrameLayout）

FrameLayout是最简单的一个布局对象。它被定制为你屏幕上的一个空白备用区域，之后你可以在其中填充一个单一对象。所有的子元素将会固定在屏幕的左上角；你不能为FrameLayout中的一个子元素指定一个位置。后一个子元素将会直接在前一个子元素之上进行覆盖填充，把它们部份或全部挡住（除非后一个子元素是透明的）。

（3）相对布局（RelativeLayout）

相对布局可以理解为某一个元素为参照物，来定位的布局方式。和LinerLayout的排列规则不同，RelativeLayout是一个允许子视图相对于其他兄弟视图或是父视图显示的视图组(通过ID指定)。每个视图的位置能够指定它相对于兄弟(比如在其他视图的左边或是下边)或是父视图(这里是指相对布局容器，比如底部对齐、中间偏左)的位置。

（4）百分比布局（PercentFrameLayout）

　　前面介绍的3种布局，满足了绝大多数场景的页面设计需求，但只有LinearLayout支持使用layout\_weight属性来实现按比例指定控件大小的功能，其他两种布局都不支持。

为此，Android引入了一种全新的布局方式来解决此问题—百分比布局。在这种布局中，我们可以不再使用wrap\_content，match\_parent等方式来指定控件大小，而是允许直接指定控件在布局中所占的百分比，这样的话就可以轻松实现平分布局甚至是任意比例分割布局的效果了。

### 2.2.4 SQLite数据库

Android系统内置了数据库SQLite。SQLite是一款轻量级的关系型数据库，它的运算速度非常快，占用资源很少，通常只需要几百kb的内存就足够了，因而特别适合在移动设备上使用。SQLite不仅支持标准的SQL语法，还遵循了数据库的ACID事务，所以只要你以前使用过其他的关系型数据库，就可以很快地上手SQLite。而SQLite又比一般的数据库简单的多，它甚至不用设置用户名和密码就可以使用了。

Android为了让我们能够更加方便地管理数据库，专门提供了一个SQLiteOpenHelper帮助类，借助这个类就可以非常简单地对数据库进行创建和升级。

## 2.3 XMPP协议解析

拒绝服务攻击因为其纷繁复杂的攻击种类，很难用单一标准去进行分门别类，可以根据自动化程度分类[7]，可以用攻击的网络模型来分类，也可以用攻击在IOS模型的第几层来分类。本文通过按攻击时的主要技术进行分类。

### 2.3.1 XMPP协议简介

XMPP（Extensible Messageing and Presence Protocol：可扩展消息与存在协议）是目前主流的四种IM（IM：instant messaging,即时消息）协议之一，其他三种分别为：即时信息和空间协议(IMPP)、空间和即时信息协议(PRIM)、针对即时通讯和空间平衡扩充的进程开始协议SIP(SIMPLE)。

在这四种协议中，XMPP是最灵活的。XMPP是一种基于XML的协议，它继承了在XML环境中灵活的发展性。它有以下优点：

1.XMPP 协议是公开的，由JSF开源社区组织开发的。XMPP 协议并不属于任何的机构和个人，而是属于整个社区，这一点从根本上保证了其开放性。

2. XMPP 协议具有良好的扩展性。在XMPP 中，即时消息和到场信息都是基于XML 的结构化信息，这些信息以XML 节(XML Stanza)的形式在通信实体间交换。XMPP 发挥了XML 结构化数据的通用传输层的作用，它将出席和上下文敏感信息嵌入到XML 结构化数据中，从而使数据以极高的效率传送给最合适的资源。基于XML 建立起来的应用具有良好的语义完整性和扩展性。

3.分布式的网络架构。XMPP 协议都是基于Client/Server 架构，但是XMPP协议本身并没有这样的限制。网络的架构和电子邮件十分相似，但没有结合任何特定的网络架构，适用范围非常广泛。

4. XMPP 具有很好的弹性。XMPP 除了可用在即时通信的应用程序，还能用在网络管理、内容供稿、协同工具、档案共享、游戏、远端系统监控等。

5.安全性。XMPP在Client-to-Server通信，和Server-to-Server通信中都使用TLS (Transport Layer Security)协议作为通信通道的加密方法，保证通信的安全。任何XMPP服务器可以独立于公众XMPP网络（例如在企业内部网络中），而使用SASL及TLS等技术更加增强了通信的安全性。如下图所示：

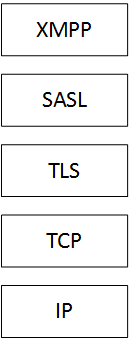


图2.3XMPP安全性

### 2.3.2 XMPP协议内容

XMPP传输的是与即时通讯相关的指令。在以前这些命令要么用2进制的形式发送（比如QQ），要么用纯文本指令加空格加参数加换行苻的方式发送（比如MSN）。而XMPP传输的即时通讯指令的逻辑与以往相仿，只是协议的形式变成了XML格式的纯文本。这不但使得解析容易了，人也容易阅读了，方便了开发和查错。而XMPP的核心部分就是一个在网络上分片断发送XML的流协议。这个流协议是XMPP的即时通讯指令的传递基础，也是一个非常重要的可以被进一步利用的网络基础协议。所以可以说，XMPP用TCP传的是XML流。

所有从一个client到另一个client的jabber消息和数据都要通过xmpp server。

1. client连接到server
2. server利用本地目录系统的证书对其认证
3. client制定目标地址，让server告知目标状态
4. server查找，连接并进行相互认证
5. client间进行交互

主要的XMPP 协议范本及当今应用很广的XMPP 扩展：

RFC 3920 XMPP：核心。定义了XMPP 协议框架下应用的网络架构，引入了XML Stream（XML 流）与XML Stanza（XML 节），并规定XMPP 协议在通信过程中使用的XML 标签。使用XML 标签从根本上说是协议开放性与扩展性的需要。此外，在通信的安全方面，把TLS 安全传输机制与SASL 认证机制引入到内核，与XMPP 进行无缝的连接，为协议的安全性、可靠性奠定了基础。Core 文档还规定了错误的定义及处理、XML 的使用规范、JID（Jabber Identifier，Jabber 标识符）的定义、命名规范等等。所以这是所有基于XMPP 协议的应用都必需支持的文档。

RFC 3921：用户成功登陆到服务器之后，发布更新自己的在线好友管理、发送即时聊天消息等业务。所有的这些业务都是通过三种基本的XML 节来完成的：IQ Stanza（IQ 节）, Presence Stanza（Presence 节）, Message Stanza（Message 节）。 RFC3921 还对阻塞策略进行了定义，定义是多种阻塞方式。可以说，RFC3921 是RFC3920 的充分补充。两个文档结合起来，就形成了一个基本的即时通信协议平台，在这个平台上可以开发出各种各样的应用。

XEP-0030 服务搜索。一个强大的用来测定XMPP 网络中的其它实体所支持特性的协议。

XEP-0115 实体性能。XEP-0030 的一个通过即时出席的定制，可以实时改变交变广告功能。

XEP-0045 多人聊天。一组定义参与和管理多用户聊天室的协议，类似于Internet 的Relay Chat，具有很高的安全性。

XEP-0096 文件传输。定义了从一个XMPP 实体到另一个的文件传输。

XEP-0124 HTTP 绑定。将XMPP 绑定到HTTP 而不是TCP，主要用于不能够持久的维持与服务器TCP 连接的设备。

XEP-0166 Jingle。规定了多媒体通信协商的整体架构。

XEP-0167 Jingle Audio Content Description Format。定义了从一个XMPP 实体到另一个的语音传输过程。

XEP-0176 Jingle ICE（Interactive Connectivity Establishment）Transport。ICE传输机制，文件解决了如何让防火墙或是NAT（Network Address Translation）保护下的实体建立连接的问题。

XEP-0177 Jingle Raw UDP Transport。纯UDP 传输机制，文件讲述了如何在没有防火墙且在同一网络下建立连接的。

XEP-0180 Jingle Video Content Description Format。定义了从一个XMPP 实体到另一个的视频传输过程。

XEP-0181 Jingle DTMF（Dual Tone Multi-Frequency）。

XEP-0183 Jingle Telepathy Transport Method。

### 2.3.3 XMPP地址格式

一个实体在XMPP网络结构中被称为一个接点，它有唯一的标示符jabber identifier(JID)，即实体地址，用来表示一个Jabber用户，但是也可以表示其他内容，例如一个聊天室．一个有效的JID包括一系列元素：(1)域名(domain identifier)；(2)节点(node identifier)；(3)源(resource identifier)．它的格式是node@domain/resource，node@domain，类似电子邮件的地址格式．domain用来表示接点不同的设备或位置，这个是可选的，例如a在Server1上注册了一个用户，用户名为doom，那么a的JID就是doom@serverl，在发送消息时，指明doom@serverl就可以了，resource可以不用指定，但a在登录到这个Server时，fl的JID可能是doom@serverl、exodus(如果a用Exodus软件登录)，也可能是doom@serverl/psi(如果a用psi软件登录)．资源只用来识别属于用户的位置或设备等，一个用户可以同时以多种资源与同一个XMPP服务器连接。

XMPP中定义了 3个顶层XML元素: Message、Presence、IQ，下面针对这三种元素进行介绍。Message用来表示传输的消息，当用户发送一条消息时。就会在流的上下文中插入一个Message元素，中间有用户发送的相关信息；Presence用来表示用户的状态。当用户改变自己的状态时。就会在数据流的上下文中插入一个Presence元素，用来表示用户现在的状态；IQ用来表示一种请求，响应机制，从一个实体发送请求，另外一个实体接受请求并响应。

（1）<Message>

用于在两个jabber用户之间发送信息。Jsm(jabber会话管理器)负责满足所有的消息，不管目标用户的状态如何。如果用户在线jsm立即提交;否则jsm就存储。

To :标识消息的接收方。

from : 指发送方的名字或标示(id)o

Text: 此元素包含了要提交给目标用户的信息。

结构如下所示:

<message to= [‘lily@jabber.org/contact’](mailto:%E2%80%98lily@jabber.org/contact%E2%80%99) type =’chat’>

<body> 你好，在忙吗</body>

</message>

（2） <Presence>

用来表明用户的状态，如：online、away、dnd(请勿打扰)等。当用户离线或改变自己的状态时，就会在stream的上下文中插入一个Presence元素，来表明自身的状态．结构如下所示：

<presence>

From =‘lily @ jabber.com/contact’

To = ‘yaoman @ jabber.com/contact'

<status> Online </status>

</presence>

<presence>元素可以取下面几种值:Probe :用于向接受消息方法发送特殊的请求.subscribe:当接受方状态改变时，自动向发送方发送presence信息。

（3）< IQ >

一种请求／响应机制，从一个实体从发送请求，另外一个实体接受请求，并进行响应．例如，client在stream的上下文中插入一个元素，向Server请求得到自己的好友列表，Server返回一个，里面是请求的结果．

<iq > 主要的属性是type。包括:Get :获取当前域值。Set :设置或替换get查询的值。Result :说明成功的响应了先前的查询。Error: 查询和响应中出现的错误。

结构如下所示:

<iq from =‘lily @ jabber.com/contact’id=’1364564666’ Type=’result’>

# 第三章 系统需求分析

## 3.1客户端分析

### 3.1.1注册和登录模块

注册模块：用户从客户端输入账号，邮箱和密码这三个字段，在提交到服务器之前，对这三个字段的格式和类型进行检测判断，比如邮箱的格式，以及密码的长度限制，在满足要求后，提交到服务器进行注册功能。

登录模块：用户注册后，会把用户信息保存在服务器的数据库中，在进行登录操作后，会从服务器的数据库中查找是否有该用户存在，如果是，则登录成功，否则登录失败。

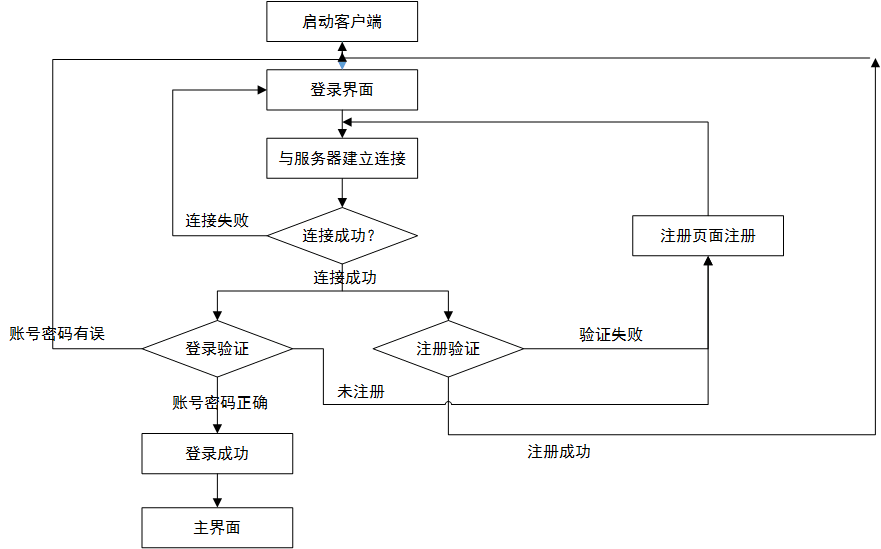


图3.1注册登录模块

### 3.1.2好友模块

好友模块：主要包括查找好友，新增好友，删除好友等功能。登录成功的用户，可以在好友搜索页面添加自己想要加的好友姓名，如果存在，则可以显示出来，并添加为好友，否则显示该好友不存在

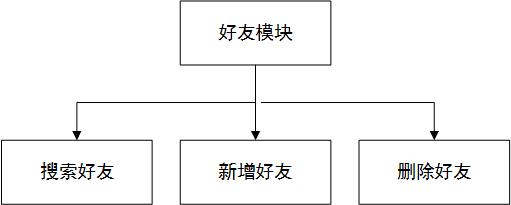


图3. 2好友模块

### 3.1.3消息模块

聊天模块：用户登陆后，在选择自己的好友后，可以进行一对一的聊天。用户可以发送文字，表情，语音，图片等，聊天记录保存在客户端。在下次打开聊天页面时，会显示历史记录。

（1）发送信息

在添加好友后，用户可以选择任意一个好友进行文字对话。在通信录展示页，点击要交谈的用户，打开聊天界面。然后在会话框中输入自己想要发送的信息，点击发送按钮，就会发送出去。可以发送文字，图片，语音等。

（2）接收信息

登陆成功后，如果有其他好友发来的信息，会有提示音，提示用户查看，在首页，会出现其他好友发来的信息，显示条数。用户可以点开这个界面，查看具体信息。

（3）处理历史信息

在聊天过程中，每次用户发出聊天的内容或者收到聊天的内容，会保存在Android自带的SQLite数据库中，在下次访问该聊天界面时，会从数据库中查询历史记录，并展示出来。

### 3.1.4 信息修改模块

用户可以在注册成功后，修改自己的相关信息，主要包括两个模块：修改密码模块和修改基本信息模块。在修改密码时，用户需要输入自己的原始密码，以及两次新密码。在修改基本信息时，用户可以修改自己的头像，昵称等。

## 3.2服务端分析

### 3.2.1登录功能

用户必须登录成功后才能进入到系统中，下面是登录的流程图：

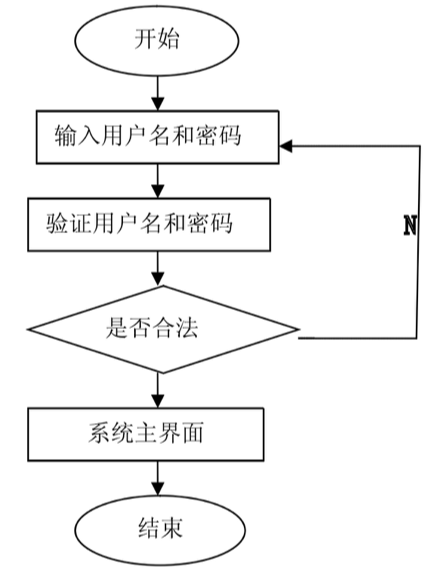


图3.3登录功能

### 3.2.2界面设置

用户可以通过web页面直观的感受到用户的在线情况，可以对这些用户进行在线的增删改查操作

### 3.2.3消息设置

服务器要对客户端发送过来的数据进行解析，按照注册，登录，聊天等不同功能的不同数据要进行响应，并返回数据进行验证。而且，系统要支持离线消息的发送，让离线消息在服务器上有一定时间的缓存，待用户上线在进行发送操作，超过一定时间则删除不再处理。

# 第四章 系统设计与实现

## 4.1服务器设计

XMPP Server拥有众多的实现方案，其中，较为成熟，且使用广泛的是ejabberd, jabberd 2.x, Openfire, Tigase。Openfire是基于JAVA的解决方案，基于XMPP协议、开源的实时协作的服务器。Openfire具有跨平台的能力。因此，我选择了openfire这个服务器端。

Openfire的通信处理基于Apache MINA框架实现。Apache MINA是一个网络应用程序框架，用来帮助用户简单地开发高性能和高可靠性的网络应用程序。它提供了一个通过[Java](http://lib.csdn.net/base/java)NIO在不同的传输例如TCP/IP和UDP/IP上抽象的事件驱动的异步API。

### 4.1.1系统启动流程

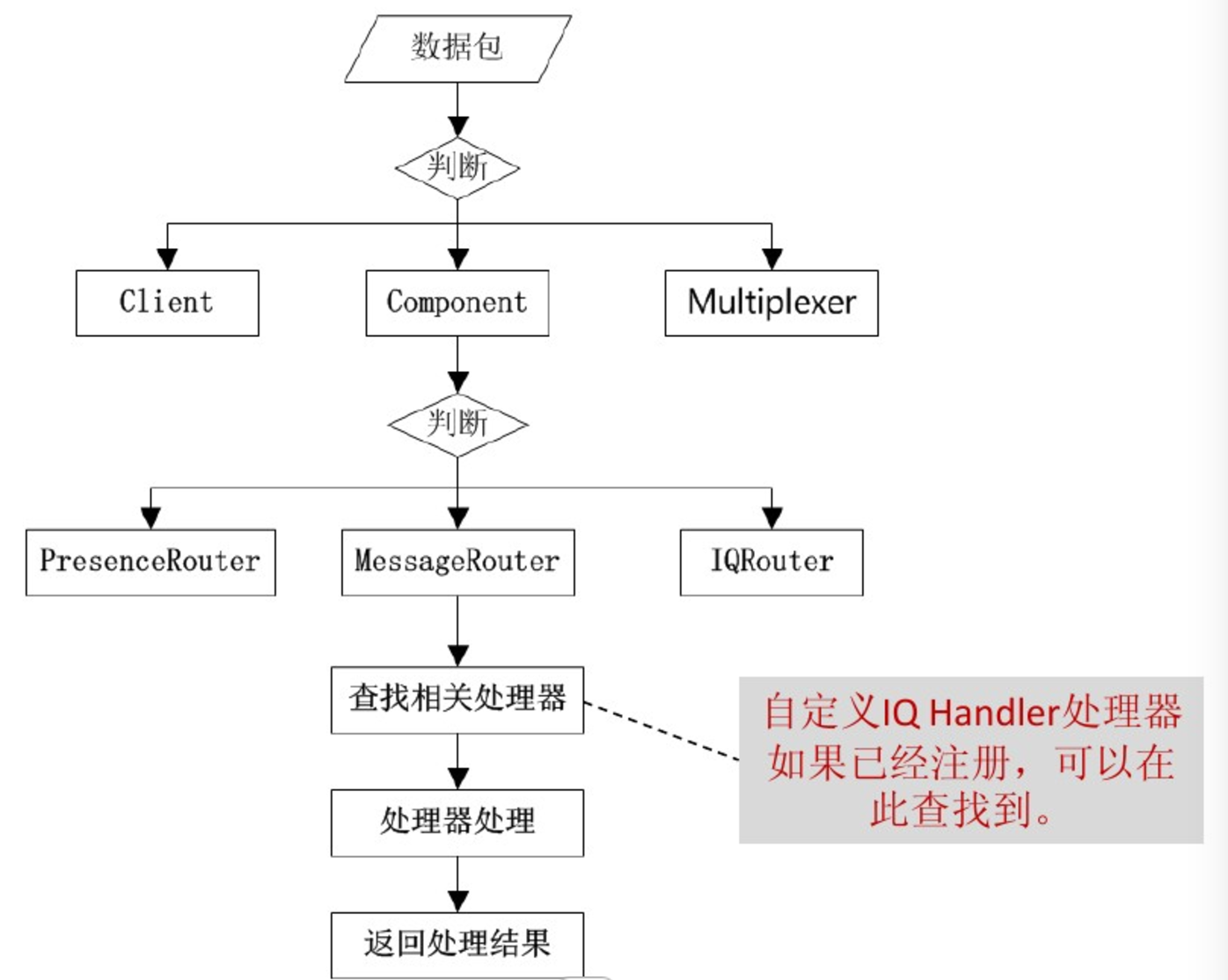
系统启动时调用ServerStarter类中的start()方法，通过反射加载org.jivesoftware.openfire.XMPPServer类文件，创建实例时调用其构造函数，在其构造函数中调用其start()方法实际启动服务应用程序。Start()方法中首先调用verifyDataSource()方法验证并确保数据库可以访问，然后会调用loadModules();initModules();startModules();方法来对Module接口的实现类的各子类进行操作，依次完成模块的加载、初始化和启动操作。

loadModules()方法中会调用loadModule(String module)方法通过反射加载各模块类，参数字符串module为对应的模块核心处理类的类名,如AdHocCommandHandler。现以AdHocCommandHandler为例对接下来的处理流程进行说明。通过loadModule创建AdHocCommandHandler类实例时调用其构造函数，在构造函数中初始化了其私有AdHocCommandManager对象。在initModules()时调用AdHocCommandHandler实例的initialize(XMPPServer server)方法对其私有属性对象进行初始化。然后调用start()方法，调用addDefaultCommands方法添加命令并启动命令（通过调用startCommand(AdHocCommand command)方法实现）。

### 4.1.2网络处理

消息监听服务：SSL等监听服务的调度在ConnectionManagerImpl类中实现。ConnectionManagerImpl.createClientSSLListeners()方法启动SSL监听

消息封装：信息处理采用XML节的方式传递信息，消息封装通常采用IQ、Message、Presence。

图4.1Openfire消息包接受处理流程

### 4.1.3数据库处理

Openfire的数据库处理采用直接调用JDBC 的方式。核心类为org.jivesoftware.database.DbConnectionManager。数据库的处理与业务处理耦合，没有划分出专门的业务逻辑层。

通常直接调用XXManager中的实例方法，XXManager中又调用的是对应的接口XXProvider的方法，实际操作在该接口的实现类中实现。实现类是动态绑定的（默认的实现类通常命名规则为DefaultXXProvider），在运行时根据ofproperty表中对应配置项值选择。下面以添加用户组为例进行说明。

    首先获得GroupManager的一个实例，在调用其构造函数时调用initProvider()方法，在该方法中获取数据库中配置项的值，若不为空则根据该值通过反射机制获取GroupProvider接口的实现类实例对象；若为空则以DefaultGroupProvider作为GroupProvider接口的实现类并创建实例对象，然后调用GroupProvider.createGroup(String name)方法完成业务操作。

### 4.1.4页面处理

Openfire没有采用现在很流行的技术架构（SSH），只使用JSP+JavaBean，但是它有自己的系统设计，就连日志都是自己做的，没有使用我们熟悉的log4j。

现有的Openfire管理控制台可采用插件方式进行扩展（详见插件开发说明部分介绍），页面采用Jsp方式实现，页面直接调用业务处理逻辑类（通常命名为XXManager）的实例方法，通常通过request对象封装的方式传递页面展现判定变量，常出现本页跳转。每个插件可定义自己的Servlet类和web.xml及web-custom.xml配置文件。

    采用装饰框架方式展现页面，decorator页面有两个，即src/web/decorators目录下的两个页面main.jsp和setup.jsp。采用自定义的admin标签实现，标签库admin.tld放置在src/web/WEB-INF目录下，标签解析类放置在org.jivesoftware.admin包下，有SidebarTag、SubnavTag、SubSidebarTag、TabsTag四个解析类。在调用loadPlugin()方法进行插件加载时，解析插件的plugin.xml配置文件，将获取的相关信息封装在AdminConsole类的generatedModel对象中，后期通过插件解析类提取该对象中的数据并配合sitemesh装饰器进行页面展现。

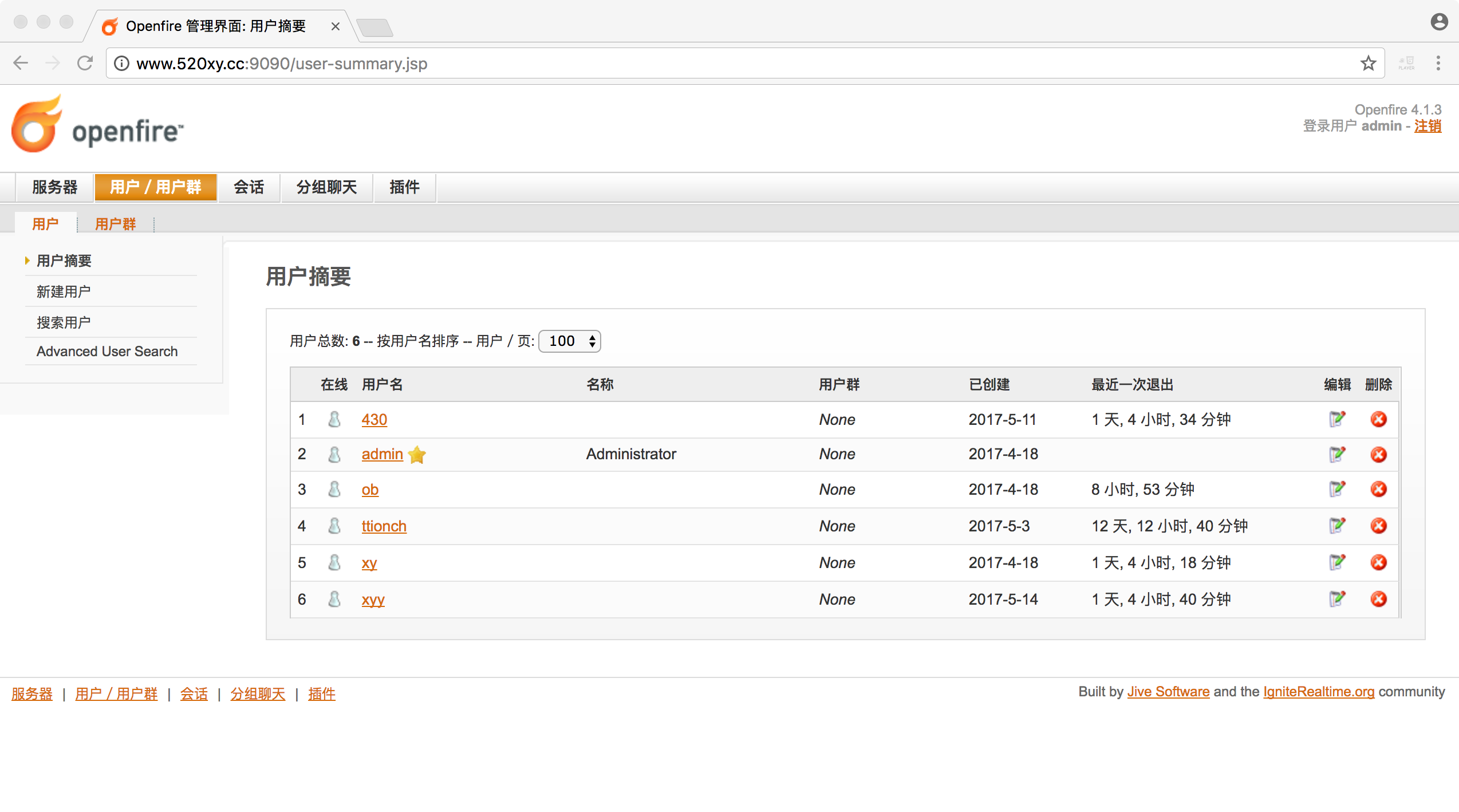


图4.2 Openfire界面

## 4.2客户端设计

Smack是一个开源，易于使用的XMPP（jabber）客户端类库。Asmack是Smack库的Android版本,虽然Smack在PC上可以工作的很好,功能也很强大,但在Android平台上有一些问题,而导致这些问题的原因是Android精简了Java的类库,以至Smack使用的部分类库在Android平台上无法找到,所以Smack不能直接在Android平台上使用。所以在客户端中，使用Asmack这个开源库来完成XMPP的相关操作。

### 4.2.1连接服务器

用户进行任何操作的前提是，客户端连上服务器。客户端通过服务器的ip地址，端口号，域，来连接到服务器。

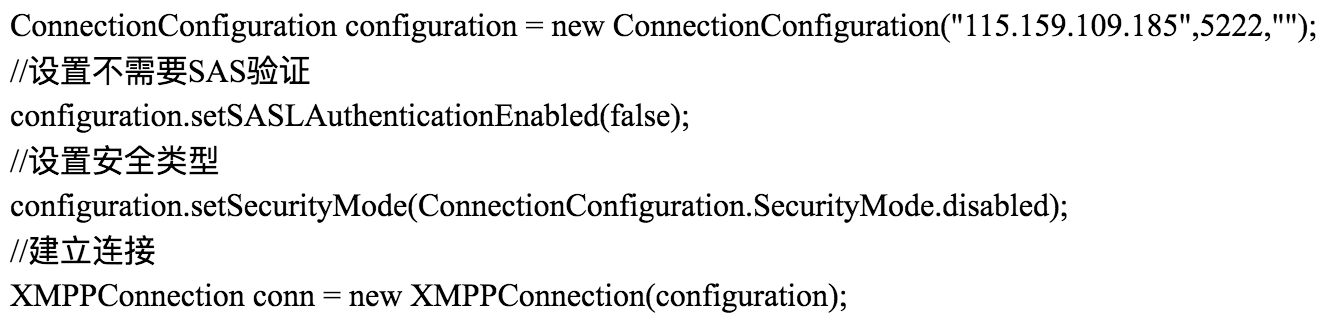


图4.3连接服务器的代码实现

### 4.2.2用户注册

在注册页面，用户需要按照要求填写信息，包括账号，邮箱和密码，用户填写的邮箱必须符合邮箱的规范，密码要求6位以上，而且要输入两次进行确认，防止出错。在通过检测没问题后，会给服务端发送一个注册请求。

服务端会检验用户名是否重复，如果没有重复，系统会提示“登录成功”，并客户端进入登录状态；否则，提示“登录失败”，让用户再次注册。下图展示了用户发送给服务端的注册请求。

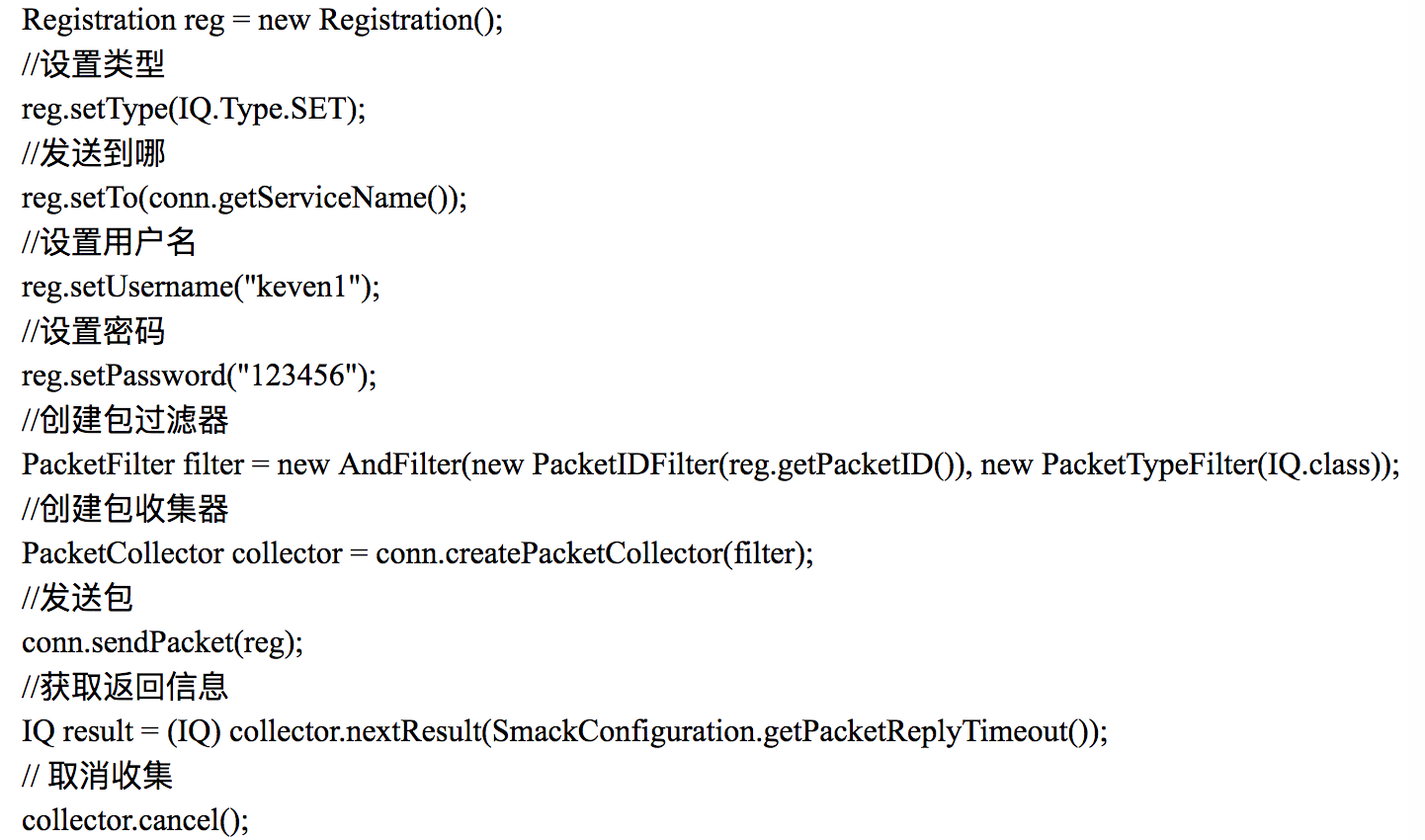


图4.4用户注册的代码实现

### 4.2.3 用户登录

用户在注册完后，在登录界面输入用户名和密码，接着点击登录按钮进行登录。客户端会先在本地检验用户名和密码是否为空，如果为空，会提示用户完善用户名和密码后，再进行登录。如果没有问题，会向服务器发送登录请求，服务器会验证密码和账号是否正确，然后返回一个响应流。客户端根据响应流来决定用户是否登录成功。以下是登录的代码：

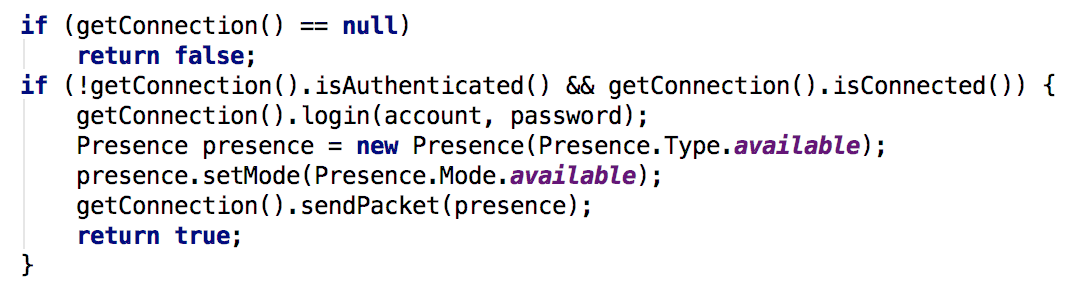


图4.5用户登录的代码实现

### 4.2.4 好友管理

好友管理总共有3个模块：好友列表，添加好友和删除好友。用户登录后，点击通信录，会向服务器发送请求，获取最新的好友列表，展示出来。想添加其他好友时，点击添加好友按钮，会出现搜索好友界面，输入关键词用户名，会出现列表，点击发送添加好友列表，只有对方同意添加时，两人才能成为好友。删除好友时，直接点击该好友，然后移除通信录即可。

（1）好友列表

（2）添加好友：用户首先要通过搜索获取好友列表，在输入关键字后，会向服务端发送请求，安装了search插件的服务端，会根据该请求返回列表。



图4.6搜索好友的代码实现

Form.getSearchForm()这个方法要求的参数是一个服务名，openfire默认的搜索服务是search.+域名。Form.setAnswer();这个方法意思是，把某个字段设成true就会在那个字段里搜索关键字，search字段设置要搜索的关键字，什么不输入不会返回。

Openfire添加好友申请通过发送Presence包实现，其中Presence包的Type含有七个状态：subscribe订阅、subscribed同意订阅、unsubscribe取消订阅、unsubscribed拒绝订阅、unavailable下线、probe探测、error错误。Roster对象，可以理解为保存好友的花名册。客户端会向服务端发送roster set的请求，然后服务端会返回一个roster result的响应，并向要添加好友的客户端发送roster push请求。

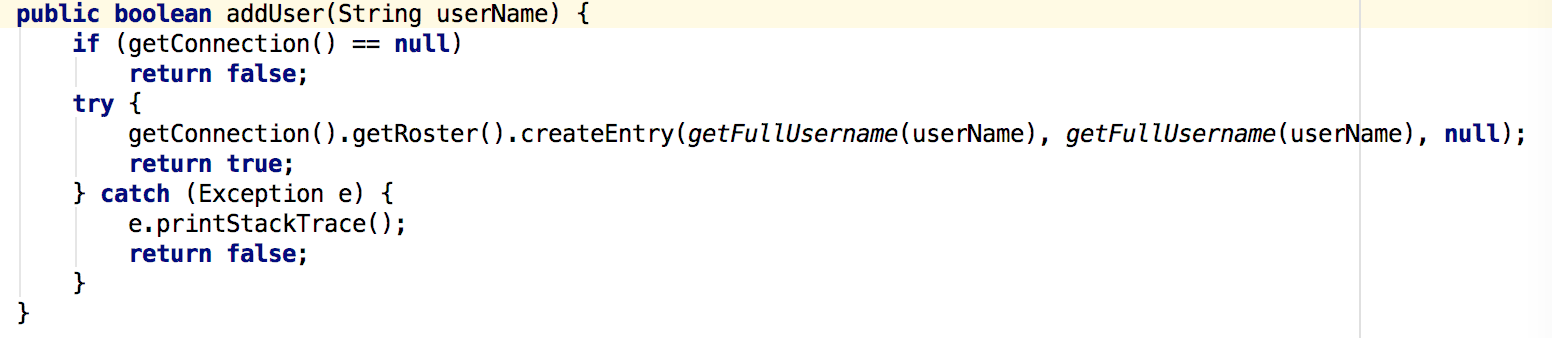


图4.7添加好友的代码实现

（3）删除好友：点击移除通信录时，客户端会向服务端发送状态为unsubscribe取消订阅的Presence包，服务器收到这个请求，会对数据库进行更新，返回一个iq result响应。

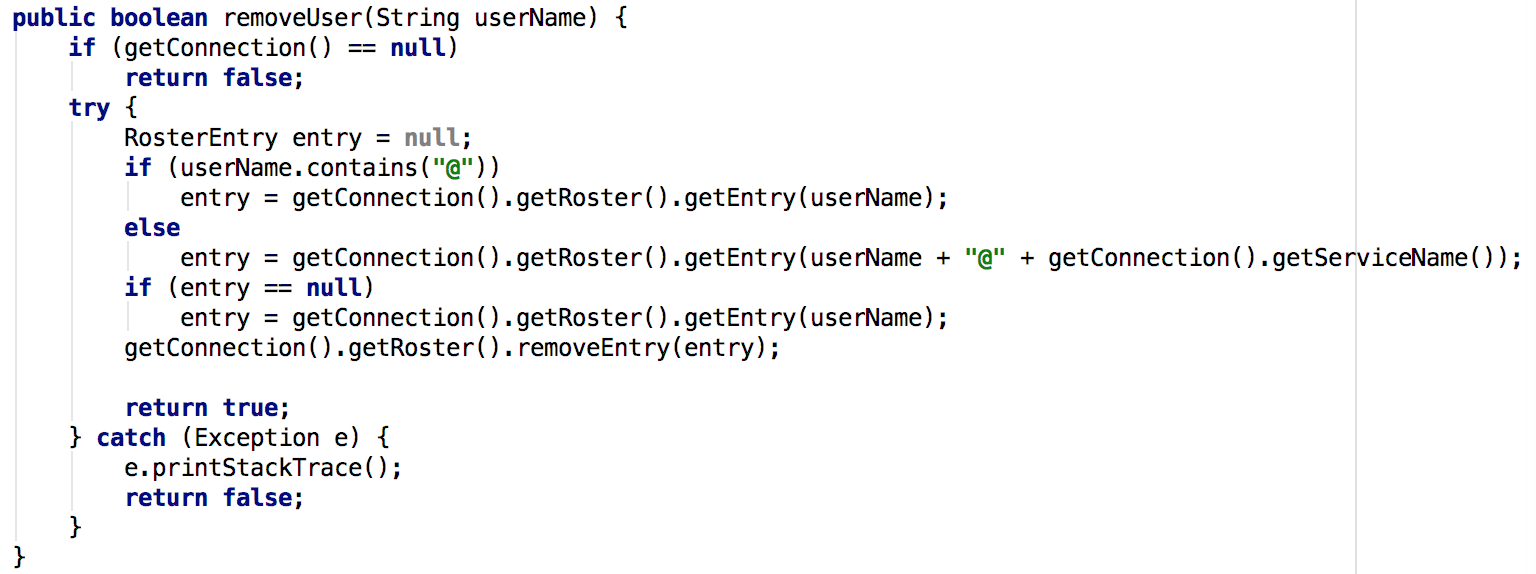


图4.8删除好友的代码实现

### 4.2.5消息传递

消息的发送主要是通过sendMessage()方法实现的。客户端将信息保存在chat对象中，然后通过sendMessage方法发送给服务器，服务器从中获得要发送的好友地址，发送给好友。接收消息是通过事件监听器来实现的，Asmack中拥有处理接收到packet的api，包括packetCollector和PacketListener这两个。



图4.9发送消息的代码实现

### 4.2.6数据库设计

在设计该系统时，需要进行一些数据的持久化处理。Android内置了数据库SQLite。由于本系统是C/S架构，客户端和服务端都要在本地保存数据，而服务端选用的是开源的openfire，会在mysql中自动生成服务端所需数据库。客户端则需要保存在SQLite中，下面是客户端所需的数据表：



表4.1信息保存表



表4.2好友表



表4.3新消息表

# 第五章 **功能测试**

本论文在XMPP协议的基础上，研究实现了基于Android的即时通信软件。在开发过程中，使用java语言进行开发，运用Android Studio进行开发，客户端使用基于XMPP实现的Asmack包来实现XMPP的各项功能，服务器则选择了开源的openfire服务器。最后，实现了可以进行用户登录注册，好友管理，消息发送与接收的Android客户端。

## 5.1用户注册测试

用户需要填写账号，邮箱，密码进行注册。客户端会判断字段是否为空，邮箱格式，以及两次密码是否一致。这个功能有四个测试用例：测试用例1是注册界面；测试用例2是当有字段为空时，会提醒输入该字段；测试用例3是会测试邮箱的格式是否正确；测试用例4是会检测两次密码是否正确。如果上述问题都不存在，则会注册成功。

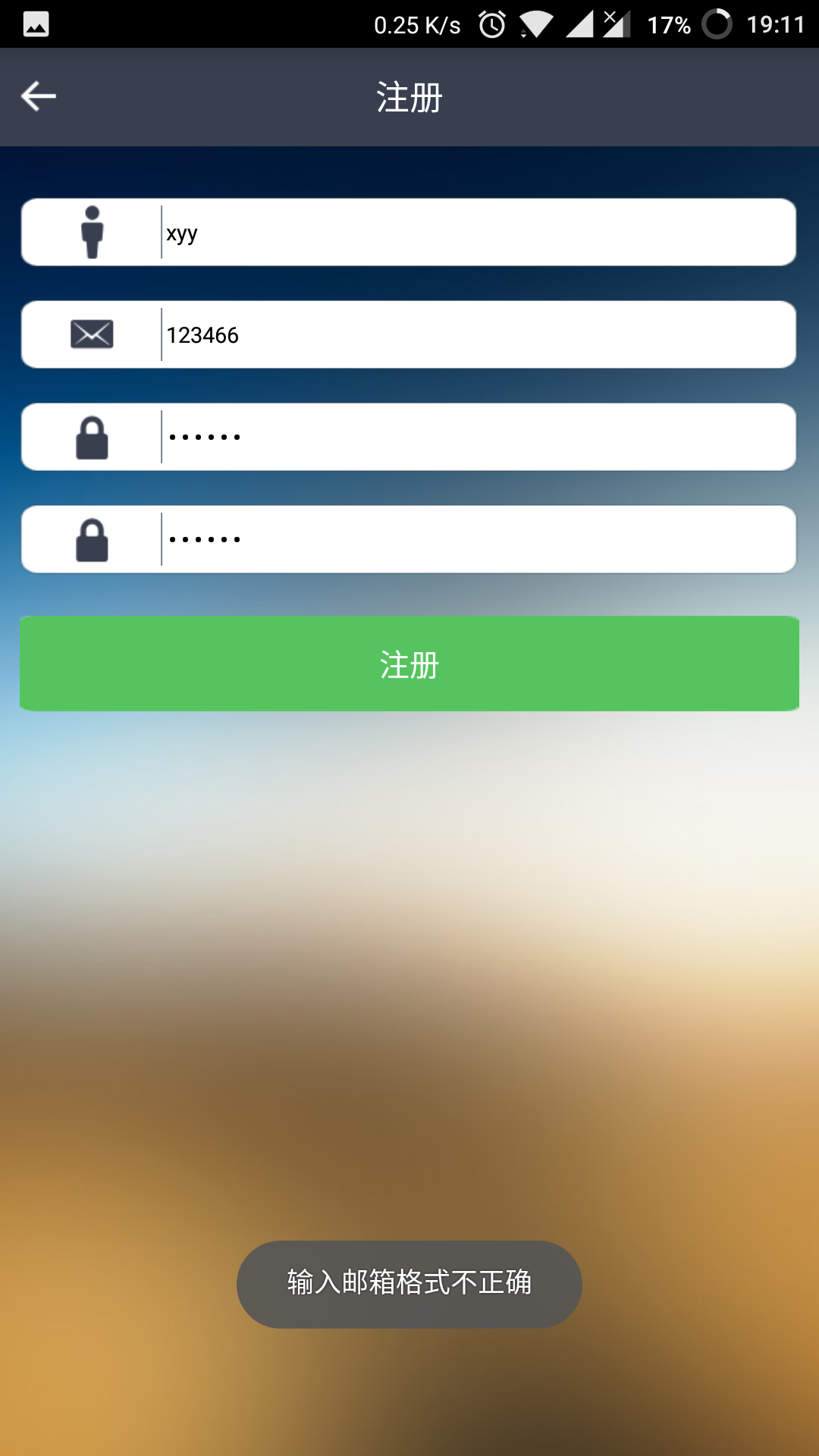
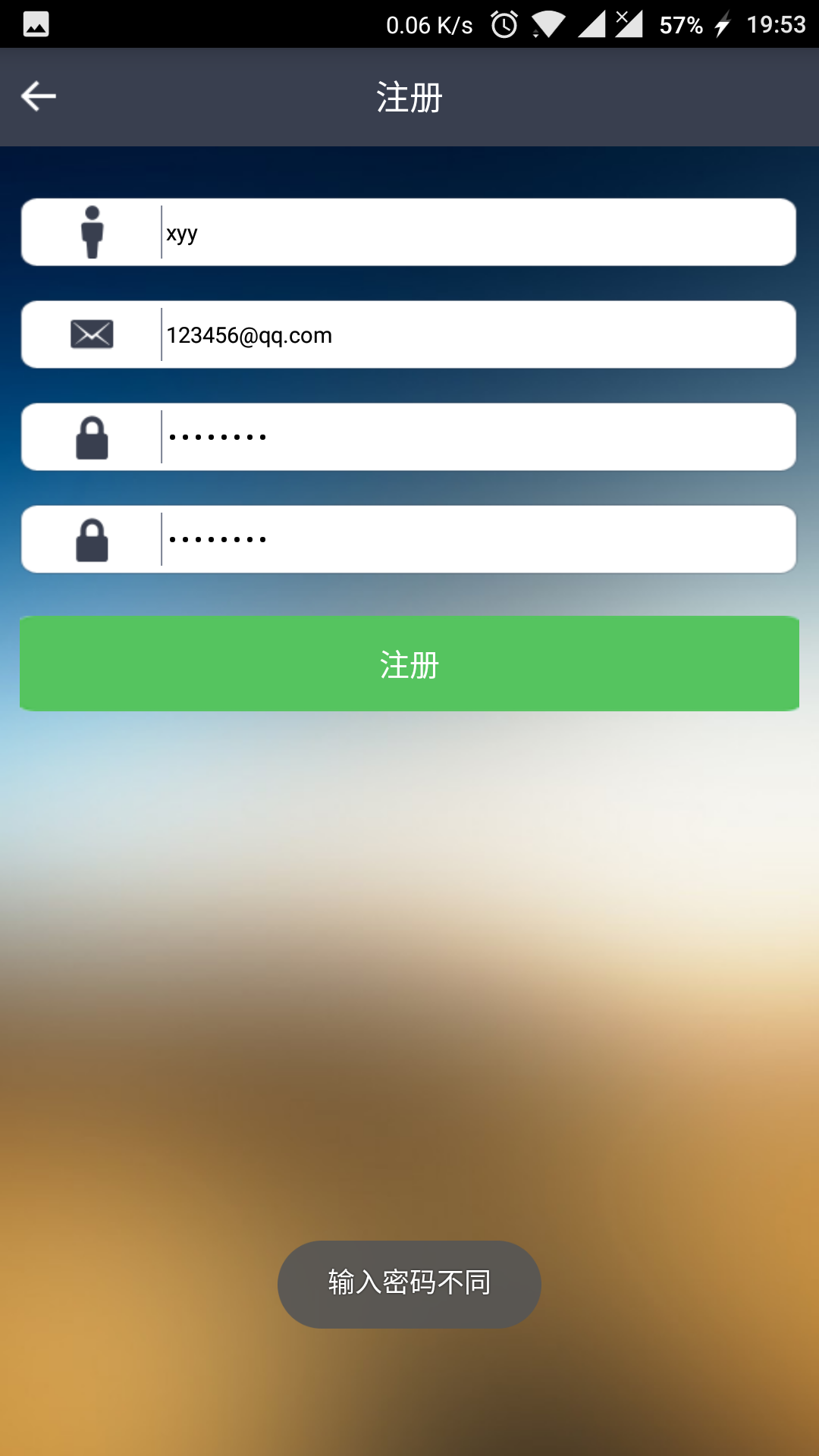
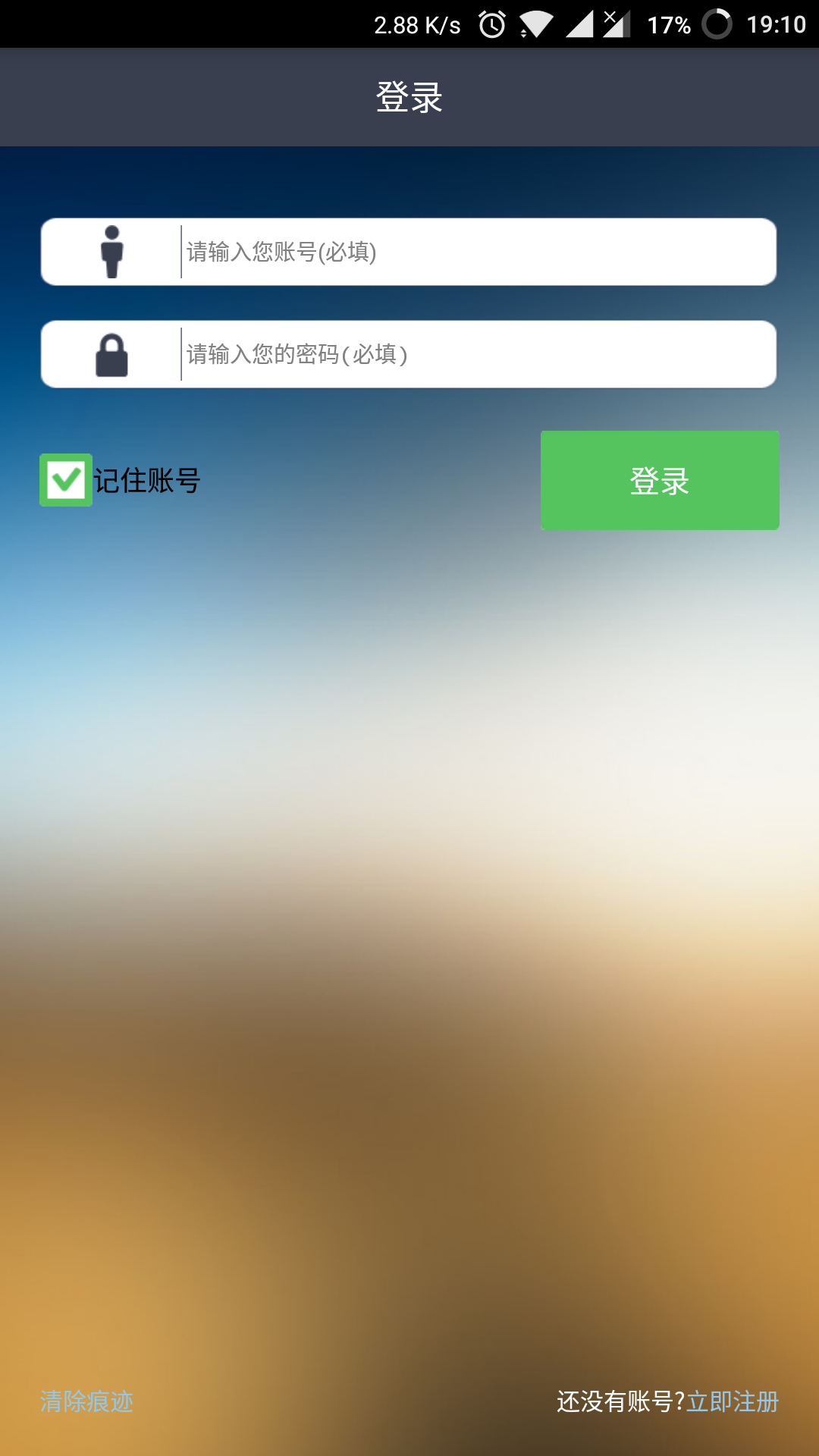
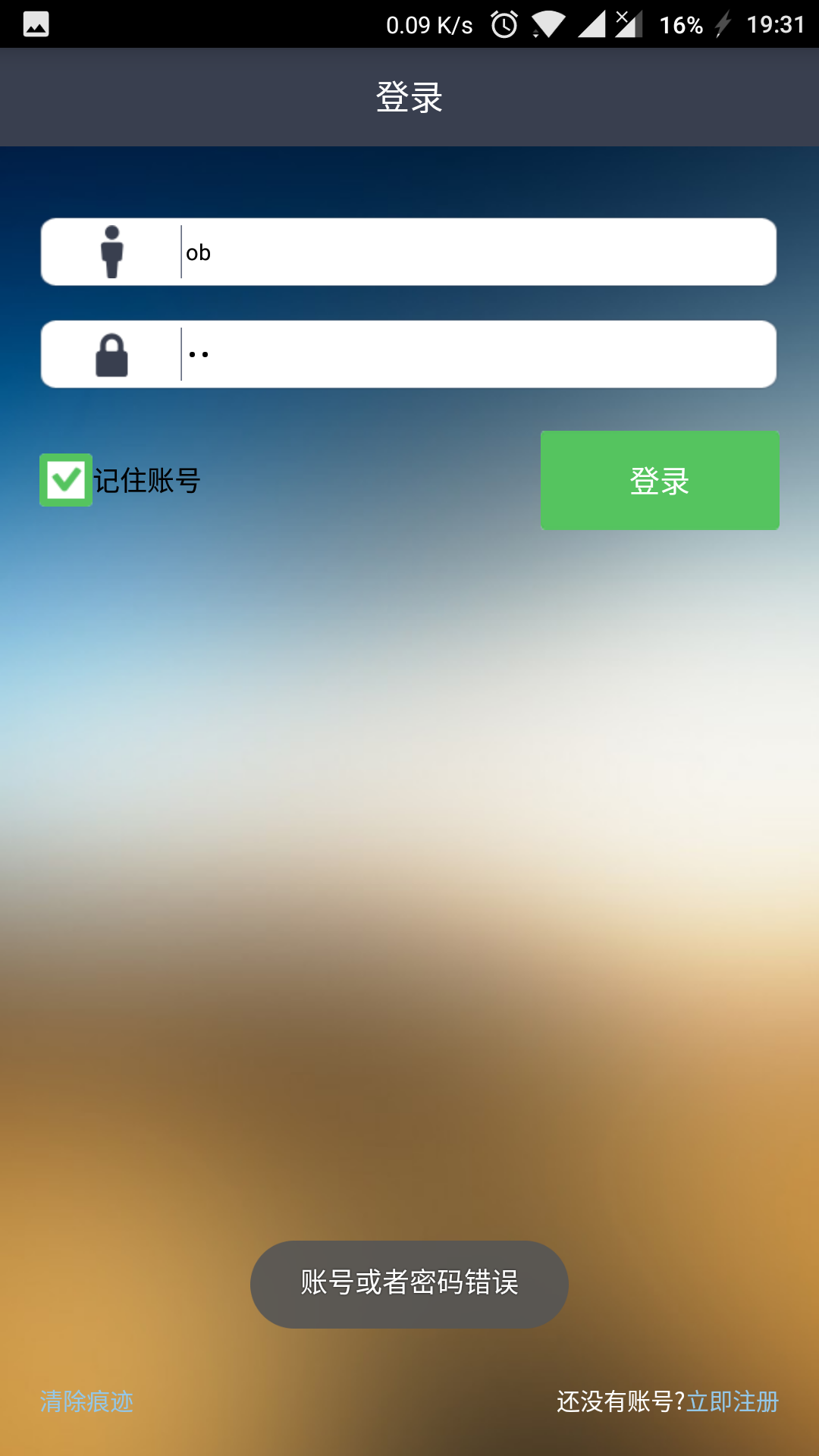
 

图5.1注册功能演示

## 5.2用户登录测试

用户在注册完后，进入登录界面，输入账号和密码。如果正确，则会进入系统，否则提示“账号和密码错误”。该部分有三个测试用例：测试用例1是进入到的登录界面；测试用例2是当账号和密码输错时，会提示；测试用例3是登录成功后，进到的界面。

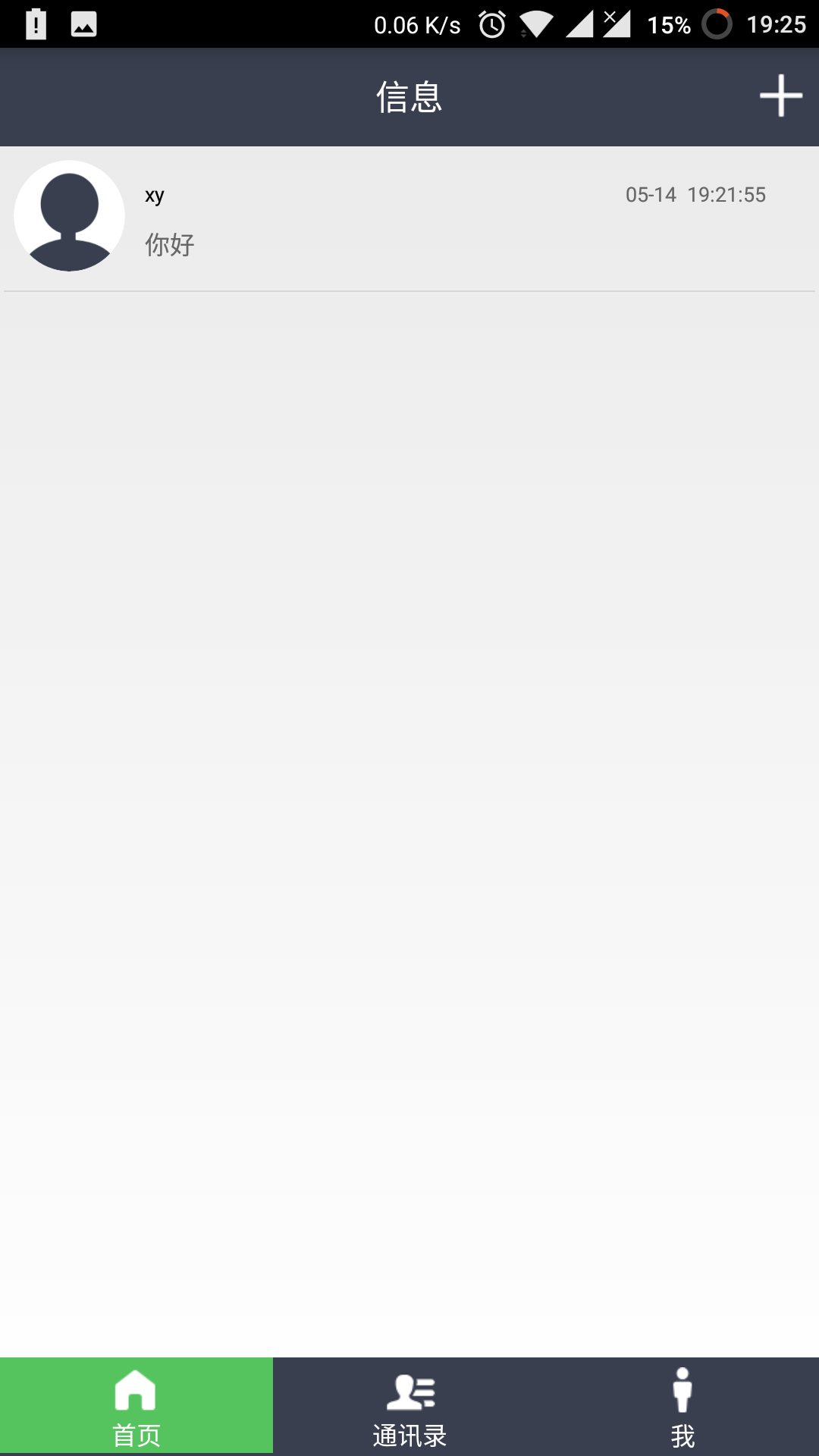
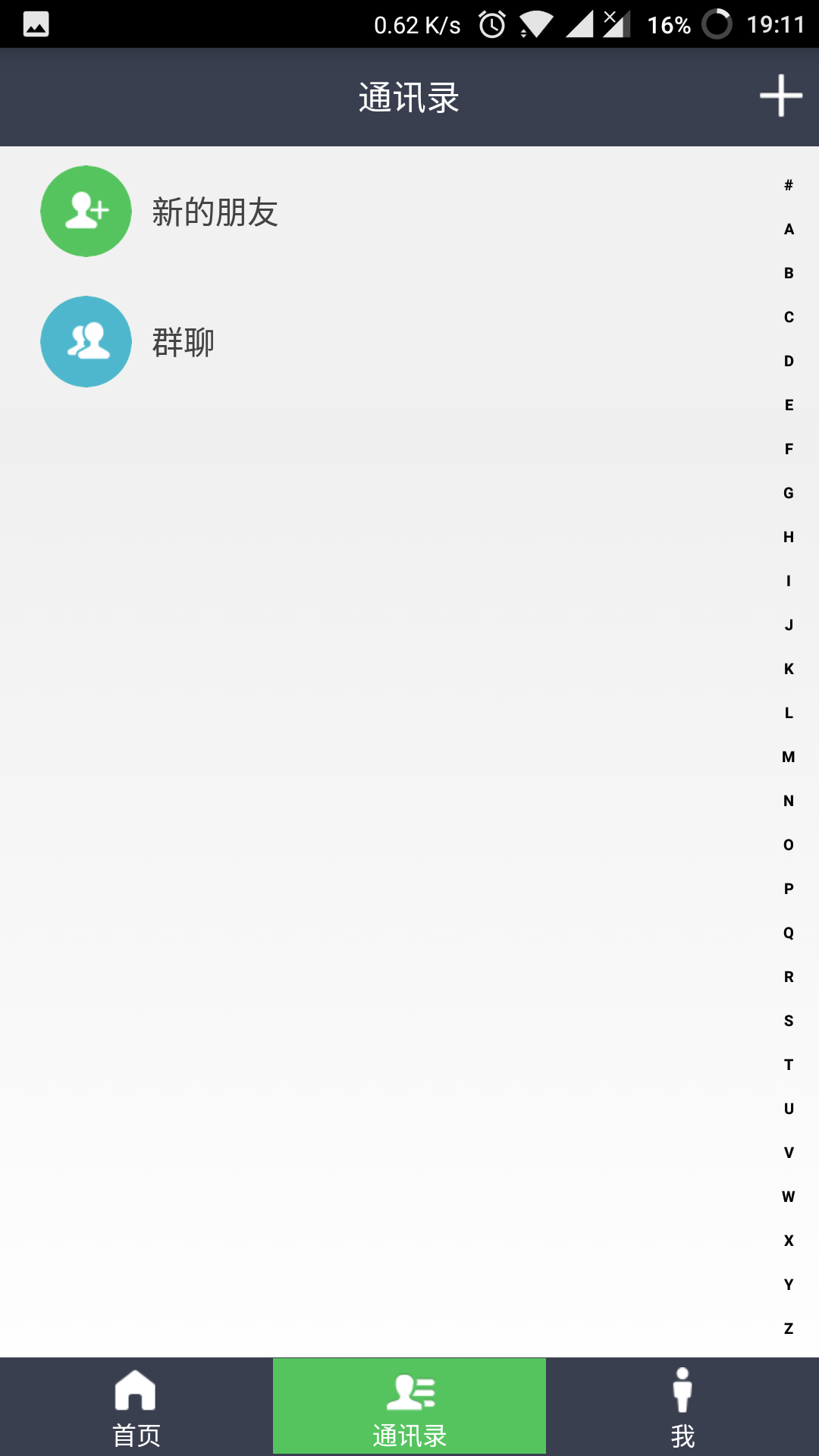
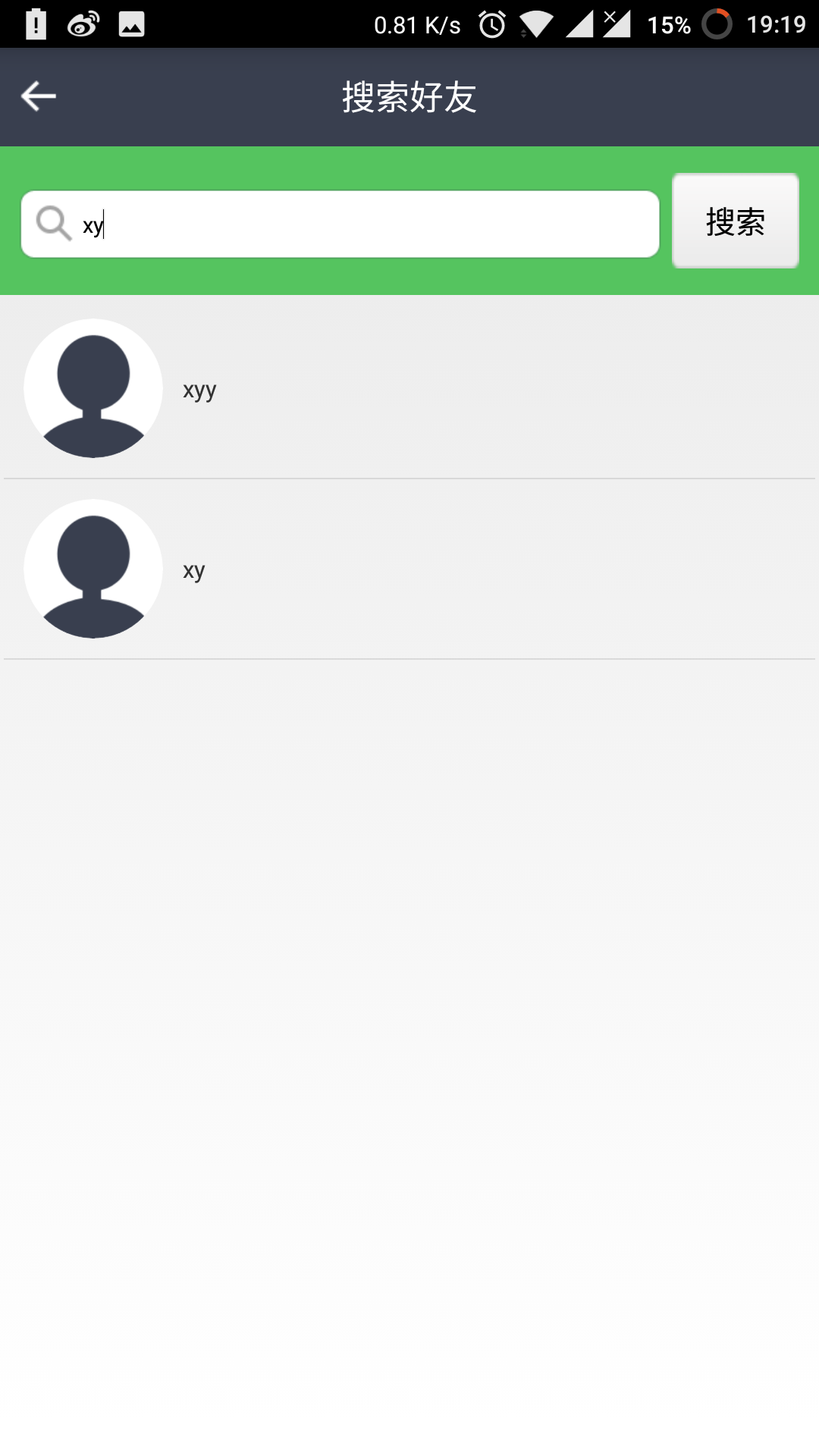
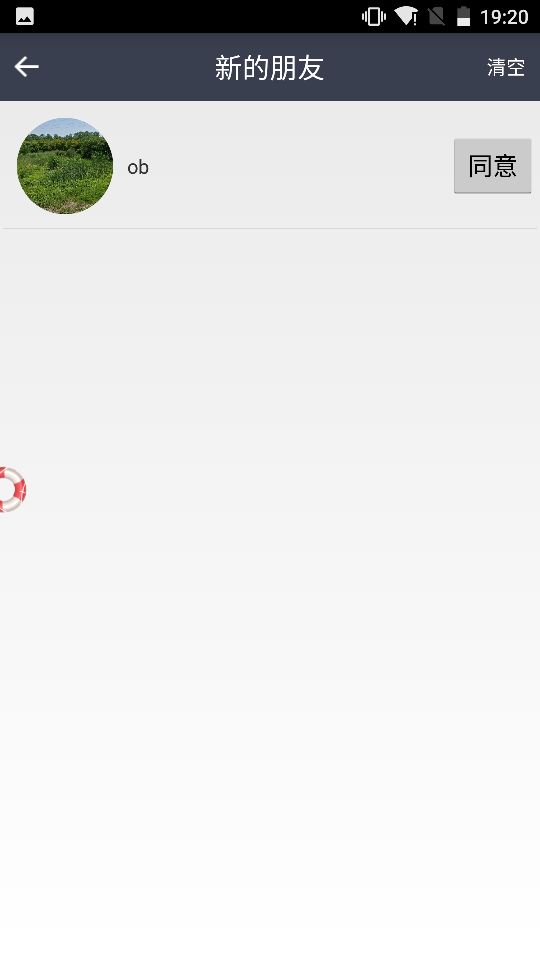


图5.2登录功能演示

## 5.3好友测试

好友功能有三大功能：好友添加，好友列表和删除好友功能。用户在通信录界面，点击添加好友，然后输入要加的好友名，点击添加到通信录。然后经过对方同意后，两人就成为好友。总共有六个测试用例，分别是搜索好友界面，添加好友界面，同意添加好友界面，通信录列表界面和删除界面。

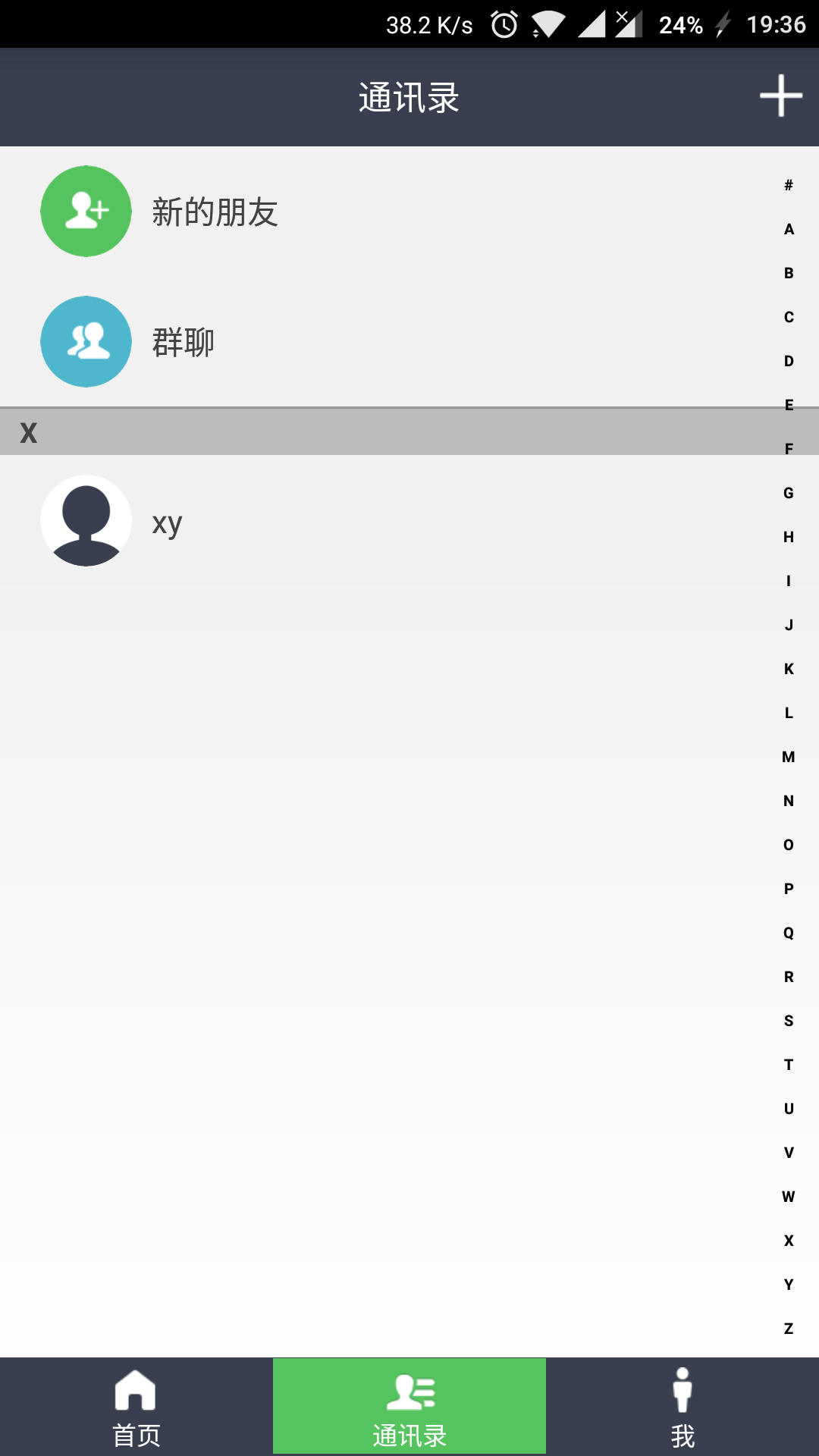
 

图5.3好友管理功能演示

## 5.4聊天测试

两人加过好友后，可直接进行对话，可以发送文字，图片和语音等。有两个测试用例：分别是两人的聊天。

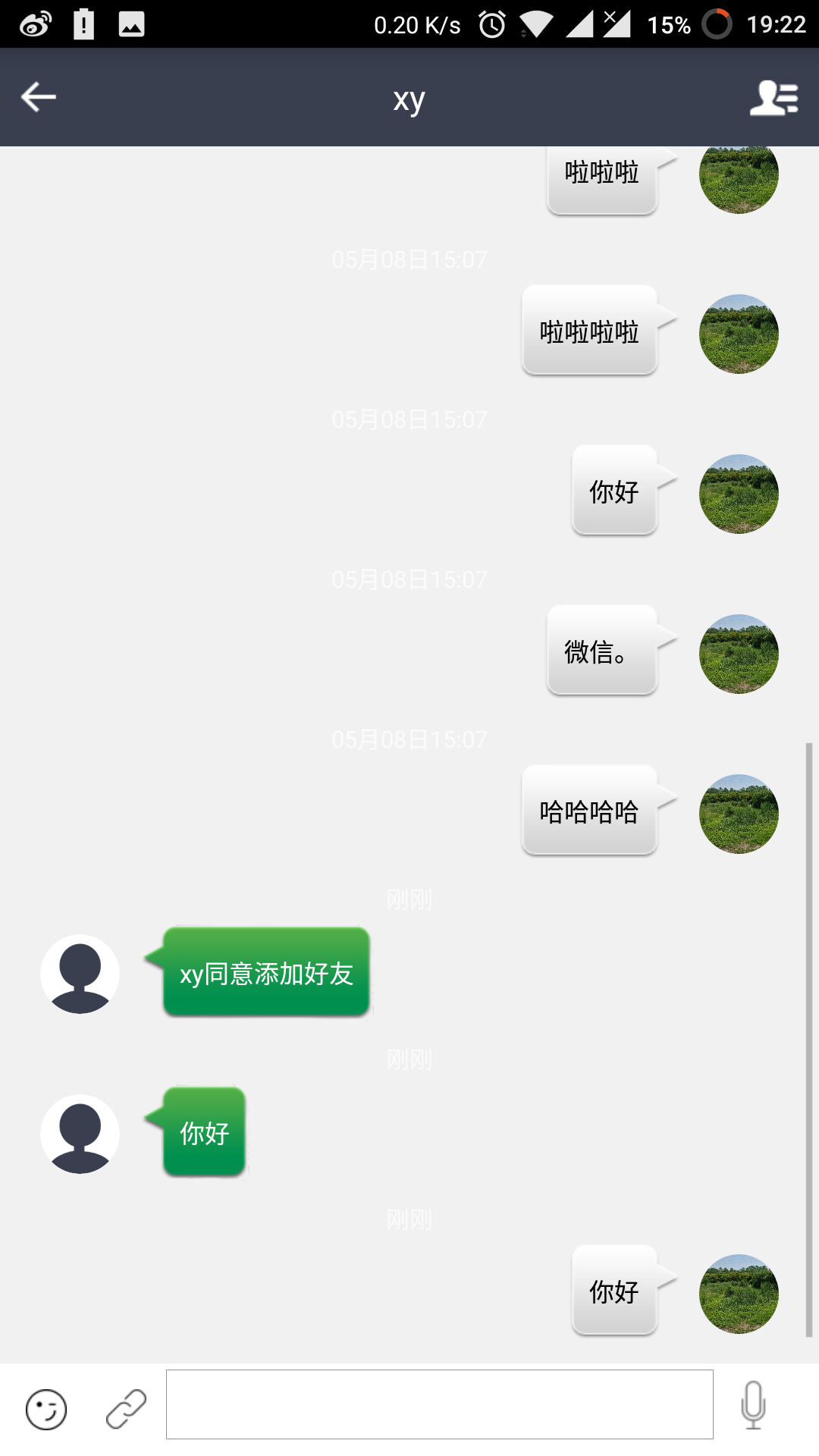
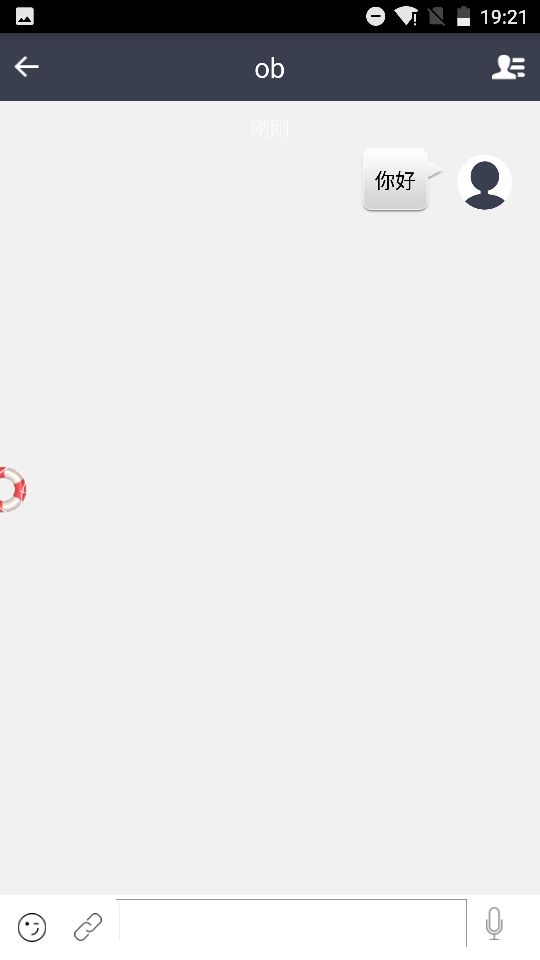
 

图5.4聊天功能演示

# 第六章 总结与展望

## 6.1总结

本文主要研究了Android的架构和XMPP协议，结合两者知识完成即时通信软件，实现了文本，图片的即时转发功能。可以满足人们的日常所需。

我进行了如下工作：

（1）去图书馆查阅资料，上网搜索，了解了基于XMPP的即时通信软件的前景和研究背景，技术实现，国内外的应用以及意义等。

（2）掌握java方面的基础知识。在其基础上，学习android方面的知识。了解android的活动activity，理解其原理，和如何使用。了解android的各种组件，实践并掌握。

（3）完成这个客户端的界面布局，设计。使用android listview控件，实现界面的上下左右滑动功能。运用Layout布局，在实际的界面中灵活的使用每种布局结构，构造出需要的界面。修改界面的颜色，完善界面更美观。

（4）选择开源的openfire作为自己的服务器。

（5）完成客户端的后台设计。客户端定位于Android手机平台。采用XMPP作为即时通讯协议，运用基于XMPP的开源库Asmack包来实现用户注册，用户登录，好友管理和消息转发等各项功能。

## 6.2展望

如今，随着移动网络的快速发展，基于Android的即时通信软件有越来越大的前景，在中国这个人口众多的国家，更应如此。本文讲述了自己研究和实现即时通信软件的全过程。但由于自己的水平问题，以及XMPP的缺陷，还有一些问题还没解决：

（1）数据负载太重：随着通常超过 70％的 XMPP 协议的服务器的数据流量的存在和近60％的被重复转发， XMPP 协议目前拥有一个大型架空中存在的数据提供给多个收件人。新的议定书正在研究，以减轻这一问题。  
（2）没有二进制数据： XMPP 协议的方式被编码为一个单一的长的 XML 文件，因此无法提供修改二进制数据。因此， 文件传输协议一样使用外部的 HTTP。如果不可避免， XMPP 协议还提供了带编码的文件传输的所有数据使用的 Base64。至于其他二进制数据加密会话（ encrypted conversations）或图形图标（ graphic icons）以嵌入式使用相同的方法。

# 结束语

四年的大学生涯即将迎来终点，而毕设则是检测我们四年的学习知识的平台。

从开始选择论文题目到系统的实现，再到论文的完成，每一步对我来说都是新的尝试与挑战，这也是我在大学期间独立完成的最大的项目。在这段时间里，我学到了很多知识也有很多感受，从对Android和XMPP的一无所知的状态，我开始了独立的学习和实践，查看相关的资料，让自己头脑中模糊的概念逐渐清晰，使自己的作品从搭建到完成开始一步步完善起来，每一次改进都是我学习的收获，每一次实践的成功都会让我兴奋好一段时间。如今，我马上步入社会，这次毕设的经历给我前进的路上点亮了一盏明灯。

# 致 谢

本论文是在我的导师李养群老师的亲切关怀和悉心指导下完成的，他严谨的科学态度、治学精神和精益求精的作风，深深地感染和激励着我。从课题的选择到项目的最终完成，老师都始终给予我细心的指导和不懈的支持。李老师不仅在学业上给我以精心指导，同时还在思想上、工作上给我以无微不至的关怀，在此谨向李养群老师致以诚挚的谢意和崇高的敬意！

大学四年匆匆，这四年我成长了很多，无论是校内的学习或社团活动，亦或是校外的社会实践和实习，我所经历的每一天每一件事，都让我万分庆幸认识了生命里这么多重要的人，感谢一路帮助我的老师、同事、同学和其他校外的朋友，感谢你们在我脆弱时给我安慰，在我迷茫时指引我方向，在我开心时陪我欢笑，在我哭泣时给我肩膀依靠。学习和生活以及完成论文撰写的过程中，正是有了你们的帮助与建议，才让我可以更加顺利地完成学业。待走出校门，愿诸君一切都好。

# 参考文献

[1]李昊. 基于XMPP协议的Android即时通信软件的设计与实现[D].东华理工大学,2016.

[2]朱姝. 基于XMPP协议的Android手机即时通讯软件研究与实现[D].南京邮电大学,2015.

[3]周士雄. 基于XMPP协议的移动平台即时通讯系统的设计与实现[D].哈尔滨工业大学,2013.

[4]袁宾奇. 基于XMPP的跨平台即时通讯软件库的设计与实现[D].南京大学,2013.

[5]高明鹏. 基于XMPP协议的Android手机即时通信应用研究与实现[D].南昌大学,2012.

[6]黄伟敏. 基于XMPP协议的Android即时通信系统设计[J]. 电子设计工程,2011,(08):57-59.

[7]周文琼,王乐球,周桐,周春光. 基于XMPP的企业即时通信系统研究与应用[J]. 吉林大学学报(信息科学版),2010,(01):106-111.

[8]付莎. 基于XMPP协议企业级IM的研究与实现[D].西南交通大学,2009.

[9]施济瑜,苗放,王华军,李刚. 基于XMPP协议文件传输的研究与实现[J]. 计算机测量与控制,2009,(04):732-733+741.

[10]陈武. 基于XMPP的企业级即时通信系统的研究与实现[D].武汉理工大学,2009.

[11]剧忻,苗放. 基于MINA开发高性能网络应用程序——以实现XMPP协议Openfire3.3.3为例[J]. 重庆工学院学报(自然科学版),2008,(10):121-125+160.



附录A 基于XMPP的Android即时通信系统使用说明

1 引言

1.1编写目的

编写本使用说明的目的是为了描述本软件的运行环境及其所能实现的功能，以便使用者了解本软件的使用范围和使用方法，通过此软件满足日常的使用需求，并在使用中找出软件的bug和有待提升的地方，为软件的维护和更新提供必要的信息。

1.2参考资料

Android使用说明，QQ使用说明

2 软件概述

2.1软件用途

本软件是用来进行即时通信的，主要包括三个模块：注册登录、好友管理以及聊天等。用户可以通过此软件实现文字，图片，语音的发送与接收，满足日常的通信需求。

2.2软件运行

本软件是运行在Android为系统的智能手机上的，在软件安装后，直接点击相应图标，就可以使用该软件。

2.3系统配置

本软件要求Android版本为4.0.1及以上，至少3M的存储空间。在使用时，需要连接WiFi，或使用数据流量，否则会登录不上。

2.4软件结构

软件由注册登录、好友管理、聊天、信息管理等几大模块组成。

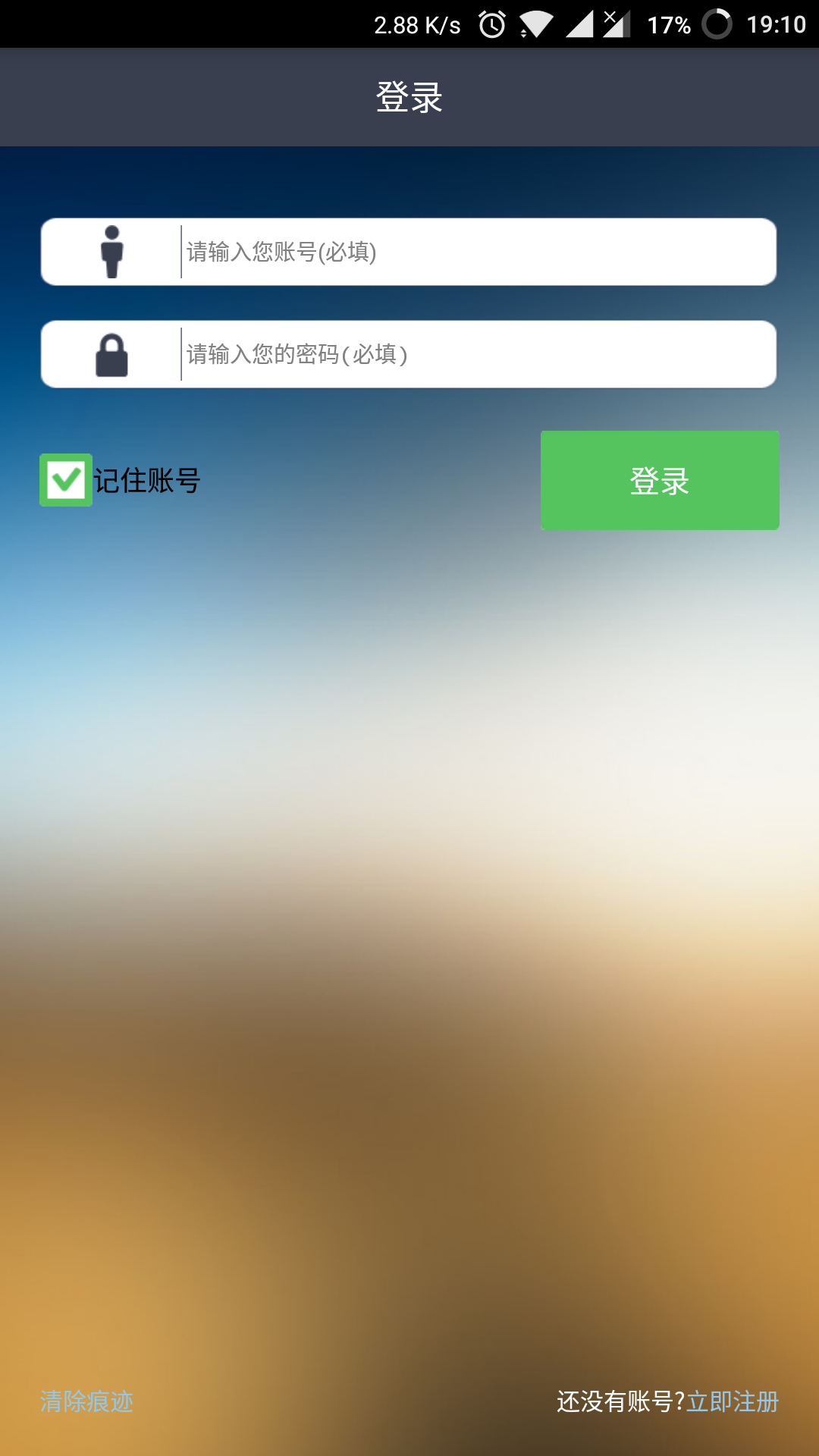
3 软件使用过程

3.1软件安装

从应用市场下载本软件的APP，然后安装即可。

3.2运行步骤

点击软件图标，运行软件。软件运行后效果如下：



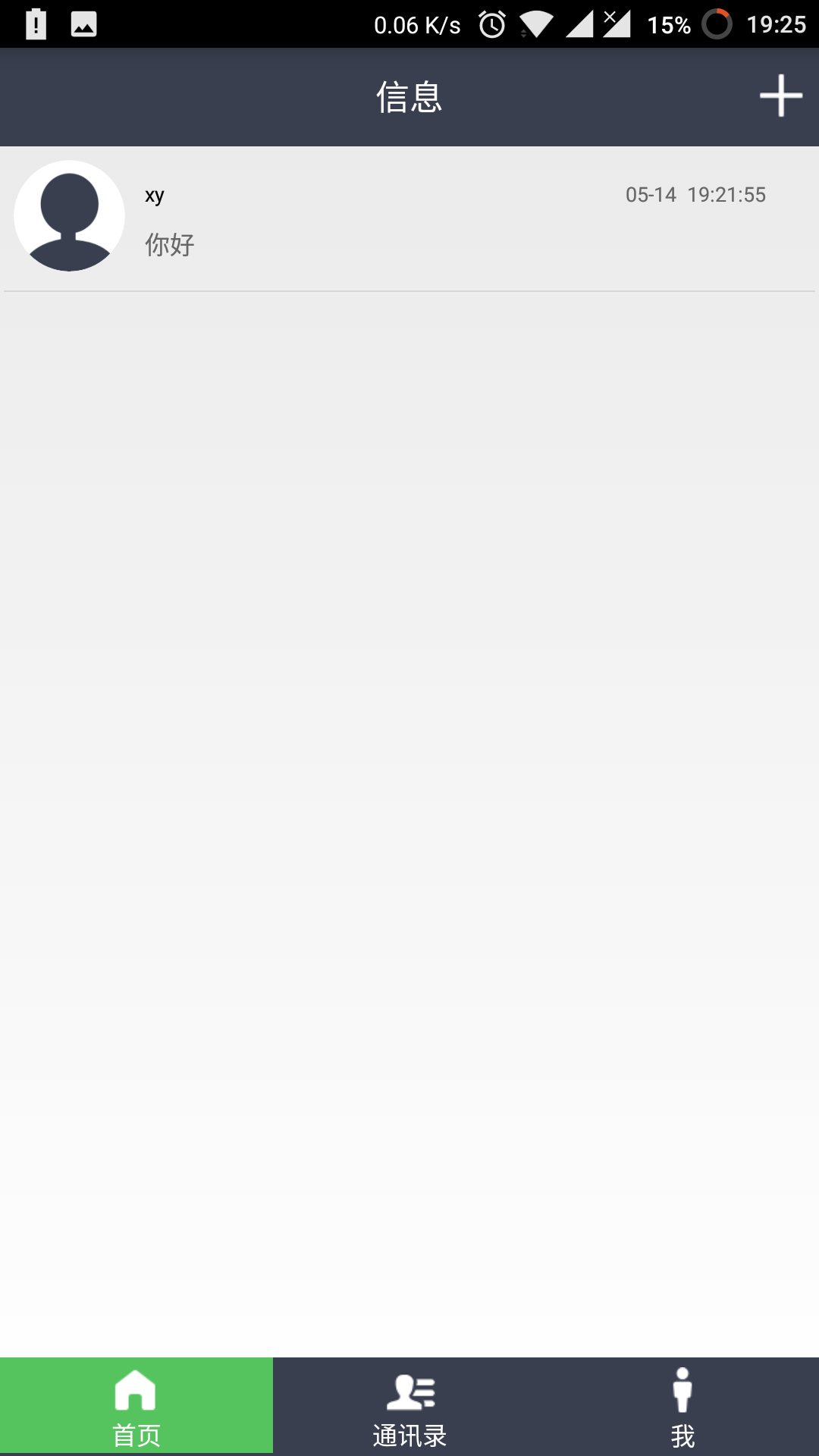
图A1

该界面为软件的登录界面，也是软件打开时的界面。用户需要将注册过的账号密码输入，就可以进入该系统。若输入错误，则会显示账号密码错误，需重新输入。如果还没有账号密码，则需点击右下角的注册按钮，会弹出以下界面：



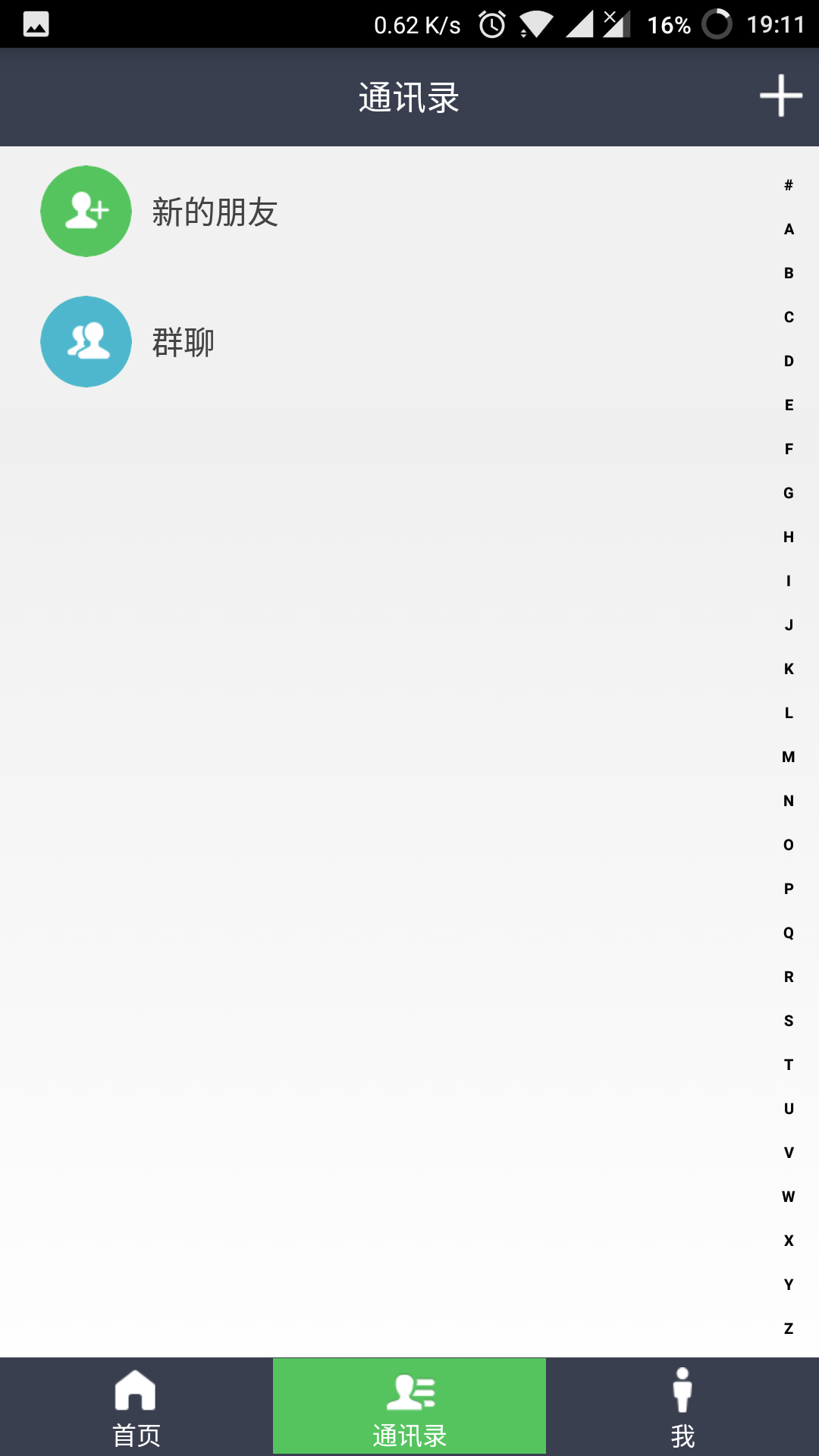
图A2

用户需要在此进行注册。在注册过程中，需要注意几点，所有的字段都不能为空，邮箱的格式要满足xxxx@xxx.xxx的格式需求，两次密码要相同。如果没有其他问题，会注册成功，并进入系统：



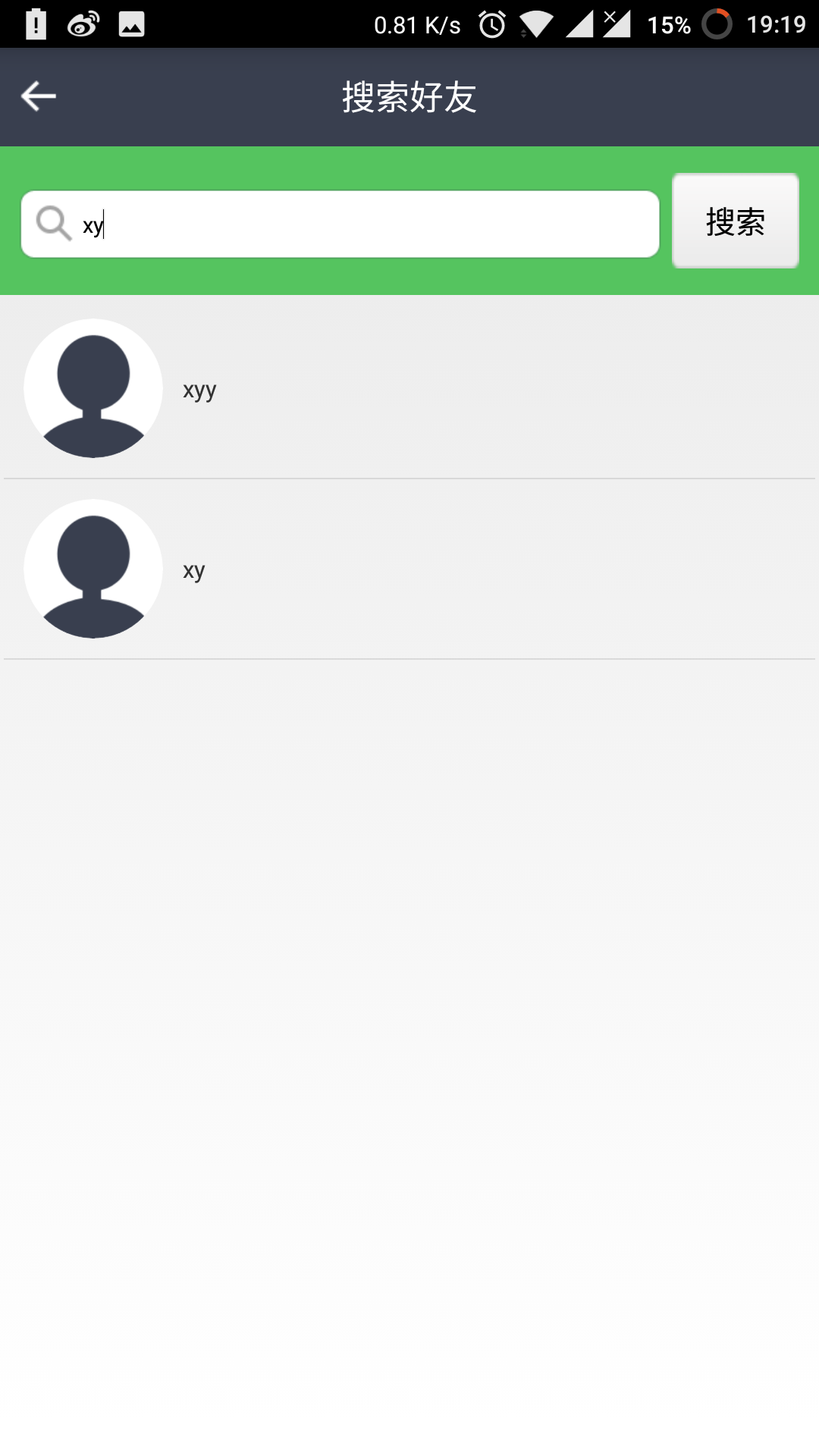
图A3

该界面底部有三个按钮，分别为首页，通信录和我这三个按钮，分别对应着聊天列表，通信录管理，以及信息管理这三个界面。默认的页面是首页（聊天列表），如果想要进行聊天，首先需要添加好友。点击正下方的通信录按钮，出现以下页面：



图A4

点击右上角的添加按钮，进入搜索好友的界面：



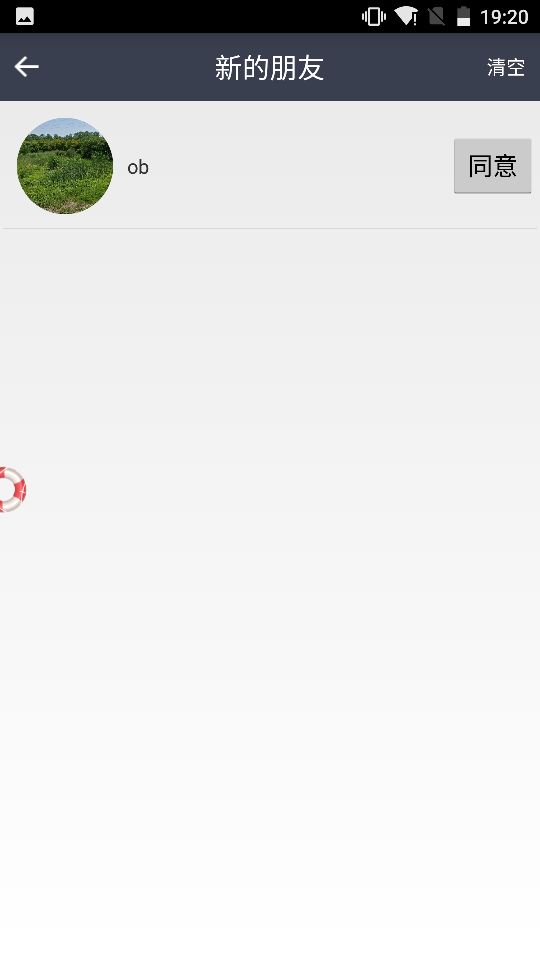
图A5

在搜索栏中输入想要添加好友的姓名，点击搜索按钮，下方就会出现与你输入内容有关的用户列表，选择一个进行点击操作，会出现以下界面：



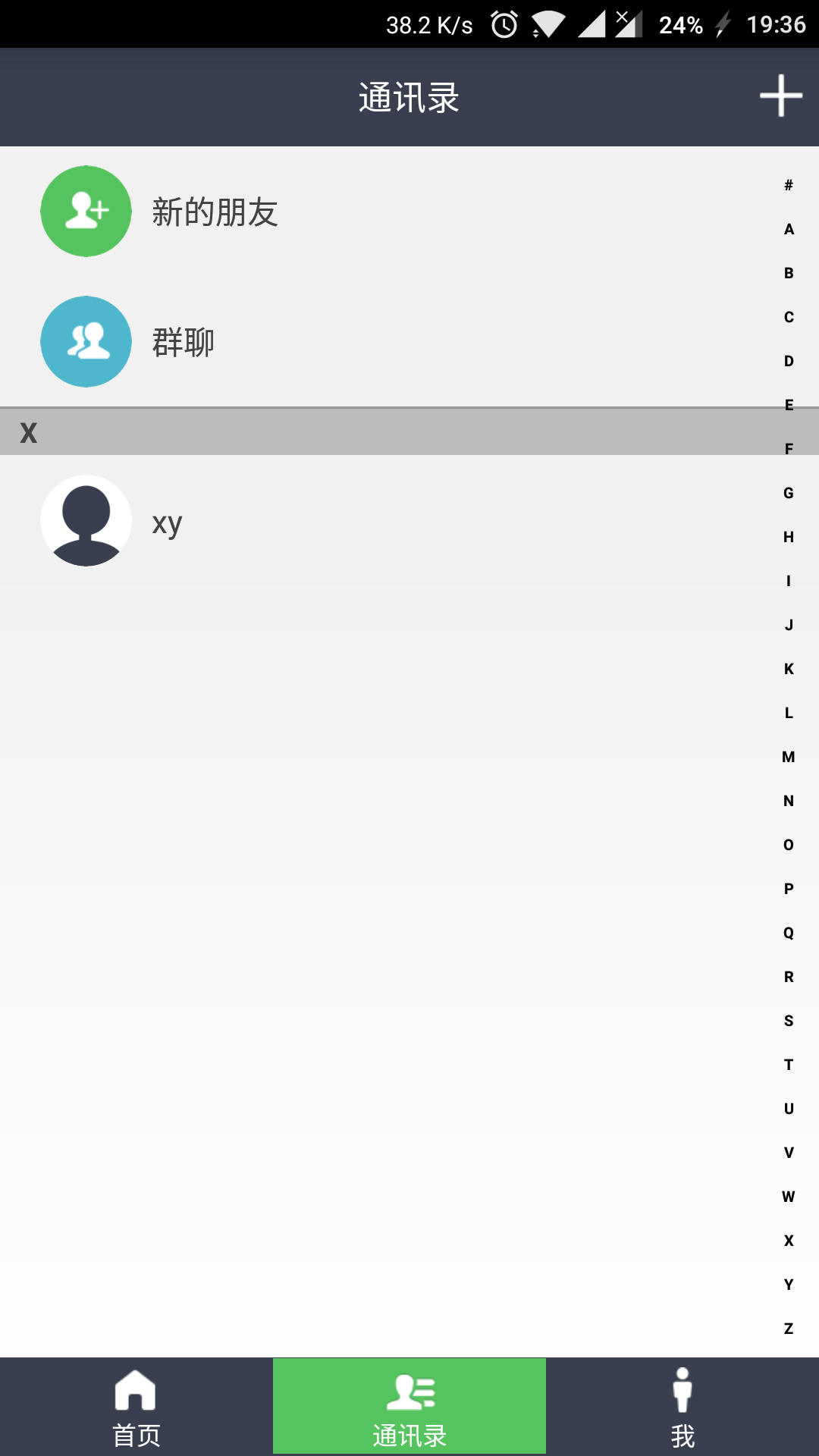
图A6

然后点击“添加到通信录”按钮，会像你要添加的好友发一个添加好友请求，如下图所示：



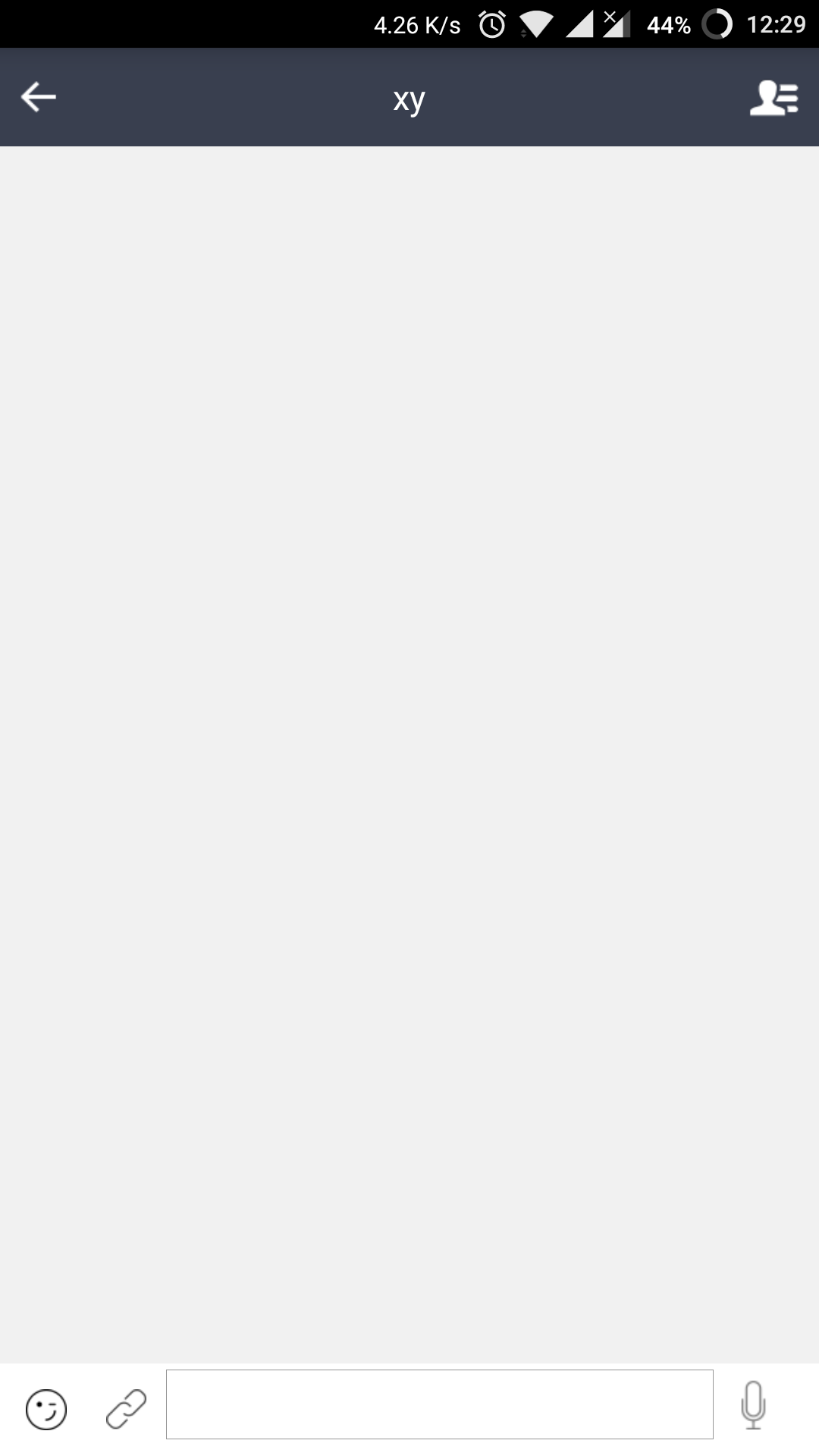
图A7

你要添加的好友的新的朋友了，就会出现你的请求了。点击同意按钮，你们两个人就会成为好友，下图为添加成功后的界面：



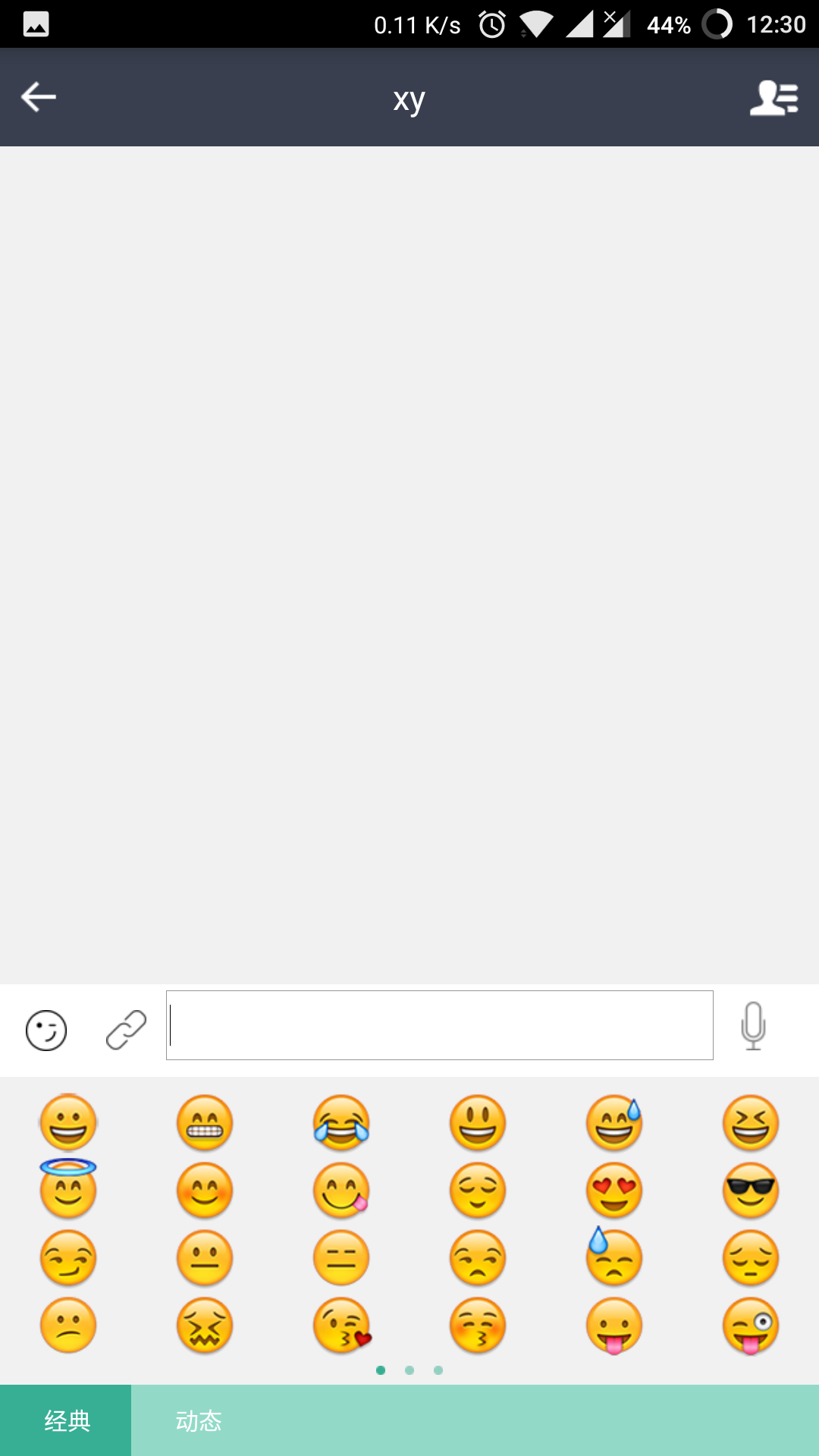
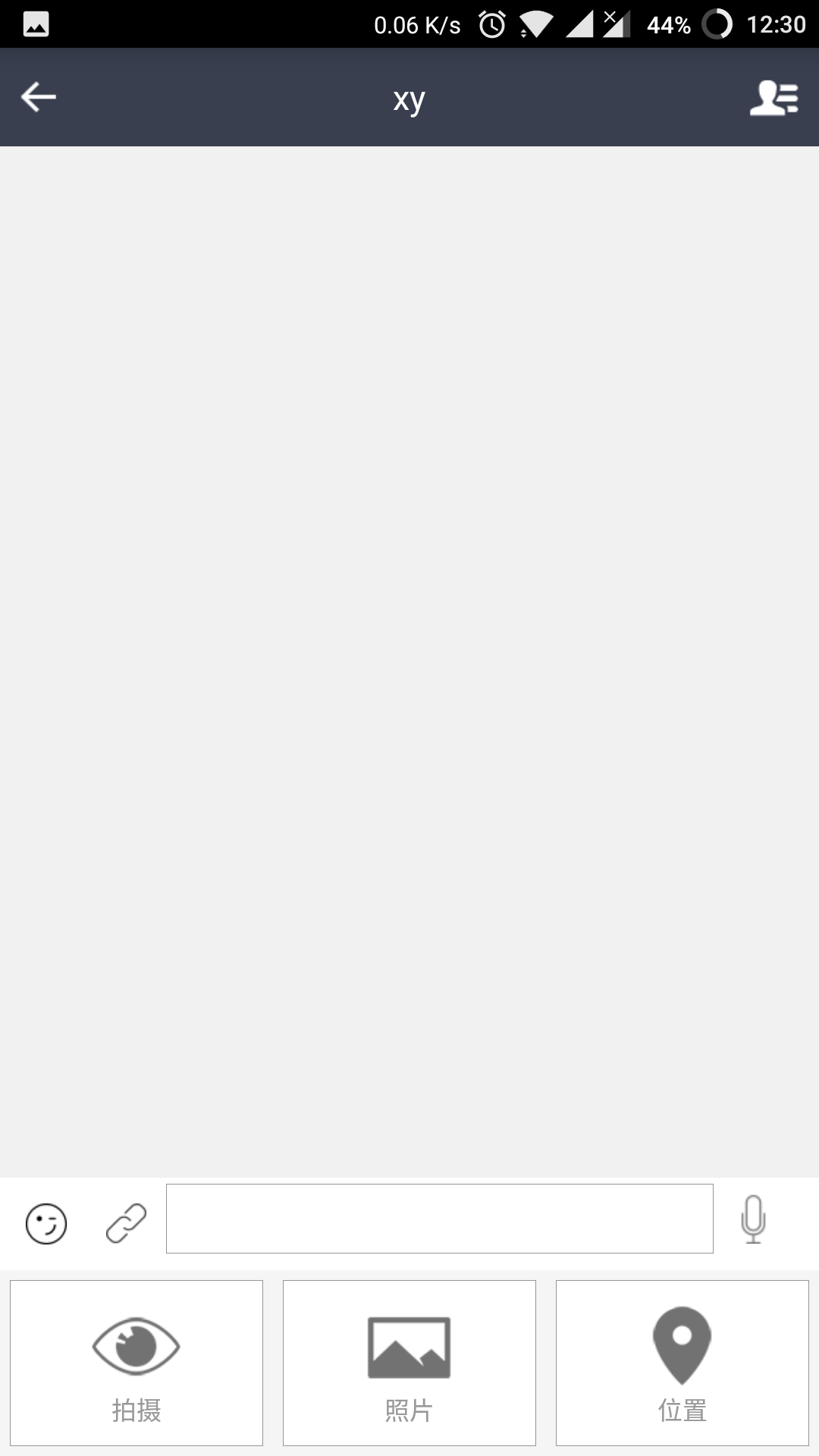
图A8

在这个时候，你们两人就可以进行聊天了，点击该好友，会进入聊天界面：



图A9

在这个页面中，你可以发送文字，图片，表情，和语音等：

图A10

如果你想删除该好友，只需要在与该好友的聊天界面中点击右上角的图标，会弹出一下界面：



图A11

这时，你只需要点击“移除通信录”按钮，就可以删除该好友。除此以外，你也可以对自己的个人信息进行修改，包括头像，昵称，手机号等，如下图所示：



图A12