

PaperPass专业版检测报告

简明打印版

比对结果（相似度）：

总体：17 %（总体相似度是指本地库、互联网的综合比对结果）

本地库：14 %（本地库相似度是指论文与学术期刊、学位论文、会议论文数据库的比对结果）

期刊库：9 %（期刊库相似度是指论文与学术期刊库的比对结果）

学位库：12 %（学位库相似度是指论文与学位论文库的比对结果）

会议库：1 %（会议库相似度是指论文与会议论文库的比对结果）

互联网：4 %（互联网相似度是指论文与互联网资源的比对结果）

编号：5921572C67105SDG3

版本：专业版

标题：基于XMPP协议的Android即时通信软件的设计与实现

作者：徐阳

长度：20426 字符(不计空格)

句子数：758句

时间：2017-5-21 17:00:28

比对库：学术期刊、学位论文（硕博库）、会议论文、互联网资源

查真伪：<http://www.paperpass.com/check>

句子相似度分布图：



本地库相似资源列表（学术期刊、学位论文、会议论文）：

- 相似度：1 % 篇名：《基于XMPP协议的Android手机即时通讯软件研究与实现》
来源：学位论文 南京邮电大学 2015 作者：朱姝
- 相似度：1 % 篇名：《基于UML的地质灾害数据采集系统建模》
来源：学术期刊 《科教导刊-电子版（上旬）》 2015年5期 作者：张自强
- 相似度：1 % 篇名：《基于XMPP协议的即时通讯系统在三维之家平台中的研究与应用》
来源：学位论文 东南大学 2013 作者：沈江
- 相似度：1 % 篇名：《Android平台基于XMPP协议的即时通信系统的设计与实现》
来源：学位论文 东华大学 2015 作者：王琰楠
- 相似度：1 % 篇名：《随机波动率的波动率模型》
来源：学位论文 上海交通大学 2011 作者：何鲁宁

6. 相似度：1 % 篇名：《一类非线性抛物方程粘性解的存在唯一性及其在图像中的应用》
来源：学位论文 上海交通大学 2006 作者：季婕
7. 相似度：1 % 篇名：《带有随机噪声的T-S模型非线性系统控制设计》
来源：学位论文 杭州电子科技大学 2012 作者：刘朋朋
8. 相似度：1 % 篇名：《基于SIP协议的Android即时通讯应用软件的研究与开发》
来源：学位论文 南京邮电大学 2014 作者：田武星
9. 相似度：1 % 篇名：《High Spectral Power Supercontinuum...》
来源：学位论文 长春理工大学 2015 作者：AcnerCaminoBenech
10. 相似度：1 % 篇名：《基于android系统的即时通讯系统设计与实现》
来源：学位论文 山东大学 2014 作者：杜辽
11. 相似度：1 % 篇名：《一个基于WebRTC技术的即时通信系统的设计与实现》
来源：学位论文 华中科技大学 2014 作者：刘浪宇
12. 相似度：1 % 篇名：《PDCA循环在本科毕业论文（设计）质量提升中的应用》
来源：学术期刊 《教育教学论坛》 2013年7期 作者：李适 傅冬和 肖文军 谭君 张盛
13. 相似度：1 % 篇名：《非齐次相对论欧拉方程组Cauchy问题整体熵解的存在性》
来源：学位论文 上海交通大学 2006 作者：王安娇
14. 相似度：1 % 篇名：《Public International Law Aspect on...》
来源：学位论文 对外经济贸易大学 2015 作者：GerashchenkoOleksandr
15. 相似度：1 % 篇名：《基于Android的即时通讯系统的设计与实现》
来源：学位论文 山东大学 2014 作者：孟超

互联网相似资源列表：

1. 相似度：2 % 标题：《XMPP 基于XMPP的即时通讯解决方案 - Leo.cheng -...》
<http://www.cnblogs.com/lcw/p/3505294.html>
2. 相似度：1 % 标题：《XMPP协议 XMPP协议的优点跟缺点_读书人》
<http://www.reader8.cn/jiaocheng/20131030/2153133.html>

全文简明报告：

毕业设计（论文）原创性声明

本人郑重声明： { 77 %：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。 } { 82 %：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。 } { 81 %：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

{ 54 %：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。 } { 51 %：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。 { 83 % : 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。 } { 66 % : 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。

{ 71 % : 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

第一章 绪论

1.1 课题研究的背景

随着互联网的迅速发展,电脑端互联网已达到了瓶颈,更加方便廉价的移动互联网开始占据人们的生活。据有关数据统计,截止2013年底,中国手机网民的数量已经超过5亿,而且增长速度并没有下降,在总网民中达到了81%。 { 48 % : 正是移动设备的廉价,以及4G网络的普及,越来越多的人开始使用移动互联网。 } { 47 % : Google为此推出了在移动端上使用的系统Android, , 其开放性吸引着越来越多的人来使用。 }

{ 42 % : Android是运行于Linux虚拟机上的操作系统,最大的特点便是其开放性,越来越多的开发者投入了Android的怀抱, } 产生了许多许多的开源代码,以及第三方的应用程序,使得Android系统在众多人的努力下茁壮成长, 如今已经有了7个Android的版本。

{ 63 % : 即时通讯(IM)随着互联网的高速发展,越来越流行。 } 许多大公司也都推出了各自的即时通信软件,如QQ,微信等,这些软件的出现大大方便了我们的日常生活。相比于传统的通信方式,即时通信的功能愈发强大,文字,图片,语言等都能传递,其中最大的特点便是其即时性,你无需等待太长时间。 英特网已经成为了传输信息的桥梁。

{ 47 % : XMPP(可扩展消息处理现场协议)是基于XML(可扩展性标记语言)的协议,它的出现促进了即时通信的发展,加强了服务器在即时通信中的作用。 } 而且,在业内形成了统一的标准,使用这个协议,用户可以跨平台地向其他用户发送消息。

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,人们可以很方便地完成信息的传递,节省了传统的打电话,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

{ 49 % : 即时通信(IM),从字面上来理解,就是即时发送和接收消息。 } 不同于传统的通信方式,比如邮件等,即时通信是即时的。 而且,可以随时了解对方的在线状态,非常方便。 无论是国内,还是国外,都有很多热门的应用。

如今,在中国最火的通信软件便是微信。 { 79 % : 微信是腾讯公司在2011年推出的免费软件。 } { 48 % : 微信不仅支持消息的转发,还可以发送语音图片,以及视频对话。 } { 51 % : 到2016年的下半年,中国超过90%的智能手机用户使用了微信。 } 而且在国外也有大量的应用,超过200个国家在使用,支持20种语言。

在国外,Skype最受欢迎。 { 82 % : 它具备即时通信的大部分功能,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。 } 此外,用户可以实现实时的视频语音对话,连传统的打电话,发短信等服务也支持。

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。 { 43 % : 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

{ 42 % : 随着智能终端设备的普及，以及网络质量的提升，移动网络用户的数量增长迅速，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。 { 71 % : 即时通讯 (IM) 是一个终端服务，允许两人或多人在因特网上即时地进行消息传递、语音与视频交流。 }

如今，即时通信软件比如微信，Skype都掀起了一股热潮。但这些软件的通信软件都存在着一个缺点，就是使用了内部的协议，不同的软件之间不能相互通信。 { 42 % : 而XMPP的出现，提出了一个统一的通信标准，有助于解决这些问题。 }

{ 45 % : 在操作系统方面，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，越来越多的开发者投入了Android的怀抱，产生了许许多多的开源代码，以及第三方的应用程序，使得Android系统在众人的努力下茁壮成长。

{ 44 % : 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。 { 54 % : 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。

{ 52 % : 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java，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。

{ 43 % : 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

{ 42 % : Java是面向对象的，不仅仅汲取了C++的各种优点，还抛弃了C++里难理解的多继承、指针等概念，所以Java语言简单方便，灵活易用。} { 64 % : 作为面向对象编程的代表，Java语言极好地实现了面向对象理论，允许程序员更加优雅地进行编程操作。}

Java语言的特性有三个：封装、继承、多态。

2.1.2 设计模式

{ 42 % : 设计模式是前人反复实践，大量经验的总结，是一种思想。} { 46 % : 使用设计模式，可以增强代码的可读性，提高代码的可靠性，有助于解决复杂的问题。} { 48 % : Java中有许多的设计模式，比如装饰器模式、代理模式、工厂模式、单例模式等，下面着重介绍几种设计模式。}

{ 53 % : (1) 单例模式是Java的众多设计模式中被广泛使用的一种。} 单例模式的使用，使某个类在一定时间内只有一个实例存在，防止多个实例同时存在，大大提高了性能，解决了多个实例对象引起的问题。 { 47 % : 单例模式的典型应用便是管理器和控制器。}

{ 46 % : 单例模式主要有三种实现方式，分别为饿汉模式，懒汉模式，双重校验锁这三种。}

a.饿汉模式

```
public class Singleton{  
  
    private static Singleton instance = new Singleton();  
  
    private Singleton(){}  
  
    public static Singleton newInstance(){  
  
        return instance;  
    }  
}
```

b.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;

        }

    }

}
```

c.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) {

                    instance = new Singleton();

                    return instance;

                }

            }

        }

    }

}
```

(2) 代理模式也是一种设计模式。代理的作用是通过另外的方法访问想要访问的对象，可以调用其中的方法。而且，也可以在不改变这个对象的基础上，增加额外的方法。代理可分为两种：静态代理，和动态代理。

静态代理： { 40%：在实现相同接口或继承同一个父类的前提下，代理对象可以对对象进行增加功能操作。 } 但每次都要创建类实现接口或继承父类，显得非常麻烦。这时，我们可以使用动态代理的方法。我们可以使用JDK自带的接口，动态地对代理对象进行功能扩展，而不需要新增一个代理类。

2.2 Android介绍

2.2.1 Android的架构

Android是基于Linux的开源操作系统，是由Google开发实现的。Android的核心依赖Linux系统的内核，因此拥有很多Linux的优点，可以在ARM，X86，MAC等不同结构的机器上运行。

Android的系统架构是分层的，可分为四个层： { 78 % : 应用程序、应用程序框架、系统运行库和linux核心。 }

(1) 应用程序

由于Android是基于Java的，所以运行在Android上的应用程序都是使用java来实现的。Android上的通话，短信，浏览器都属于是Android上的应用程序。正是由于Java的跨平台，应用程序可以不编译，就直接使用。 { 43 % : 而且，Android的开放性，也吸引了无数的Android开发者，编写了无数的第三方Android应用程序，满足了用户的日常所需。 }

(2) 应用程序框架

开发者可以使用核心的应用程序的接口。这个接口中包含着许多服务，比如Android自带的视图View，访问数据的内容提供者，以及管理资源的资源管理器。开发者可以使用这些模块来更快速地实现自己的功能，用户也可以替换为自己喜欢的应用程序。

(3) 系统运行库

a)程序库： { 42 % : Android中除了java语言实现的应用程序外，还存在着一些用C/C++语言实现的运行库。 } 很多组件都在使用着这些运行库。比如媒体库，SQLite数据库都是C语言实现的。

b)Android 运行库： Android运行库里都是放着一些核心的东西。 { 42 % : 每个程序都会在自己的虚拟机中运行，也就是说Android可同时运行多个虚拟机来执行应用程序。 }

(4) Linux 内核

{ 48 % : Android 的核心系统服务都是依赖于Linux的内核来实现的，高可靠的安全性，优秀的内存管理服务，稳定的网络协议栈等都是依据Linux来实现的。 } 除此以外，Android也对其做了一些修改，比如在进程通信，以及节省电力方面。 { 57 % : 运行在Android上的一个个应用程序就好像是Linux中的进程。 }

图2.1Android架构

2.2.2 Android之Activity生命周期

Android组件有最基本常见的四大组件，Activity是其中之一。Activity是用来和用户交互的，用户可以使用Activity来交互实现某项任务。 { 41 % : 在Android开发中，创建Activity后，用户通过调用setContentView()方法，系统会在这个Activity里指定一个布局界面View，这个界面的作用就是给用户进行交互。 } Android系统中管理Activity是通过操作Activity栈来实现的，而Activity管理自己则是通过Activity自带生命周期的方法来进行操作的。 下

面就讲述了Activity的生命周期的大体过程。

1. onCreate()： Activity在第一次被创建时，会调用该方法。 在这个方法里，你需要将一些初始化的操作完成： 加载View，数据绑定到事件等。 如果是第一次创建，接下去会调用onStart()方法，如果是停止后重新显示，那么会调用onRestart()这个方法。
2. onStart()： 当前Activity对用户由不可见变为可见时，会调用该方法。 当这个方法调用完，会接着调用onResume()这个方法。
3. onResume()： Activity会在用户将要和用户交互时，调用此方法。 在进行该方法后，此时的活动会处在运行状态。
4. onPause()： 当系统要启用其他活动或者恢复之前活动时，会调用这个方法。 onPause()方法启动时，占用系统的资源比如CPU资源，会被回收释放，一些比较重要的数据则会被保留下来。 {40%： 这个过程需要在极短时间内完成，不然会影响其他活动。 }
5. onStop()： 当其他活动启动掩盖该Activity，该活动变得不可见时，会调用该方法。 这个方法是紧接着onpause()之后执行的。 如果当前活动恢复，会启动onRestart()方法； 否则，会销毁掉。
6. onDestroy()： 调用该方法，活动会被销毁。
7. onRestart()： {49%： 当活动由onStop()变得停止状态，需要重新启动时，会调用该方法。 } 调用完后，该活动会恢复运行。

图2.2 Activity生命周期

2.2.3 Android之布局

Android里有多种多样的控件，这些控件的有机结合，就可以形成丰富的Android界面。 为了让这些控件在界面上按照自己的想法排列，Android引入了多种布局方式。 布局是一种容器，它可以放置许多控件，控件的位置可以按照一定规则进行改变，然后就可以将页面变得精美。 而且，布局除了放置控件外，还可以在里面嵌套其他布局，通过多种不同的布局，就可以实现比单一布局更加复杂的页面。 接下来，详细讲解了四种的基本布局。

(1) 线性布局--LinearLayout

线性布局，从字面意思理解，就是从上往下一个一个的排列在界面上，就像html中的div标签一样。 LinearLayout又可分为两种： 垂直布局和水平布局。 垂直布局的意思就是，每个元素都是按从上往下的方式排列的，每个元素独占一行； 水平布局的意思是，所有元素都在同一行中，每个元素都是从左到右这个顺序进行排列的。 在线性布局中，weight也是一个重要属性。 在垂直布局时，代表了这个控件占总高度的比重； 在水平布局时，代表的是该控件在宽度上占的比重； 每个比重相当于该控件的weight除以所有的weight。

(2) 帧布局--FrameLayout

FrameLayout应该算是所有布局里比较简单的。 {47%： 在默认情况下，所有的控件将会固定在界面的左上角

，每个元素相互覆盖重叠。} FrameLayout里提供了一种对其方式layout_gravity，可以设置其为left或right来实现左对齐，或右对齐。

(3) 相对布局--RelativeLayout

相对布局就是说，相对于一个元素，可以按照这个元素为参照物，进行排列。可以以父类为参照物，也可以以同级元素为参照物，来进行上下左右各个方面的相对排列，比如居中，右下角等布局。

(4) 百分比布局--PercentRelativeLayout

百分比布局是Android为了解决控件大小问题，而引入的。在上面三种布局中，只有线性布局中，有个weight属性可以按比例指定控件的大小。百分比布局针对上面的帧布局和相对布局，分别定义了PerencetFrameLayout和PerencetRelativeLayout，为这两个布局分别提供控制控件大小的功能。需要注意的一点是，这个布局需要自己引入，否则不能使用。

2.2.4 SQLite数据库

SQLite是Android系统中自带的数据库，无需自己引入。和mysql数据库相比，SQLite虽然比较轻量，但结果却是更快的运行速度，更小的内存使用，在移动设备这种内存本来就不大的设备上使用，非常有优势。{44%：而且，SQLite数据库支持SQL语法，连数据库的ACID特性都完美支持。}配置SQLite也非常简单，因为SQLite相当于零配置。但由此也带来了一些缺点，比如并发慢，对数据库进行增删改操作相比而言也比较慢。

SQLite数据库分为了三个模块：编译器(Compiler)，虚拟机(Virtual Machine)和后端(Back-End)。编译器的作用是对传进来的sql进行语法检查，如果检查无误，会发送给虚拟机进行处理。虚拟机接收到这些数据后，会解释这些字节码，来进行数据的增删改查操作。而后端则是管理这些数据的，通过B+树，缓存等技术，来更迅速的实现各种操作。下图就是SQLite的架构

图2.3 SQLite架构

2.3 XMPP协议解析

2.3.1 XMPP协议简介

{43%：在当今世界，热门主流的即时通信协议有四种，XMPP协议便是其中之一。}XMPP是由Jabber转变而来，是开源的通信协议。它的最大特点便是基于XML实现的。它有以下优点：

1. XMPP协议是开源的，任何人都能得到XMPP的源码。经过无数人的验证和修改，XMPP协议变得更加完善，不足之处也在慢慢减少。而且，其开放性也让业界有了统一的规范。

2. XMPP的扩展性非常好。{46%：由于XMPP协议是基于可扩展性标记语言XML的，所以信息在交换时是以XML的格式存在的。}XML的使用强化了数据的传输，用元素和属性来描述数据，而不使用其显示方法，致使无论是怎样的数据，都可以保存在信息中，提高了XMPP协议的扩展性。而且比起传统的方式少去了字符串解析步骤，节省了大量时间。

3.分布式架构。XMPP 协议是基于C/S 架构的，和其他使用p2p的通信软件不同，消息主要是由服务器来传递的，这点和电子邮件的传递过程十分相似。而且，XMPP的三个元素之一的网关是异构的，使之实现了不同用户之间的互联互通。

4. XMPP 使用范围广。XMPP 不仅仅可以使用在即时通信领域上，而且在游戏，视频对话，办公协同工具等领域也有很大的使用前景。

5.安全性。 { 45 % : XMPP在客户端和服务端，或者服务器与服务器通信中使用TLS和SASL协议来加密解密通信过程。 } TLS主要的作用是保证传输通道的安全性，SASL作用是鉴别用户的权限的。这两个协议的使用大大增强了XMPP的通信安全，在内部网之外的网络也有很强的保密性。如下图所示：

图2.3XMPP安全性

2.3.2 XMPP协议内容

XMPP是基于XML协议的，传输的是XML格式的数据。其他通信软件的传输内容则各不相同，比如腾讯QQ是使用二进制的数据进行传输的，MSN则是使用纯文本加其他符号来实现传送的。而XMPP传输的XML格式的信息，不但省去了传统的字符串解析工作，而且通俗易懂，能一眼看出每部分的用处和功能，简化了开发，方便了查错。由此可见，传输XML的协议是XMPP协议的基础，也是核心所在。

{ 42 % : 每一个从客户端发出的消息，都要经过服务器来进行转发，才能到达另一个客户端。 } { 54 % : 下面就讲解了客户端到客户端的传输过程： }

{ 52 % : 1) 客户端要连接到服务器，这是下面操作的前提。 }

{ 44 % : 2) 服务器会对客户端发送的连接请求进行认证，利用本地的证书来对照，失败则不通过。 }

{ 51 % : 3) 客户端确定目标地址，通过服务器获得目的地址的实时状态。 }

{ 43 % : 4) 服务器通过目的地址来查找另一个客户端，进行连接并验证相互的身份。 }

5) 客户端之间就可以相互进行通信。

XMPP开源到现在，拥有了许许多多的协议范本，别人也以XMPP为基础做了很多扩展，下面介绍几种重要的协议范本。

(1) RFC 3920：这是XMPP协议的核心部分。这个范本规定了XMPP使用XML协议来实现信息的传输，大大提高了XMPP的开放性以及扩展性，定义了一种统一的规范。这个范本也规定使用TLS和SASL来加密通信过程，提高了XMPP的安全可靠性。JID的定义，和错误样式也是在这部分定义的，所以，这个是核心所在，所以基于XMPP的都要实现该范本。

(2) RFC 3921：在用户连接服务器后，所做的操作比如好友的添加删除，消息发送都是该范本定义的。通过三种不同格式的XML字节来完成上面的这些功能，IQ节是用来请求响应的，Presence节是用来表明用户的在线或不在线状态的，Message节则是用来传递消息的。阻塞策略也是在该部分定义的，到如今已有多种方式。和

RFC3920结合，就可以实现简单的即时通信软件了。

(3) 除了上面这两种比较重要的范本外，还有许多。比如，XEP-0045是用来实现多人聊天的，XEP-0096是用来实现文件传输的，XEP-0167则是用来发送语音的，其他的还有很多，就不一一介绍了。

2.3.3 XMPP地址格式

每一个用户都有一个唯一标识符JID。JID标识符的作用很大，不仅可以表示一个用户，而且还可以表示类似聊天室等其他内容。JID的格式有严格的规定，必须由域名，节点以及源这三部分组成。格式也必须为节点@域名/源这种形式，前半部分和电子邮件的地址格式相类似。举个例子，当用户在服务器s上注册了一个名称为name的账号后，这个用户的JID就是name@s。以后每次与服务器进行通信，都会使用这个JID，后面的源则会跟着由于登录设备的不同而发生改变。用户的源可以有多种不一样的。

XMPP协议中定义了三种XML元素：Message、Presence、IQ。Message主要负责消息的传输。{49%：用户每发送消息时，就会新增一个Message标签，里面插入用户想要发送的消息。} Presence主要用来负责用户状态，随着用户的上线或下线，都会新增一个Presence标签，里面存放着用户的上线或下线的状态。IQ则是用来负责客户端与服务器之间的请求和响应的，服务端和客户端进行交互，都会产生IQ标签，存储请求或存储状态。下面详细介绍这三种元素：

(1) Message元素

主要是负责用户之间的消息的传输。Message首先会发送给服务器，服务器根据数据包中的目的地址，找到该用户，判断其状态，在线则会立即发送出去，{40%：否则会存储在本地，通过触发器来判断其在线时，再发送出去。} 里面有三种属性。To：用来表示接收该消息的用户。from：指发送该消息的用户。Text：用来保存要发送的内容。

Message元素的格式如下所示：

```
[message to= ' niyi@xy.cc/content ' type = ' chat ' ]
```

```
[body]我有件事要请教你[/body]
```

```
[/message]
```

(2) Presence元素

主要是负责用户的上下线状态。{43%：当用户上线或下线时，都会产生一个Presence元素，用来表明用户的在线状态。有三个元素：} To和from属性和Message相似。status：用来保存在线状态。

Presence元素的格式如下所示：

```
[presence]
```

```
From = ' niyi @ xy.cc/contact '
```

To = ' xy @ xy.cc/contact '

[status] Online [/status]

[/presence]

(3) IQ元素

{ 53 % : 主要是用来负责客户端与服务器之间的请求和响应的。 } { 51 % : 服务端和客户端进行交互时, 会进行IQ元素的相互转发。 } { 44 % : 比如, 客户端搜索好友时, 会像服务器发送IQ请求, 服务端处理后, 会返回一个IQ响应。 }

IQ元素的属性有from, id, type三种。 最重要的是type属性, 用来区分是请求还是响应。 From是该请求或响应的来源。 Id则是该请求或相应的id

IQ元素的格式如下所示:

[iq from = ' niyi@ xy.cc/content ' id= ' 654321 ' Type= ' get ']

第三章 系统需求分析与设计

3.1系统需求分析

3.1.1系统整体需求分析

{ 43 % : 根据用户的基本需求, 本系统最终的目标是做出一个即时通信软件。 } { 66 % : 用户可以通过这款软件进行即时通信。 } { 50 % : 下面我对该软件具备的功能进行详细描述。 }

(1) 登录注册

用户只有在注册过后, 才能登录此系统。

若用户没有注册账号, 可以先进行注册, 注册成功后会自动跳转到主界面, 注册失败会给出相关提示。 若用户已经注册过该系统, 可以直接进行登录, 登录成功会跳转到系统主界面, 登录失败会给出相关提示。

当用户登录成功后, 也可以退出该系统, 选择其他账号登录。

(2) 好友管理

a.添加好友

用户在登录成功后, 可以搜索要添加的好友姓名, 选择添加该好友, 如果好友不存在, 则会进行提示。 在进行添加后, 要等对方确定后, 才会成为好友。

c.查看好友列表

{ 100 % : 用户可以进入到通讯录查看自己的好友列表。 } 好友列表展示每个好友的头像和姓名

e.删除好友

如果不想要该好友，你可以选择删除该好友。

(3) 即时通讯功能

a.发送消息

{ 68 % : 进入聊天界面，用户可以输入文字，图片等，点击发送，就可以给好友发送文字消息。 } 自己的消息在右边，别人的在左边。

b.查看聊天记录

{ 49 % : 聊天记录会保存在本地，打开该好友的聊天界面，可以看到与该好友之前的聊天记录。 }

3.1.2用例分析

{ 48 % : 对于软件来说，用例图(Use Case Diagram)是软件由需求分析到实现完成的第一步，它的作用是描述人们对系统进行何种操作。 } { 82 % : 用例视图显示谁为用户、用户希望系统提供怎样的服务，以及用户需要为系统提供的服务，以使用户更容易理解这些元素的用途，也便于开发人员最终实现这些元素。 } { 80 % : 用例图在各种开发实践中被广泛使用，但是它最常用的是用来描述系统以及子系统。 }

为了更加清晰的展示，根据需求分析，分析出系统的具体操作步骤及功能，从而确定系统用例图，具体描述如下，系统用例主要包括注册登录功能，好友管理功能，聊天功能和个人信息管理功能等。

图3.2系统总体用例图

系统的各项功能描述如下：

(1)用例1

用例名： 注册/登录

行为者： 用户

目的： 让用户有该系统的账号，让其可以登录。

过程描述：

{ 47 % : a.进入Andorid系统主界面，点击注册进入注册页面； }

b.进入注册界面后，填写账号，密码等信息；

c.数据无误后，用户注册成功，然后就可以进行登录。

(2)用例2

用例名： 好友

行为者： 用户

目的： 可以添加好友，删除好友，来让用户可以选择人聊天。

过程描述：

{ 48 % : a.登录进入Andorid系统主界面，点击下方的通信录； }

b.进入通信录管理界面；

(3)用例3

用例名： 添加好友

行为者： 用户

目的： 让用户可以和更多的人聊天。

过程描述：

{ 51 % : a.在通信录管理界面，点击右上方的添加按钮； }

{ 50 % : b.进入搜索好友界面，输入想要添加的好友姓名，点击该用户； }

c.进入添加按钮，添加成功。

(4)用例4

用例名： 删除好友

行为者： 用户

目的： 让用户删除不想聊天的好友。

过程描述：

a.点击该好友，进入聊天界面；

{ 62 %： b.点击右上角的按钮，进入详细信息界面，点击移出按钮； }

c.删除好友成功。

(5)用例5

用例名： 聊天

行为者： 用户

目的： 用户和用户之间可以即时通信。

过程描述：

a.在添加好友后，可以点击该好友；

b.进入聊天界面，你可以发送文字，图片和语音等；

c.你的好友可以收到，并进行回复。

(6)用例6

用例名： 信息管理

行为者： 用户

目的： 对自己的信息进行修改。

过程描述：

a.进入我的界面；

b.可以修改自己的详细信息。

3.2系统设计

3.2.1系统总体结构设计

{ 54 %： 在结合了以上我对基于Android的即时通信软件的需求分析，我对该系统的整体结构进行了设计： }

图3.2系统总体架构图

3.2.2功能模块设计

(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

(2) 好友模块

好友模块： { 59 %： 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) 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。 { 46 %： 在下次打开聊天页面时，会显示历史记录。 }

a.发送信息

{ 45 %： 在添加好友后，用户可以选择任意一个好友进行文字对话。 } 在通讯录展示页，点击要交谈的用户，打开聊天界面。 { 51 %： 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。

b.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。

c.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， { 48 %： 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。 }

(4) 信息修改模块

{ 50 % : 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: } { 61 % : 修改密码模块和修改基本信息模块。 } { 61 % : 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。

3.2.3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。 { 44 % : 客户端则需要保存在SQLite中, 下面是客户端所需的数据表: }

信息存储表

字段名类型主键描述

idnumber是信息id

usernamevarchar否对方的名称

headvarchar否头部信息

msgvarchar否消息

sendDatevarchar否发送时间

inOrOutnumber否发送或接受

whosvarchar否谁的

好友关系表

字段名类型主键描述

idnumber是好友id

usernamevarchar否对方名称

sendDatevarchar否发送时间

isDealnumber否删除状态

whosvarchar否谁的

好友关系表

字段名类型主键默认值

idnumber是好友id

msgIdnumber否消息id

msgCountnumber否消息数量

whosMsgvarchar否谁的消息

第四章 系统设计与实现

4.1服务器设计

XMPP 服务器拥有大量的实现方案，其中，较为成熟，且使用广泛的是ejabberd，jabberd 2.x，Openfire这几种服务器。 { 58 %：在这些中，Openfire是基于JAVA，基于XMPP协议、开源的实时协作的服务器。 } 因此，Openfire服务器具有跨平台的优点。 所以，我选择了openfire这个服务器来作为自己的服务端。

Openfire的通信处理是基于Apache MINA框架实现。 { 75 %：Apache MINA是一个网络应用程序框架，作用是使用户开发高性能和高可靠性的网络应用的过程变的更简单。 } { 81 %：它提供了一个通过Java NIO在TCP/IP和UDP/IP等不同的传输协议上抽象的事件驱动的异步API。 }

4.1.1系统启动流程

{ 68 %：系统启动时调用ServerStarter类中的start()方法，通过反射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。 } { 64 %：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。 }

{ 57 %：然后，服务器会通过5222端口监听客户端的连接。 } { 51 %：每次连接，服务器都会进行认证操作。 } { 45 %：认证失败，会返回错误代码给客户端，否则认证通过，开始一个会话。 }

图4.1连接服务器认证

4.1.2消息处理

{ 68 %：在服务器的消息处理中，首先，Openfire服务器需要启动一个基于TCP/IP的监听服务，用以接收客户端传过来的XML流文件。 } 这个过程在XMPPServer类的start()方法中进行。 其中的socketAcceptor是作为一个服务端的接收器，是mina框架为我们封装好的一个socketserver。 我们为socketAcceptor添加了一个过滤器，XMPPCodeFactory，这个类将过滤xmpp相关请求，加以处理，我们再看同一个类的另外一个方法startClientListener()。

然后，启动了监听服务器，来监听所有发送到服务器5222端口的数据。 { 96 %：至此，从客户端到服务器端的packet传递结束。 }

图4.2Openfire消息包接受处理流程

4.1.3数据库处理

{ 100 % : Openfire的数据库处理采用直接调用JDBC 的方式。 } 安装openfire时, 可以选用mysql数据库。MySQL是一种关系型数据库, 关系数据库将不同数据保存在不同的数据表中, 而不是将所有数据放在同一个仓库内, { 53 % : 增加了增删改查速度并提高了操作的灵活性。 } 在填写完JDBC驱动程序, 数据库URL, 用户名, 密码后, 然后再执行配置就可以成功了。 { 43 % : openfire服务器会在mysql数据库里自动新建所需的表和数据。 }

图4.3Openfire数据库配置

4.2客户端设计

为了更加方便的操作XMPP, 我选择了asmmack这个开源库。 { 61 % : Smack是一个开源, 易于使用的XMPP客户端类库。 } Asmmack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, { 100 % : 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, } { 98 % : 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Asmmack这个开源库来完成XMPP的相关操作。

4.2.1连接服务器

{ 51 % : 用户进行任何操作的前提是, 客户端连上服务器。 } { 59 % : 客户端通过服务器的ip地址, 端口号, 域, 来连接到服务器。 }

图4.4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。 { 40 % : 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。 { 48 % : 下图展示了用户发送给服务端的注册请求。 }

图4.5用户注册的代码实现

4.2.3 用户登录

{ 65 % : 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。 } { 47 % : 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。 } { 44 % : 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。 } { 46 % : 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4.6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。 { 46 %： 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。 } { 50 %： 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。 { 45 %： 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4.7搜索好友的代码实现

{ 100 %： Form.getSearchForm()这个方法要求的参数是一个服务名，openfire默认搜索服务是search.+域名。 }
{ 99 %： 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。 }

{ 100 %： 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。 { 43 %： 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4.8添加好友的代码实现

(3) 删除好友： { 43 %： 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， } { 47 %： 服务器收到这个请求，会在数据库中进行操作，然后返回一个 iq result响应。 }

图4.9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。 { 41 %： 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4.10发送消息的代码实现

第五章 功能测试

{ 88 %： 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。 { 45 %：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

{ 55 %：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； { 51 %：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

{ 59 %：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： { 65 %：好友添加，好友列表和删除好友功能。 } { 50 %：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

{ 40 %：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

{ 41 %：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。 { 43 % : 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。 { 46 % : 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： { 100 % : 随着通常超过 70% 的 XMPP 协议的服务器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。 } { 100 % : 新的议定书正在研究，以减轻这一问题。 }

(2) 没有二进制数据： { 100 % : 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。 { 100 % : 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。 } { 100 % : 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。 }

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

{ 47 % : 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。
{ 51 % : 若输入错误，则会显示账号密码错误，需重新输入。 } { 42 % : 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。 { 48 % : 点击正下方的通信录按钮，出现以下页面： }

图A4

{ 62 % : 点击右上角的添加按钮，进入搜索好友的界面： }

图A5

{ 41 % : 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

{ 40 % : 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

{ 45 %：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

{ 46 %：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