

文本复制检测报告单(全文标明引文)

№:ADBD2018R_2018053015312720180530154849440174291476

检测时间:2018-05-30 15:48:49

检测文献: 53141512_罗宏宇_物联网工程_基于Android的聊天系统设计与实现

作者: 罗宏宇

检测范围: 中国学术期刊网络出版总库

中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库

中国重要会议论文全文数据库

中国重要报纸全文数据库

中国专利全文数据库

图书资源

优先出版文献库

大学生论文联合比对库

互联网资源(包含贴吧等论坛资源)

英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)

港澳台学术文献库

互联网文档资源

CNKI大成编客-原创作品库

个人比对库

时间范围: 1900-01-01至2018-05-30

检测结果

总文字复制比: **11.1%**

跨语言检测结果: **0%**

去除引用文献复制比: **9.5%**

去除本人已发表文献复制比: **11.1%**

单篇最大文字复制比: **3%**

重复字数: [3508]

总段落数: [8]

总字数: [31565]

疑似段落数: [6]

单篇最大重复字数: [952]

前部重合字数: [77]

疑似段落最大重合字数: [2357]

后部重合字数: [3431]

疑似段落最小重合字数: [32]



指标: ☐ 疑似剽窃观点 ☒ 疑似剽窃文字表述 ☐ 疑似自我剽窃 ☐ 疑似整体剽窃 ☐ 过度引用

表格: 0

公式: 0

疑似文字的图片: 0

脚注与尾注: 0

0% (0)

中英文摘要等 (总3007字)

1.8% (32)

第1章绪论 (总1748字)

2.5% (45)

第2章安卓相关技术介绍 (总1798字)

12.4% (618)

第3章系统需求分析与结构设计 (总4980字)

20.4% (2357)

第4章系统设计与实现_第1部分 (总11531字)

0% (0)

第4章系统设计与实现_第2部分 (总6295字)

4.3% (39)

第5章系统测试 (总915字)

32.3% (417)

第6章总结与展望 (总1291字)

(注释: 无问题部分 文字复制比部分 引用部分)

1. 中英文摘要等

总字数: 3007

相似文献列表 文字复制比: 0%(0) 疑似剽窃观点: (0)

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

吉林大学学士学位论文(设计)承诺书

本人郑重承诺: 所提交的学士学位毕业论文(设计), 是本人在指导教师的指导下, 独立进行实验、设计、调研等工作

基础上取得的成果。除文中已经注明引用的内容外，本论文（设计）不包含任何其他个人或集体已经发表或撰写的作品成果。对本人实验或设计中做出重要贡献的个人或集体，均已在文中以明确的方式注明。本人完全意识到本承诺书的法律结果由本人承担。

学士学位论文（设计）作者签名：

2018年5月20日

摘要

基于Android的聊天系统设计与实现

在21世纪发展的今天，人们相互之间交流的方式呈现多元化的特点，而随着手机移动系统的火爆以及摩尔定律的作用，即时通讯技术在近十年得到了非常大的发展，因此学习一个基本聊天系统的设计与实现，对工作或者学习是大有裨益的。

要开发一个即时通讯系统软件，最首要的就是选择一个恰当的开发平台，自从谷歌宣布放弃Eclipse ADT的维护更新支持后，功能强大的Android Studio就默认成为开发安卓应用程序的首选。而经过市场调研，功能完善的环信第三方服务为整个即时通讯系统提供了强大的动力，而同时也为该设计确保了可行性，简单地说，就是整个系统采用C/S的设计模式，开发者主要完成客户端的设计，通过环信服务器实现不同用户间的随时交互，而对于服务器端的设计，开发者只需调用一些第三方函数便可以轻松实现想要的功能。对于客户端的设计，将采用主流的Java面向对象语言，这既是由Android Studio的内在语言支持所决定的，又是由Java本身特性符合Android开发思想的特点所决定的。开发过程中不可避免的要与数据库进行交互，而Android系统内置的SQLite轻量数据库便是该设计最好的选择。

在经过细致而严谨的需求分析调研后，论文对客户端各个功能模块进行了剖析，客户端将要实现欢迎模块、登录注册模块、消息列表模块、好友列表模块、注销用户模块、添加好友模块、聊天模块以及后台消息模块八个功能不同模块。因为工作时间有限，本文也在最后给出了未来该系统的改进意见，方便在下一步工作中获得更好的用户体验。

关键字：即时通讯，SQLite，C/S，Android，Java

Abstract

Design and Implementation of Chat System Based on Android

In the development of the 21st century, the way in which people interact with each other presents a plurality of characteristics. With the popularity of the Android system and the role of Moore's Law, instant messaging technology has developed greatly in the past decade. Therefore, the design and implementation of a basic chat system must be studied. Learning Android system is worthwhile.

To develop an instant messaging system software, the most important thing is to choose an appropriate development platform. Since Google announced its abandonment of Eclipse ADT's maintenance and update support, the powerful Android Studio has become the default choice for developing Android applications. After market research, a fully functional third-party service provided a powerful driving force for the entire instant messaging system, and at the same time ensured the feasibility of the design. In short, the entire system adopts the C/S design model. The developer mainly completes the design of the client and realizes the interaction between different users at any time through the ring letter server. For the server-side design, the developer can easily realize the desired function by simply calling some third-party functions. For the client's design, the mainstream Java object-oriented language will be used, which is determined by the Android Studio's internal language support, and is determined by the nature of Java in line with the characteristics of Android development. It is inevitable to interact with the database during the development process, and the SQLite lightweight database built into the Android system is a good idea.

On the basis of comprehensive requirements analysis, the paper analyzes the various functional modules of the client, the client will implement the welcome module, login registration module, message list module, buddy list module, logout user module, add friend module, chat module And eight different modules of the background message module. Because of the limited working time, this article also provides suggestions for future improvement of the system to facilitate a better user experience in the next step.

Keywords：Instant Messaging，SQLite，C/S，Android，Java

目录

第1章绪论	1
1.1 研究背景	1
1.2 论文工作	2
1.3 论文组织	2
第2章安卓相关技术介绍	4
2.1 安卓体系介绍	4
2.2 安卓发展现状介绍	5
2.3 安卓开发介绍	5

2.3.1 安卓应用开发特色	5
2.3.2 安卓开发平台特点	6
第3章系统需求分析与结构设计	7
3.1 大体业务流程简述	7
3.2 系统概要设计	8
3.2.1 系统物理结构设计	10
3.2.2 系统逻辑结构设计	10
3.3 各功能模块需求分析	12
3.3.1 欢迎界面	12
3.3.2 用户注册登录	12
3.3.3 消息列表	12
3.3.4 好友列表	13
3.3.5 用户注销	13
3.3.6 添加好友	13
3.3.7 好友聊天交互	13
3.3.8 后台服务	14
3.4 本章小结	14
第4章系统设计与实现	15
4.1 总体设计	15
4.2 服务器端的设计与架构	15
4.3 客户端整体结构实现	17
4.4 客户端各功能模块具体设计与实现	21
4.4.1 用户登录注册功能设计与实现	21
4.4.2 获取好友列表设计与实现	22
4.4.3 添加好友功能设计与实现	23
4.4.4 删除好友功能设计与实现	26
4.4.5 用户聊天功能设计与实现	27
4.4.6 用户注销功能设计与实现	29
4.4.7 后台消息功能设计与实现	30
4.4.8 数据库功能设计与实现	30
4.5 本章小结	31
第5章系统测试	32
5.1 测试工具及环境	32
5.2 测试流程及结果	32
5.2.1 用户登录注册功能测试	32
5.2.2 用户添加好友功能测试	33
5.2.3 好友聊天交互功能测试	33
5.3 本章小结	34
第6章总结与展望	35
6.1 工作总结	35
6.2 下一步工作	35
参考文献	36
致谢	38

2. 第1章绪论		总字数：1748
相似文献列表 文字复制比：1.8%(32) 疑似剽窃观点：(0)		
1	54130822_张琼_软件工程_前端工程化的研究与实践 张琼 - 《大学生论文联合比对库》 - 2017-05-31	1.8% (32) 是否引证：否
2	54130804-康锴-软件工程-基于Flink的社区发现算法的设计与实现 康锴 - 《大学生论文联合比对库》 - 2017-06-01	1.8% (32) 是否引证：否

3	54130822_张琼_软件工程_前端工程化的研究与实践 张琼 - 《大学生论文联合比对库》 - 2017-06-05	1.8% (32) 是否引证：否
4	7_康锴_基于Flink的社区发现算法的设计与实现 康锴 - 《大学生论文联合比对库》 - 2017-05-24	1.8% (32) 是否引证：否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第1章绪论

1.1 研究背景

随着科技的不断发展、社会生产力水平的不断提升，手机的功能在近十年得到很大地完善，处理能力和硬件配置都得到质的提升，于是人们的交互方式呈现多元化，不仅仅局限于打电话的方式，因此，即时通讯软件如雨后春笋般繁荣起来，据统计，截止至今年5月份，微信和QQ的月活跃用户相加总和超过17亿，超过目前中国人口的总人数，因此，不难看出即时通讯软件的火爆程度。这些即时通讯软件功能强大且丰富，均有不同的定位群体，部分用户群体甚至选择多个即时通讯软件。在移动通信的发展过程中，目前发展的趋势为互联网逐步从pc端转向手机平板转向各种移动端。目前社会上存在的两大移动端系统是IOS和Android，而其中，由于安卓系统开源的特性，得到了快速发展[1]。据统计，安卓操作系统的国内市场占有率已超过八成，IOS却只占比20%。

其实，即时通讯软件出现已有比较长的时间了，自1998年第一代即时通讯软件出现后，便如雨后春笋般快速发展着。同时，人们对的需求大幅上涨，即时通讯系统的角色早已经不仅仅是一个单纯的聊天工具，而是逐渐演变成为了一个内容丰富、功能齐全的信息应用平台，近两年直播行业异军突起，对即时通讯软件性能提出了更高的要求。目前，市场上已经出现了不少专门提供即时聊天功能的第三方服务，如环信融云等，这极大地降低了开发即时通讯软件的门槛，这些第三方服务只把必须的用户信息（例如用户ID、头像、昵称等）保存在第三方的服务器上，而用户关键记录（密码、通讯记录等）则保存在自己的服务器或者本地数据库，为开发人员提供了便利，极大地减轻应用程序开发成本，目前不少企业都采取这种开发模式，未来这种模式将进一步普及。

在二十一世纪的今天，开发效率永远是放在第一位的，跟其他行业一样，开发也是一个前人栽树后人乘凉的过程，合理利用第三方服务，将自己精力集中在更重要的地方，这应该是每个开发人员的共识，但不是说个不加理解的加以应用，而是要从提供的服务中梳理出整个功能的实现框架，并举一反三地应用到其他项目中，形成一个良性循环。因此，掌握一个利用第三方服务来开发即时通讯软件的技术是很有必要的，不仅满足未来的技能要求，更能从优秀的第三方服务中学到很多规范的设计模式、封装技巧等。

1.2 论文工作

本文的主要工作有以下几点：

- * 分析安卓系统的现状，阐明安卓开发体系及核心思想，将其思想特点体现在实际编程当中，解释其系统的可行性。
- * 分析现阶段几种常见的集成第三方服务的编程模式的特点，总结该编程模型的优势，并将该编程模式与传统的编程模式作比较，分析解释其优越性的原因，根据第三方服务的文档，力争熟悉每一个服务的模块，分析其采用的通讯协议等。
- * 在Android Studio编程平台上完成作品的代码部分，设计程序模型执行过程，在真机和安卓虚拟机上完成程序的调试，并给出程序的类图、流程图、逻辑示例图以及部分核心实现代码。

1.3 论文组织

本文的后序章节按如下的方式进行组织：

- * 第二章：给出了一些Android系统的基础知识，同时也简单地叙述了Android系统发展历史，对Android各版本的市场份额做出比较，便于开发者选择合适的开发版本，最后，还对AndroidStudio这个强大的Android开发工具做出了介绍，分析了这个工具的优点。
- * 第三章：一方面，从各模块功能需求上要求设计的即时通讯系统具有用户注册登录、添加删除好友、发送接收信息等功能，系统阐述了各功能的设计要点，分析一些编程技巧，确定本系统的可行性。另一方面，根据系统需要对系统进行了概要设计，主要包括逻辑结构设计和物理结构设计，列出数据、消息在客户端和服务端的传输图，画出系统的拓扑关系图，分析采用该设计结构的好处。
- * 第四章：比较系统地阐述了系统的架构设计，分别给出了系统总体架构、服务端结构、客户端模块结构。同时还剖析了即时通讯系统客户端的功能模块的重点技术，给出了具体的实现代码。
- * 第五章：展示了几个核心模块的测试过程及结果，检验了模块是否达到预期目标，让用户更易理解其过程。
- * 第六章：比较全面地对本文工作做了一个较为详细的总结，并且对系统的未来所需要的改进提出了一些意见。

3. 第2章安卓相关技术介绍		总字数：1798
相似文献列表 文字复制比：2.5%(45) 疑似剽窃观点：(0)		
1	基于Android智能终端的多协议接入网关 刘俊杰;王番;- 《可编程控制器与工厂自动化》 - 2013-01-15	2.5% (45) 是否引证：否

第2章 安卓相关技术介绍

2.1 安卓体系介绍

Android体系结构的思想是分层架构 [2]，如下图所示，从上到下主要可以分为四层架构：应用层、应用框架层、系统运行库层、linux内核层。

图 2-1 Android系统架构分层图

应用层(Application)：在此层上，包含了所有用户下载安装或内置的应用程序，系统自带的短信、照相机、手电筒等程序，又或是自己安装的QQ、微信等程序，均属于该层。该层提供了一些核心应用的程序包，使得各个程序能够在Android系统上和谐共处。

应用框架层(Application Framework)：开发者主要是跟本层打交道，每一个开发人员都是从该层一步一步地编写整个程序的。该层提供了构建应用程序时可能用到的各个API，合理使用这些API是每一个开发者的必修课。

系统运行库层(Android Runtime & Libraries)：系统库包含九个子系统，Android系统的主要支持就是由这九个子系统提供的。而Android运行时包含两个子系统，主要提供了一些核心库，同时也兼容了大多数Java所需要的功能函数，更为强大的是，该层拥有一个专门针对移动设备的虚拟机，极大地优化了移动设备的运行性能。

Linux内核层(Linux kernel)：其实，linux内核层是Android系统核心服务的基础，它为每一个Android设备提供了最基础的底层驱动支持。

2.2 安卓发展现状介绍

在15年前，安迪·鲁宾等人创立了Android公司，这是Android系统的源头，虽然该公司之后被谷歌收购，但谷歌仍旧保持原有团队不便，终于在2008年，正式发布了Android 1.0，随之打开了安卓系统快速发展的大门，接下来的几年，谷歌以疯狂的速度不断更新Android系统，除了出现的少数失败版本外，几乎均获得了巨大的成功。随着2017年10月份微软宣布Windows Phone系统不再是发展重点后，手机系统上出现了苹果IOS系统和谷歌Android系统“二分天下”的局面[3]，如今的Android系统无论是流畅性还是续航性等方面，均得到了显著提升，目前最新的Android系统是2017年8月发布的Android Oreo(8.0)，下表展示了目前市场上主要的Android系统版本和详细信息[4]。

表 2-1 各Android系统版本市场信息表

2.3 安卓开发介绍

在本小节中，将会比较详细地介绍Android系统的开发特色以及Android系统开发平台特点。

2.3.1 安卓应用开发特色

简单来说，Android应用开发特色主要包括以下几方面：

1. 四大组件[5]：Android系统的四大组件分别是活动、服务、广播接收器以及内容提供者。活动是所有程序的根本，服务为程序在后台运行提供了可能，广播接收器提供了对事件的响应功能，而内容提供者使得独立程序间的数据交互不再是难事。
2. 丰富的控件系统：Android系统所拥有的系统控件五花八门，使得开发者编写一个漂亮的交互界面是一件再简单不过的事，同时开发者也可以根据项目要求自定义一些控件。
3. SQLite数据库：Android系统独具匠心地嵌入了这种便捷高效的关系型数据库，开发者只需通过SQL语法或者一些函数接口，可以很方便地进行数据库的操作，让存储、读取数据变得非常方便。
4. 强大的多媒体：安卓系统所特有五花八门的多媒体服务，使应用程序更能满足现代人们的需要。
5. 地理位置定位：与PC相比，这是移动设备最大的亮点之一，与强大的地图功能结合起来，能轻而易举地做出创意十足的应用程序。

2.3.2 安卓开发平台特点

Android Studio是谷歌公司开发的一款Android集成开发工具，现如今，几乎所有Android程序都是在此平台上开发完成的。具体来说，它在IDEA的基础上，又提供了一些额外的功能[6]：

1. 基于强大的构建系统Gradle的支持；
2. 具有Android所特有的重构和快速修复功能；
3. 所特有的错误提示工具可以让开发者轻松解决版本兼容等常见问题；
4. ProGuard和应用签名在该工具上均得到支持；
5. 界面编辑器的功能非常强大，可以便捷在窗口中拖动UI控件并且能进行效果预览。

4. 第3章系统需求分析与结构设计 总字数：4980

相似文献列表 文字复制比：12.4%(618) 疑似剽窃观点：(0)

1	基于android系统的即时通讯系统设计与实现 杜辽(导师：杨公平) - 《山东大学博士论文》 - 2014-04-20	10.4% (518) 是否引证：是
2	基于J2EE平台和MVC模型的综合素质测评系统的设计和实现	0.7% (33)

杜垚(导师：吴跃;高枫) - 《电子科技大学博士论文》 - 2011-04-07		是否引证：否
3	web 实训 - 《互联网文档资源 (http://wenku.baidu.c) 》 - 2017	0.6% (32)
4	在线订餐系统设计 计晓刚 (51002011004) - 《大学生论文联合比对库》 - 2014-05-05	是否引证：否
		0.6% (30)
		是否引证：否

原文内容 **红色文字**表示存在文字复制现象的内容; **绿色文字**表示其中标明了引用的内容

第3章系统需求分析与结构设计

3.1 大体 业务流程简述

根据用户需求，本论文主要完成，对于Android手机即时通讯软件系统的设计与实现，利用环信第三方服务，将实现三个页面，分别为消息列表页面、好友列表页面以及用户注销的设置页面，其中默认的主页面是消息列表页面。本即时通讯系统将实现用户登录注册功能、用户注销功能、信息交互功能等传统即时通讯所必须的功能。对于一个新用户，**首先进入的是注册登录界面，用户首先要输入用户名密码进行注册**，然后才能登录，登录完成后转向消息列表页面，通过点击底部三个按钮实现页面的切换，在好友列表页面默认为空，通过点击右上角的加号实现好友的添加，别人同意后便互相成为了好友，可以开始即时通讯了，第三方服务默认集成了拍照、图片、语音、文字的聊天方式，而这些文件都是通过HttpPost类的相关方法实现传输[7]。在用户设置注销界面可以注销当前用户，退出本系统再打开会默认登录上一个未被注销过的登录成功的用户。

具体用户示例图如下

图3-1 用户操作示例图

1. 用户注册：新用户通过输入想申请的ID号、密码完成用户的注册，若ID号已被注册则会报错。
2. 用户登录：只有用户ID与密码能正确匹配上，才能实现用户登录，不然会报错。
3. 消息管理：在此页面可以实现删除消息条目、点击消息条目进入聊天的功能，同时在全局注册了消息提示的广播。
4. 好友管理：这部分比较复杂，集成了第三方的服务页面，用户可以看到当前好友列表，可以删除好友，可以通过点击右上角加号实现添加好友，可以点击好友条目进入聊天，可以根据上方好友邀请的红点状态判断是否有新的好友添加信息。
5. 注销用户：实现注销当前已登录的用户。

界面顺序图：

本即时通讯软件系统**通过以下几个主要界面实现：欢迎界面、用户登录注册界面、消息列表界面、好友列表界面、注销用户界面、添加好友界面、接受邀请界面、**即时通话界面。具体图例如下：

图3-2 即时通讯系统页面切换图

3.2 系统概要设计

在本小节中，将会对本即时通讯系统做出整体结构设计，主要从物理结构设计和逻辑结构设计两方面入手，依循C/S设计模式，给出环信服务器端、客户端、数据库端（本地服务器端）的连接问题以及传输数据的模式。

该系统总体上可以分为以下三个部分：

环信服务器端：环信服务器是本系统中最核心的部分，环信服务器经过十多年的沉淀已经取得很好的运行性能，环信服务器允许日活用户（即在线用户）不超过100个，对于每一个注册登录的用户，都会有一个固定的环信ID与之对应，便于用户管理，其中用户收发的消息都是先通过本地服务器（如果有的话），再通过环信服务器，最后环信服务器通过保存的用户列表来进行转发。环信服务器通过基于TCP研发的二进制私有协议——msync使得数据传输更方便高效，也更省流量，环信IM后台主要架构如下图所示：

图 3-3 环信服务器架构

环信服务器主要分为IM服务器和REST服务器，IM服务器用于保存全部用户列表，管理用户，而REST服务器主要用于保存用户的好友列表，消息交换也是通过REST服务器，环信服务器不会保存用户的关键信息，环信服务器只有app的两个数据：环信ID和环信ID所对应的密码，其他的信息只会保存一段很短的世界，所有要想实现长时间段的聊天记录等信息的保存，则必须要保存在app自己的服务器中，环信后台服务API是基于OAuth2.0标准和RBAC的权限模型的[7]。

本地数据库：本地数据库主要保存了在本地上登录过的用户信息，比如邀请信息、用户好友列表、用户账户等，相当于替代了app服务器，具有快速获取用户数据、实现常用数据长期存储的功能。本地数据库采用的是AndroidStudio内嵌的SQL微型数据库，主要实现了联系人表、邀请信息表、用户账号信息表三张表，在未来的信息交互中，客户端和服务端会频繁地与本地数据库打交道。

客户端：客户端工作的时候首先要向环信服务器发送请求数据，环信服务器将获得的环信ID和密码校验成功后，服务器便可知客户端将有数据与之交互，客户端通过改进的TCP协议与服务器端进行数据传输，客户端的功能还包括对通信系统的初始化，当打开软件的时候，就在后台通过GPRS网络实现客户端与服务端的连接，环信服务端作为通信中枢，可以与多个客户端进行连接，从而可以实现多个客户端间的实时通讯。

3.2.1 系统物理结构设计

该即时通讯软件的物理结构设计如下图所示，可以看到，该系统主要仍采用典型C/S设计结构，主要由三个不同的主体组成：服务器端、用户客户端、好友客户端[8]，其中服务器端起着中心枢纽的作用，方便用户客户端与好友客户端**之间的信息交互，而用户客户端和好友客户端所实现的功能几乎一样，即用户注册登录、添加好友、信息接收发送等功能，该系统的物理**

结构拓扑图类似于局域网的星型拓扑图结构，如图所示：

图 3-4 即时通讯系统物理拓扑结构

3.2.2 系统逻辑结构设计

该Android应用客户端采用先进成熟的MVC的设计模式，具体介绍如下：

MVC(Model-View-Controller)模式：这是一种被开发者广泛采用的设计模式，该模式将应用程序软件分为三个主要开发部分——模型层、视图层以及控制层，每一层都是互相独立的，项目代码根据所产生的作用分配到不同的层次上，从而可以减少开发过程中代码冗余、混乱等问题，为程序的阅读提供了便利，有利于程序的协同开发。通过参考学习这种先进的程序开发模式，对我们以后工作学习是大有裨益的。

模型层(Model)：在本软件系统开发中，模型层主要由数据库、数据库操作类、全局事件监听、用户bean类以及一个很特别的数据-模型层全局类组成，该层主要用来处理在程序中出现的数据库操作问题，网络的连接问题和数据库管理问题等。全局事件监听主要具有获取联系人广播、邀请广播，发送广播的功能，而数据-模型层全局类则具有获取全局上下文、全局线程池、用户账号数据库等功能。

视图层(View)：由字面意思可以看出，视图层的主要功能便是用来显示界面，它用以显示数据的输入界面和模型数据等。在Android开发中，视图层主要由Activity组件、Dialog及一些Service组成，页面布局文件多采用XML文件来编写，使用时可以根据控件的ID非常便捷的引入并执行相关操作，Android系统可以很方便的实现Java与JavaScript之间的通信。Android提供了很多功能强大的控件，通过改变控件的属性可以实现非常精美的UI界面，另外，开发者编写出专属的自定义控件也不失是一个好选择。

控制层(Controller)：控制层负责的功能众多，既包括对用户可见部分，同时也包括不可见部分，可以对用户展示流畅的页面跳转，还可以在后台管理用户数据的交互，在整个应用中起着核心作用。在整个Android应用程序的开发中，AndroidManifest.xml扮演者控制器的角色，该文件对系统所需要的各种组件、视图、权限进行了相应的配置，同时也为系统的API的版本做出了要求[9]。

系统客户端结构图如下所示：

图 3-5 系统客户端结构图

3.3 各功能模块需求分析

在本小节中，将会给出客户端各个功能模块的需求分析，列出每个模块所需要完成的功能，为下一章的具体实现奠定基础。

3.3.1 欢迎界面

当点击程序的时候，将会打开程序，首先会有一段长约2秒的欢迎界面，该界面由吉林大学LOGO和开发者姓名组成，主题界面颜色以白色为主。若上一个已登录过的用户并没被注销，则此时将会自动登录上一个用户，并且欢迎界面结束后自动转入消息列表页面，否则将转入用户登录注册界面。

3.3.2 用户注册登录

用户注册登录界面，主要由两个输入框和两个按键组成，一个输入用户ID，一个输入用户密码，一个按键实现注册，一个按键实现登录。对于一个新用户，必须先在ID和密码上填写正确的字符串并点击注册按钮才能在本数据库及环信服务器创建相应的用户序号。在用户完成注册后便可以进行登录操作了，输入用户ID及正确的密码后，点击登录按钮即可实现用户登录操作。若当前用户ID没经过注册则会弹出报错信息，密码与用户ID不匹配同样也会弹出报错信息，用户ID或密码为空也会弹出报错信息。只有正确的用户ID和正确用户密码相匹配后才能实现用户的登录，此时会在环信服务器更新用户状态，在本数据库则会根据用户ID号创建相应的User用户数据库。

3.3.3 消息列表

消息列表界面主要由ListView控件和消息条目Item组成，底部同时会有三个按键用以实现消息列表、联系人列表、用户注销三个界面的切换。该页面会显示最近用户聊天的消息条目，同时若有未读的新消息条目则会自动置顶并且在右上方有新消息数目提示。当用户点击通知栏上的消息条目时则会跳转到聊天界面，同时会清除原来的新消息数目提示。

3.3.4 好友列表

用户通过点击底部的第二个按键切换到好友列表界面，该界面主要由顶部标题栏的加号、新邀请提示按键、以及好友界面组成，其中好友界面按首字母字典顺序展示，旁边还有各字母顺序方便查找，这一主要的好友界面是集成了环信SDK上的UI界面。用户可以通过点击右上角的加号跳转到添加好友界面，当用户收到好友邀请信息时，新邀请提示按钮旁边会出现小红点，用户可以通过点击新邀请提示按键进入邀请管理界面，在好友界面用户可以通过长按好友条目操作实现删除好友功能，添加或删除好友后均要更新环信服务器上的数据和本地服务器的数据，用户也可以通过单机好友条目进入聊天页面。当用户好友较多时，则可以通过在界面上放的搜索栏输入好友ID或在右方根据第一个字符的字典顺序，实现好友的快速查找。

3.3.5 用户注销

该界面主要由注销按钮和一些未实现的设置组成，设置因未实现于是用setting字符串模拟，显示在主要部分上，底部是一个红色的注销按钮，点击此按钮即可注销当前用户，同时跳转到用户登录注册界面。

3.3.6 添加好友

该界面主要由用户ID的输入框和查找按钮组成，在下方空白处有好友的邀请条目，该条目默认是不可见的，只有用户输

入了欲添加的用户ID并点击查找后才会可见，点击发送按键即可发送添加好友邀请广播。

3.3.7 好友聊天交互

该界面主要是由环信UI模块帮我们完成，用户可以通过在聊天输入框输入消息并点击发送即可实现与好友即时通讯。通讯并不仅仅局限于文字交流，也可以发送语音、图片、表情，用户通过点击喇叭按键即可切换到语音输入模式，若第一次则会有获取权限提示，根据提示可以完成语音聊天，用户也可以点击输入框旁边的圆脸按钮发送表情，点击加号按钮可以发送图片等等，同样地，这些操作也要获取相应的权限。界面中部以Listview的方式展示用户间的聊天记录，若消息条目太多也可以通过上拉查看历史聊天记录。

3.3.8 后台服务

该聊天软件因为集成了环信SDK而实现了后台消息通知功能，当软件在后台运行时，若有新消息收到则会在手机上，伴随着提示音在通知栏上弹出一个对话框，用户可以通过下拉手机主界面点击软件通知进入到系统的聊天界面，如果有添加好友的消息也会产生相应的通知对话框。

3.4 本章小结

本章首先从系统应用的业务流程出发，接着按照系统物理结构和逻辑结构两方面，刻画出系统的大致设计，所给出的流程图可以使用户更加明确系统的操作过程。最后对该系统的主要功能模块：欢迎界面、用户登录注册、聊天交互等进行了较为详细的需求说明，为软件的架构实现提供了良好的基础。

本章同时从MVC设计模式出发，将应用程序分为三个主要部分，解释各部分的**功能作用。然后从系统的物理结构和逻辑结构两个方面对系统设计做出了较为详细的介绍，物理结构主要是给出了**系统的物理拓扑图，该拓扑图类似于局域网中的星型拓扑结构，位于中央的服务端可以同时与多个手机客户端相连。而系统的逻辑结构则是从MVC设计模式出发，较为详细地介绍了客户端的逻辑结构划分，采用“分而治之”的思想，将一整个大的聊天系统细化到三个主要的部分，使系统易于控制、延展和分配资源。

5. 第4章系统设计与实现_第1部分		总字数：11531
相似文献列表 文字复制比：20.4%(2357) 疑似剽窃观点：(0)		
1	Android平台下的简易泛在社交系统的设计与实现 陈彩云 - 《大学生论文联合比对库》 - 2017-06-02	8.3% (952) 是否引证：否
2	信计1301_1310820209_饶建雄 饶建雄 - 《大学生论文联合比对库》 - 2017-05-16	8.1% (938) 是否引证：否
3	基于移动终端的沂水景点智能导游系统设计与实现 王洁 - 《大学生论文联合比对库》 - 2016-05-24	4.2% (486) 是否引证：否
4	基于移动终端的沂水景点智能导游系统设计与实现 王洁 - 《大学生论文联合比对库》 - 2016-05-26	4.2% (486) 是否引证：否
5	基于Android开发的掌上医院系统 杨爱华;丘斌; - 《电子技术与软件工程》 - 2014-09-15	3.7% (428) 是否引证：否
6	2011202912_王鑫_Android课程表设计论文 王鑫 - 《大学生论文联合比对库》 - 2016-01-04	3.2% (366) 是否引证：否
7	基于Android的汽车保养系统的设计与实现 马杰 - 《大学生论文联合比对库》 - 2017-06-02	2.8% (322) 是否引证：否
8	基于Java的常见二维码自动生成系统的设计与开发 余淼 - 《大学生论文联合比对库》 - 2015-04-19	2.7% (306) 是否引证：否
9	基于Android平台的移动终端GPS研究 倪红军; - 《计算机技术与发展》 - 2012-05-10	2.5% (290) 是否引证：否
10	基于Android的Google Map服务系统开发 吴元君; - 《合肥学院学报(自然科学版)》 - 2014-05-15	2.5% (290) 是否引证：否
11	Android手机趣味APP开发之账号信息管理 汪永松; - 《电脑编程技巧与维护》 - 2015-02-03	2.5% (290) 是否引证：否
12	Android图片浏览器的实现 曾杰; - 《电脑编程技巧与维护》 - 2015-04-18	2.5% (290) 是否引证：否
13	中小规模电商系统移动端应用开发框架研究 马海庆; - 《电子技术与软件工程》 - 2016-03-25 0	2.5% (290) 是否引证：否
14	智能手机个性化主题设计与开发 罗孟;周瑞秋;陶卫平; - 《大众科技》 - 2016-09-20	2.5% (290) 是否引证：否
15	基于Android的智能手机室内导航路径规划设计	2.5% (290)

林育曼;潘敏洁;卓伟鹏;饶浩; - 《电脑与电信》 - 2017-08-10	是否引证：否
16 AR开发中的Unity 3D与原生代码交互技术	2.2% (248)
魏三强; - 《重庆理工大学学报(自然科学)》 - 2017-11-15	是否引证：否
17 基于Android平台的计算机理论考试宝典	1.5% (176)
秦文文; - 《电脑编程技巧与维护》 - 2014-05-03	是否引证：否
18 201092295_张博_基于Android 的新闻APP的设计与实现	1.0% (120)
张博 - 《大学生论文联合比对库》 - 2014-06-12	是否引证：否
19 李智-11101018-软件工程	0.7% (79)
李智 - 《大学生论文联合比对库》 - 2015-06-04	是否引证：否
20 软件1108-1111611621-曹嘉伟	0.5% (57)
曹嘉伟 - 《大学生论文联合比对库》 - 2015-05-19	是否引证：否
21 基于Android的美食推荐与交友APP设计与实现	0.3% (40)
覃春燕 - 《大学生论文联合比对库》 - 2016-04-28	是否引证：否
22 高校医务室综合管理系统的设计与实现	0.3% (31)
覃晓琼 - 《大学生论文联合比对库》 - 2015-04-09	是否引证：否

原文内容 **红色文字**表示存在文字复制现象的内容; **绿色文字**表示其中标明了引用的内容

第4章系统设计与实现

4.1 总体设计

本系统的设计实现了Android平台上的即时通讯，多个不同的手机客户端可以通过填写正确的用户ID和密码登录到环信服务器端，便可以方便快捷地收发信息，由此得出，本设计系统的整体架构采用了当前较成熟、先进的客户端/服务端(C/S)[10]体系结构。本设计的服务端采用的是环信第三方服务器，该服务器主要完成管理服务器端与客户端连接的任务，以及准确无误地交付消息的任务。

现阶段，大多数Android手机都是通过LTE(4G)标准进行数据传输，当连接上环信服务器后，便可以通过服务器实现两个或多个Android客户端之间的信息交互。在信息交换的过程中，客户端与服务器端的连接协议是TCP协议，而通信协议则是最简单原始的HTTP协议，在整个移动网络中，移动端始终接入无线网络，无线信号经转换后变为有线信号，继而可以通过HTTP协议实现与服务器端的连接。

服务器端与客户端的连接示意图如下：

图 4-1 服务器端与客户端的连接示意图

4.2 服务器端的设计与架构

环信服务器端的作用主要功能是完成管理服务器端与客户端连接，以及要准确无误地在不同客户端间交付消息[11]。环信服务器主要分为IM服务器和REST平台服务器，IM服务器功能主要是管理用户的上线情况，而REST服务器主要是暂存用户的交互数据记录等。**REST服务方式提供了环信服务端的接口，REST API基于最简单的HTTP协议[12]，在各种编程语言中都提供了良好的支持，调用REST API，本质上就是发送HTTP请求，主要通过其POST和GET请求交互数据。**

因为环信服务器是环信公司自己的私有产品，因此本节只能给出环信服务器的整体架构概述。环信服务器是一个支持多租户的云服务平台，且是支持“用户”——“APP”两级结构的，即在平台中，每个平台注册用户ID之间的数据都是严格相互隔离的，而每个平台注册的用户ID内部的每个APP直接的数据，也是严格相互隔离的。最后，因为环信提供的是REST服务，并且前面也说明了，REST本质就是通过HTTP协议的四种常用操作方法来操作资源(URL)[13]，所示可以看到在下面环信的REST API[14]中，就体现了该思想。

假设一个APP在环信上注册的ID为easemob-demo，然后该ID下有一个APP系统叫做chatdemoui，那么环信的REST API如下所示：

获取这个APP下的所有用户：

Path : /easemob-demo/chatdemoui/users HTTP Method : GET Request Headers : { Authorization : Bearer \${token} }
Path : /easemob-demo/chatdemoui/users HTTP Method : GET Request Headers : { Authorization : Bearer \${token} }

获取这个APP下一个用户stliu的用户详情：

Path : /easemob-demo/chatdemoui/users/stliu HTTP Method : GET Request Headers : { Authorization : Bearer \${token} }
Path : /easemob-demo/chatdemoui/users/stliu HTTP Method : GET Request Headers : { Authorization : Bearer \${token} }

在这个APP下创建一个新的用户：

Path : /easemob-demo/chatdemoui/users HTTP Method : POST Request Headers : { Content-Type : application/json, Authorization : Bearer \${token} } Request Body : { "username": "stliu1", "password": "123456" }
Path : /easemob-demo/chatdemoui/users HTTP Method : POST Request Headers : { Content-Type : application/json, Authorization : Bearer \${token} } Request Body : { "username": "stliu1", "password": "123456" }

删除一个用户：

Path : /easemob-demo/chatdemoui/users/stliu HTTP Method : DELETE Request Headers : { Authorization : Bearer \${token} }
Path : /easemob-demo/chatdemoui/users/stliu HTTP Method : DELETE Request Headers : { Authorization : Bearer \${token} }

\$(token)} }

从以上的URL操作规则中，不难看出“注册用户”-“APP”-“实际用户”之间的层层递进的关系。

4.3 客户端整体结构实现

本文所设计的即时通讯系统的客户端的功能较全面，基本实现了传统即时通讯系统上的所有功能：用户注册登录、添加好友、聊天交互等功能。但对比当今流行的即时通讯软件，功能还是稍显单一，不过系统客户端的设计具有模块化的特征，模块化最大的优势就是，开发人员可以根据需求，可以很方便地扩展增加系统的功能，满足现实的需要。

根据系统的初步需求分析，本系统可以按照MVC设计模式分为以下几个模块包：Controller模块：包含activity包、adapter包和fragment包，Model模块：包含bean包、database包、dao包和全局事件监听类，Utils模块：包含一个常量类constant和一个工具类sputils类。在AndroidStudio平台下的APP开发过程中，每一个APP的开发都会有一个较紧密的工程结构，通过使用这种工程结构，可以方便地对程序中用到的组件进行管理，为程序开发提供了便捷。

平常的Android源码项目文件基本都是会被分在这几个文件夹下面：src文件夹、gen文件夹、res文件夹以及AndroidManifest.xml文件等[15]，简要介绍下各文件夹的内容：

1、src文件夹：该文件夹主要包含了项目程序中所应用到的源代码包，每个包都包含有不同的类，这些不同的类经过相互合作使得每个包都能完成不同的功能，例如：activity包实现主要的页面功能，adapter包主要实现listview的适配器功能，fragment包实现主页面下三个页面的切换功能，model包建立了程序中所用到的数据模型，db包实现了数据库操作的相关功能等等。

2、gen文件夹：R.java文件是该文件夹下最核心的文件之一，所有res目录下的XML描述文件、图像等资源都包含在此文件内，该文件在项目建立的时候自动生成，而且在开发过程中也会随着文件、图像等资源的改变而自动修改，原则上来说不需要手动进行修改。R.java中自动产生的R类，为项目文件中涉及到的所有界面、图像、字符等各式各样的资源提供了相应的编号。编译的时候编辑器也会查看这个R文件，如果编译过程中某些资源没有使用到，则那些资源不会被编译进去，这种方式极大地减少了手机应用程序的空间开销。

3、res资源文件夹：在该文件夹下，包括的文件夹都有一些差别，子文件不同则内容也会不一样。drawable文件夹里保存着程序需要用到的图片，分为hdpi、xhdpi、xxhdpi等以适应不同分辨率的需求[16]。layout文件夹里放置了各种xml布局文件，这些xml文件通过设置不同控件的不同属性，向用户展现出精美的交互界面。values文件夹保存程序代码中所用到的所有字符串资源，这样做的好处是将布局与字符串分离出来，方便开发过程中的更新修改。

4、AndroidManifest.xml文件：每一个安卓应用程序都少不了这个文件，该文件在安卓系统的应用中声明了重要的信息，信息系统必须具备之前，它可以运行任何应用程序的代码。除其他事项外，清单中还执行了以下操作：

1 它声明了应用程序为访问受系统保护的API接口或者与其他应用程序交互时所必须申请的权限。

2 它负责为应用程序的每个Java包命名。

3 它表明了应用程序组件的活动、服务、广播接收机、内容提供商等信息，这些声明可以让安卓系统知道这些组件的名称以及运行的前提条件等信息[17]。

4 它决定了进程将运行的应用组件，同时也决定了其他程序与之交互是所需要的各项权限[18]。

5 它给出了当前Android API的应用程序需要的最低安卓系统型号。

6 它列出了各项库，每个应用程序必须与之形成一一配对[19]。

该客户端在AndroidStudio集成开发环境下的工程目录结构如下所示：

图 4-2 工程目录结构图

该聊天系统主要应用到的视图包括：登录注册页面、添加好友页面、聊天页面、注销页面、邀请信息页面等，视图包内所含文件结构如下所示：

图 4-3 视图控制包结构图

功能模块包括登录注册、添加好友、邀请信息、注销、欢迎界面等操作，具体结构图如图4-4所示：

图 4-4 功能包结构图 图 4-5 模型-数据包结构图

模型控制层包括数据库的创建、修改、删除等操作，也包括全局线程类和全局事件监听类等。具体结构图如图4-5所示：

目前与服务器端的主流即时通讯协议有四种：IMPP、SIP、SIMPLE、XMPP[20]。而在这四种协议中，本系统采用XMPP即时通讯协议，XMPP协议全称可扩展消息处理现场协议，它是一种基于XML的近端串流式即时通讯协议，于是乎，基于此协议的应用都具有非常强的可扩展性，在可预见的未来，XMPP协议会理所当然的同TCP/IP、HTTP等协议一起成为Internet标准。该协议的主要内容是通过定义了三个不同的角色——客户端、服务器、网关，构建出一种通信可以在这三者中任意两者传输的模式，通过该模式特点，可以得出基本的网络形式是客户端通过TCP/IP协议连接到服务器端，接着在上面进行XML传输[21]。该协议具有分布式、安全、可扩展、弹性佳、多样性的优点，拥有巨大生命力，该协议的结构示意图如下所示：

图 4-6 XMPP协议结构示意图

4.4 客户端各功能模块具体设计与实现

在本小节中，会给出客户端各功能模块的设计与具体实现，但由于本文篇幅限制，只会给出最核心的代码实现部分，而略去一些常规的、易理解的代码，同时还会适当指出一些可能会忽略的要点以及错误，有利于改善代码书写习惯。

4.4.1 用户登录注册功能设计与实现

注册：启动程序后，会有一段2秒左右的欢迎界面，接着会转入登录界面，在登录界面，每一个新用户通过输入正确的用户名和密码，点击注册按钮，便可以在环信服务器注册一个新用户，当注册成功时，界面还会出现一个Toast提示，提示成功创建新用户，若该用户名ID已被注册，则会提示错误信息，或者没有联网也会弹出错误信息。

登录：当用户输入正确的用户ID和与之匹配的密码后，点击登录按钮，即可实现用户的登录，如果登录成功，则会有提示说明登录成功并转入会话页面，若用户名为空或密码为空则登录失败，或者密码与所输入的ID不匹配也会登录失败，此时也会有相应的Toast提示登录失败，并附上登录失败的原因。

功能具体流程图如下所示：

图 4-7 用户注册(左)登录(右)功能操作流程

4.4.2 获取好友列表设计与实现

该模块主要负责获取用户的好友信息和展示用户好友列表，当用户登录成功后，用户可以通过点击主界面底部的联系人按钮进入好友列表界面，好友列表界面主要由一个listview和一个界面按键构成，而listview界面主要由环信SDK内置的XML文件实现，在此界面上，用户可以根据字典顺序查看当前登录用户的所有联系人信息，如果列表为空则表示当前用户没有添加过好友，此时可以通过点击界面左上角的扩展按键实现添加好友功能。在此页面下，用户可以通过点击任意一个好友用户即可进入互动聊天界面，上方的好友邀请用来展示接收到的好友邀请信息，点击可转入好友邀请信息界面。

获取好友列表这个功能是在程序启动时自动完成的，它不需要用户操作，主要包含了一个listview用以展示好友列表信息，实现的程序界面如下图所示：

图 4-8 用户好友列表界面

因为环信自带的SDK已经帮我们实现了这个复杂的listview界面，因此我们只需要在此基础上根据实际情况添加自己的布局文件，于是我们需要添加上方的好友邀请条目和一个用于提醒的红点，布局核心代码如下所示：
`//添加加号
titleBar.setRightImageResource(R.drawable.ease_blue_add); //添加头布局 View headerView = View.inflate(getActivity(),
R.layout.header_fragment_contact, null); listView.addHeaderView(headerView); //获取红点对象 iv_contact_red =
(ImageView) headerView.findViewById(R.id.iv_contact_red); //获取邀请信息条目的对象 ll_contact_invite = (LinearLayout)
headerView.findViewById(R.id.ll_contact_invite); //添加加号 titleBar.setRightImageResource(R.drawable.ease_blue_add);
//添加头布局 View headerView = View.inflate(getActivity(), R.layout.header_fragment_contact, null);
listView.addHeaderView(headerView); //获取红点对象 iv_contact_red = (ImageView)
headerView.findViewById(R.id.iv_contact_red); //获取邀请信息条目的对象 ll_contact_invite = (LinearLayout)
headerView.findViewById(R.id.ll_contact_invite);`

欲实现此功能，首先得在全局模型类Model中打开一个子线程，用于连接环信服务器，获取环信服务器上登录用户的所有好友信息，获取后要将数据转成Userinfo用户类，接着将其显示在主页面上的listview中，同时也要将好友保存到本地数据库上，方便其他功能的调用。

客户端代码具体实现如下所示：

```
//获取了所有的好友的环信ID List<String> hxsids =  
EMClient.getInstance().contactManager().getAllContactsFromServer(); //将获得的List对象转换为一个个userinfo对象 for  
(String hxid : hxsids){ UserInfo userInfo = new UserInfo(hxid); contacts.add(userInfo); //保存到数据库上  
Model.getInstance().getDbManager().getContactTableDao().saveContacts(contacts, true); //刷新页面 refreshContact(); //获  
取了所有的好友的环信ID List<String> hxsids = EMClient.getInstance().contactManager().getAllContactsFromServer(); //将获  
得的List对象转换为一个个userinfo对象 for (String hxid : hxsids){ UserInfo userInfo = new UserInfo(hxid);  
contacts.add(userInfo); //保存到数据库上Model.getInstance().getDbManager().getContactTableDao().saveContacts(contacts,  
true); //刷新页面 refreshContact();
```

其中对于刷新页面的方法refreshContact()的实现如下所示：

```
//获取数据 List<UserInfo> contacts = Model.getInstance().getDbManager().getContactTableDao().getContacts(); //设置  
数据 Map<String, EaseUser> contactsMap = new HashMap<>(); //转换数据 for (UserInfo contact : contacts){ EaseUser  
easeUser = new EaseUser(contact.getHxid()); contactsMap.put(contact.getHxid(), easeUser); //刷新页面 refresh(); //获取数  
据 List<UserInfo> contacts = Model.getInstance().getDbManager().getContactTableDao().getContacts(); //设置数据  
Map<String, EaseUser> contactsMap = new HashMap<>(); //转换数据 for (UserInfo contact : contacts){ EaseUser easeUser  
= new EaseUser(contact.getHxid()); contactsMap.put(contact.getHxid(), easeUser); //刷新页面 refresh();
```

4.4.3 添加好友功能设计与实现

用户可以通过好友列表界面右上角的扩展符号进入添加好友界面，接着便可以通过常见的传统添加好友操作，向目标用户发送一个添加好友邀请信息，当好友登录上线后便可以收到这个添加邀请信息，好友可以根据自己需要来处理邀请信息。

添加好友功能是整个即时通讯系统的基础功能之一，当用户需要添加好友时，客户端将需要添加的好友ID通过无线网络和有线网络发送到环信服务器端，环信服务器将收到的数据进行解析，并进入数据库查找欲添加的用户ID，更新欲添加用户的好友邀请人、原因等信息，然后向该用户发送一个邀请信息广播，该信息主要包含了好友邀请人、原因信息。

当欲添加的用户上线后，便会从服务端收到一个包含好友邀请人、原因的邀请信息广播，此时在该用户的好友邀请按键旁会出现一个红点，以示提醒，用户点击好友邀请按键后便会进入好友邀请信息列表界面，这里可以看到每个邀请信息的邀请人和原因的信息，用户可以选择接受邀请或者拒绝邀请，选择接受邀请则会发送一个广播[22]给环信服务器，环信服务器此时必须更新这两个用户数据库的信息，使两用户信息保持同步，本地邀请信息数据库也得更新，同时邀请信息列表也会进行刷新；而选择拒绝邀请同时也会发送一个广播给环信服务器，环信服务器此时会向原用户发送拒绝邀请信息广播，本地邀请信息数据库也会更新，将欲添加用户的邀请信息列表更新。

具体的操作流程图如下所示：

图 4-9 添加好友功能操作流程

根据以上对添加好友功能的描述，可以很容易地得出该模块的代码实现，下图给出添加好友的界面展示：

图 4-10 用户添加好友界面

实现此界面很简单，只需要一个输入栏、两个按钮、一张图片。界面实现的核心代码如下所示：

```
<EditText android:id="@+id/et_add_edit" android:layout_width="0dp" android:layout_height="match_parent"
android:layout_weight="10" /><Button android:id="@+id/bt_add_find" android:layout_width="35dp"
android:layout_height="match_parent" android:layout_weight="1" android:text="查找" /><EditText
android:id="@+id/et_add_edit" android:layout_width="0dp" android:layout_height="match_parent"
android:layout_weight="10" /><Button android:id="@+id/bt_add_find" android:layout_width="35dp"
android:layout_height="match_parent" android:layout_weight="1" android:text="查找" />
```

在客户端中，添加好友的逻辑操作应包括：

- 1、在联系人页面的扩展符号上实现一个跳转到添加好友页面的Intent；
- 2、在添加好友页面上实现查找功能；
- 3、实现添加按键功能；

根据此思路，添加好友的实现代码如下所示：

```
// 添加加号的点击事件 Intent intent = new Intent(getActivity(), AddContactActivity.class); startActivity(intent); // 添加加号
的点击事件 Intent intent = new Intent(getActivity(), AddContactActivity.class); startActivity(intent);
// 去服务器判断当前查找的用户是否存在，默认都存在 userInfo = new UserInfo(name); // 更新UI
rl_add.setVisibility(View.VISIBLE); tv_add_name.setText(userInfo.getName()); // 去服务器判断当前查找的用户是否存在，默
认都存在 userInfo = new UserInfo(name); // 更新UI rl_add.setVisibility(View.VISIBLE);
tv_add_name.setText(userInfo.getName());
```

```
// 环信服务器添加好友EMClient.getInstance().contactManager().addContact(userInfo.getName(), "添加好友");// 环信服
务器添加好友EMClient.getInstance().contactManager().addContact(userInfo.getName(), "添加好友");
```

除了实现添加好友界面外，还必须实现联系人页面的好友邀请条目操作，具体操作逻辑如下：

- 1、点击该条目进入邀请信息列表界面；
- 2、对收到的邀请信息进行处理；
- 3、更新联系人页面和邀请信息数据库。

根据此操作逻辑，功能实现的核心代码如下所示：

```
// 红点处理 iv_contact_red.setVisibility(View.GONE); SpUtils.getInstance().save(SpUtils.IS_NEW_INVITE, false); // 跳转
到邀请信息列表页面 Intent intent = new Intent(getActivity(), InviteActivity.class); startActivity(intent); // 红点处理
iv_contact_red.setVisibility(View.GONE); SpUtils.getInstance().save(SpUtils.IS_NEW_INVITE, false); // 跳转到邀请信息列表
页面 Intent intent = new Intent(getActivity(), InviteActivity.class); startActivity(intent);
// 接受邀请EMClient.getInstance().contactManager().acceptInvitation(invitationInfo.getUser().getHxid()); // 数据库更新
Model.getInstance().getDbManager().getInviteTableDao().updateInvitationStatus(invitationInfo.InvitationStatus.INVITE_ACC
EPT, invitationInfo.getUser().getHxid()); // 拒绝邀请
EMClient.getInstance().contactManager().declineInvitation(invitationInfo.getUser().getHxid()); // 数据库变化
Model.getInstance().getDbManager().getInviteTableDao().removeInvitation(invitationInfo.getUser().getHxid()); // 接受邀请
EMClient.getInstance().contactManager().acceptInvitation(invitationInfo.getUser().getHxid()); // 数据库更新
Model.getInstance().getDbManager().getInviteTableDao().updateInvitationStatus(invitationInfo.InvitationStatus.INVITE_ACC
EPT, invitationInfo.getUser().getHxid()); // 拒绝邀请
EMClient.getInstance().contactManager().declineInvitation(invitationInfo.getUser().getHxid()); // 数据库变化
Model.getInstance().getDbManager().getInviteTableDao().removeInvitation(invitationInfo.getUser().getHxid());
```

4.4.4 删除好友功能设计与实现

在好友列表界面，用户可以通过长按好友条目，打开删除好友目录，然后选择删除按钮即可实现删除好友功能。

要实现删除好友功能，首先得创建一个删除好友的布局，然后用户长按条目后使该布局可见，在用户点了删除按钮后，系统要向环信服务器发送一个删除好友的广播，然后环信服务器对收到的信息解析后，查找这两个用户的数据库，将彼此分

别从对方数据库的好友列表中删除，最后要实现更新好友列表的操作，此时也得更新本地好友列表的数据库，然后才能根据本地好友列表数据库实现用户好友列表的更新，自此，一个完整的删除好友功能才算彻底完成。

指 标
疑似剽窃文字表述
1. 系统的应用中声明了重要的信息，信息系统必须具备之前，它可以运行任何应用程序的代码。除其他事项外，清单中还执行了以下操作： 1 它声明了应用程序

6. 第4章系统设计与实现_第2部分	总字数：6295
相似文献列表 文字复制比：0%(0) 疑似剽窃观点：(0)	

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

由此不难看出该功能的逻辑操作应包括：

- 1、 创建一个删除好友提示布局；
- 2、 从本地数据库和环信服务器将用户与该好友的关系更新。

根据此操作逻辑，功能实现的核心代码如下所示：

```
//ContextMenuInfo接口有两个实现类，AdapterContextMenuInfo有position属性，强转获得EaseUser对象，获取环信idint position = ((AdapterView.AdapterContextMenuInfo) menuInfo).position;EaseUser easeUser = (EaseUser) listView.getItemAtPosition(position);//添加布局getActivity().getMenuInflater().inflate(R.menu.delete, menu);//ContextMenuInfo接口有两个实现类，AdapterContextMenuInfo有position属性，强转获得EaseUser对象，获取环信idint position = ((AdapterView.AdapterContextMenuInfo) menuInfo).position;EaseUser easeUser = (EaseUser) listView.getItemAtPosition(position);//添加布局getActivity().getMenuInflater().inflate(R.menu.delete, menu);
```

//从环信服务器删除联系人EMClient.getInstance().contactManager().deleteContact(mHxid);//本地数据库的更新Model.getInstance().getDbManager().getContactTableDao().deleteContactByHxid(mHxid);//从环信服务器删除联系人EMClient.getInstance().contactManager().deleteContact(mHxid);//本地数据库的更新Model.getInstance().getDbManager().getContactTableDao().deleteContactByHxid(mHxid);

4.4.5 用户聊天功能设计与实现

这是本系统的核心功能，无论用户是点击好友列表的任意一个好友，还是点击消息列表的消息条目，都会进入用户聊天界面，该页面主要由环信SDK提供，在处理逻辑上调用环信的监听函数，根据监听返回值刷新聊天记录，用户发送消息的步骤跟传统即时通讯软件的操作一样。用户在此聊天界面不仅仅可以发送文字，还能发送语音、图片、表情[23]，根据发送数据的类别，消息列表也会根据其数据类别调用不同的发送方法，进而在聊天中显示不同的样式。发送语音需要点击输入栏左边的麦克风按钮，此时会向用户获取麦克风权限，接着长按按钮并说话，然后松开即可发送语音[24]；而发送图片则需要点击输入栏右边的扩展功能按键，在弹出的目录中选择从相册或者照相机获取图片，此时也会向用户获取照相机权限和访问相册权限，选择图片后即可发送。聊天模块采用UDP协议作为文字语音信息的实时传输协议[25]，该软件系统还可以根据实际需求，在扩展栏上增加多种聊天功能。具体的操作流程如下图所示：

图 4-11 收发消息功能流程图

对于页面的布局问题，环信SDK已经为我们提供了该页面的设计以及相关操作，我们要做的就是根据自己的需求更改界面元素或更改功能逻辑。该界面布局主要由一个listview用以显示消息条目、一个输入框用以输入发送信息和三个按键用以分别实现表情、扩展功能、语音消息，消息编写完毕后按发送键即可实现发送，用户聊天界面如下图所示：

图 4-12 用户聊天界面

因为环信sdk已经帮我们做好一个很完善的页面，因此我们只需很简单地在ChatActivity上创建一个EaseChatFragment，接着向该EaseChatFragment设置一些参数，即可实现聊天界面以及聊天逻辑。具体代码实现如下所示：

```
//创建一个会话的fragmentEaseChatFragment easeChatFragment = new EaseChatFragment();mHxid = getIntent().getStringExtra(EaseConstant.EXTRA_USER_ID);//因为EaseChatFragment中的onActivityCreated方法需要fragmentArgs参数，所以必须得设置easeChatFragment.setArguments(getIntent().getExtras());//创建一个事务更改ChatActivity的页面FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();transaction.replace(R.id.fl_chat, easeChatFragment).commit();//创建一个会话的fragmentEaseChatFragment easeChatFragment = new EaseChatFragment();mHxid = getIntent().getStringExtra(EaseConstant.EXTRA_USER_ID);//因为EaseChatFragment中的onActivityCreated方法需要fragmentArgs参数，所以必须得设置easeChatFragment.setArguments(getIntent().getExtras());//创建一个事务更改ChatActivity的页面FragmentTransaction transaction =
```

```
getSupportFragmentManager().beginTransaction();transaction.replace(R.id.fl_chat, easeChatFragment).commit();
```

当然，由于Android系统版本的内容要求不同，Android 7.0以上版本不仅需要在AndroidManifest.xml上声明所用到的权限，并且还须在系统中动态获取权限，这就对使用环信语音聊天、发送图片等操作带来不便，因此我们在集成了环信SDK后还得根据客户的版本设计不同的权限获取机制。

4.4.6 用户注销功能设计与实现

当用户进入用户主要页面时，可以看到一个红色的Logout按键，当点击此按键，当前用户便会下线并转入用户登录注册界面，同时客户端也会发送一个广播给环信服务器，告诉该用户已经下线。如果用户在没有注销的前提下退出了程序，则下次打开程序时会默认登录上一个用户，而不会先跳转到用户登录注册界面再跳转到用户主页面，极大地节省了用户登录的时间。

在用户注销界面，注销按钮上要出现当前登录用户ID，方便用户知道注销的用户是谁，当用户点击此按钮后，系统程序会首先去环信服务器上更新用户状态，使其成为未登录状态，接着会在全局数据-模型管理类下获取数据库管理类，关闭数据库管理类，然后开启一个跳转到注册登录页面的activity，结束MainActivity。具体代码实现如下所示：

```
// 在button上显示当前用户名称 bt_setting_out.setText("退出登录 ( " + EMClient.getInstance().getCurrentUser() + " ) ");//
登录环信服务器退出登录 EMClient.getInstance().logout(false, new EMCallBack() { // 关闭DBHelper
Model.getInstance().getDbManager().close(); // 回到登录界面 Intent intent = new Intent(getActivity(), LoginActivity.class);
startActivity(intent);// 结束MainActivity getActivity().finish();// 在button上显示当前用户名称 bt_setting_out.setText("退出登录
( " + EMClient.getInstance().getCurrentUser() + " ) ");// 登录环信服务器退出登录 EMClient.getInstance().logout(false, new
EMCallBack() { // 关闭DBHelper Model.getInstance().getDbManager().close(); // 回到登录界面 Intent intent = new
Intent(getActivity(), LoginActivity.class); startActivity(intent);// 结束MainActivity getActivity().finish();}
```

4.4.7 后台消息功能设计与实现

当用户没有关闭系统而直接点击Home键时，系统将会在后台静默运行，此时若有新消息，则系统会伴随着提示音，在通知栏弹出一个对话框，当用户点击此对话框后便会跳转到消息列表界面，这个功能默认是开启的，用户无需任何操作，未来根据需要提供关闭后台消息提醒的功能。

为调用环信SDK的后台消息功能，我们需要根据环信内置的后台消息模块EaseNotifier，做出适合自己系统的修改。在实现用户消息列表页面的时候，通过创建一个EMMessageListener对象，并在环信服务器上注册，即可实现一个简单的后台消息提醒功能。具体代码实现如下所示：

```
private EMMessageListener emMessageListener = new EMMessageListener() { @Override public void
onMessageReceived(List<EMMessage> list) { //设置数据 EaseUI.getInstance().getNotifier().onNewMesg(list); //刷新页面
refresh();};//监听会话消息EMClient.getInstance().chatManager().addMessageListener(emMessageListener);private
EMMessageListener emMessageListener = new EMMessageListener() { @Override public void
onMessageReceived(List<EMMessage> list) { //设置数据 EaseUI.getInstance().getNotifier().onNewMesg(list); //刷新页面
refresh();};//监听会话消息EMClient.getInstance().chatManager().addMessageListener(emMessageListener);
```

4.4.8 数据库功能设计与实现

在本系统中，数据库是一个非常重要的部分，无论是添加好友、删除好友、好友邀请、注销用户等功能都需要对本地数据库进行操作，而因为数据库的变化不能体现在页面的变化上，所以常常会被忽视。客户端的数据库一般直接采用系统内嵌的轻量级数据库SQLite，对于数据库的功能设计，主要集中在用户账号数据库、邀请信息数据库、联系人数据库三个数据库上，各数据库的实现过程类似，先要创建一个Table类，用以保存各数据库的建表语句；接着要有一个各数据库的TableDao类，用以实现每个数据库的建立、更新操作；最后要建立一个DBManager类用以对三个数据库进行管理。具体实现代码如下所示：

```
//建表语句public static final String CREATE_TAB = "create table " + TAB_NAME + " (" + COL_HXID + " text primary
key," + COL_NAME + " text," + COL_NICK + " text," + COL_PHOTO + " text," + COL_IS_CONTACT + " integer);";// 获取数据库
连接 SQLiteDatabase db = mHelper.getReadableDatabase();// 执行查询语句 String sql = "select * from " +
ContactTable.TAB_NAME + " where " + ContactTable.COL_IS_CONTACT + "=1";//执行更新语句
db.update(InviteTable.TAB_NAME, values, InviteTable.COL_USER_HXID + "=?", new String[]{hxID});// 创建数据库表的语句
db.execSQL(UserAccountTable.CREATE_TAB);//建表语句public static final String CREATE_TAB = "create table " +
TAB_NAME + " (" + COL_HXID + " text primary key," + COL_NAME + " text," + COL_NICK + " text," + COL_PHOTO +
" text," + COL_IS_CONTACT + " integer);";// 获取数据库连接 SQLiteDatabase db = mHelper.getReadableDatabase();// 执行查
询语句 String sql = "select * from " + ContactTable.TAB_NAME + " where " + ContactTable.COL_IS_CONTACT + "=1";//执行
更新语句db.update(InviteTable.TAB_NAME, values, InviteTable.COL_USER_HXID + "=?", new String[]{hxID});// 创建数据库
表的语句 db.execSQL(UserAccountTable.CREATE_TAB);
```

4.5 本章小结

本章比较详细地介绍了聊天系统的各项功能的总体设计，包括服务器端的设计和客户端的设计，从用户注册、用户登录、聊天信息交互等不同模块，给出了客户端的主要功能设计流程图，为理解客户端的功能实现提供了方便。同时还给出了客户

7. 第5章系统测试		总字数：915
相似文献列表 文字复制比：4.3%(39) 疑似剽窃观点：(0)		
1	基于Qt的即时通讯软件设计 邓清沅 - 《大学生论文联合比对库》 - 2016-06-08	4.3% (39) 是否引证：否
原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容		

第5章系统测试

5.1 测试工具及环境

因为该应用是面向移动端，所以至少需要两台安卓手机的参与，因此选定的环境是1台安装了Android Studio的PC机以及两台安卓系统手机，PC机通过模拟器即可模拟出一台安卓手机，另外，安卓手机和PC端均要实现连接网络的操作后，才能开始测试程序。

5.2 测试流程及结果

将程序应用分别安装在虚拟机和手机上，并打开相应的网络设置，即可开始系统的设置，因部分功能对于用户是不可见的，所以该系统只测试重要的几个功能模块。

5.2.1 用户登录注册功能测试

测试界面如下所示：

图 5-1 用户登录注册测试图

测试内容：

- 1、 输入用户名密码。
- 2、 分别点击注册按钮和登录按钮。

测试结果：

1、 当第一次输入用户名密码时，点击登录键弹出错误信息，点击注册键弹出注册成功信息，再次点击注册则弹出错误信息。

2、 输入不正确的用户名密码时点击登录弹出错误信息。

3、 输入正确的用户名密码时登录成功并切换到会话列表界面。

测试结论：用户登录注册功能实现成功。

5.2.2 用户添加好友功能测试

测试界面如下所示：

图 5-2 用户添加好友测试图

测试内容：

- 1、 输入ID后点击查找。
- 2、 点击添加按钮。
- 3、 分别点击接受按钮和拒绝按钮。

测试结果：

- 1、 下方出现好友条目。
- 2、 出现提示信息并发送邀请。
- 3、 接受邀请和拒绝邀请提示出现，并发送广播。
- 4、 更新好友条目成功。

测试结论：用户添加好友功能实现成功。

5.2.3 好友聊天交互功能测试

测试界面如下所示：

图 5-3 好友聊天交互测试图

测试内容：

- 1、 点击好友条目进入聊天页面。
- 2、 输入想要发送的信息。
- 3、 点击发送按键。

测试结果：

- 1、 成功发送消息。
- 2、 成功接收消息。
- 3、 成功展示消息条目。

测试结论：好友聊天交互功能实现成功。

5.3 本章小结

经过PC机上的模拟测试和安卓真机的真实测试，可以得出该系统的各项功能均达到了预期的要求。本章给出了三个重点模块的测试过程，用简洁的文字和图表，使测试变得清晰明了。

8. 第6章总结与展望

总字数：1291

相似文献列表 文字复制比：32.3%(417) 疑似剽窃观点：(0)

1	54130804-康锴-软件工程-基于Flink的社区发现算法的设计与实现 康锴 - 《大学生论文联合比对库》 - 2017-06-01	32.3% (417) 是否引证：否
2	54130806_申浩杰_软件工程_基于智能算法的工业生产故障预测与诊断方法的研究 申浩杰 - 《大学生论文联合比对库》 - 2017-05-31	17.1% (221) 是否引证：否
3	基于遗传算法的卫星导航系统测距码设计 段峰(导师：田日才) - 《哈尔滨工业大学博士论文》 - 2013-07-01	2.2% (29) 是否引证：否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

第6章总结与展望

6.1 工作总结

本文首先从即时通讯的现状着手，分析了即时通讯的研究背景以及研究的可行性，给出了论文的组织结构。紧接着介绍了当前Android技术的发展现状、为什么选择的原因以及开发环境的选择介绍。然后在第三章中对系统的需求和系统结构设计进行了深刻的解读，从系统的物理结构和逻辑结构两方面给出了系统的概要设计，重点介绍了环信服务器与客户端的交互方式，提供了一个良好的视图，同时概括出了整个系统的大体业务流程，并重点对各个功能模块进行较详细的需求分析，为接下来的代码实现提供了坚实的基础。而在本文最核心的第四章中，不仅对服务器端做出了非常详尽的介绍，还根据环信服务器端的设计模式，阐明了客户端各模块的设计与实现，并且对于用户易犯的错误或者可能忽略的点，本文均一一指明了，还给出了部分用户-系统交互界面图。在最后的第五章，则给出了比较重要的三个模块的测试样例。

6.2 下一步工作

由于时间有限，本文设计的系统仍有许多可以改进完善的地方，下一步工作可以从以下几方面着手：

* 本文没有完成群组聊天功能的设计，同样这也可以通过调用环信SDK实现，但由于设计的过程较复杂，需要实现的页面、数据库操作、环信用户更新等功能稍复杂，在尝试群组集成的时候出现了一个错误，在有限的时间内没有解决，因此在下一步工作当中可以认真解决该错误，实现群组聊天的功能集成。

* 在用户设置界面只放置了一个用户注销按钮，在下一步工作中可以根据系统的需要增加更多的设置功能，如改变后台通知、勿扰模式、导出聊天记录、改变用户头像等功能，增加这些功能是改善用户体验最好的方式之一，未来可以参考一些优秀的开源项目，使系统得到进一步优化。

参考文献

- [1] 杜辽. 基于android系统的即时通讯系统设计与实现[D]. 山东大学, 2014.
- [2] 吴文思. 基于安卓的系统架构及程序开发探索[J]. 电子技术与软件工程, 2014(19):76.
- [3] 吴潇. Android与iOS胜算几何[N]. 通信产业报, 2016-08-29(011).
- [4] <https://developer.android.com/about/dashboards/>
- [5] 陈伟杰. Android的四大组件[J]. 电脑迷, 2018(03):161.
- [6] 郑振思. Android手机运动健康APP的设计与实现[D]. 华中科技大学, 2016.
- [7] Ammar H. Ali, Ali M. Sagheer. Design of an Android Application for Secure Chatting[J]. International Journal of Computer Network and Information Security(IJCNIS), 2017, 9(2).
- [8] 徐磊. 基于Android即时通讯系统的设计与实现[D]. 大连理工大学, 2016.
- [9] 马光远. 冷链物流网的android手机客户端系统设计与实现[D]. 山东大学, 2014.
- [10] 丁倩. 基于Android平台局域网即时通讯软件的设计与实现[D]. 西安电子科技大学, 2013.
- [11] 曾红李. 松耦合嵌入式软件系统设计与实现[J]. 计算机光盘软件与应用, 2012(24):168-169.
- [12] Xuetao Wei, Michael Wolf. A survey on HTTPS implementation by Android apps: Issues and countermeasures[J]. Applied Computing and Informatics, 2016
- [13] 林汝泽, 徐媛媛, 方凯, 熊松泉. 基于HTTP协议的Android手机数据同步实现[J]. 信息通信, 2013(01):96.
- [14] 彭凤凌, 虞先国, 王洪辉, 蒲建华. Android手机与内网PC通信系统的设计与现实[J]. 计算机工程与设计, 2013, 34(07):2333-2337.
- [15] 郭霖. 第一行代码[M]. 人民邮电出版社：郭霖，2014, 8.
- [16] 曹义成. 移动分布式自行车自助借还系统设计与实现[D]. 广东工业大学, 2015.

- [17] 于泽平. 基于PhoneGap和Sencha的移动Web应用开发研究[D]. 复旦大学, 2013.
- [18] 李肖强. 基于对象引用图的Android恶意软件检测系统设计与实现[D]. 哈尔滨工业大学, 2015.
- [19] 鲍美英, 申晋祥. Android中网络通信研究及聊天系统设计[J]. 山西大同大学学报(自然科学版), 2016, 32(06): 3-5+12.
- [20] 杨继承. 基于XMPP协议的Android即时消息APP的设计与实现[D]. 浙江工业大学, 2015.
- [21] 范乃如. 基于XMPP协议的即时通讯教育平台的研究[D]. 复旦大学, 2012.
- [22] 周雯. Android广播监听机制的两种实现方法[J]. 现代计算机, 2013(16): 48-49+69.
- [23] 陈岑. 基于Android移动应用的网络图片传输与存储的设计与实现[D]. 北京邮电大学, 2014.
- [24] 刘文琦. 基于Wi-Fi Direct语音传输的语音压缩算法的研究与实现[D]. 西安电子科技大学, 2014.
- [25] 吴忠明, 黄辰, 朱小丹. 基于局域网的手机即时通讯聊天软件的设计与实现[J]. 物联网技术, 2016, 6(02): 50-51+104.

致谢

光阴荏苒，眨眼间本科学生时代即将结束。回首过往的岁月，感慨万千。四年求学的道路不是一帆风顺的，经历过的成功是我的光荣和荣幸，经历过的失败和挫折对我而言是人生的经验与历练，二者都是成长道路上最美丽的风景。值此论文完成之际，我要衷心感谢这一路上帮助过我的老师、同学、朋友和家人。

首先，我要感谢我的父母，感谢父母养育了我，同时正是他们的高瞻远瞩，让我得以进入吉林大学这所双一流学校，如果当初选择错过了这所大学，也就遇不到这么优秀的老师和同学，自身实力也就不会像今天一样。

接着，我最要感谢我的毕业论文导师，刘桂霞教授，正是她的兢兢业业，才得以让我完成这个论文设计，得以体现出大学四年的知识成果。刘老师待人真诚，和蔼可亲，在毕业设计的过程中给予了我很多帮助。

我还要感谢我的室友，不管是学习上，还是日常生活中，他们都给我带来了莫大的帮助。我们在一起的这四年里，大家一直敞开心扉，无话不谈，情同手足。还有计算机学院的同学们，每次消息的通知，期末考前的资料共享，都和他们的勤恳努力分不开。谢谢在我有需要的时候总能帮我度过难关。

感谢吉林大学为我们这些莘莘学子创造的典雅清幽的学习环境，国际化的学术平台。感谢吉林大学软件学院为我们提供的大量实验硬件设备和自由开放的学习氛围。

最后，感谢各位评审专家给出的宝贵意见，以及在学术上的指导和帮助。

指 标

疑似剽窃文字表述

1. 致谢

光阴荏苒，眨眼间本科学生时代即将结束。回首过往的岁月，感慨万千。四年求学的道路不是一帆风顺的，经历过的成功是我的光荣和荣幸，经历过的失败和挫折对我而言是人生的经验与历练，二者都是成长道路上最美丽的风景。值此论文完成之际，我要衷心感谢这一路上帮助过我的老师、同学、朋友和家人。

首先，我要感谢

2. 不管是学习上，还是日常生活中，他们都给我带来了莫大的帮助。我们在一起的这四年里，大家一直敞开心扉，无话不谈，情同手足。还有计算机学院的同学们，每次消息的通知，期末考前的资料共享，都和他们的勤恳努力分不开。谢谢在我有需要的时候总能帮我度过难关。

感谢吉林大学为我们这些莘莘学子创造的典雅清幽的学习环境，国际化的学术平台。感谢吉林大学软件学院为我们提供的大量实验硬件设备和自由开放的学习氛围。

最后，感谢各位评审专家给出的宝贵意见，以及在学术上的指导和帮助。

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例

2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例

4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的

6.红色文字表示文字复制部分；绿色文字表示引用部分

7.本报告单仅对您所选择比对资源范围内检测结果负责



✉ amlc@cnki.net

🌐 <http://check.cnki.net/>

👤 <http://e.weibo.com/u/3194559873/>

<http://check.cnki.net>