

毕业设计（论文）检测系统  
文本复制检测报告单 (全文标明引文)

No: BC20230601185755362661517

检测时间: 2023-06-01 18:57:55

篇名: 基于Netty与Spring Boot的高性能即时聊天软件设计与实现

作者: 刘娟 (20192727)

指导教师: 邝祝芳 (教授)

检测机构: 中南林业科技大学

文件名: 6 正文 (刘娟).docx

检测系统: 毕业设计 (论文) 检测系统 (毕业设计 (论文) 管理系统)

检测类型: 毕业设计论文

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  
源代码库  
CNKI大成编客-原创作品库

时间范围: 1900-01-01至2023-06-01

## 检测结果

去除本人文献复制比: 7.8%

跨语言检测结果: -

去除引用文献复制比: 6.9%

总文字复制比: 7.8%

单篇最大文字复制比: 1.3% (基于Netty框架实现呼叫中心实时统计系统)

重复字数: [3328]

总段落数: [5]

总字数: [42442]

疑似段落数: [5]

单篇最大重复字数: [548]

前部重合字数: [402]

疑似段落最大重合字数: [1860]

后部重合字数: [2926]

疑似段落最小重合字数: [51]



文字复制部分 6.9%  
引用部分 0.9%  
无问题部分 92.2%

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

相似表格: 0

相似公式: 没有公式

疑似文字的图片: 0

10.2% (1048)	10.2% (1048)	基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第1部分 (总10286字)
18.3% (1860)	18.3% (1860)	基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第2部分 (总10157字)
2.9% (254)	2.9% (254)	基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第3部分 (总8891字)
1.2% (115)	1.2% (115)	基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第4部分 (总9643字)
1.5% (51)	1.5% (51)	基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第5部分 (总3465字)

(注释: ■ 无问题部分 ■ 文字复制部分 ■ 引用部分)

## 1. 基于Netty与Spring Boot的高性能即时聊天软件设计与实现\_第1部分

总字数: 10286

### 相似文献列表

去除本人文献复制比: 10.2%(1048) 去除引用文献复制比: 9.8%(1009) 文字复制比: 10.2%(1048) 疑似剽窃观点: (0)

1	基于SpringCloud的论坛系统的设计与实现 - 《大学生论文联合比对库》- 2019-04-27	2.7% (273) 是否引证: 否
2	基于Java-Web的酒店预订系统的设计与实现 XX刘子锋 - 《高职高专院校联合比对库》- 2019-06-24	2.0% (205) 是否引证: 否
3	基于Java-Web的酒店预订系统的设计与实现 XX刘子锋 - 《高职高专院校联合比对库》- 2019-06-28	2.0% (205) 是否引证: 否
4	基于Java-Web的酒店预订系统的设计与实现 XX刘子锋 - 《高职高专院校联合比对库》- 2019-06-11	2.0% (205) 是否引证: 否
5	基于SpringCloud的南工论坛系统的设计与实现 王侯程 - 《大学生论文联合比对库》- 2019-05-23	2.0% (205) 是否引证: 否
6	房屋中介系统的设计与实现 赖鑫 - 《大学生论文联合比对库》- 2019-05-24	2.0% (205) 是否引证: 否
7	基于Spring Cloud的南工论坛系统的设计与实现+任怡帆+1515925291+软件工程 任怡帆 - 《大学生论文联合比对库》- 2019-05-29	2.0% (205) 是否引证: 否
8	基于Java Web的酒店预订系统的设计与实现 刘子锋 - 《大学生论文联合比对库》- 2019-06-04	2.0% (205) 是否引证: 否
9	基于Java Web的酒店预订系统的设计与实现 刘子锋 - 《大学生论文联合比对库》- 2019-06-11	2.0% (205) 是否引证: 否
10	基于Java Web的酒店预订系统的设计与实现 刘子锋 - 《大学生论文联合比对库》- 2019-06-19	2.0% (205) 是否引证: 否
11	基于Java Web的酒店预订系统的设计与实现 刘子锋 - 《大学生论文联合比对库》- 2019-06-28	2.0% (205) 是否引证: 否
12	南方学院学生消费预警系统分析与设计 姚栩 - 《大学生论文联合比对库》- 2019-05-29	2.0% (205) 是否引证: 否
13	基于Java的养老服务系统的设计与实现 王冠霖 - 《大学生论文联合比对库》- 2019-05-31	1.9% (195) 是否引证: 否
14	13101316-吴元维-易鼎天成有限公司绩效考核系统 吴元维 - 《大学生论文联合比对库》- 2017-06-09	1.8% (183) 是否引证: 否
15	200523004273   11603990801_詹啟华_基于微信点餐系统的设计与实现 詹啟华 - 《大学生论文联合比对库》- 2020-05-25	1.7% (179) 是否引证: 否
16	基于Uni-app的移动集团专线售前支撑系统的设计与实现 刘森(导师: 董永生;陈萍) - 《河南科技大学硕士论文》- 2019-12-01	1.7% (179) 是否引证: 否
17	基于Spring MVC的智享车位平台 李海燕 - 《大学生论文联合比对库》- 2017-04-16	1.6% (165) 是否引证: 否
18	教学空间系统的设计与实现 王健 - 《大学生论文联合比对库》- 2018-06-04	1.4% (139) 是否引证: 否
19	7257066_陈振鑫_外卖系统的设计与实现 陈振鑫 - 《大学生论文联合比对库》- 2019-06-10	1.4% (139) 是否引证: 否
20	基于SSM的校园信息网 张弛 - 《大学生论文联合比对库》- 2020-05-17	1.2% (120) 是否引证: 否
21		1.1% (113)

	桂林特产销售系统的分析设计与实现		
	蒋林平 - 《大学生论文联合比对库》 - 2016-05-12	是否引证:	否
22	桂林特产销售系统的分析设计与实现	1.1% (113)	
	蒋林平 - 《大学生论文联合比对库》 - 2016-05-23	是否引证:	否
23	软件工程201401班刘鸿201401420154	1.1% (113)	
	- 《大学生论文联合比对库》 - 2018-05-22	是否引证:	否
24	软件工程201401班曾铁权201401420115	1.1% (113)	
	- 《大学生论文联合比对库》 - 2018-05-22	是否引证:	否
25	软件工程201401班曾铁权201401420115	1.1% (113)	
	曾铁拳 - 《大学生论文联合比对库》 - 2018-05-25	是否引证:	否
26	基于java的在线水果超市的设计与实现	1.1% (110)	
	王帆 - 《大学生论文联合比对库》 - 2019-05-20	是否引证:	否
27	Windows环境下的贪吃蛇游戏的设计与实现	1.0% (107)	
	刘晟志 - 《大学生论文联合比对库》 - 2015-05-16	是否引证:	否
28	基于Android客户端的智能家居App设计	0.6% (63)	
	凌敏;罗浩武;邹阿金;黄健明; - 《现代计算机》 - 2019-09-25	是否引证:	否
29	基于评论数据的文本关键词提取和可视化方法	0.6% (60)	
	林素红;杜玉红; - 《信息与电脑(理论版)》 - 2022-12-10	是否引证:	否
30	基于Hadoop集群的运维管理系统设计与实现	0.4% (39)	
	景科华(导师:李贺;张百军) - 《西安电子科技大学硕士论文》 - 2021-06-01	是否引证:	是
31	基于碎片化应用的微型学习研究	0.3% (33)	
	朱学伟;朱昱;徐小丽; - 《现代教育技术》 - 2011-12-15	是否引证:	否
32	中小企业局域网即时通信的设计与实现	0.3% (32)	
	张弛(导师:范录宏;邵虹) - 《电子科技大学硕士论文》 - 2015-03-01	是否引证:	否
33	家居信息网站的设计与实现	0.3% (29)	
	张家慧 - 《大学生论文联合比对库》 - 2017-04-17	是否引证:	否

原文内容

分类号: 密级:

学校代号: 10538 学号: 20192727

本科毕业(学士学位)设计

基于Netty与Spring Boot的高性能即时聊天软件设计与实现

学生姓名: 刘娟

指导教师: 邝祝芳教授

培养学院: 计算机与信息工程学院

年级专业: 19级计算机科学与技术

提交日期: 2023年6月1日

中南林业科技大学

毕业设计原创性声明

本人郑重声明: 所呈交的设计是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外, 本设计不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体, 均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

学生签名: 刘娟日期: 2023 年 6 月 1 日

毕业设计版权使用授权书

本毕业设计作者完全了解学校有关保留、使用设计的规定, 同意学校保留并向国家有关部门或机构送交设计的复印件和电子版, 允许设计被查阅和借阅。本人授权中南林业科技大学可以将本设计的全部或部分内容编入有关数据库进行检索, 可以采用影印、缩印或扫描等复制手段保存和汇编本设计。

本设计属于

1、保密, 在年解密后适用本授权书。

2、不保密√。

(请在以上相应方框内打“√”)

学生签名: 刘娟日期: 2023 年 6 月 1 日

导师签名: 日期: 年月日

# 基于Netty与Spring Boot的高性能即时聊天软件设计与实现

## 摘要

随着互联网高速发展，如今的网络通讯系统逐渐完备，目前市场上开发一款即时聊天软件的技术框架趋于成熟稳定。一款好的即时聊天软件不仅是在功能方面的多样化更是在于性能方面的快捷且高效、安全且稳定。本文基于Netty与Spring Boot框架实现一个高性能的即时聊天软件。主要是利用Java语言的Spring Boot框架搭建后台服务，结合Netty框架实现用户之间消息的快速发送，达到一个即时聊天的效果，主要工作包括如下：

1. 基于vue3与typeScript实现一个聊天软件的PC客户端界面。在客户端中底层利用WebSocket来实现网络通讯的连接以及用户之间消息的传递。
2. 基于vue搭建一个即时聊天软件的后台管理系统，并做出用户权限限制处理，结合后端Netty框架的心跳机制实现对网络连接的实时监控以及聊天软件实时数据的展示
3. 利用Spring Boot框架来搭建后台管理以及客户端后端服务，结合Netty框架、Zookeeper以及RabbitMQ利用TCP协议完成消息的发送，提高软件的高并发能力，通过自定义消息格式解决消息传输的半包、粘包问题。
4. 实现单聊与群聊，消息形式包括文字、表情、图片、文件，同时单聊支持音视频通话功能，支持离线、在线消息的存储与传输。

关键词：即时聊天软件;Spring Boot;Netty;WebSocket;

Design and Implementation of High Performance Instant Chat Software Based on Netty and [Spring Boot](#)

## Abstract

With the rapid development of the Internet, today's network communication systems are gradually becoming complete, and the technical framework for developing an instant messaging software on the market is becoming mature and stable. A good instant messaging software not only has diverse functions, but also fast, efficient, secure, and stable performance. This article implements a high-performance instant messaging software based on Netty and the Spring Boot framework. The main task is to use the Spring Boot framework of Java language to build backend services, and combine it with the Netty framework to quickly send messages between users, achieving an instant chat effect. The main tasks include:

1. Implement a PC client interface for chat software based on Vue3 and typeScript. Utilize WebSocket at the bottom of the client to achieve network communication connections and message transmission between users.
2. Build a backend management system for instant messaging software based on Vue, handle user permission restrictions, and combine the heartbeat mechanism of the backend Netty framework to achieve real-time monitoring of network connections and display of real-time data in the chat software
3. Use the Spring Boot framework to build backend management and client backend services, and use the Netty framework, Zookeeper, and RabbitMQ to send messages using the TCP protocol, improving the software's high concurrency ability. By customizing message formats, solve the problem of half packet and sticky message transmission.
4. Implement single chat and group chat, with message formats including text, emoticons, images, and files. At the same time, single chat supports audio and video call functions, and supports the storage and transmission of offline and online messages.

Keywords: Instant messaging software; Spring Boot; Netty; WebSocket;

## 目录

1 绪论.....	1
1. 1 研究背景.....	1
1. 2 国内外现状及趋势.....	2
1. 3 研究意义.....	3
1. 4 研究特点.....	3
2 相关技术及工具.....	5
2. 1 开发与运行环境.....	5
2. 2 关键技术.....	5
3 需求分析.....	14
3. 1 可行性分析.....	14
3. 2 功能需求.....	15
3. 3 非功能需求.....	20
3. 4 数据需求.....	21
4 系统设计.....	30
4. 1 功能结构.....	30
4. 2 功能流程.....	39
4. 3 数据库设计.....	41
4. 4 子系统设计.....	48
5 详细设计.....	51
5. 1 系统界面.....	51
5. 2 具体技术实现方法.....	62
6 系统测试.....	76



6. 1 测试环境以及测试条件.....	76
6. 2 模块功能测试.....	76
6. 3 测试总结.....	81
结论.....	82
参考文献.....	82
致谢.....	84

## 1 绪论

### 1. 1 研究背景

#### 1. 1. 1 研究依据

随着移动互联网的不断发展，移动通讯从当初的3G到后来的4G，甚至是如今的5G，人们的通讯方式不再局限于距离、设备难以携带等因素，从当初的书信、报纸、电视、广播等途径发生了巨大的进步。互联网的到来将世界连成一个整体，如今人与人之间的通讯在移动互联网的加持下可以做到远距离高效的文字交流、邮件来往、实时对话，甚至是面对面视频聊天。随着5G时代到来，互联网的进一步发展，人们的通讯环境更加稳健快速，通讯方式也更加多元化。其中像QQ、微信、腾讯会议等通讯软件都是互联网技术下发展的产物，与其他传统媒体相比，此类工具有着资讯容量大、即时变化强、使用简便、成本低的优势。

通讯交流作为人们日常生活中的一项基本需求，在逐步发展的科技力量下对于通讯软件的要求不止在于方式的快捷，更加着眼于通讯软件的性能、功能、安全以及用户体验的方面的加强。一个高效的聊天软件，不仅节约了时间、资金还提高了网络利用率。本文主要是由此启发采用Netty与Spring Boot框架，结合Mybatis、Redis、MySQL以及SpringCloud相关第三方组件实现一款较高性能的即时聊天软件。旨在利用Java软件编程技术，动手实现整个设计，了解通讯软件的实现原理，同时对我国目前流行的Java开发方面相关技术的学习与掌握，结合当下的互联网通讯软件反思不足，提出一定的优化措施。

#### 1. 1. 2 研究目的

本文关键在于Java后端方面的实现，利用Netty以及Spring Boot框架实现一款高性能、高可靠性的即时聊天软件，同时通过加入微服务等第三方相关技术为整个软件设计提高并发能力，满足人们对数据实时准确且可靠的需求。

目前，基于Internet的通讯工具已经足够完善了，但是为了更加满足用户便利的同时增加软件的受众，从而提高软件吸引力，需要不断扩展更加智能化的功能。在互联网这样的大环境下，即时聊天软件对于购物、游戏、社交、线上交易等第三方服务集成越多，用户的使用环境以及信息安全越不容易得到保障，而网络信息安全一直以来都是互联网环境下人们关心的重要因素。所以，给即时聊天软件做减法，开发一款只注重即时聊天功能的软件，或者将软件受众范围缩小，例如小区、公司、校园，甚至是楼层，这样在一定的程度上减小了远距离传输数据丢失的风险、网络延迟等因素的影响，从而达到更加快速高效且安全的效果。不仅如此，即时聊天软件的用户更具针对性，功能细节的实现更加简易，用户群体较稳定，需求量自然也是可观的。本文所设计的高性能即时聊天软件关键在于对消息数据进行即时发送的同时满足高并发情况下软件的可用性并保证消息的可靠性。整个软件的设计还涉及后台管理系统，从用户角度以及开发者的角度去审视一款聊天软件的优劣，在消息发送形式方面做出一定优化效果，为用户提供简约且好用的体验感。不仅如此，随着互联网的快速发展，即时聊天软件已经成为现代人办公、娱乐、学习必不可少的工具，它与www、email共同成为互联网的主流，作为一名计算机的学生，学习和掌握一定的网络编程技术很有必要。本研究能锻炼开发者的编程设计能力，扎实Java专业知识，学以致用解决开发过程中的难点以及性能的优化问题，有所收获。

### 1. 2 国内外现状及趋势

追溯即时聊天软件的发展，早在20世纪60年代，美国的大学中已经诞生了一个电脑辅助教学系统——柏拉图系统。当时的聊天建立在用户双方在线且使用终端进行通讯的情况下，双方之间发送的内容都会在终端中显示，包括对方对每个字符的编辑、删除以及修改的过程，这是真正意义上的“即时通讯”。随着计算机操作系统的发展，产生了基于UNIX/Linux的即时通讯工具，但它只能在局域网环境下才能进行通讯。1996年，三个以色列人维斯格、瓦迪和高德芬格开发了世界上第一个即时通讯软件ICQ，意为“我在找你”——“I Seek You”[1]，把即时通讯带入了互联网大舞台。但是在当时即时通讯软件的发展热潮下，由于每个通讯软件之间使用的协议不一样，导致彼此之间无法互通，需要用户同时运行多个即时聊天软件或者使用支持多协议的终端软件才能够实现聊天，如Gaim、Trillian、Jabber等。

ICQ诞生后不久，中国的即时通讯软件开始蓬勃发展。当时马化腾创立了深圳市腾讯计算机系统有限公司，模仿ICQ，在ICQ前加上字母“Q”，即“Opening I Seek You”，意为开放的ICQ。后于2000年，由于版权侵权丑闻，OICQ更名为QQ。随着QQ在中国的巨大浪潮，现代出现了大量的即时通讯软件，不断创新，功能与性能逐渐趋于完善，社交、娱乐相结合，形成了一个以即时通讯为基础的庞大的产业链。随着后来智能手机以及互联网的快速发展，各个国家都诞生了具有代表性的通讯软件，例如美国的WhatsApp、日本的LINE、韩国的Kakao Talk以及中国的WeChat。现代的即时聊天软件在大数据以及人工智能的影响下演进成可以根据用户数据分析用户喜好的社交软件或者是镶嵌在某些电商平台提供客服服务以及微信、支付宝这类支持线上交易支付功能的产品。不仅如此，即时聊天已经成为目前市面上大部分主流软件不可或缺的一项功能。相对于近代即时通讯而言，现代即时聊天软件细化了产品使用领域，不仅有基于个人使用的即时通讯工具，还有面向企业的即时通讯工具。企业通讯软件更强调信息安全、功能实用、系统稳定与可扩展能力，有的企业甚至实现自研，量身定制，不仅便捷了组织管理还极大提高了企业内部的沟通效率。

在即时通讯发展的五十年以来，即时聊天软件已经成为人们生活中一种必备的工具。就目前来看，国内外的即时通讯系统基本趋于完善，整个格局也基本划定。我国目前的即时通讯系统在国际上也占有一定的主导地位，基于各种开发语言和技术实现的即时聊天软件层出不穷。现发展阶段，各国的即时通讯聊天软件都大同小异，在这个智能信息化的5G时代背景下，随着人工智能的逐步成熟，如今的发展趋势主要是从用户使用的便捷程度、功能领域细化、用户群体的针对性、通信方式的多样化以及产品的智能安全性等角度优化。

在整体局势的把控上，我国目前即时通讯云服务是[发展的一个热门方向\[2\]](#)。总结中国即时通讯的发展趋势，各开发商将继续面对客户需求展开更加惨烈的争夺，而竞争的焦点也从对个人用户的竞争偏向于对公司客户的竞争。腾讯在中国个人用户领域的领头作用短期内将很难被撼动。未来网络领域的主要竞争方式将是信息服务与增值业务之间的比拼，而三网融合与互联网的全

面开放将是未来的发展趋势,网络公司要做的是随着网络领域的迅速变革以及时满足需要,把业务做细做专,同时还要懂得扬长避短,也需要同时依靠运营商的实力,采用“竞合”的经营模式共同做大业务。

### 1. 3 研究意义

本文的设计与实现主要从理论与现实这两个方面的意义出发。

理论意义方面:据了解,即使是像QQ、微信这样被如此广泛使用的成功软件案例,也还是会存在有遗漏消息或者消息传输延时的情况,尤其是多平台消息互通的情况下,信息的安全更是重中之重。Netty作为一个Java的框架,底层使用 NIO技术的异步时间驱动的开源框架,使用简单,具有良好的稳定性、安全可靠、功能强大且易扩展,用它来实现一个Java网络编程的实时系统是十分可取的。同时使用Spring Boot框架开发“约定优于配置”,可实现快速整合第三方技术,对于整个软件其他技术的整合与运用提供了很大的便利,简化了配置代码量,更加注重业务代码的实现,有利于提高开发效率。

现实意义方面:即时聊天软件目前市面上的运用也是十分广泛的。就个体而言,能够有效地保证个人信息的安全性,对于企业而言,在公司规模很大的情况下,便于员工之间的进行有效即时的沟通,同时企业与客户之间的对话也可以借助这样一款软件实现快捷高效安全的交流。例如现在很多APP或者小程序提供的客服服务,利用这样一个即时聊天的工具可以实现与用户的稳定高效的交流,给用户提供帮助、为用户解决问题,提升用户对于整个产品的满意度。

综上所述,开发一款具有较高性能的即时聊天软件有利于把握互联网通讯技术的发展趋势,具有一定的研究和学习意义。

### 1. 4 研究特点

聊天软件又称为即时通讯软件,通过即时通讯技术来实现在线聊天、交流的软件。目前对于聊天软件的设计主要有两种架构形式[3]。

1. C/S架构:客户端/服务端形式,用户使用需要安装特定的客户端软件[4],这类形式的产品主要有:QQ、微信、Skype、新浪UC、百度HI、钉钉、MSN等。

2. B/S架构:浏览器/服务端形式,采用这种架构用户无需下载客户端,借助互联网媒介就可以实现在线聊天,代表产品主要有:53KF、Website Live、live800等。

本文设计的即时聊天软件采用的是B/S架构,从高性能且即时的角度出发,相对注重软件并发能力以及数据传递可靠性保证,主要具有以下特点:

1. 简化聊天环境:主要包括单聊、群聊,消息传输形式为文字、表情、图片、文件,结合WebRTC与WebSocket技术实现支持单聊语音及视频聊天。不涉及过多个人信息的设置,简化聊天环境,更好的保护个人隐私。

2. 后台系统易扩展:支持多端登录的同时实现了数据同步的功能,设计基于Netty框架搭建的即时通讯系统(Instant Messaging System),通过TypeScript编写代码封装对接后端IM系统作为SDK,实现任意软件客户端与服务之间只需通过调用SDK就可以完成聊天系统的接入。

3. 高并发能力较高:Spring Boot整合Zookeeper采用一致性Hash的负载均衡策略实现网络通讯连接IP以及端口的分配,整合RabbitMQ利用多线程处理保证高并发情形下软件可用性。

4. 采用TCP通讯协议以及保证消息可靠:通过Redis以及MySQL来保证消息的持久化,设计存储sequence字段记录多端之间数据同步的同时结合RabbitMQ保证消息传递的有序性,采用私有协议封装传输数据防止Netty的半包、粘包问题,采用消息确认响应发送方式避免同一条消息的重复发送。

5. 利用Netty心跳机制,实现自动重连,超时断连等功能,避免造成服务资源的浪费。

### 2 相关技术及工具

#### 2. 1 开发与运行环境

1. 开发环境:利用Java语言采用jdk1.8版本进行开发;

2. 开发工具:采用IDEA 2021进行服务端开发,利用VSCode进行客户端及后台前端页面开发;

3. 数据库:MySQL Server 8.0 + redis-5.0.14;

4. Spring Cloud三方组件:zookeeper-3.4.8 + RabbitMQ-server-3.11.9;

5. 服务器:

(1) CPU:要求1GHz以上,推荐2GHz;

(2) 内存:1G以上,推荐2G;

(3) 硬盘:要求250G,推荐500G;

(4) 显存:推荐1G或以上;

6. 操作系统:推荐Windows 10或以上版本。

7. 浏览器:Google Chrome 14、Microsoft Edge 10等可支持WebSocket的浏览器。

#### 2. 2 关键技术

##### 2.2.1 Java语言

Java是一种通用的、基于类的、面向对象的编程语言。Java的底层是基于C++实现的,利用Java语言编写的程序不受平台限制,具有一次编译到处运行的特点[5]。Java是快速、安全和可靠的。Java语言主要具有以下几种特点:

1. 面向对象:面向对象编程主要包含封装、继承、多态、抽象这四个概念。

2. 与平台无关且可移植:Java提供了在不同平台下的解释运行环境,Java代码无需经过修改即可在另一台机器上运行。

3. 简单:不具有指针,运算符重载,多重继承,显式内存分配等复杂功能。

4. 健壮:Java语言吸收了C/C++语言的优点,健壮可靠。Java编译器能够尽早检查出程序中的错误,同时Java具有垃圾回收机制,能进行异常处理和内存分配。

5. 具有较高的安全性:自动回收垃圾,强制类型检查,取消指针。没有指针则无法进行数组的越界访问,一旦越界就会出现ArrayIndexOutOfBoundsException报错。所以在Java中无法进行堆栈破坏或缓冲区溢出这类安全攻击。

6. 多线程:Java允许并发执行程序的两个或更多部分,以最大程度地利用CPU。

7. 开源:Java语言封装的代码是对外开放的。

在Java 开发中有JDK、JRE以及JVM这三个重要概念和工具。



1. JDK (Java Development Kit) : Java的开发工具包。JDK是提供给Java开发人员使用的, 包含Java的开发工具和JRE。其中的开发工具包括: 编译工具 (javac.exe) 打包工具 (jar.exe) 等。JDK中包含javac指令能将. java文件编译成.class文件。

2. JRE (Java Runtime Environment) : Java运行环境。JRE包括Java虚拟机和Java程序所需的核心类库等, 能够运行已经开发好的Java程序。

3. JVM (Java Virtual Machine) : Java虚拟机。它只认识.class为后缀的文件并将.class文件中的字节码指令进行识别并调用操作系统向上的API完成动作。JVM是java能够跨平台的核心机制。

其中JDK, JRE和JVM三者之间的联系, 如图2-1所示。

JDK = JRE + 开发工具集 (例如Javac编译工具等)

JRE = JVM + Java SE 标准类库

图2-1 jdk、jre、jvm关系图

总的来说就是使用JDK API完成Java程序开发, 通过JDK编译指令将Java文件编译成字节码文件并交给JRE去运行, JVM解析字节码文件, 将它们映射到CPU指令集或操作系统调用, 实现跨平台。

Java语言JDK版本应用主要有三大平台体系:

1. Java SE (Java Platform Standard Edition) 。以前称为J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。

2. Java EE (Java Platform Enterprise Edition) 。以前称为J2EE。开发企业级的应用

3. Java ME (Java Platform Micro Edition) 。以前称为J2ME。Java ME 为在移动设备和嵌入式设备 (比如手机、PDA、电视机顶盒和打印机) 上运行的应用程序提供一个健壮且灵活的环境。

本文作为一款应用软件的实现主要就是基于Java SE版本下完成开发。

### 2.2.2 Spring Boot框架

说起Spring Boot框架, 首先需要对Spring框架有一定的了解。

Spring框架是一款轻量级的Java开发框架, 对外开源。Spring框架提供了IOC容器、控制反转、依赖注入以及一些其他模块, 具有简化代码量、便捷开发, 节省时间, 提高效率等优点。Spring可以管理Web层、业务层、Dao层以及持久层的组件, 配置各种Bean, 目的是为了了解企业级应用开发的业务逻辑层和其他各层的耦合问题[6]。Spring能够解决开发中的基础性问题, 使开发者更加专注于应用程序的开发。

虽然Spring框架提供了全面的基础框架, 但Spring框架的配置较为复杂, 每使用一项第三方jar包就需要进行相关的XML文件配置, 降低了编程效率。同时编写基于SSM框架的项目需要配置Tomcat, 需要把封装好的war包放到Tomcat容器运行。

基于Spring的复杂的配置问题, Spring Boot诞生。Spring Boot是一个微服务框架, 延续了Spring框架的核心思想IOC和AOP, 而且更加简单, 功能更加丰富, 性能更加稳定而健壮。

根据官方介绍, Spring Boot是所有基于Spring开发的项目的起点[7]。Spring Boot的设计是为了尽快的运行Spring 应用程序并且尽可能减少配置文件[8]。Spring Boot框架遵循“约定优于配置”的原则, 即约定采用默认配置, 当配置与约定不一致时, 开发者可以对约定进行自定义替换配置。这样直接简化了Java项目的启动以及实现第三方技术的快速整合。

Spring Boot框架具有以下4大特征:

1. Spring Boot Starter: 对常用的依赖分组进行整合到一个依赖中, 能够一次性对常用依赖添加到项目的Maven或Gradle构建中[9]。

2. 使编码变得简单, Spring Boot采用Java Config的方式对Spring进行配置, 并且提供了大量的注解, 极大的提高了工作效率。

3. 自动配置: Spring Boot的自动配置特性利用了Spring对条件化配置的支持, 合理地推测应用所需的bean并自动化配置他们[10]。

4. 使部署变得简单, Spring Boot内置了三种Servlet容器, Tomcat, Jetty, undertow。只要有Java的运行环境就可以启动Spring Boot项目, Spring Boot的项目可以打包成一个jar文件。

总体而言, Spring Boot是Spring的完善和扩展, 为便捷开发, 方便测试和部署, 提高效率而诞生的框架技术。目前Spring Boot作为Java开发的一款主流框架, 相对Spring而言是更好的选择。

### 2.2.3 TCP协议

在介绍Netty框架之前, 先了解什么是TCP协议。

TCP协议全称传输控制协议 (Transmission Control Protocol), 是一种面向连接的、可靠的、基于字节流的传输层通信协议[11], 必须对数据的传输进行控制。TCP协议的优点主要在于对可靠性的保证。TCP协议的可靠通信的实现主要通过两种方式来保证:

1. 在传输无差错的情况下, 采取ACK 回复

在每收到一个正确的、符合次序的片段之后, 就向发送方 (也就是连接的另一段) 发送一个特殊的TCP片段, 用来知会 (ACK, acknowledge) 发送方已收到该数据片段, 这个特殊的 TCP片段叫做ACK回复。如果一个片段序号为L, 对应ACK 回复有回复号L+1, 也就是接收方期待接收的下一个发送片段的序号。

2. 当传输出现差错时, 进行超时重传

如果发送方在一定时间等待之后, 还是没有收到ACK回复, 那么它推断之前发送的片段一定发生了异常。

指 标

疑似剽窃文字表述

### 1. Spring Boot

## Abstract

With the rapid development of the Internet,

- 以前称为J2SE。它允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。
- 编码变得简单, Spring Boot采用Java Config的方式对Spring进行配置, 并且提供了大量的注解, 极大的提高了工作效率。
- Java的运行环境就可以启动Spring Boot项目, Spring Boot的项目可以打包成一个jar文件。

## 2. 基于Netty与Spring Boot的高性能即时聊天软件设计与实现\_第2部分

总字数: 10157

### 相似文献列表

去除本人文献复制比: 18.3%(1860) 去除引用文献复制比: 14.6%(1488) 文字复制比: 18.3%(1860) 疑似剽窃观点: (0)

1	<u>基于Netty框架实现呼叫中心实时统计系统</u> 潘馨(导师: 刘惠) - 《西安电子科技大学硕士论文》 - 2014-12-01	5.4% (548) 是否引证: 否
2	<u>源代码( package com.uplooking.netty; import io.netty.b... )</u> - 《源代码库 ( <a href="https://raw.githubusercontent.com">https://raw.githubusercontent.com</a> ) 》 - 2020	4.1% (415) 是否引证: 否
3	<u>聊天系统的设计与实现</u> 张龙(导师: 李凤岐) - 《大连理工大学硕士论文》 - 2015-02-20	3.6% (367) 是否引证: 是
4	<u>支持面部再造的视频聊天系统的设计与实现</u> 卢昱昊 - 《大学生论文联合比对库》 - 2020-05-28	3.5% (358) 是否引证: 否
5	<u>仿微信单页应用的设计与实现.</u> 成佳红 - 《大学生论文联合比对库》 - 2019-04-30	1.8% (179) 是否引证: 否
6	<u>52354_孟庆喜_山西大学互联网+众创平台即时通讯模块开发</u> 孟庆喜 - 《大学生论文联合比对库》 - 2019-04-12	1.4% (144) 是否引证: 否
7	<u>基于Web的室内甲醛浓度远程实时监测系统</u> 窦猛 - 《大学生论文联合比对库》 - 2019-05-26	1.3% (134) 是否引证: 否
8	<u>基于智慧校园下的食堂消费系统的设计</u> 张群; - 《工业控制计算机》 - 2020-12-24	1.3% (133) 是否引证: 否
9	<u>基于微信小程序的校园二手书交易系统的实现</u> 陈泽集 - 《大学生论文联合比对库》 - 2018-05-11	1.3% (133) 是否引证: 否
10	<u>凌纬鸿_软件开发_5720153025_基于web的网上聊天系统</u> 软件开发 - 《大学生论文联合比对库》 - 2019-05-24	1.3% (130) 是否引证: 否
11	<u>WebSocket技术在医院智能语音输入中的应用</u> 周莉; - 《电子世界》 - 2020-06-30	1.3% (129) 是否引证: 否
12	<u>2013578441_杨智洪</u> 杨智洪 - 《大学生论文联合比对库》 - 2017-05-24	1.2% (117) 是否引证: 否
13	<u>基于容器集群的云开发环境研究与实现</u> 由伟希 - 《大学生论文联合比对库》 - 2017-05-22	1.1% (116) 是否引证: 否
14	<u>Z09415105-戴逸辉-殷旭东</u> 戴逸辉 - 《大学生论文联合比对库》 - 2019-06-03	1.1% (116) 是否引证: 否
15	<u>煤矿安全监控分站模拟软件设计</u> 邢少妹; 刘海峰; - 《工矿自动化》 - 2019-12-12 1	1.1% (115) 是否引证: 否
16	<u>基于WebSocket协议的特征匹配聊天系统</u> 潘哲; 邓芙蓉; 韩明正; 刘彩营; - 《科技经济导刊》 - 2020-05-25	1.1% (115) 是否引证: 否
17	<u>W201607198_何凡_基于WebSocket的安全即时通讯系统的设计与实现</u> 何凡 - 《高职高专院校联合比对库》 - 2019-05-22	1.1% (115) 是否引证: 否
18	<u>基于元编程的接口测试与文档自动化生成系统设计与实现</u> 邵子奇 - 《大学生论文联合比对库》 - 2017-03-22	1.1% (115) 是否引证: 否
		1.1% (115)



19	131003610117-巩文华-基于WebRTC的多媒体聊天室 巩文华 - 《大学生论文联合比对库》 - 2017-05-24	是否引证: 否
20	基于元编程的接口测试与文档自动化生成系统设计与实现 邵子奇 - 《大学生论文联合比对库》 - 2017-05-26	1.1% (115) 是否引证: 否
21	支持文件动态更新的密文搜索算法 黄浪 - 《大学生论文联合比对库》 - 2018-05-27	1.1% (115) 是否引证: 否
22	编程技术社区的设计与实现 戴逸辉 - 《大学生论文联合比对库》 - 2019-04-26	1.1% (115) 是否引证: 否
23	众云聊天室 许晨 - 《大学生论文联合比对库》 - 2019-04-29	1.1% (115) 是否引证: 否
24	飞剑即时通讯软件项目的设计与实现 陈相如 - 《大学生论文联合比对库》 - 2019-05-26	1.1% (115) 是否引证: 否
25	毕设-查重 赵坤 - 《大学生论文联合比对库》 - 2019-05-29	1.1% (115) 是否引证: 否
26	赵坤-毕设-V5.0.1 赵坤 - 《大学生论文联合比对库》 - 2019-05-29	1.1% (115) 是否引证: 否
27	飞剑即时通讯软件项目的设计与实现 陈相如 - 《大学生论文联合比对库》 - 2019-06-05	1.1% (115) 是否引证: 否
28	20152480203_陈相如_信息工程学院_飞剑即时通讯软件的设计与实现 陈相如 - 《大学生论文联合比对库》 - 2019-06-13	1.1% (115) 是否引证: 否
29	人力资源管理系统设计与实现论文 廖景豪 - 《大学生论文联合比对库》 - 2019-04-24	1.1% (115) 是否引证: 否
30	高并发下基于Actor模型抽奖系统设计与实现 泮关森 - 《大学生论文联合比对库》 - 2017-03-22	0.8% (84) 是否引证: 否
31	即时通讯系统的设计与研究 王明高 - 《大学生论文联合比对库》 - 2018-05-21	0.8% (82) 是否引证: 否
32	第38讲 对比Java标准NIO类库, 你知道Netty是如何实现更高性能的吗? - 道客巴巴 - 《互联网文档资源 ( <a href="http://www.doc88.com">http://www.doc88.com</a> )》 - 2019	0.7% (74) 是否引证: 否
33	基于Android的社交APP的设计 陈嘉琪 - 《大学生论文联合比对库》 - 2017-05-24	0.6% (65) 是否引证: 否
34	210517014640   11703080423_陈浩东_农产品交易平台的开发 陈浩东 - 《大学生论文联合比对库》 - 2021-05-17	0.6% (57) 是否引证: 否
35	16044728_张凯_基于Web的理财管理系统的设计与实现 张凯 - 《大学生论文联合比对库》 - 2020-05-25	0.3% (30) 是否引证: 否
原文内容		

发送方会重新发送之前发送过的分组, 直到收到该片段对应的ACK回复[12]。超时重传的实现需要设置一个超时计时器, 来设置发送方等待ACK回复的时间。

不仅如此, TCP还提供流量控制来控制发送方的发送频率, 进行拥塞控制来防止过多的数据植入到网络中导致网络性能变坏。所以本文设计的即时聊天软件的实现采用TCP协议可以有效的保证消息在传输层传递的可靠性, 减少在传输层对消息安全、完整、可靠等方面的处理。同时TCP对传输的字节流形式不做任何解释, 这样以便于即时聊天软件发送消息形式多样化的拓展。

#### 2.2.4 Netty框架

Netty是一个高性能、异步事件驱动的NIO框架, 基于Java NIO提供的API实现[13]。Netty 除了支持传输层的 UDP、TCP、SCTP协议, 也支持 HTTP(s)、WebSocket 等多种应用层协议, 它并不是单一协议的API。作为一个异步NIO框架, Netty的所有IO操作都是异步非阻塞的[19], 通过Future-Listener机制, 用户可以方便的主动获取或者通过通知机制获得IO操作结果。

##### 1. 什么是NIO?

Java NIO全称Java Non-blocking IO, 意为Java同步非阻塞IO[14], 基于Reactor模型实现。下面简单了解NIO与BIO的区别:

(1) BIO: 同步阻塞IO, 当出现一个连接就需要创建一个线程来处理该请求, 并且在客户端产生请求的时候, 在得到请求处理结果之前会造成客户端下一次请求的阻塞。在高并发场景下, 当产生大量客户端请求的时候, 这种IO会造成服务端线程过多导致服务端负载过高最后甚至崩溃的同时还会使大量后置客户端请求阻塞无法得到处理响应, 大大影响程序的性能、造成机

器资源的浪费以及降低用户的体验感，所以BIO在高并发场景下并不适用。

(2) NIO：利用单线程处理连接，即使在单线程的情况下也能同时处理大量的客户端请求，不会造成其他客户端请求的阻塞。NIO的I/O多路复用技术使得NIO单线程就可以实现非阻塞IO，通过一个selector线程来不断的轮询大量的channel（通道），每次就获取一批有事件的channel[15]。当出现客户端请求时selector直接接收，并交由对应的channel进行处理，再由channel启动线程来处理请求，这样就使selector能够专注接收客户端请求而无需自己处理，避免处理过程中造成其他请求的阻塞。而且由于NIO是单线程并且只有在连接/通道真正有读写事件发生时(事件驱动)，才会进行读写，避免了多线程占用内存以及线程之间切换导致资源浪费，很大程度上的减少了系统的开销。

## 2. Netty的优点

Netty框架的底层基于NIO实现的同时在NIO等类库上进行了很多的改进与优化，保证高吞吐量、低延时、高可靠性。相对而言Netty具有很多优点：

- (1) API使用简单，学习成本低。
- (2) 功能强大，内置了多种解码编码器，支持多种协议。
- (3) 性能高，对比其他主流的NIO框架，Netty的性能最优[15]。
- (4) 社区活跃，发现BUG会及时修复，迭代版本周期短，不断加入新的功能。
- (5) Dubbo、Elasticsearch都采用了Netty，质量得到验证。

### 2.2.5 利用Netty框架实现即时聊天

接下来用一个小案例在Spring Boot框架下利用Netty实现一个简单的聊天室功能：

1. 搭建一个Spring Boot工程，在该工程下导入Netty依赖。
2. 创建一个DiscardServer类，用来配置并建立Netty连接：

```
public class DiscardServer {  
    private int port; //连接端口号  
    public DiscardServer(int port) { this.port = port; }  
    public void run() throws Exception { //设置主线程池线程数量与工作线程池线程数  
        EventLoopGroup bossGroup = new NioEventLoopGroup(1); //主线程池  
        EventLoopGroup workerGroup = new NioEventLoopGroup(); //工作线程池  
        try {ServerBootstrap b = new ServerBootstrap();  
            b.group(bossGroup, workerGroup)  
            .channel(NioServerSocketChannel.class)  
            .childHandler(new ChannelInitializer<SocketChannel>() {  
                @Override  
                public void initChannel(SocketChannel ch) throws Exception {  
                    Charset gbk = Charset.forName("GBK");  
                    ch.pipeline().addLast("encoder", new StringEncoder(gbk)); //byteBuf编码  
                    ch.pipeline().addLast("decoder", new StringDecoder(gbk)); //byteBuf解码  
                    ch.pipeline().addLast(new DiscardServerHandler()); //接收处理消息  
                    //添加心跳机制，监控连接情况，超时断开  
                    //ch.pipeline().addLast(new IdleStateHandler(3, 0, 0));  
                    //ch.pipeline().addLast(new HeartbeatHandler());  
                }  
            })  
            .option(ChannelOption.SO_BACKLOG, 128)  
            .childOption(ChannelOption.SO_KEEPALIVE, true);  
            ChannelFuture f = b.bind(port).sync(); //连接端口  
            f.channel().closeFuture().sync(); //等待socket服务关闭  
        } finally { //关闭线程池  
            workerGroup.shutdownGracefully();  
            bossGroup.shutdownGracefully();  
        }  
    }  
}
```

3. 创建DiscardServerHandler类继承ChannelInboundHandlerAdapter类作为Netty消息接收处理器，主要通过重写channelActive()来监听websocket连接，重写channelRead()来处理通道连接读取的消息，重写channelInactive对连接断开时进行处理。

```
public class DiscardServerHandler extends ChannelInboundHandlerAdapter {  
    static Set<Channel> channelList = new HashSet<>(); //记录存储通道  
    @Override //通道活跃  
    public void channelActive(ChannelHandlerContext ctx) throws Exception {  
        channelList.forEach(e->{//通知其他人，本人上线  
            e.writeAndFlush("[客户端]" + ctx.channel().remoteAddress() + "上线了");  
        });  
        channelList.add(ctx.channel());  
    }  
    @Override //通道读取消息  
    public void channelRead(ChannelHandlerContext ctx, Object msg) throws Exception {  
        String message = (String) msg;
```

```

channelList.forEach(e->{//分发给聊天室内的所有客户端
if(e != ctx.channel()){
e.writeAndFlush("[客户端] " + ctx.channel().remoteAddress() + ": " + message);}
});
}
@Override // channel 处于不活跃的时候会调用
public void channelInactive(ChannelHandlerContext ctx) throws Exception {
channelList.remove(ctx.channel());
channelList.forEach(e->{//通知其他人, 本人下线
e.writeAndFlush("[客户端]" + ctx.channel().remoteAddress() + "下线了");});
}
}

```

4. 创建Starter启动类配置连接端口, 启动Netty服务, 利用网络调试助手根据配置的端口号建立连接, 就可以实现两个客户端之间的即时通讯。效果如图2-2所示。

图2-2 Netty简易即时聊天实现

5. 为了减少造成不必要的连接资源的浪费, 还可以创建一个实现ChannelInboundHandlerAdapter 的HeartbeatHandler类, 实现心跳机制用来检测连接状态, 连接超时不活跃就进行断开连接处理用来节省服务器资源。

```

public class HeartbeatHandler extends ChannelInboundHandlerAdapter {
int readTimeOut = 0;
@Override
public void userEventTriggered(ChannelHandlerContext ctx, Object evt) throws Exception {
IdleStateEvent event = (IdleStateEvent) evt;
if(event.state() == IdleState.READER_IDLE){
readTimeOut++;}
if(readTimeOut >= 3){// 超时超过3次, 断开连接
ctx.close();}
}
}

```

以上就是关于Netty框架用来实现局域网下即时聊天的简单流程及效果, 本文的整个聊天软件核心就是在此基础上根据接收到的消息类型做出不同的处理, 并利用Spring Boot框架整合其他三方技术所搭建的一个更加复杂的IM服务系统。

## 2.2.6 WebSocket

WebSocket是HTML5开始提供的一种浏览器与服务器进行全双工通讯的网络技术, 属于应用层协议。它基于TCP传输协议, 并复用HTTP的握手通道[16]。

WebSocket使得客户端和服务端之间的数据交换变得更加简单, 允许服务端主动向客户端推送数据。在WebSocket API中, 浏览器和服务端只需要完成一次握手, 两者之间就直接可以创建持久性的连接[17], 并进行双向数据传输。相较HTTP协议, WebSocket具有支持双向通信, 更灵活, 更高效, 可扩展性更好等优点[18]。

图2-3 HTTP与WebSocket请求响应对比图

## 2.2.7 WebRTC

WebRTC是一个开源项目, 旨在使浏览器能够为实时通信(RTC)提供简单的JavaScript接口。简单地说, 这意味着为浏览器提供JS即时消息接口。此接口创建的通道与WebSocket不同, WebSocket连接浏览器和服务端之间的通信而WebRTC通过一系列信号在浏览器之间建立了一个通道, 可以在不经过服务器的情况下发送任何数据。WebRTC实现了MediaStream, 它通过浏览器调用设备的摄像头和麦克风, 允许在浏览器之间传输音频和视频。

同时, WebRTC是一个出色的多媒体框架, 支持跨平台(如Android和IOS), 使Android和IOS设备作为终端设备能够像浏览器一样即时通信。

WebRTC的具有以下几点主要功能:

1. 用于音视频实时互动
2. 用于游戏、即时通讯、文件传输等等
3. 传输、音视频处理(回音消除、降噪等)

WebRTC核心API有三个: MediaStream、RTCPeerConnection与RTCDataChannel。

1. **MediaStream:** 通过设备的摄像头及话筒获得视频、音频的同步流
2. **PeerConnection:** 用于构建点对点之间稳定、高效的流传输的组件
3. **DataChannel:** 能够使得浏览器之间(点对点)建立一个高吞吐量、低延时的信道, 用于传输任何数据

本文主要是利用WebRTC这三个核心API通过JavaScript调用接口结合WebSocket对音视频的同步流进行传输交互, 来实现即时聊天软件的音视频聊天的功能。

## 3 需求分析

### 3.1 可行性分析

现代软件工程不可或缺的重要组成部分之一就是做可行性分析。对于一款应用软件的开发, 为了避免在实现阶段造成很对不必要的资源、时间、精力、人力等成本的浪费, 确保用最小的开销在一定的时间范围内确保整个方案的完成度, 对问题以及方案的实施做出认真、严谨且详细的可行性分析是十分有必要的。以下从经济、技术、操作着三个方面对可行性做出分析, 最终得出结论认为设计实现一款高性能即时聊天软件是可行的。

#### 3.1.1 经济可行性分析



整个软件开发所涉及的核心技术Java语言其框架Spring Boot与Netty都是开源的，甚至是整合Spring Cloud第三方组件：Zookeeper、RabbitMQ等以及数据库Redis、MySQL都是无需自费的。成本方面主要考虑的是提供能够支撑起软件高并发场景的服务器资源以及开发软件过程中时间与精力的付出。由于这是一个较为轻量的聊天软件，实现的功能虽然简化，但是在性能与功能相持且受众自由度较好的情况下，还是有很多用户愿意使用这样一款软件的。而且本软件作为一个毕业设计开发的产品，并不以盈利为目的，对经济收益方面并无要求，更多的是能够锻炼自身的编程思维，扎实个人的专业水平，增长关于IO网络编程的能力。

### 3.1.2 技术可行性分析

经过认真分析与思考，决定用Java语言来完成这款即时聊天软件。Java作为一门面向对象的高级编程语言，具有跨平台、可移植性强、安全且简单、性能高且支持分布式的同时多线程机制支持高并发场景，其语言的健壮性可以避免出错导致程序崩溃，底层的Java虚拟机实现了垃圾自动回收机制避免了内存泄漏，有效的保证了程序的稳定与安全。并且Java语言的J2SE平台，作为Java2平台标准版，支持桌面应用和服务应用，SE和相关组件技术提供了丰富的用户接口、操作性、多功能性、便携性和安全性。不仅如此，Java语言Spring Boot框架基于“约定优于配置”特性可以实现快速搭建项目，简化第三方技术的整合，大大的减少程序员的开发步骤。Netty框架的使用保证了程序的高并发能力，用户之间信息的传递通过Netty进行传递，快捷且高效，同时还可以将Netty接收到的消息异步放入消息队列给到应用服务利用多线程进行其他相关的业务处理，这样有利于对消息的持久化以及可靠性做出进一步保障。

整个聊天软件使用MySQL对数据进行持久化存储，利用Redis 作为缓存采用Java语言作为开发语言，以QQ、微信等即时聊天软件的排版以及功能为例，具有成熟的先例。

### 3.1.3 操作可行性分析

Java可实现的产品类型十分广泛，如手机游戏、中间件、软件、网站、电脑游戏，以及现在流行的安卓手机App等，都是由Java语言编写的。不仅如此Java语言提供可视化API以及丰富的库类资源，对于简单的功能只需要通过简单的继承就可以完成实现，继承的同时利用重写可以做到对功能特点性拓展，便于实现自定义开发。Java语言所编写的程序通过JVM编译称字节码文件后，可以做到无视平台，具有强大的可移植性，这样对于任何的Java开发的产品可以做到不做任何的修改就能够被移植到支持Java虚拟机的操作系统上。这款软件作为一个小型的局域网即时聊天软件，只需要一台以上连入局域网的电脑即可运行。无论是软件开发语言及技术还是硬件设备及开发成本来看都满足开发实践的条件。

## 3.2 功能需求

### 3.2.1 功能划分

软件客户端部分以及后台管理系统部分的功能模块主要由如图3-1所示。

图3-1 聊天软件功能模块设计图

### 3.2.2 功能描述

软件的功能设计侧重于客户端功能实现，后台管理系统主要是为管理员提供对用户数据的快速便捷管理。下面通过用例图分别描述用户客户端以及后台管理需要实现的主要功能：

一、客户端主要整体功能实现主要如图3-2所示。

图3-2 聊天软件客户端整体数据用例图

#### 1. 客户端登录与注册模块

注册模块主要功能用例图展示如图3-3所示。

图3-3 聊天软件客户端登录注册数据用例图

(1) 注册：用户注册需要填写昵称设置登录密码，并且进行手机或者是邮箱验证，验证码发送后用户接收并输入验证码，完成验证后可直接进行登录操作。

(2) 登录：登录功能的有三种方式，用户默认输入账号和密码登录，忘记密码的情况下还可以使用手机或者是邮箱通过验证码登录。

#### 2. 客户端用户模块

客户端用户模块主要功能用例图展示如图3-4所示。

图3-4 聊天软件客户端用户模块数据用例图

(1) 个人中心：本账号用户查看和编辑个人基本信息，包括头像设置与更换。

(2) 账号设置：设置中心为用户提供账号管理操作，主要包括：密码修改、密码找回、绑定邮箱、绑定手机号、注销账号功能。每一项操作都需要严格进行用户信息安全认证，以免他人随意篡改或者信息泄漏，保证用户数据安全性。

(3) 通知：出于提高用户体验感与净化聊天环境考虑，提供用户反馈与用户举报功能。不仅如此，利用通知公告为用户提供与管理交互能力，让用户问题有所解决，同时开发者可以吸收用户意见，提高软件功能与性能开发。

(4) 退出登录：用户退出客户端返回登录，本用户下线同时好友获取用户下线信息。

#### 3. 客户端好友模块

客户端好友模块主要功能用例图展示如图3-5所示。

图3-5 聊天软件客户端好友模块数据用例图

(1) 查找群及用户：方便用户快速定位好友或者群聊，同时非好友或者非群成员关系可以通过查找获取对方信息并建立好友及群关系。

(2) 好友操作：用户简介页面，用户可以查看好友或其他用户基本信息，但无法修改他人基本信息，好友关系下可以设置备注。用户与好友之间的操作主要是好友关系的建立与解除，所以需要包括以下功能：添加好友、删除好友、拉黑好友。此外，添加好友申请功能，尊重用户意愿，基于用户个人设置添加好友限制，添加好友时需额外通过申请审批。

#### 4. 客户端群聊模块

客户端群聊模块主要功能用例图展示如图3-6所示。

图3-6 聊天软件客户端群聊模块数据用例图

群聊功能不仅要考虑用户对群聊基本的增删改查操作，在进行操作逻辑处理时还需要根据群成员角色而设计不同的操作权

限。总体而言，主要分为群信息管理与群成员管理两类：

(1) 群信息管理：创建群聊、申请入群、退出群聊是每位普通用户都可实现的操作，同时群内成员都具有邀请其他成员入群的权利。但对群资料编辑，普通群成员只能修改本人相关的群信息，而管理员以上无修改限制。此外，群主只能操作解散群聊而不具备退出群聊的能力。同时群聊也需要具有入群申请审批功能，而审批人需为管理员或群主。

(2) 群成员管理：普通群成员是无法进行群成员管理的，只能对群内成员资料进行查看。管理员则可以对普通成员进行禁言以及踢出群聊处理。而群主则可以对群内任何人进行管理操作，不仅如此还额外具有设置管理员、转移群主的能力。

### 5. 客户端消息文件

客户端消息文件模块主要功能用例图展示如图3-7所示。

图3-7 聊天软件客户端消息文件模块数据用例图

一款聊天软件的核心功能是用户之间的消息传递，本模块主要基于用户操作聊天框与消息数据两个方面考虑，基本需要具有以下功能：

(1) 普通文本消息发送：最基础的消息传递形式，其中包括对表情的发送。

(2) 发送与接收文件：好友双方可以选择以文件的形式进行消息传递，用户选择本地文件或图片发送给对方，接收方收到消息，可以对文件进行基本查看与下载处理。

(3) 音视频通话：好友之间单聊可以进行语音通话以及视频通话，通话条件是双方必须同时在线，保证实时通讯。

(4) 消息记录搜索：帮助用户快速排查定位消息。

(5) 消息未读：提示用户消息接收，增加用户体验感。

(6) 聊天会话框置顶与删除：聊天会话置顶，方便用户定位好友聊天框，快速开启聊天。还提供删除操作，一般是用户好友关系不存在或者为非群聊成员无法发送消息时，会话框失效就可进行删除操作。

二、后台管理系统功能实现主要如图3-8所示。

图3-8 聊天软件后台主要功能数据用例图

### 1. 后台系统首页

后台系统首页数据主要包括内容如图3-9所示。

图3-9 聊天软件后台系统首页数据用例图

后台系统首页主要是对软件数据进行一个展示处理，主要包含管理员信息简要展示、用户角色类型分布百分比、软件用户量以及用户消息总数等数据的可视化展示。

### 2. 后台系统管理

后台系统管理数据主要包括内容如图3-10所示。

图3-10 聊天软件后台系统管理数据用例图

后台系统管理为管理员提供了数据库核心数据可视化操作，主要包括以下几个功能需求：

(1) 用户角色管理：由于整个即时聊天软件的设计涉及用户以及管理员，需要对客户端与后台管理的登录用户进行区别，所以需要对只用该软件的所有用户进行一个角色分配管理。主要是管理员角色与普通用户角色。

(2) 用户信息管理：管理员可以对使用软件的用户进行整体统一性的管理，为用户配置角色，查看用户在线状态或者是控制用户状态是否启动。

(3) 消息记录：管理员对用户消息进行查看，可以为管理员对用户通讯行为有一定的掌控，为用户反馈的处理提供依据。对于不良的消息内容管理员可以控制删除某条用户信息。

(4) 消息文件：对用户使用软件产生的文件进行管理，包括用户头像、群头像以及用户发送消息产生的文件。管理员可以对用户违规文件进行清理处理。

(5) 通知公告：用户反馈以及用户举报产生的信息都会在这个模块由管理员进行处理，处理后的用户反馈管理员将给出反馈处理结果并发送通知给用户，同时管理员也可以进行通知的创建向指定用户发送通知或者是向全体客户端用户发送公告。

(6) 系统参数：对系统所有内置参数进行展示，管理员可以直观的对系统重要参数进行配置，新增，修改或者是删除处理。

(7) 操作日志：记录所有用户使用软件的操作日志以及管理员使用后台的操作记录，掌握用户对软件的使用行为。用户日志只允许查看无法进行随意的删除处理。

### 3. 后台账号中心

后台系统管理账号中心数据主要内容如图3-11所示。

图3-11 聊天软件后台系统管理数据用例图

账号中心主要是对登录管理员账号进行管理，管理员可以进行密码的修改以及管理员基本信息的编辑与修改，同时由于后台允许管理员进行操作，所以管理员拥有创建新管理员账号的权力，当然也可以是普通用户账号，但一般管理员不进行此操作，普通用户由用户自行使用客户端创建账号。

### 3. 3 非功能需求

#### 3.3.1 性能需求

即时聊天软件的性能需求主要如下：

1. 保证数据的安全：避免对数据库的非法操作，造成软件功能出错。需要对用户密码，电话以及邮箱等重要信息做加密以及解密技术处理。

2. 时间特性：大部分的操作允许在一定的响应时间范围内完成都可接受。

3. 适应性：平台的一致性，操作界面简介舒适，用户操作便捷，对硬件的要求不高。

#### 3.3.2 运行需求

聊天系统的运行需求如下所示。

1. 用户界面：采用B/S架构，对用户而言操作更加便利，但是支持鼠标操作而无法实现键盘操作。

2. 硬件接口：客户端支持连接到网上，服务器支持奔腾以上的机器。

3. 软件接口：客户端推荐使用Windows操作系统，服务器数据使用MySQL与Redis。

3. 4 数据需求

3.4.1 数据流分析

首先对本款即时聊天软件的数据流进行以下的主要划分。

1. 顶层数据流图：如图3-12所示。

指 标		
疑似剽窃文字表述		
<div>1. Netty 除了支持传输层的 UDP、TCP、SCTP协议，也支持 HTTP(s)、WebSocket 等多种应用层协议，它并不是单一协议的 API。</div> <div>2. 简单地说，这意味着为浏览器提供JS即时消息接口。此接口创建的通道与WebSocket不同，WebSocket连接浏览器和服务器的通信</div> <div>3. 可以在不经过服务器的情况下发送任何数据。WebRTC实现了MediaStream，它通过浏览器调用设备的摄像头和麦克风，允许在浏览器之间传输音频和视频。</div> <div>4. 手机游戏、中间件、软件、网站，电脑游戏，以及现在流行的安卓手机App等，都是由Java语言编写的。不仅如此Java</div>		
3. 基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第3部分		总字数：8891
相似文献列表		
去除本人文献复制比：2.9%(254) 去除引用文献复制比：2.9%(254) 文字复制比：2.9%(254) 疑似剽窃观点：(0)		
1	张伯虎-基于Web的BTA连锁酒店物资管理系统的设计与实现 - 《大学生论文联合比对库》- 2016-05-24	0.5% (41) 是否引证：否
2	国网福州供电公司电力工程项目管理系统的分析与设计 陈志光(导师：周华) - 《云南大学硕士论文》- 2016-09-01	0.4% (34) 是否引证：否
3	基于深度学习框架的小样本视频分类方法的研究与实现 何江龙(导师：张忠宝) - 《北京邮电大学硕士论文》- 2021-06-01	0.4% (32) 是否引证：否
4	彩铃业务病毒式营销互联网社区应用框架的设计与实现 钟辉辉(导师：王纯) - 《北京邮电大学硕士论文》- 2012-12-16	0.3% (31) 是否引证：否
5	社交网络中基于关系强度的用户群体发现研究 邓钟晟(导师：陈德华) - 《东华大学硕士论文》- 2014-11-01	0.3% (31) 是否引证：否
6	12621336_邓俊_基于Web的政府门户网站设计与实现 严枚 - 《大学生论文联合比对库》- 2018-05-03	0.3% (31) 是否引证：否
7	药品管理系统的设计与实现 马润珍 - 《大学生论文联合比对库》- 2018-03-26	0.3% (31) 是否引证：否
8	中国银行山东省分行CRM系统开发与应用研究 梁佳(导师：李长云) - 《哈尔滨理工大学硕士论文》- 2016-03-01	0.3% (30) 是否引证：否
原文内容		

图3-12 顶层数据流图

2. 一层数据流图：如图3-13所示。

图3-13 一层数据流图

3. 二层数据流图：分模块对一层数据流图的六个数据处理进行更为细节的数据流展示。

(1) 用户登录，如图3-14所示。

图3-14 二层用户登录数据流图

(2) 用户管理，如图3-15所示。

图3-15 二层用户管理数据流图

(3) 好友管理，如图3-16所示。

图3-16 二层好友管理数据流图

(4) 群聊管理，如图3-17所示。



图3-17 二层群聊管理数据流图

(5) 消息收发, 如图3-18所示。

图3-18 二层消息收发数据流图

(6) 聊天会话管理, 如图3-19所示。

图3-19 二层聊天会话管理数据流图

### 3.4.2 数据字典

1. 数据处理字典, 如表3-1所示。

表3-1 数据处理字典表

2. 数据存储字典, 如表3-2所示。

表3-2 数据存储字典表

3. 数据流字典, 如表3-3所示。

表3-3 数据流字典表

表3-3 数据流字典表续表

### 3.4.3 ER图

软件整体的ER图设计如图3-20所示。

图3-20 软件整体E-R图

本软件ER图设计主要包括14个实体, 14个实体之间具有以下关联:

1. 系统参数实体: 主要用于帮助管理员操作后台系统时对服务系统参数进行管理与设置。

2. 用户绑定角色: 管理员能对每个用户进行角色绑定, 一个角色可以绑定多个用户。

3. 用户与好友: 如果用户与用户之间存在好友关系, 那么好友关系信息会存储在好友关系表中, 其中一个用户可以存在多个好友用户。

4. 用户添加好友发送好友申请: 根据接收方用户设置, 当用户进行添加好友时需要先发送好友申请并通过审批, 每位用户可以发送多条好友申请。

5. 用户操作系统产生日志: 操作日志记录用户各种操作行为。

6. 用户已加入群聊: 用户与群聊之间存在多对多的成员关系, 群成员关系作为中间数据, 即每位用户可以加入多个群聊, 同时每个群聊可以包含多位用户成员。

7. 用户申请入群发送群聊申请: 根据群信息设置, 用户加入群聊时需先发送群聊申请用通过管理员审批, 每位用户可以发送多条群聊申请。

8. 用户反馈通知: 用户可以对其他用户进行举报或者向软件管理员发送反馈通知。每位用户可以发出多条反馈通知。

9. 用户创建聊天会话: 用户与用户之间进行通信发送消息之前需要创建聊天会话框, 一个用户可以创建多个聊天会话框。

10. 用户发送单聊消息: 用户与单聊消息记录之间存在一对多的关系。即用户可以向多为特定好友发送多条单聊消息, 产生单聊消息记录。

11. 用户发送群聊消息: 用户与群聊消息记录、群聊与群聊消息记录之间都存在一对多的关系。群成员通过群聊向群内其他成员发生消息, 产生群消息记录。

12. 消息体对应消息记录: 分为单聊与群聊, 消息记录不包含消息内容, 但消息体包含。一条单聊消息体对应两条消息记录, 分别归属发送方与接收方。而一条群聊消息体就对应一条消息记录, 归属方为群聊。

13. 文件记录对应消息体: 用户发送的文件消息需要额外记录在文件记录中, 文件消息可重复发送, 所以一条文件记录可以对应多条消息体。

14. 管理员管理系统参数: 每位管理员用户在后台系统可以对所有系统参数进行管理。

其中每个实体类的具体属性如图3-21至3-27所示。

图3-21 用户与角色实体类图

图3-22 好友申请与群聊申请实体类图

图3-23 消息实体与聊天会话框类图

图3-24 操作日志与组群实体类图

图3-25 通知与好友实体类图

图3-26 单聊消息记录与群聊消息记录实体类图

图3-27 文件记录与系统参数实体类图

## 4 系统设计

### 4.1 功能结构

本软件的设计采用的是B/S架构, 属于C/S架构的一种改进。用户只需要通过浏览器无需下载软件就可以使用软件进行即时聊天。用户通过对客户端页面进行操作, 前端利用axios向服务端发送请求进行处理。客户端以及后台的页面展示效果由前端利用vue3实现, 软件的主要逻辑功能由服务端利用Spring Boot框架与Netty框架进行请求响应处理, 结果返回给前端进行展示。B/S架构图如图4-1所示。

图4-1 即时聊天软件B/S架构图

本研究软件设计整体架构如图4-2所示。

图4-2 即时聊天软件设计架构图

对本文即时聊天软件的设计主要分为三个部分: 用户客户端、管理员后台系统以及核心服务端处理。软件的主要功能通过客户端以及后台管理直接操作。客户端用户主要由核心单聊群聊操作、个人信息修改、好友关系操作以及群聊关系操作等功能实现。后台系统管理员实现对软件数据进行管理, 主要包括用户数据、聊天信息数据、系统参数、重要文件管理等操作。服务端则负责处理客户端用户请求以及后台管理员请求后分别将响应结果返回给客户端与后台管理。软件整体系统功能结构如图4-3所示。

图4-3 即时聊天软件功能结构图

#### 4.1.1 登录注册功能

任何一款软件的设计都离不开登录与注册功能的实现，本文即时聊天软件登录注册功能如图4-4所示。

图4-4 聊天软件客户端登录功能结构图

如上图所示，登录与注册的设计分为客户端登录注册与后台登录两部分：

1. 用户登录：一般来说，用户登录功能默认提供的是使用账号以及密码进行登录。此外本软件从用户的角度出发，结合用户忘记密码的情形设计了根据用户绑定的手机号或者邮箱通过发送验证码进行软件登录使用功能，为用户登录使用软件进行双重保险。

2. 用户注册：当账户不存在时，用户可以通过手机号或者邮箱接收验证码，设置登录密码验证成功后完成用户账号注册，注册成功后可以直接进行登录操作。

3. 后台登录功能时提供软件管理员登录使用，用户账号无法进行登录操作。后台管理员的登录相对比较简单，直接通过输入账号密码登录即可。

#### 4.1.2 客户端用户模块

客户端用户模块只要功能如图4-5所示。

图4-5 聊天软件客户端用户模块功能结构图

客户端用户通过点击自己用户头像出现个人中心页面，本页面可进行个人用户基本信息的查看与编辑以及用户头像的更换。至于关于用户账号安全的关键信息，需要点击设置图标来进行对用户密码、手机号更改以及邮箱绑定等操作。主要有以下几点：

1. 头像更换：在用户个人中心点击更换头像按钮可以选择本地图片进行头像更换。

2. 用户资料编辑：在用户个人中心界面点击编辑按钮即可对用户个性签名、生日、地址、昵称，性别、好友认证方式等信息进行修改，保存成功后用户信息修改完成。

3. 修改密码：侧边栏设置按钮选择修改密码进行修改密码操作，在设置新密码之前需要选择根据手机号或者是邮箱认证方式对密码进行修改，新密码设置完需要发送认证验证码完成验证才可完成密码修改，防止其他人恶意操作修改用户密码。

4. 找回密码：侧边栏设置按钮选择找回密码，也需要选择邮箱或者是手机号认证方式，直接根据选择的认证方式获取验证码，完成验证即可出现关于用户密码信息的弹出框。

5. 绑定手机号：如果用户在注册账号的时候通过手机号注册，那么更改手机号绑定的时候需要对原手机号进行认证，认证成功后才可设置新的手机号，并对新手机号进行验证码验证成功后完成更换手机号绑定。如果用户之前并没有绑定过手机号，那么直接就可以设置新手机号并通过验证即可完成绑定。

6. 绑定邮箱：本功能的实现主要与绑定手机号功能类似，如果用户为绑定邮箱，直接设置新邮箱号并接受输入验证码完成验证即可设置成功，否则就需要先对原邮箱进行认证才可进行新邮箱绑定。

7. 注销账号：这个功能较为严谨，需要用户通过对绑定的邮箱或者手机号完成认证才能执行，一旦用户进行注销账号操作，该用户直接退出软件回到登录界面，但改用户账号不再具有登录能力，且关于该用户信息全部被删除。

8. 用户反馈：侧边栏点击通知公告按钮，出现通知公告界面，用户可以在该页面最上方点击用户反馈按钮向后台系统提出用户请求或者是给出用户建议，后台管理员接收到消息通知会对用户反馈做出处理并给出反馈通知，这个设计给与用户能够与管理员进行交互的功能，让软件更好的为用户服务，提高用户体验感。

9. 系统通知：用户客户端在有接收到一些重要信息通知时会弹出系统公告提示框，同时用户也可以在通知公告界面查看关于该用户的所有反馈信息、用户举报信息以及系统通知。

#### 4.1.3 客户端好友模块

客户端用户之间进行即时聊天需要建立好友关系，好友模块功能设计主要如图4-6所示。

图4-6 聊天软件客户端好友模块功能结构图

客户端好友模块主要设计包括以下几部分功能设计：

1. 好友查找：用户通过搜索栏输入搜索内容进行用户信息搜索，搜索的结果可以是满足搜索信息的用户账号，用户昵称的模糊匹配。这样可以帮助用户实现对非好友用户的查找以及好友用户的快速定位。

2. 好友备注：在好友简介页面点击好友昵称旁边的修改按钮重新输入对好友的备注确认提交之后完成对好友备注的修改。

3. 删除好友：在好友简介页面，如果该用户与本用户之间存在好友关系就可以点击删除好友按钮，删除该用户与本用户之间的的好友关系，不存在好友关系之间的用户将无法进行即时聊天。

4. 拉黑用户：无论是否存在好友关系都可以将该用户进行来黑操作，被拉黑的用户如果与本用户之间将不再具有即时聊天功能，并且也无法进行添加好友操作，想要恢复通讯交流只能由拉黑用户取消对被拉黑用户的拉黑操作。

5. 添加好友：不存在好友关系之间的用户可以在对方的好友简介中通过添加好友按钮，根据对方添加好友认证方式的设置进行添加好友操作。如果对方添加好友无需认证，即可直接进行好友的添加，双方之间直接建立好友关系，否则，就需要先发送好友申请，发送申请的同时可以添加附言以及对该用户的好友备注设置。发送的好友申请可以通过点击在侧边栏的好友申请按钮查看我方接收以及发送的好友申请，查看进度以及结果。只有当对方通过我方好友申请双方之间才可建立好友关系，即完成好友添加。

6. 用户举报：通过点击好友简介的右上角的红色感叹号可以对该用户进行举报操作，举报用户需要输入举报理由，提交即可完成对该用户的举报。举报的信息以及点击在侧边栏的公告按钮进行进度或者处理反馈查看。

#### 4.1.4 客户端群聊模块

客户端用户之间的交流方式不仅可以通过建立好友关系进行单聊，还可以由用户创建群聊并添加群成员进行群聊。该群里面的所有用户之间都可以进行通讯，无论成员之间是否存在好友关系，这就是即时聊天群聊功能。本软件的群聊功能设计如图4-7所示。

图4-7 聊天软件客户端群聊模块功能结构图

客户端群聊模块功能涉及的内容相对比较多，主要涉及以下几个部分：

1. 创建群聊：任何用户都可以通过在群聊列表最上方的创建群聊按钮开始创建群聊，群主对群聊进行设置，包括群聊名称、群聊类型、禁言类型、群聊人数、申请入群方式设置、设置管理员、群聊普通成员邀请、群介绍以及群公告等设置之后即可完成群聊创建，被邀请和设置的管理员直接入群，成员之间直接可以通过该群进行群聊交流。

2. 申请入群：在搜索栏也可以通过群账号或者群昵称进行搜索，点击群搜索结果打开群简介，非群成员可以点击申请入群按钮进行入群操作。根据群入群方式设置：直接入群、禁止入群以及管理员审批入群。如果是需要管理员审批入群，则先提交入群申请，经过该群管理员通过之后才算加入群聊。

3. 群信息编辑：只有管理员或者群主才有资格编辑修改群资料信息，主要包括对群头像、群简介、群公告、入群方式、群禁言、群成员数等内容的修改与设置。

4. 群成员邀请：该群中的任何成员都有资格邀请自己的好友进入群聊。管理员或者房主操作操作，被邀请的成员根据该群入群方式直接加入群聊或者是禁止入群，非管理员操作，根据群入群方式设置在须经管理员审批情况下，通过管理员审批才可完成入群。

5. 群昵称设置：任何成员在群简介界面可以直接设置及修改自己在本群的群昵称。

6. 群成员管理：该功能主要面对群管理员及以上的用户，普通成员只能查看其他群成员的信息，而群管理员及以上则可以对群内某成员进行禁言时间设置以及群成员踢出群聊处理。不仅如此，群主还可进行群主转让操作以及管理员设置操作。

7. 退出群聊：除了群主以外其他群内成员可以直接通过点击群简介页面的退出群聊按钮进行退出群聊操作，退出群聊的成员将不再具备在该群发言的能力。

8. 解散群聊：只有群主才有资格解散群聊，被解散的群聊数据将进行删除处理，群内的任何成员将无法再通过该群进行通讯交流。

#### 4.1.5 客户端消息文件发送模块

一款即时聊天软件的重点就是在于用户之间的信息通讯交流，其中包括消息的传播形式，本即时聊天软件客户端消息文件模块的功能设计如下图4-8所示。

图4-8 聊天软件客户端消息文件模块功能结构图

客户端用户通过点击聊天会话列表的文本框或者通过在好友简介以及群聊简介中点击发消息按钮，开启聊天界面，在输入框中输入发送内容，点击发送按钮或者按回车都可以进行消息发送。本即时聊天软件为所有单聊包括群聊都提供了以下功能：

1. 消息未读：在聊天会话列表会对最新的未读消息进行小红点标记，当点击聊天页面发送消息时默认消息已查看，这是未读小红点消失。

2. 文本消息：本软件的设计文本消息不仅可以发送文字还可以在发送框上方选择个性表情进行表情的发送，完成编辑即可按回车或者点击发送按钮都可以完成对文本消息的发送。

3. 图片消息：在发送文本框上方点击图片按钮，可以对该用户本地图片进行选择发送，图片会先在文本框预览，用户决定发送时再按回车或者发送按钮对图片进行发送。已经发送或者接收的图片可以在消息显示框点击该图片对图片进行预览及查看。

4. 文件发送：文件的发送处理与消息的发送处理有点不同，用户在选择想要发送的文件之后不会在发送文本框中进行预览处理，而是选择完文件直接就会对该文件进行发送。发送过去的文件将会显示该文件的名称以及大小。对方无法立刻查看文件内容，需要对方接收并下载到本地才可进行查看。

5. 文件下载：对于接收到的文件消息，点击文件消息对该文件进行文件预览或者下载操作，用户可以选择路径将文件下载到本地进行查看。

#### 4.1.6 客户端音视频聊天模块

软件客户端只提供了单聊模块的音视频聊天功能，主要功能设计如图4-9所示。

图4-9 聊天软件客户端音视频聊天模块功能结构图

本软件最大的难度以及设计突破点就是结合Spring Boot、Netty、WebSocket、WebRTC实现了音视频聊天功能主要包括以下两个设计点：

1. 语音聊天：语音聊天功能的使用除了无法与视频聊天同时进行，其他功能都可以。也就是说可以一边进行语音聊天一边进行文本聊天。因为语音聊天的并不会占用整个聊天界面。

2. 视频聊天：跟语音聊天不同，视频聊天需要占用聊天界面，所以在视频聊天结束之前无法进行其他的聊天操作，并且由于WebSocket本地媒体流在界面刷新之后会进行重置，所以当未正常挂断就结束视频聊天会被视为本次聊天异常中断，不会对本次聊天进行计时处理。

音视频聊天的大概流程如图4-10所示。

图4-10 聊天软件客户端音视频聊天流程图

#### 4.1.7 后台系统首页数据

后台系统首页主要是设计展示系统数据，主要展示内容如图4-11所示。

图4-11 聊天软件后台系统首页功能结构图

由上图可知本模块的设计主要包括对管理员简要信息展示、系统用户总数、系统在线用户数以及用户发送消息总数的展示。其中发送消息数以及用户总数展示采用折线图进行一个更加直观性的数据展示。不仅如此，本页面还对用户类型进行了一个百分比分布统计，让管理员更清晰的对本系统用户分布有一个总结性的了解。

#### 4.1.8 后台系统管理模块

后台系统首页主要是设计展示系统数据，主要展示内容如图4-12所示

图4-12 聊天软件后台系统管理功能结构图

后台管理系统的系统管理模块一共分为七个子模块，分别是角色管理、用户管理、消息记录、消息文件、通知公告、系统参数以及操作日志。

1. 角色管理：这个模块只要是对使用软件的每一位用户进行角色分配管理，通过区分普通用户以及管理员角色用户来限制用户权限。本款软件只支持用户使用客户端、管理员使用后台。除了基础的增删改查操作，这里需要对用户进行一个同意的角



色分配绑定，为每个用户分配一个角色，通过客户端创建的用户默认为普通用户，未分配角色的用户在使用后台或者是客户端操作时会造成错误响应。如果角色删除，该角色下的所有用户将会被置为非角色用户需要重新进行角色绑定。

2. 用户管理：管理员有权对软件使用的所有用户进行统一管理，同时管理员有权修改用户信息，禁止该用户行为以及注销某用户的操作。由于这里涉及到用户隐私以及用户重要信息的操作，所以一般管理员使用该模块来对用户行为进行约束，情形严重者直接由管理员进行账号注销操作，而不是擅自随意修改用户信息。

3. 消息记录：本模块管理员对用户聊天行为进行管理，对于违规或不良的消息，管理员有权进行该条消息记录删除处理操作，但是无法进行修改操作。

4. 消息文件：本模块对软件系统用户产生文件进行统一管理，除了包括用户聊天产生的图片与文件信息还包括用户头像以及群头像数据管理。管理员可以下载用户文件进行查看或者是对图片进行预览操作。对于用户产生的违规或不良的图片或文件管理员有权进行删除处理。

5. 通知公告：本模块功能设计主要是为了进一步提高用户体验，净化通讯环境，让管理员与用户之间存在消息通讯，用户遇到问题能够有所方式寻求管理员解决。主要包括用户举报信息，用户建议，管理员需要对用户发布的通知进行处理，处理完成的通知才可进行通知发布给指定用户或者是所有用户。管理员可以对任何一条通知公告进行删除操作，但是管理员无法修改用户反馈通知，但是又管理员自创建的系统通知或者是公告是可以编辑修改的。

6. 系统参数：主要是用来给管理员提供一个直观性的系统内部设置参数展示。

7. 操作日志：记录每一位用户使用软件的操作行为，跟踪管理用户行为，保障用户操作执行的同时一旦软件系统出现报错操作也可以快速定位错误，对症下药。

4.1.9 后台账户中心模块

后台账号中心主要是设计展示系统数据，主要展示内容如图4-13所示

图4-13 聊天软件后台账号中心功能结构图

后台管理账号中心模块的功能主要是给管理员提供一个账号修改、新增的操作能力，由于服务端用户默认为普通用户，所以能够登录后台的其他管理员用户账号只能由已有的管理员进行创建才有效。如果管理员忘记了密码也可以在本模块进行密码修改操作，管理员的密码修改没有客户端复杂，所以在创建账号和修改密码时管理员都需要慎重操作。

4.1.10 退出登录模块

本文即时聊天软件退出登录功能如图4-14所示

图4-14 聊天软件退出登录功能结构图

用户账号的退出，无论时管理员操作后台系统还是用户操作客户端系统，都是将判定用户状态由在线转变为离线，同时要删除页面保存关于本次登录用户的Session，并且操作执行成功后直接退回初始登录页面进行重新登陆操作。这里需要注意避免管理员直接关闭页面进行系统退出造成用户在线状态出错，正确的应该是点击头像进行系统退出操作。

4.2 功能流程

4.2.1 登录注册

登录注册功能流程图如图4-15所示。

图4-15 聊天软件登录注册功能流程

4.2.2 客户端功能流程图

客户端功能流程图如图4-16所示。

图4-16 聊天软件客户端功能流程图

4.2.3 后台功能流程图

即时聊天软件后台管理系统功能流程图如图4-17所示。

图4-17 聊天软件后台功能流程图

4.3 数据库设计

4.3.1 逻辑模型

即时聊天软件设计数据库逻辑模型如下图4-18所示。

图4-18 逻辑模型图

4.3.1 物理模型

即时聊天软件设计数据库逻辑模型如下图4-19所示。

图4-19 物理模型图

4.3.1 数据表

即时聊天软件数据表设计主要如表4-1至4-14所示。

表4-1 user表

表4-2 conervation表

表4-3 friendship表

表4-4 friendship request表

表4-5 group表

表4-6 group member表

表4-7 group message表

表4-8 group request表

表4-9 file表

表4-10 message body表

表4-11 message表

表4-12 param表

表4-13 role表

表4-14 notification表

4. 4 子系统设计

4. 4. 1 TCP模块

本软件的设计主要核心在于后台服务端代码，软件最重要的功能是通过Netty框架采用TCP协议结合前端使用WebSocket技术实现消息即时传递。同时为了提高软件并发能力整合第三方组件RabbitMQ以及Zookeeper，数据库结合Redis作为缓存提高数据请求响应效率。服务端TCP模块的类图如4-20所示。

图4-20 服务端TCP模块类图

根据图中数据，对以下核心类做简要介绍，如表4-15所示。

表4-15 TCP模块类功能表

表4-15 TCP模块类功能表续表

4. 4. 2 客户端时序图

软件客户端与服务端时序图如图4-21所示。

图4-21 客户-服务端时序图

5 详细设计

5. 1 系统界面

本章节将会对软件的界面以及核心功能实现进行详细介绍，首先从以下几个模块对软件的界面进行展示。

指 标	
疑似剽窃文字表述	
1. 系统管理功能结构图 后台管理系统的系统管理模块一共分为七个子模块，分别是角色管理、用户管理、消息记录、消息文件、	
4. 基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第4部分	总字数：9643
相似文献列表	
去除本人文献复制比：1.2%(115) 去除引用文献复制比：1.2%(115) 文字复制比：1.2%(115) 疑似剽窃观点：(0)	
1 基于云加密系统的即时通讯技术研究与应	0.5% (45)
叶奔(导师：黄浩) - 《华中科技大学硕士论文》 - 2020-11-01	是否引证：否
2 G计101-王菊	0.4% (36)
王菊 - 《大学生论文联合比对库》 - 2014-06-13	是否引证：否
3 基于JSP的宠物店管理系统	0.3% (30)
周炎炎 - 《大学生论文联合比对库》 - 2016-04-14	是否引证：否
原文内容	

5. 1. 1 客户端登录与注册

客户端登录界面如图5-1所示（以默认账号密码登录为例）：

图5-1 客户端登录界面展示图

客户端注册界面如图5-2所示（这里以手机号验证注册为例）：

图5-2 客户端注册界面展示图

5. 1. 2 客户端聊天主页面及其子页面

（1）主页面，登录成功之后进入聊天主页面如图5-3所示。

图5-3 客户端聊天主页面展示图

页面整体设计分为左中右三部分：

1. 左：侧边栏，分为六个图标按钮。从上至下依次为用户头像用来展示个人中心、好友申请、组群申请、通知公告、个人设置以及退出登录。

2. 中：搜索与列表，分为上下两部分。上部分为文本输入框，根据输入内容对用户或组群信息进行搜索。搜索结果会在下半部分列表框进行展示。下部分为展示列表框，列表最上方三个图标分别展示聊天会话、好友以及组群，默认展示已存在的聊天会话框。。

3. 右：展示框，页面左中部分固定不变，而展示框根据用户操作进行换展示不同的内容。

（2）个人中心，主页面左侧栏点击用户头像，展示个人信息。如图5-4所示。

图5-4 客户端个人中心页面展示图

（3）好友申请，主页面左侧栏点击好友申请按钮，展示好友申请。如图5-5所示。

图5-5 客户端好友申请页面展示图

（4）组群申请，主页面左侧栏点击组群申请，展示组群申请。如图5-6所示。

图5-6 客户端组群申请页面展示图

(5) 通知公告, 主页面左侧栏点击通知公告按钮, 展示通知公告内容。如图5-7所示。

图5-7 客户端通知公告页面展示图

(6) 个人账号设置, 主页面左侧栏点击设置按钮, 进行账号设置。如图5-8、5-9所示(这里以修改密码为例):

图5-8 客户端个人设置-密码修改页面展示图

(7) 用户/群聊搜索, 在搜索栏输入搜索信息, 展示搜索匹配结果。如图5-9所示。

图5-9 客户端用户/群聊搜索展示图

(8) 好友/用户简介, 如图5-10所示。

图5-10 客户端好友简介展示图

(9) 组群简介, 如图5-11所示。

图5-11 客户端组群简介展示图

(10) 单聊/群聊界面, 如图5-12、5-13所示。

图5-12 客户端单聊展示图

图5-13 客户端群聊展示图

(11) 聊天记录, 点击单聊/群聊组件的聊天记录按钮进行聊天记录查询, 如图5-14所示。

图5-14 客户端聊天记录展示图

(11) 音视频聊天, 点击单聊组件的右上角音视频聊天按钮进行即时音视频聊天, 如图5-15、5-16所示。

图5-15 客户端语音聊天展示图

图5-16 客户端视频聊天展示图

### 5.1.3 后台登录页面

后台登录界面如图5-17所示。

图5-17 客户端登录页面展示图

### 5.1.4 后台管理主页面及其子页面

(1) 后台管理首页, 如图5-18所示。

图5-18 后台系统首页展示图

(2) 后台系统管理—用户角色页面, 如图5-19所示。

图5-19 后台系统用户角色页面展示图

(3) 后台系统管理—用户信息页面, 如图5-20所示。

图5-20 后台系统用户信息页面展示图

(4) 后台系统管理—消息记录页面, 如图5-21所示。

图5-21 后台系统消息记录页面展示图

(5) 后台系统管理—消息文件页面, 如图5-22所示。

图5-22 后台系统消息文件页面展示图

(6) 后台系统管理—通知公告页面, 如图5-23所示。

图5-23 后台系统通知公告页面展示图

(7) 后台系统管理—系统参数页面, 如图5-24所示。

图5-24 后台系统系统参数页面展示图

(8) 后台系统管理—操作日志页面, 如图5-25所示。

图5-25 后台系统操作日志页面展示图

(9) 后台管理系统账号中心页面, 如图5-26所示。

图5-26 后台系统账号中心页面展示图

## 5.2 具体技术实现方法

本即时聊天软件核心为后台服务端处理, 服务端设计分为两个部分: 应用服务端与TCP服务端。其流程如图5-27所示。

图5-27 服务端处理流程图

### 5.2.1 应用服务端—聊天会话操作

聊天会话模块, 功能实现接口如表5-1所示。

表5-1 聊天会话接口表

(1) 聊天列表展示: 调用getConversationList接口, 通过sql查询语句获取当前用户所有聊天会话, 结果先按顶置降序排列, 再按创建时间降序排列。这里需要判断是否有未读消息。

(2) 单个聊天会话获取: 用户点击某个聊天会话框后需获取该会话框详细信息。根据好友用户id或者是组群id进行查询。这里做已读消息处理。

(3) 删除聊天会话: 从会话列表中删除与某位好友或某群的聊天会话框。

(4) 顶置/取消顶置聊天会话: 前端传入需更新的顶置状态以及数据主键编号, 调用更新会话框接口, 更新顶置数据, 0为非定制, 1为顶置状态。

(5) 会话同步: 多端登录模式下, 需要进行不同端会话数据同步处理, 采取分批量同步, 限制每次拉取同步数据最大值, 根据isCompleted字段判断是否完成同步。

(6) 消息未读/已读处理: 获取会话列表时比较数据库中sequence字段数据与Redis中该用户同步conversationSeq字段数据大小, sequence>conversationSeq说明有未读消息, 客户端对未读消息做小红点标记。当用户点击获取该聊天会话时, 对Redis中conversationSeq与MySQL中sequence字段数据同步, 表示已读取所有已收消息。

### 5.2.2 应用服务端—好友操作

好友模块, 功能实现接口如表5-2所示。



表5-2 好友操作接口表

- (1) 用户查找：调用/searchFriend接口，通过sql查询语句获取所有满足搜索条件的用户数据，结合好友关系表扩充是否为好友信息。
- (2) 添加好友：调用/addFriend接口，先校验双方用户信息是否存在，根据用户添加好友设置判断是否需要进行好友申请，需要就先发送好友申请，否则直接建立好友关系。
- (3) 好友备注：前端封装修改信息对象，调用/updateFriend接口，通过sql语句对好友关系表中备注字段进行修改。
- (4) 删除好友：前端封装from\_id与to\_id数据，调用/deleteFriend接口，修改数据库中关系字段状态为被删除状态，数据删除标志为1。删除成功利用RabbitMq发送消息给被删除用户（数据给到Netty服务端进行处理）。
- (5) 获取好友列表：客户端调用/getAllFriendShip接口，sql语句获取当前用户所有存在好友关系的用户。这里好友关系采用双向校验，即双方都存在正常好友关系才算好友。
- (6) 好友/用户简介：前端封装from\_id与to\_id调用/getRelationWithInfo接口，获取其他用户的详细数据，并封装isFriend字段判断两者是否为好友关系，是则封装friendShip字段记录好友关系详细信息，否则friendShip为空。
- (7) 用户拉黑/取消拉黑：拉黑用户调用/addBlack接口，利用sql语句修改好友关系表中black字段为2，取消拉黑则调用/deleteBlack将black修改回1。
- (8) 好友数据同步：多端登录模式下，对好友数据同步处理，采取分批量拉取好友同步数据，限制每次拉取最大值，根据isCompleted字段判断是否完成同步。
- (9) 获取好友申请：好友申请展示，前端传参userId调用/getFriendRequest接口获取from\_id或to\_id为当前userId的所有好友申请数据，结果返回给前端。
- (10) 好友申请：只有接收方为我方的好友申请需要经过申请，前端封装审批状态数据调用/approveFriendRequest接口对好友申请数据状态进行修改（1为同意，2为拒绝）。

### 5.2.3 应用服务端—组群操作

群聊模块，功能实现接口如表5-3所示。

表5-3 组群操作接口表

表5-3 组群操作接口表续表

- (1) 群查找：调用/searchGroup接口，通过sql查询语句获取所有满足搜索条件的群聊数据，结合群成员数据表封装用户是否为群成员。
- (2) 创建群聊：前端封装创建群聊表单数据，调用/createGroup接口，实现group表中数据新增的同时遍历邀请成员Id列表新增group\_member表数据。
- (3) 群简介：调用/getGroupInfo接口，结合用户数据表、好友关系表、组群表、群成员数据表封装群详细信息返回。
- (4) 群资料编辑：群管理员才有权限修改群信息，调用/updateGroup接口，判断请求用户群角色，非管理员返回错误，否则根据前端传参数据修改群公告、群简介、入群方式、群禁言等群信息，并利用RabbitMq发送通知给群内其他成员。
- (5) 获取已加入群聊列表：客户端调用/getJoinedGroup接口，通过sql语句结合group表以及group\_member表，获取当前用户所加入的所有群聊信息。
- (6) 解散群聊：由群主调用/destroyGroup接口传参groupId，删除group表以及group\_member表中所有关于该群聊数据，利用RabbitMq发送通知给群内其他成员。
- (7) 群主转让：群主选择群内成员进行群主转让，调用/transferGroup接口修改group表该群群主字段以及修改group\_member表中群主与被转让群主成员的群角色role字段（群主为2，管理员为1，普通成员为0）。
- (8) 群聊数据同步：多端登录模式下，对群聊数据同步处理，采取分批量拉取群聊同步数据，限制每次拉取最大值，根据isCompleted字段判断是否完成同步。
- (9) 获取群消息记录：前端传参groupId，调用/getGroupChatMessage查询所有关于该群的消息数据，根据消息类型（图片、文件、文本、语言、视频）进行不同数据封装返回。
- (10) 群头像上传：群管理员修改群头像，调用/uploadLogo接口对群头像字段数据进行修改，同时将头像文件上传至服务器。
- (11) 群成员邀请/申请入群：前端封装群成员Id列表，调用v1/group/member/add接口，先对操作用户做群身份判断，非群成员或者普通群成员根据群设置执行入群逻辑或先发送入群申请，管理员及以上直接执行被邀请人入群逻辑，并利用RabbitMq发送通知给被邀请人以及群管理员。
- (12) 退出群聊/踢出群聊/解散群聊：调用/exit接口，执行退出群聊逻辑，修改group\_member中该成员状态以及删除标志。群主调用/remove接口，执行群删除以及群成员删除逻辑，两表操作采用事物处理。
- (13) 管理员设置/群昵称修改/群成员禁言：根据前端封装群成员修改数据，通过/update接口对群成员角色判断，普通成员只能修改群昵称，群主可以对群内成员数据进行管理员设置，修改group\_member表数据。群主及群管理员调用/speak接口对群内成员执行禁言处理，根据禁言时间判断该成员是否处于禁言状态。
- (14) 群申请/审批：前端传入userId，调用/getGroupRequest获取该用户已加入的所有群聊，获取关于每个群聊的角色信息，如果为管理员就获取该群下所有的群申请信息。否则就获取关于本userId的所有群申请信息。群管理员才有资格审批其他用户的入群申请，前端传入审批人、申请人以及审批状态等信息，调用/approveGroupRequest接口，对群申请表数据进行修改，最后通过RabbitMq发送通知给审批人以及其他群管理员。

### 5.2.4 应用服务端—消息模块

消息模块，功能实现接口如表5-4所示。

表5-4 消息处理接口表

- (1) 获取单聊消息记录：根据from\_id与to\_id获取本用户与好友所有消息记录并根据不同的消息类型进行不同的数据封装。
- (2) 上传消息文件：前端封装文件大小、url、发送方与接收方等信息调用后端/uploadMsgFile处理，将文件上传至服务

器以及mysql对文件消息数据的存储。

(3) 下载文件：调用/downloadFile接口，利用文件IO流将文件从服务端下载到本地。

(4) 管理员消息管理：后台管理系统消息分页展示、系统首页消息数据统计、消息文件分页展示、管理员操作消息删除分别由接口 / getMessageByPage、/getDataOfWeek、/delete、/getMessageFileByPage、/deleteMsgFile实现。

(5) 离线消息同步：多端登录模式下，对用户离线消息进行同步处理，根据Redis缓存sequence对数据库中未同步消息进行拉取，限制每次拉取最大值，根据isCompleted字段判断是否完成同步，每一次同步完成修改Redis中sequence字段记录。

#### 5.2.5 应用服务端—角色管理

系统角色模块，功能实现接口如表5-5所示。

表5-5 角色管理接口表

以上**操作全由管理员处理，只有管理员拥有操作后台系统绑定用户角色的能力**，具体实现主要是批量对用户表中角色字段进行修改。其他功能就是对用户角色表的增删改查。

#### 5.2.6 应用服务端—系统管理

系统管理模块，功能实现接口如表5-6所示。

表5-6 系统管理接口表

表5-6 系统管理接口表续表

系统管理涉及系统参数与系统通知两个**模块**。

(1) **系统参数管理：系统参数是针对管理员设计对系统配置数据进行**可视化管理。具体涉及对系统参数的增删改查。

(2) 用户反馈/用户举报/管理员通知：客户端用户调用/create接口实现用户反馈或者用户举报申请创建，数据保存在sys\_params表中，由管理员查看并调用/handle接口将数据进行处理之后调用/send接口将处理通知反馈给用户，同时管理员也可以进行自行创建面向全体客户端用户的系统公告以及面向某些用户的系统通知。

#### 5.2.7 应用服务端—用户模块

用户模块，功能实现接口如表5-7所示。

表5-8 用户模块接口表

表5-7 用户模块接口表续表

(1) 用户登录/注册：根据前端调用接口/login传过来的数据获取登录类型loginType以及登录方式类型type做出判断及操作。loginType为1表示登录，为2表示注册，type为1表示默认账号密码登录，type为2表示手机号验证登录或注册，type为3表示邮箱验证登录或注册。手机号以及邮箱验证码数据存储在redis用户账号信息存储在MySQL，经过数据校验成功后台生成用户专属签名以及获取zookeeper注册IP及端口返回数据给前端。前端利用typescript语言根据返回的端口与IP建立WebSocket连接并发送登录消息到后台TCP服务由Netty接收消息并处理。

登录以及注册功能实现了QQ邮件验证，实现步骤主要如下：

1) 开启QQ邮箱POP3/IMAP 或者 SMTP/IMAP 服务。

2) 创建工具类MailUtils，调用方法sendMail()参数为发送邮箱以及验证码，当用户点击获取验证码时调用该方法即可。

MailUtils类结构如图5-28所示。

图5-28 MailUtils类图

(2) 管理员登录/登出：管理员用户直接调用/adminLogin接口进行登录操作，根据userId查询用户信息判断用户角色，非管理员无法登录。然后进行账号面膜校验，成功则修改用户状态为在线。登出操作调用/adminLogout接口实现对用户在线数据为离线状态的修改。

(3) 个人中心：调用/getSingleUserInfo接口根据userId获取关于该用户的详细信息数据返回给前端进行回显。

(4) 用户编辑：个人中心实现对用户昵称、地址、个性签名等数据的修改，通过前端封装数据调用/modifyUserInfo接口实现对用户基本数据的修改。

(5) 用户头像更换：用户头像修改独立于用户信息编辑，需要调用/uploadLogo接口利用文件IO实现头像图片数据向服务器地址的上传。

(6) 用户账号管理：主要包括用户密码找回、密码修改、手机号更换与邮箱绑定更换。首先需要进行信息认证，用户通过/getCode选择验证方式获取验证码后调用/getCertify进行手机号/邮箱认证。调用/modifyUserInfo接口对密码进行修改，而邮箱更换与手机号更换要复杂一步，修改之前需要对原信息进行认证。调用/findPassWord接口实现对密码数据的查找。

(7) 后台用户管理：管理员操作。主要包括后台系统通过/getUserByPage接口与/getSingleUserInfo接口获取用户分页数据展示单用户数据编辑回显。调用/modifyUserInfo接口管理员身份直接进行修改而无需进行信息验证。/create接口提供管理员创建管理员账号权限，/changeStatus接口管理员具有启用/禁言客户端用户的权限，如果被禁用客户端用户在线，则发生通知，提醒用户下线处理。/getUserPercentage与/getDataOfWeek分别根据用户角色与创建时间对用户数据进行统计返回数据给前系统首页进行展示处理。

#### 5.2.8 TCP服务端—IM登录/登出

Netty接收到WebSocket发送的登录消息，先对消息进行多端登录校验，如果登录设备系统发生冲突或者是超出多端限制就会进行冲突端口的下线，不冲突的情况下正常进行登录操作。登录操作首先缓存关于该用户的Session信息到redis，在对处理该用户的channel进行静态存储，这样不同用户发送的消息通过标识该用户Session的channel处理。利用RabbitMQ发送登录消息至应用服务端，这里是为了同步MySQL用户在线状态，用时方便后台管理调用应用服务端接口实现对用户实时在线数据统计。当连接建立，应用服务端接收消息处理完登录操作完整结束。对于登出操作，TCP端直接删除redis用户Session缓存，利用RabbitMQ发送登出消息发送至应用服务端改变数据库用户状态即可。

#### 5.2.9 TCP服务端—Netty消息接收

所有基于WebSocket发送的消息，一律由Netty进行处理。在执行消息发送之前利用Feign远程调用了应用服务端接口前置对发送数据的校验，如果校验失败就无需将消息发送给应用服务进行消息存储以及转发，直接由Netty向发送方发回失败响应即可。如果校验成功就利用RabbitMQ发送消息至应用服务端，由应用服务端进行消息存储。当服务端处理完消息TCP服务端

RabbitMQ接收器接收到发送方返回结果响应消息以及接收方接收消息，再由Netty根据发送方与接收方channel分别发出这两条消息至前端WebSocket接收，前端为WebSocket配置消息监听回调，对接收到的消息进行处理展示。

5.2.10 消息可靠性保证

1. RbbitMq保证消息有序、避免遗漏

首先要保证消息有序进入消息队列进行处理，具体的处理思路是为每一为用户创建一个对应的消息处理队列专门负责对发送给该用户的消息进行处理，避免消息混乱的同时保证了消息队列中消息的有序产生与消费。其次对于处理失败的消息，调用channel.basicNack()方法将消息进行重新入队处理，避免产生消息的遗漏。

分别在message模块和group模块中创建RabbitMQ消息接收器ChatOperateReceiver、MediaChatReceiver以及GroupChatOperateReceiver通过onChatMessage()来监听接收TCP服务端发送的消息。TCP服务端RabbitMQ消息接收器数据如表5-8所示。

表5-8 RabbitMQ消息接收器数据表

2. Redis缓存解决消息重复发送

根据消息体中命令对该条消息做出相应的处理。这里以单聊消息为例，为了对用户短时间内对消息进行大量重复发送进行性能优化，每发一条消息先查缓存中是否已有该消息数据，如果存在直接发回消息，不进行MySQL存储减轻数据库压力。如果缓存中不存在该信息，就对信息进行两分钟时效缓存处理。再通过对消息重新封装利用RabbitMQ发送至消息存储模块进行异步消息存储，同时RabbitMQ发送至TCP服务两条消息包括：发回响应数据给发送方、发送消息数据给接收方。

考虑到用户信息安全问题，这里利用Base64工具类对用户消息内容进行了加密处理，存储到数据库中的数据是一个结合了加密字符串进行加密处理过的字符串。当用户进行消息查找的时候在进行消息解密操作。

3. 自定义消息体解决消息粘包与拆包

在RPC框架中，粘包和拆包问题是必须解决一个问题，因为RPC框架中，各个微服务相互之间都是维系了一个TCP长连接，由于微服务往对方发送信息的时候，所有的请求都是使用的同一个连接，这样就会产生粘包和拆包的问题。

产生粘包和拆包问题的主要原因是，操作系统在发送TCP数据的时候，底层会有一个缓冲区，例如1024个字节大小，如果一次请求发送的数据量比较小，没达到缓冲区大小，TCP则会将多个请求合并为同一个请求进行发送，这就形成了粘包问题；如果一次请求发送的数据量比较大，超过了缓冲区大小，TCP就会将其拆分为多次发送，这就是拆包，也就是将一个大的包拆分为多个小包进行发送。粘包、拆包示意图如图5-28所示。

图5-28 粘包、拆包示意图

本文采用的解决办法是通过自定义消息格式规定每次消息的接收范围都为该条消息的长度。思路就是通过自定义私有协议消息内容分为两部分：消息头与消息体，消息中的每个属性由一个int类型的表示数据长度+数据体组成。利用ByteBuf底层API readInt()先读取数据长度，通过readableBytes()获取剩余可读长度，如果大于本次消息体长度就调用readBytes()对数据进行读取（参数为获取到的数据长度）并调用markReaderIndex()将读索引更新，否则就调用resetReaderIndex()将都索引重置回读取数据长度之前，不进行此次读取，等下次数据满足长度再进行读取。这样就解决了消息发送产生的粘包、拆包问题。自定义消息格式如下表5-9所示。

表5-9自定义消息协议Message属性表

5.2.11 WebRTC实现音视频聊天

基本实现流程如图5-29所示。

图5-29 WebRTC音视频通话实现流程图

整体来说就是WebRTC 使用getUserMedia获取摄像头与话筒对应的媒体流对象MediaStream，媒体流可以通过 WebRTC 进行传输，并在多个对等端之间共享。将流对象赋值给视频元素的 srcObject，实现本地播放音视频。

1. 连接管理

主要的数据就是SDP与ICE 候选者。其中双方之间发送的offer以及answer其实就是会话描述信息RTCSessionDescription (SDP)。SDP 是各端的能力，包括音频编解码器类型、传输协议等。这些信息是建立连接是必须传递的，双方知道视频是否支持音频、编码方式是什么，都能通过 SDP 获得。而candidate就是ICE候选者，两个节点交换 ICE 候选来协商他们自己具体如何连接，一旦两端同意了一个互相兼容的候选，该候选的 SDP 就被用来创建并打开一个连接，通过该连接媒体流就开始运转。

5. 基于Netty与Spring Boot的高性能即时聊天软件设计与实现_第5部分			总字数：3465
相似文献列表			
去除本人文献复制比：1.5%(51) 去除引用文献复制比：1.5%(51) 文字复制比：1.5%(51) 疑似剽窃观点：(0)			
1	基于FDS的套型房屋家具摆放 对火灾发展及安全疏散影响研究	夏雪健 - 《大学生论文联合比对库》 - 2017-06-02	1.0% (33) 是否引证：否
2	基于FDS的套型房屋家具摆放对火灾发展及安全疏散影响研究	夏雪健 - 《大学生论文联合比对库》 - 2017-06-05	1.0% (33) 是否引证：否
3	对外汉语教材中的中国形象研究	陈宇玲(导师：何平;邱世鸿) - 《东南大学硕士论文》 - 2021-06-01	0.9% (32) 是否引证：否
原文内容			



## 2. 信令服务器

WebRTC 的 SDP 和 ICE 信息需要依赖信令服务器进行消息传输与交换、建立 P2P 连接，之后才能进行音视频通话、传输文本信息。如果没有信令服务器，WebRTC 无法进行通信。本文是基于WebSocket与Netty实现信令服务器对SDP与ICE进行数据封装为消息进行接收与发送。

## 3. 实现思路

发送方需要先创建本地视频流，创建并初始化RTCPeerConnection调用createOffer()设置本地描述同时调用onicecandidate()将offer与candidate发送给接收方。接收方收到发送方音视频请求先创建本地视频流以及初始化RTCPeerConnection。当接收方接收对方请求时将接收到的offer与candidate分别调用setRemoteDescription()与onicecandidate()进行远程视频流配置，同时将答复数据answer以及candidate发回给发送方。发送方收到接收方的答复answer以及candidate表示对方接收通话请求，并也调用setRemoteDescription()与onicecandidate()设置远程接收方视频流，这样双方之间就通过RTCPeerConnection完成了数据流交互，实现音视频通话功能。

## 6 系统测试

### 6.1 测试环境以及测试条件

本系统在个人电脑上基于Windows操作系统完成，测试需要两台以上客户端，一台服务器。服务器需要安装JDK1.8版本、Redis、MySQL、Zookeeper以及RabbitMQ。浏览器支持WebSocket，客户端带有能够支持音视频的摄像头以及麦克风设备，以便进行音视频效果展示。

### 6.2 模块功能测试

#### 6.2.1 登录与注册

(1) 客户端登录，登录成功跳转至数据加载界面后跳转至聊天主页面。如图6-1所示。

图6-1 客户端登录

(2) 客户端注册，获取验证码信息完成账号密码设置后，可直接进行登录操作（注意，这里验证码失效为三分钟，超时验证码失效）。如图6-2所示。

图6-2 客户端注册

#### 6.2.2 客户端个人设置

(1) 个人信息修改，个人中心界面进行头像修改以及个人信息修改。如图6-3所示。

图6-3 客户端个人信息修改测试图

(2) 密码找回，获取验证码信息完成账号密码设置后，可直接进行登录操作（注意，这里验证码失效为三分钟，超时验证码失效）。如图6-4所示。

图6-4 客户端个人账号找回密码测试结果图

(3) 密码修改，选择修改认证方式，获取验证码并设置新密码，验证通过完成密码修改。如图6-5所示。

图6-5 客户端个人账号修改密码测试图

(4) 更换绑定信息，这里以绑定邮箱为例，对邮箱进行验证设置操作。如图6-6所示。

图6-6 客户端个人账号邮箱绑定测试图

#### 6.2.3 客户端消息收发

(1) 文本、表情消息发送，双方进入聊天会话页面，在文本输入文本或者表情进行交流。如图6-7所示。

图6-7 客户端文本、表情消息发送测试图

(2) 图片文件消息发送，双方进入聊天会话页面，选择本地文件或者图片发送给对方，点击图片进行预览，点击文件进行下载或者预览操作。如图6-8所示。

图6-8 客户端图片预览、文件发送测试图

(3) 语音通话，点击语音通话按钮，请求与对方进行语音通话，对方接受，双方通讯成功。如图6-9所示。

图6-9 客户端单聊语言聊天测试图

(4) 视频通话，点击视频通话按钮，请求与对方进行视频通话，对方接受，双方通讯成功。如图6-10所示。

图6-10 客户端单聊视频聊天测试图

## 6.3 测试总结

测试总结如表6-1所示。

表6-1 测试总结表

结论

本文完成了基于Spring Boot与Netty框架对一个即时聊天软件基本功能的实现。虽然目前国内对于即时聊天系统的开发技术已经趋于完备，不同的开发语言使一款聊天软件具有不同的特色，例如基于C/C++、Python、Go、Android等语言都可以实现一个较好性能的聊天软件。当然也有很多利用Java语言进行开发的聊天软件取得了不错的效果，比如利用JMS(Java消息服务)技术实现的即时系统系统、基于JMF(Java媒体框架)实现的音视频聊天系统等。整体来看像这类通信系统的实现，基本离不开Socket技术，Java语言也是如此。本文在技术方面还是利用了主流的Spring Boot框架，核心点在于利用Netty框架以及使用TCP协议来实现更加安全高效的网络通讯，创新点在于利用了Zookeeper、RabbitMQ等第三方的微服务组件为软件的并发能力做出提升，同时利用了WebRTC技术支持软件实现音视频聊天功能。

虽然软件的整体完成度还是可以的，但其实一款高效强大的即时聊天软件还有很多细节以及功能方面的设计需要考虑。相比之下，本款软件可以继续深究完善的地方：

1. 页面信息细节展示、多端消息同步、消息未/已读处理、多用户之间操作同步等方面都可以做出进一步完善，
2. 经过测试发现WebRTC技术虽实现了音视频功能，但其实效果一般，可以在此基础上进行代码加强做出一定的通讯效果优化。

3. 目前还未实现群聊多用户同时音视频通话功能。

4. 后续功能的拓展可以考虑进行更多合理的组件整合以及设计更加合理的数据库结构等来进一步减轻数据库压力以及服务器资源的消耗。

最后,即时通讯作为现代生活中不可或缺的部分,随着科技创新,无论是通讯硬件技术对时延的缩减、通讯网络软件化以及AI的加入更加智能化,未来的即时通讯将会是更加安全稳健且高效的。

#### 参考文献

- [1] 张艳芳. 基于Web的企业即时通讯系统的设计与实现[D]. 河北科技大学, 2018.
- [2] 杜豪杰. 移动医疗平台即时通讯系统设计与实现[D]. 华中科技大学, 2015.
- [3] 张龙. 聊天系统的设计与实现[D]. 大连理工大学, 2015.
- [4] 卞光宇. 移动终端跨平台组件扩展技术研究与实现[D]. 北京邮电大学, 2018.
- [5] 王佳宁. 辽宁省广播电视节目监管系统设计与实现[D]. 东北大学, 2015.
- [6] 景科华. 基于Hadoop集群的运维管理系统设计与实现[D]. 西安电子科技大学, 2021.
- [7] 陈文敏. 基于微服务的教学管理云平台设计[D]. 西安电子科技大学, 2020.
- [8] 张经纬. 基于SpringBoot的高校毕业生离校系统设计与实现[D]. 西安电子科技大学, 2020.
- [9] 杨懂懂. HTTP接口自动化测试系统设计与实现[D]. 武汉邮电科学研究院, 2021.
- [10] 宋瑞栋. 轨道交通电气设备监测系统的设计与实现[D]. 北京交通大学, 2020.
- [11] 王鹏. 基于.NET框架下即时通信系统的设计与实现[D]. 哈尔滨工程大学, 2007.
- [12] 廖俊锋, 朱晓光, 周文端. 基于TCP的微波回程链路自适应优化技术研究[J]. 移动通信, 2014, 38(14): 32-36.
- [13] 袁双双. 基于RPC框架的云视频会议控制微服务设计与实现[D]. 东南大学, 2021.
- [14] 卢国宝, 陈锡爱, 胡群等. 基于NIO网络模型远程实验系统设计[J]. 中国计量大学学报, 2021, 32(04): 555-560.
- [15] 李林峰. 分布式服务框架原理与实践[M]. 北京: 电子工业出版社, 2016.
- [16] 石瑞芳. 基于区块链技术的智能合约在东北粮网中的应用研究[D]. 沈阳师范大学, 2019.
- [17] 石文浩. 软件定义架构下支持能量交换的通信协议设计与实现[D]. 北京邮电大学, 2018.
- [18] 黄智伟. 基于生成式对抗网络的视频人物唇形修改系统的设计与实现[D]. 北京邮电大学, 2021.
- [19] 潘风祥. 基于Netty的高性能异构服务间网络代理中间件的设计与实现[D]. 山东科技大学, 2019.
- [20] 熊滴, 赵红帅, 宫新飞等. 基于JAVA语言的聊天软件的设计与实现[J]. 中国科技信息, 2013. 473(12): 100+109.
- [21] 叶为正, 林声肯, 黄立轩等. 即时通讯系统的设计与实现[J]. 计算机技术与发展, 2020, 30(02): 216-220.
- [22] 张艳军, 王剑, 叶晓平等. 基于Netty框架的高性能RPC通信系统的设计与实现[J]. 工业控制计算机, 2016, 29(05): 11-12+15.
- [23] 傅瑶. Java即时通讯系统程序的设计与实现[D]. 吉林大学, 2015.
- [24] 李元君. 基于Java的即时通讯系统的设计与实现[D]. 山东大学, 2013.
- [25] 王全彬. 基于Java的网络即时通讯系统的设计与实现[D]. 电子科技大学, 2007.
- [26] 陈绘宇. 局域网的即时通信软件设计与实现[J]. 中国新通信, 2019, 21(11): 2.
- [27] 陈衍席. 基于Java网络聊天系统的设计及实现研究[J]. 通讯世界, 2016(07).
- [28] 陈盼, 黄辰. 基于Java的网络聊天系统的设计与实现[J]. 物联网技术, 2016, 6(01): 94-95.
- [29] 任琴, 孔令慧, 秦冰. 基于JAVA的多人聊天室的系统设计[J]. 时代农机, 2018, 45(01): 142.
- [30] 徐伟. 基于Java语言的在线实时聊天系统的设计与实现[J]. 数字技术与应用, 2011(05): 12+105.
- [31] 李丹, 张师毅. 基于JAVA的仿QQ聊天系统的设计[J]. 微型机与应用, 2013, 32(24): 11-13.
- [32] 向波. 基于Java的即时通讯软件的设计与实现[D]. 电子科技大学, 2007.
- [33] 霍芥廷. 即时通讯工具现状及前景展望[J]. 通讯世界, 2018, No. 332(01): 155.
- [34] 张云川. 即时通信的企业级应用研究[J]. 企业经济, 2005(06): 96-97.
- [35] 万建民. 基于Netty和Redis应对高并发场景的研究和实现[D]. 南京邮电大学, 2022.
- [36] 王鑫, 胡华昌, 许智鸿等. 基于Netty的物联网系统通信框架的实现[J]. 信息与电脑(理论版), 2020, 32(11): 177-179.
- [37] 张晗, 郭勇, 徐蕾等. 基于Java的即时通讯系统的设计研究[J]. 信息通信, 2014.
- [38] 潘进学. 基于JMS的企业即时通讯系统的设计[J]. 中国新通信, 2014, 16(15): 107-108.
- [39] 任汉秋. 基于Netty多租户的企业即时通讯系统的设计与实现[D]. 哈尔滨工业大学, 2019.
- [40] 宋伶俐. 网络即时通信系统的分析与设计[D]. 山东大学, 2010.
- [41] 王利民. 网络即时通信系统[J]. 电脑编程技巧与维护, 2008, No. 173(11): 55-57+64.
- [42] 查骏. 基于NIO的远程调用框架的设计与实现[D]. 复旦大学, 2012.
- [43] 赵将. Java语言在计算机软件开发中的应用[J]. 数字技术与应用, 2023, 41(03): 160-162.
- [44] 谭少华. Java Web管理信息系统快速开发平台的设计与实现[D]. 电子科技大学, 2018.
- [45] 王焕家. 基于Web的企业内部即时通讯系统的设计与实现[D]. 浙江大学, 2021.
- [46] 严梅. 基于Java网络聊天工具设计研究[J]. 信息记录材料, 2019, 20(11): 194-195.
- [47] 杜凤旭. 基于WebSocket协议的在线聊天系统设计与实现[D]. 东南大学, 2016.
- [48] 余春贵. 企业级即时通讯系统设计与实现[D]. 华南理工大学, 2018.
- [49] 杜瑞庆, 李一诺. 基于Java语言的即时通讯系统设计与实现[J]. 电脑知识与技术, 2022, 18(31): 29-32.
- [50] 曾照成. 基于WebRTC的视频会议系统的设计与实现[D]. 湖南:中南林业科技大学, 2016.

#### 致谢

经过几个月的努力,从完成毕业设计到结束论文写作的整个过程都让我受益匪浅。从遇到问题,到最后解决问题,交出大学生涯的最后一份答卷,我收获良多。在此我要特别感谢我的指导老师邝祝芳教授,感谢老师帮助指导让我顺利完成毕业成为

一名合格的大学生。从毕业论文的选题开始，老师都会认真耐心地与我们进行沟通，尊重我们的想法同时对此做出专业的指导以及合理的建议。导师每周都会抽出时间关注我们的进度，对于我们在毕业设计实现以及论文的撰写过程中提出的难点和问题，导师都会一一为我们分析指导并指出解决办法，不仅如此老师还会亲切的为我们缓解毕业期间带来的压力与焦虑。邝祝芳教授认真严谨的作风，待人真诚为人亲和的态度是我今后工作生活的榜样，渊博的学识是我往后继续努力前进的方向，再次郑重感谢邝祝芳教授的悉心指导与关照。

光阴如梭，从2019年9月6日踏进中南林业科技大学开启我的大学求学之旅开始到现在不知不觉四年时间已过，我的大学之路也将步入尾声。回首过往四年的学习时光，努力保证专业课程的学习之余花点时间学习前端到后面坚定选择Java方向，我已经不再是当初那个对计算机一无所知的小白了。在此我要感谢中南林业科技大学给我提供了优秀的教师资源、良好的学习环境。感谢大学期间我的所有任课老师的悉心教导，你们认真负责的教学态度，扎实的专业能力、真诚友善的待人方式都是熏陶着我得以成长和蜕变的关键。感谢我的辅导员黄昕老师，待人亲切真诚，设身处地为我们考虑，无论是生活还是学习中遇到的困难都会想办法帮我们处理。感谢我的班主任陈爱斌教授以及我的班级，班主任为人亲和，悉心指导带领班级进步，班干部们认真负责，为班级成长出谋划策付出良多时间与精力，班级同学之间共同搭建的良好学习氛围、友好的同学关系、学习和生活中互帮互助，收获知识的同时更多的是收获到真挚的友谊是我大学生涯最直观的感受。还要感谢我最好的朋友胡柔柔，同一所大学我们一直互相关照，遇到挫折总是有你在身边陪伴我，给我鼓励和安慰，一切尽在不言中，大学结束开启的是人生的新篇章，忠心地祝愿你前程似锦。感谢我的家人，是家人在我背后的支撑让我有权利往更好的方向成长，你们的鼓励与支持是我前进最大的动力。

我的论文水平可能不够优秀，还有很多不足与进步的空间。但是完成整个毕业设计论文的编写过程让我终生受益。激发自己的潜能，充分利用自己的专业知识，研究探索问题解决，认真严谨的做好每一个细节，自我满意地交出最后一份答卷，使我能够对得起自己的付出，所以我要感谢我自己。未来也会保持求学态度，塑造成为更好的自己。

最后，感谢所有对我论文给与支持和帮助的人，感谢参与本文审阅以及指导答辩的各位老师、专家及教授，谢谢大家的付出！

指 标
疑似剽窃文字表述

1. 提供了优秀的教师资源、良好的学习环境。感谢大学期间我的所有任课老师的悉心教导，你们认真负责的教学态度，

- 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. 去除本人文献复制比：去除作者本人文献后，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. 复制比：按照“四舍五入”规则，保留1位小数
6. 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
7. 红色文字表示文字复制部分；绿色文字表示引用部分（包括系统自动识别为引用的部分）；棕灰色文字表示系统依据作者姓名识别的本人其他文献部分
8. 本报告单仅对您所选择的比对时间范围、资源范围内的检测结果负责



 [amlc@cnki.net](mailto:amlc@cnki.net)

 <https://check.cnki.net/>