

# 高等教育自学考试

**毕业论文（设计）**

**基于Spring Cloud的社交聊天系统的**

**设计与实现**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **办学单位：** | **华南理工大学** | | | | | |
| **届** ●**专业：** | **2022届计算机科学与技术** | | | | | |
| **学 生：** | **孙鑫 （010621102593）** | | | | | |
| **指导教师：** | **王亮明 中级** | | | | | |
| **提交日期：** | **2023** | **年** | **3** | **月** | **1** | **日** |

# 

# 摘 要

随着信息技术的不断发展，互联网得以广泛应用，人们的社交关系逐渐由线下转向线上，如何构建一套即时通讯的陌生人社交聊天系统成为当下的热点。在这种背景之下，开发一套简易的，标准的社交聊天系统具有重大意义。其功能至少需要囊括用户注册登录、聊天、建立关系、发布动态等。基于以上要点，结合当下流行、完善、社区活跃的Spring Cloud框架，设计并实现一套使用Java语言开发的社交聊天系统。

根据社交的基本需求，对整个系统的功能性需求，非功能性需求进行分析。基于需求的分析，对整体前端、后端、数据库做整体架构设计，前端设计，后端设计，前后端接口设计，数据库设计。再依据设计，利用开发工具对系统前后端、数据库进行开发工作，将设计的功能一一实现，再将开发完成的程序打包部署。最后对系统关键功能进行演示，对整个系统做界面测试，功能测试和性能测试。

通过对社交聊天系统的测试，测试结果表示系统满足基本的功能需求，确保各个功能模块的可用性和稳定性。该系统可以为用户提供一个安全的，稳定的社交环境，对互联网产业起到促进作用。

**关键词：**陌生人社交；即时通讯；Spring Cloud

# Abstract

With the continuous development of information technology, the internet has been widely used, and people’s social relations have shifted from offline to online gradually. How to build a social chat system for strangers with instant messaging has become a hot topic at present. In this context, it is of great significance to develop a simple and standard social chat system. Its functions need to include at least user registration and login, chat, relationship building, post blog, etc. Based on the above points, combined with the current popular, perfect and active community Spring Cloud framework, design and implement a series of social chat system developed in Java language.

According to the basic needs of social interaction, analyze the functional requirements and non-functional requirements of the entire system. Based on the analysis of requirements, do the overall architecture design of the overall front-end, back-end, and database, front-end design, back-end design, front-end and back-end interface design, and database design. Then according to the design, develop the front-end and back-end of the system and the database by the development tools, realize the designed functions one by one, and then package and deploy the developed program. Finally, the key functions of the system are demonstrated, and the interface test, function test and performance test are performed on the whole system.

Through the test of the social chat system, the test results show that the system meets the basic functional requirements and ensures the usability and stability of each functional module. The system can provide users with a safe and stable social environment and promote the Internet industry.

**Key Words:** Social for strangers；instant messaging；Spring Cloud

**目 录**

[摘 要 I](#_Toc7855)

[Abstract II](#_Toc4971)

[第一章 绪论 1](#_Toc797)

[1.1研究的背景和意义 1](#_Toc394)

[1.1.1研究背景 1](#_Toc14377)

[1.1.2研究的意义 2](#_Toc28494)

[1.2绿色建筑国内外研究现状 2](#_Toc9925)

[1.2.1国外研究现状 2](#_Toc24537)

[1.2.2国内研究现状 2](#_Toc29604)

[1.3研究内容与方法 3](#_Toc8929)

[1.3.1研究内容 3](#_Toc7744)

[1.3.2研究方法 3](#_Toc30220)

[第二章 绿色建筑相关概念界定与发展现状分析 5](#_Toc8542)

[2.1绿色建筑概念 5](#_Toc5628)

[2.2绿色建筑评价体系 5](#_Toc26707)

[2.3广东地区绿色建筑发展现状分析 5](#_Toc25830)

[2.3.1绿色建筑发展概况 5](#_Toc18394)

[2.3.2绿色建筑设计标识项目概况 5](#_Toc15012)

[2.3.3广东绿色建筑设计项目现状问题思考 6](#_Toc24130)

[2.4 本章小结 6](#_Toc21160)

[第三章 绿色建筑理念下住宅建筑设计技术与策略建议 7](#_Toc7721)

[3.1绿色技术建议 7](#_Toc11834)

[3.1.1广东地区绿色住宅建设特点分析 7](#_Toc8248)

[3.1.2广东地区绿色住宅可用绿色技术建议 7](#_Toc13793)

[3.2绿色住宅建筑设计策略 7](#_Toc31594)

[3.2.1优化空间布局 7](#_Toc31473)

[3.2.2户型具体设计及标准层优化 7](#_Toc8424)

[3.2.3统筹考虑住宅全寿命周期内设计 7](#_Toc17032)

[3.3本章小结 7](#_Toc24461)

[第四章 基于《绿色建筑评价标准》二星住宅设计实例分析 8](#_Toc3462)

[4.1项目概况 8](#_Toc5237)

[4.2设计理念 8](#_Toc11592)

[4.3设计方法 8](#_Toc31068)

[4.3.1充分回应环境与气候 8](#_Toc13099)

[4.3.2设计手法上回归理性、简约 8](#_Toc18063)

[4.4基于《绿色建筑评价标准》设计要点 8](#_Toc14619)

[4.4.1节地与室外环境 8](#_Toc19899)

[4.4.2节能与能源利用 8](#_Toc13252)

[4.4.3节水与水资源利用 9](#_Toc7136)

[4.4.4节材与材料资源利用 9](#_Toc17178)

[4.4.5室内环境质量 9](#_Toc25646)

[4.5 本章小结 9](#_Toc23637)

[结 论 10](#_Toc17062)

[参考文献 11](#_Toc16532)

[致 谢 13](#_Toc7325)

（另起页：目录范例）

（标题：小二号黑体加粗，居中，1.5倍行距，两字之间空2字符，段前、段后各１行，目录为电脑自动生成）

（各章标题、结论、参考文献、附录、致谢：四号黑体，居左，1.5倍行距；各次级标题及题序：小四号宋体，1.5倍行距）

# 

# 第一章 绪论

### 1.1研究的背景

随着技术的不断发展，移动终端的普及，几乎人人都有一台自己的移动终端。而社交离不开沟通，人们的社交习惯已经从线下口头交流逐步走到线上打字交流。无论在生活还是在工作中，社交软件已成为人们重要的沟通工具，一些陌生人社交软件也是人们扩宽社交面，满足社交需求的工具。

而同时一方面随着互联网产业不断扩大，寡头垄断了熟人社交软件，另一方面人们对于隐私的重视程度日益增长，人们开始抵制寡头对个人信息的控制，这便使得重视隐私数据的小公司的产品得以分一杯羹。并且人们也需要多元化的社交软件满足不同属性的社交需求。

由此看来，一系列新的陌生人社交软件成为众望所归。此系统的研究与实现具有广泛的现实意义与应用价值。用户通过移动设备发布自己的生活动态，获得志同道合的人关注，通过聊天交流加深认识，从而与陌生人进行互动，建立社交关系，满足日常社交需求。

## 1.2研究的意义

为了满足人们对陌生人社交的需求，提供高效的安全的聊天环境，本课题开发的社交聊天系统可以有效地满足这类需求。通过本系统，用户可以认识到更多的陌生人，打破陌生人之间的社交壁垒，提高了用户的使用体验，降低了人与人的信任成本，促进了互联网产业的发展。

### 1.3研究内容

通过各种学术期刊研究文献的收集与阅读，以及社会实践相结合，分析社交聊天系统的背景与意义分析，因此针对社交关系复杂，即时通讯难构建的问题，本文研究了基于 Spring Cloud的社交聊天系统如何设计与实现，具体研究如下:

首先确里在系统中的所使用的各种技术框架的使用，其中以Spring Cloud作为后端 Web框架，使用Redis, Neo4j, Rabbit MQ为辅助的技术路线，利用软件工程中所掌握的技能，首先对整体系统进行需求分析，确定各功能模块的需求，其次进行详细的系统架构设计，前后端交互设计、API接口设计、数据库设计等。完成整体设计后，进行项目框架的搭建，各功能模块的代码实现，最后对系统进行测试，实现最终的社交聊天系统。

### 1.4研究方法

通过文献归纳与社会实践相结合的方法，通过对国内各种学术期刊等研究文献的收集与研读，以市面上前沿的技术为前提，构建一套实用的架构。

# 第二章 相关理论与关键技术

## 2.1 Spring-Cloud微服务

微服务的概念是由Martin Fowler(马丁·福勒)在2014年提出的，它是一种新兴的软件架构模式。传统的web服务架构，由无数个工程师共同开发一个项目，当业务愈发庞大，代码量也变得巨大，似的开发变得沉重。而微服务把一个大型的单体应用或服务拆分为多个支持微服务。每个微服务按照各自的功能划分开来，形成由多个责任单一的服务共同组成的一套服务。

Spring Cloud是java生态里微服务架构的实现之一，它利用Spring Boot开发便利性巧妙地简化了分布式系统基础设施的开发，它的组成部分有服务发现注册器Spring-Cloud-Eureka、配置中心Spring-Boot-Config、消息总线Spring-Boot-Bus、负载均衡Ribbon、断路器Hystrix、数据监控Spring-Boot-Health等。

## 2.1.1 注册中心Spring-Cloud-Eureka

Spring Cloud Eureka是对Netflix Eureka的封装以实现服务发现功能，它包含了Server 端和Client端。Eureka Server 提供服务注册功能，各个节点启动后，会向Eureka Server发送注册请求。并获取其他服务的信息，此后每隔一段时间向Eureka Server发送心跳包更新健康状态，同时更新其他服务的信息。过程中若有其他服务接入，则要等到下一次心跳便可获取到新的服务信息。

## 2.1.2 服务网关Spring-Cloud-Gateway

网关作为系统的唯一流量入口，封装内部系统的架构，所有请求都先经过网关，由网关将请求路由到合适的微服务,常见的有 Zuul、Spring-Cloud-Gateway、Kong。Spring-Cloud-Gateway因为使用异步非阻塞IO模型，并且支持异步，故而性能要高于其他网关实现。

## 2.2 Netty通讯模型

Netty是一个异步的、基于事件驱动的网络应用框架，用以快速开发高性能、高可靠性的网络IO程序。简化和流程化了[NIO](https://so.csdn.net/so/search?q=NIO&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/qq_36389060/article/details/_blank)的开发过程。Java NIO模型是同步非阻塞的，服务器实现模式为一个线程处理多个请求(连接)，即客户端发送的连接请求都会注册到多路复用器上，多路复用器轮询到连接有I/O请求就进行处理。



图 2-1 NIO模型

## 2.3 Redis缓存

Redis 是一款开源的、网络调用的、基于内存的、原子性的、可持久化的Key-Value存储系统。它依据Key将数据结构存储在内存中，能够存储的数据结构有String, List(ziplist), Hash(Key-Value), Set(Hash的变种，没有value), ZSet(Value-Score, 根据Score排序的Set)等。

Redis的key使用快表排序，快表是一种接近于二叉树的数据结构，时间复杂度是O(log2N)。

## 

## 2.4 Neo4j图数据库

[Neo4j](https://baike.baidu.com/item/Neo4j" \t "/Users/robertsun/Documents\\x/_blank)是一个高性能的，NOSQL图形数据库，其主要数据结构为图，图数据由节点，关系组成。Neo4j中，节点和关系可以有类型，属性，因此可以进行面向对象的。Neo4j也可以被看作是一个高性能的图引擎，该引擎具有成熟数据库的所有特性。

由于图数据结构比较接近于社交网络的抽象意义，因此使用Neo4j来存储好友关系。

## 2.5 Rabbit MQ消息队列

消息队列是一种存储中间件，它将消息存起来，再进行消费，队列具有FIFO（先进先出）的特点。消息队列可以将消息的生产者和消息的消费者隔离开来，降低耦合性。

Rabbit MQ是一个在AMQP（Advanced Message Queuing Protocol）基础上实现的，可复用的企业消息系统。它使用Erlang开发的，Erlang是一款面向并发的编程语言，设计之初用于网络通信，因此Rabbit MQ特别适应网络通信。

## 2.6 本章小结

本章简要阐述了该系统所选择的技术组成，后端使用 Spring Cloud作为 Web架构，主要用到了Spring-Cloud-Eureka作为注册中心，Spring-Cloud-Gateway作为服务网关，使用netty作为即时通讯服务。使用Redis作为缓存数据库，Neo4j用于处理好友关系，Rabbit MQ作为消息队列，便于接下来的系统设计与实现。

# 第三章 聊天系统需求分析与概要设计

## 3.1业务需求分析

一个社交聊天系统至少具有用户登录功能，社交关系功能，日常动态发布功能，好友间聊天功能，内容审核功能，以及其他非功能性需求。

### 3.1.1登录鉴权的需求分析

用户登录需要能够支持短信登录，密码登录。用户的权限包括普通用户和会员用户。为了记录用户具有的权限，需要对用户进行一些标记，并且在维持登录状态的同时体现出来。表3-1是用户短信注册用例文档：

表 3-1 用户短信注册

|  |  |
| --- | --- |
| 用例名称 | 用户短信注册 |
| 执行者 | 未注册用户 |
| 前置条件 | 所用手机号未在系统中注册 |
| 后置条件 | 填写用户信息 |
| 基本路径 | 1. 点击登录页面的注册按钮 2. 输入手机号，点击获取验证码 3. 输入验证码 4. 成功后进入用户信息界面 |

表 3-2 用户短信登录

|  |  |
| --- | --- |
| 用例名称 | 用户短信注册 |
| 执行者 | 已注册用户 |
| 前置条件 | 所用手机号已在系统中注册 |
| 后置条件 | 进入主页 |
| 基本路径 | 1. 点击登录页面的注册按钮 2. 输入手机号，点击获取验证码 3. 输入验证码 |

### 3.1.1用户及好友关系的需求分析

在陌生人社交中，应允许单边好友关系，也有双边好友关系，所以会存在以下情况:

1. 无关系
2. 单向关注
3. 双向关注

表 3-3 用户关注/取消关注

|  |  |
| --- | --- |
| 用例名称 | 用户关注/取消关注 |
| 执行者 | 注册用户 |
| 前置条件 | 用户未关注/已关注对方，对方未拉黑用户 |
| 后置条件 | 开始聊天 |
| 基本路径 | 1. 点开目标用户主页； 2. 点击“关注”/“已关注”按钮； |

### 3.1.1用户动态的需求分析

用户动态不尽作为日常生活的分享，也是一种社交名片，需要被展示给好友及陌生人查看。动态的内容通常包括文字，图片，视频。当用户发布一条动态，动态有一定几率会发布到公共区域让陌生人看到。当用户的粉丝进入关注人动态，返回所有关注人的动态给他。

表 3-5 用户发布动态

|  |  |
| --- | --- |
| 用例名称 | 用户发布动态 |
| 执行者 | 注册用户 |
| 前置条件 | 用户已注册且正常使用 |
| 后置条件 | 开放动态给粉丝或陌生人查看 |
| 基本路径 | 1. 点开动态页面右上角“发布动态”； 2. 输入文字内容，选择最多8张图片或一段视频 3. 点击“发布”按钮； |

### 3.1.1聊天模块的需求分析

在社交聊天系统中，聊天是最重要的一块。聊天是一种互动，具有很强的交互性，实时性要求比较高，当用户A与用户B发送消息，消息因在很短的时间内从A传输到B，所以需要接入实时通讯。过程如下：

1. 客户A向服务器建立连接
2. 客户A利用连接信道向服务器发送消息
3. 服务器找到与客户B的连接，向客户B主动推送客户A发送的消息
4. 如果找不到客户B的连接，则需要将数据存储到数据库中
5. 如果客户B在消息产生时不在连接状态，等到下一次连接到服务器时，服务器主动推送所有未读消息

### 3.1.1内容审核的需求分析

作为一款用户年龄跨度较大，人群受教育水平参差不齐的聊天系统，用户发布的内容动态、头像、昵称、个性签名具有较大的曝光量，其内容可能会存在不合时宜的情况，所以需要加以内容审查。随着人力成本上升，自动化必将取代人工。在系统中，有来自不同功能的内容需要审核，如何设计一套通用的审核模块是一个重点。

在实际的业务系统中，需要审核的内容有文字，图片，视频及其组合，因而需要将文字、图片、视频作为参数传给审核模块，由审核模块给出最终的结果。

## 3.2非功能性需求分析

作为一个To C的产品，需要具备较高的性能还有稳定性，可以给终端用户提供较为舒适的体验。

性能对于计算机系统来说是一项重要的评价指标，从终端用户的角度看，一款APP必须具备以下指标:

1. 接口响应时间。对于终端使用客户来说，操作等待时间不宜太长。一般来说，高频调用的接口响应时长最长不能超过1秒，0.5秒以下无感，0.3秒和0.5秒相差不大；低频接口响应时长最长不能超过3秒。较长的等待时间会降低用户体验，造成用户流失。
2. 稳定性。稳定性是一个系统在运行过程中是否能够正常地、持续地为用户提供服务，不会出现APP闪退、服务端持续不响应等异常情况。这类情况也会降低用户体验，造成用户流失。因此本系统要求网络正常的情况，在服务器的承载能力以内不出现错误与无响应。

## 3.3本章小结

本章使用用例文档详细阐述功能性需求分析，并且为确保系统的搞笑与稳定，还对非功能性需求作分析，为接下来的系统设计作骨架支撑。

# 第四章 社交聊天系统的设计

## 4.1微服务架构设计

本论文中的社交聊天系统使用了Spring Cloud中的注册中心和服务网关，组织架构上，由网关调用其他各个服务，为了减少服务之间的调用引起的IO损耗，服务之间避免使用服务调用，并且连接到同一台数据库进行数据共享。该模式最大的优势是极致的高效运行。

本论文设计的移动医疗系统采用 C/S 结构，C代表移动用户终端，S代表服务器。移动端的原声应用程序具有较高的性能，服务器具备强大的计算能力。结合两者，充分使整个系统得以高效运行。图4-1 是系统架构图：

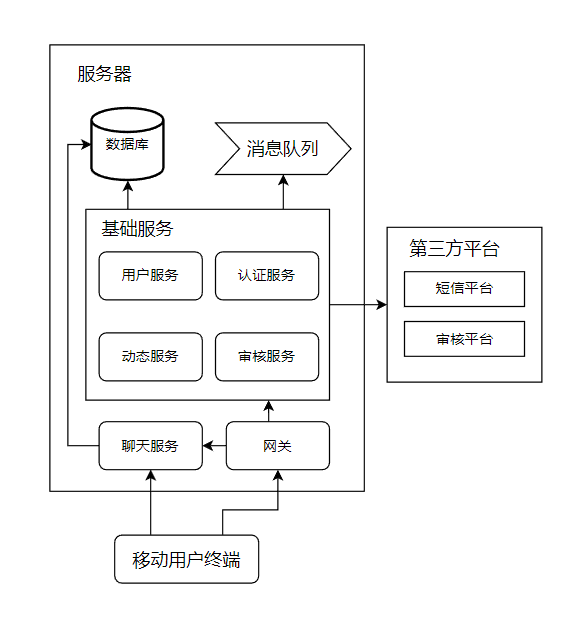


图 4-1 系统架构图

## 4.2前端设计

本论文的社交聊天系统前端设计着重于引导用户发布动态和聊天功能，使用精美、简明的视觉设计，重视用户的体验以及人为直觉的操作。目标在于使用户可以凭借直觉自由顺畅地使用APP。

用户体验：(1)用户提供手机号获取短信验证码，一旦匹配成功，则自动进入登录，如果客户没有在平台注册，则自动完成注册动作，并通知前端进入信息填写界面，用户无需完成注册。(2)查看和发布动态相关信息，用户通过点击中间的万能按钮，弹出所有需要的功能，就像呼唤助手一样，平时不用的时候可以收纳起来，不会占用过多的屏幕控件。(3)用户聊天，用户点进目标用户主页，看到对方的动态，在有一定了解的前提下，点击关注才能给对方发送消息，可以增进聊天的内容。

本系统前端主要包含以下内容:

1. 注册/登录页面：通过手机号和短信验证码或者手机号和密码方式登录，如果使用手机号和短信验证码登录的用户尚未注册，则自动进行注册后自动登录。
2. 动态广场/关注人动态：查看动态发布人，发布人标签，动态内容。
3. 聊天页面：对方头像，历史聊天记录，表情包，上传图片按钮。

图4-2是本论文系统的界面关系图，主要介绍各系统界面、界面实现的功能以及界面之间的关系。

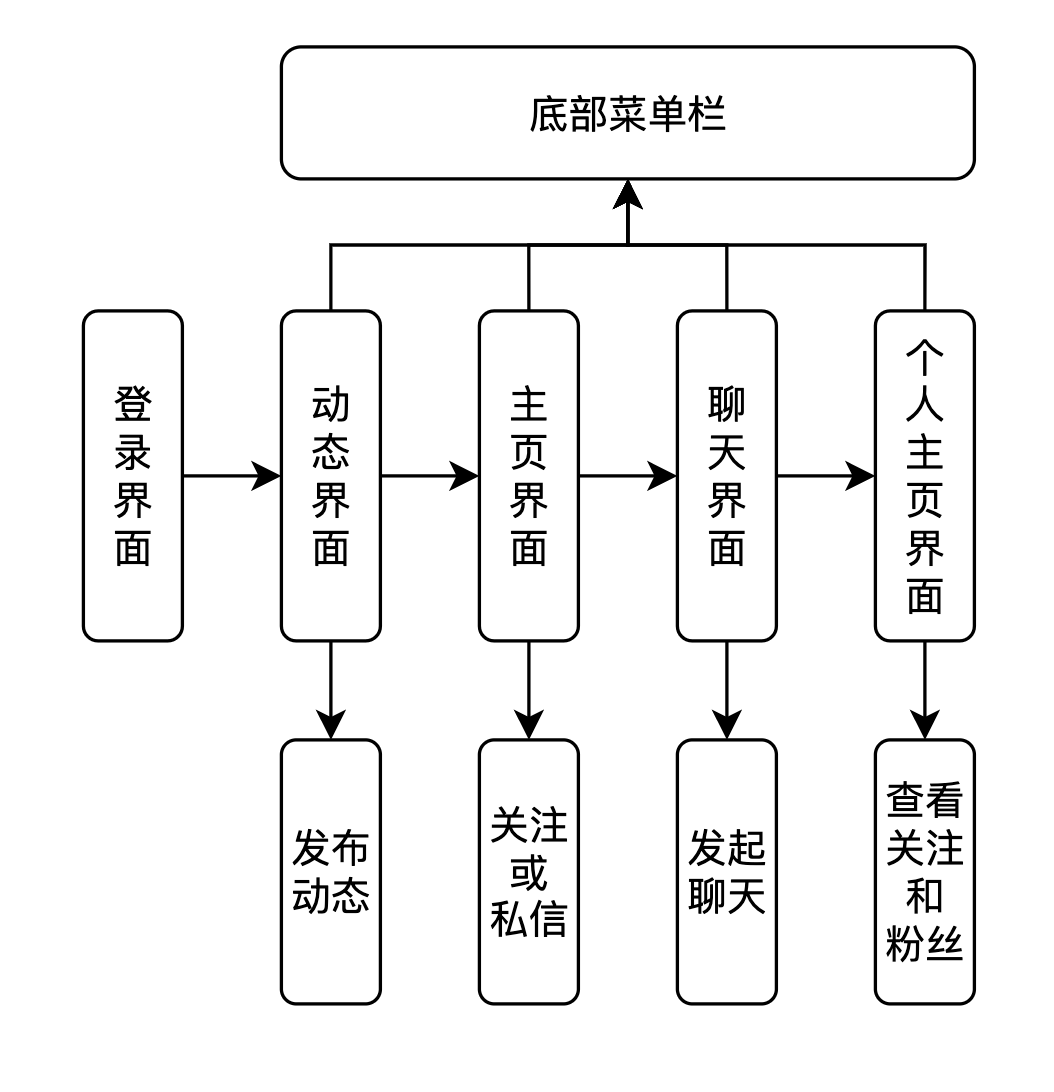


图 4-2 界面关系图

## 4.3后端设计

此小节通过文字阐述、时序图和流程图描述社交聊天系统的业务逻辑。

## 4.3.1用户注册流程

用户打开未登录的APP后，点击右上角的注册按钮，进入注册页面，提供手机号码，登录密码，确认密码，通过POST请求获取短信验证码，后端返回一个MD5加密后的短信验证码，用于前端首次校验，用户输入短信中发送的验证码后，调用POST请求给后端进行校验，成功则进入信息初始化界面，不成功前端提示错误。

用户注册流程时序图如图4-3所示

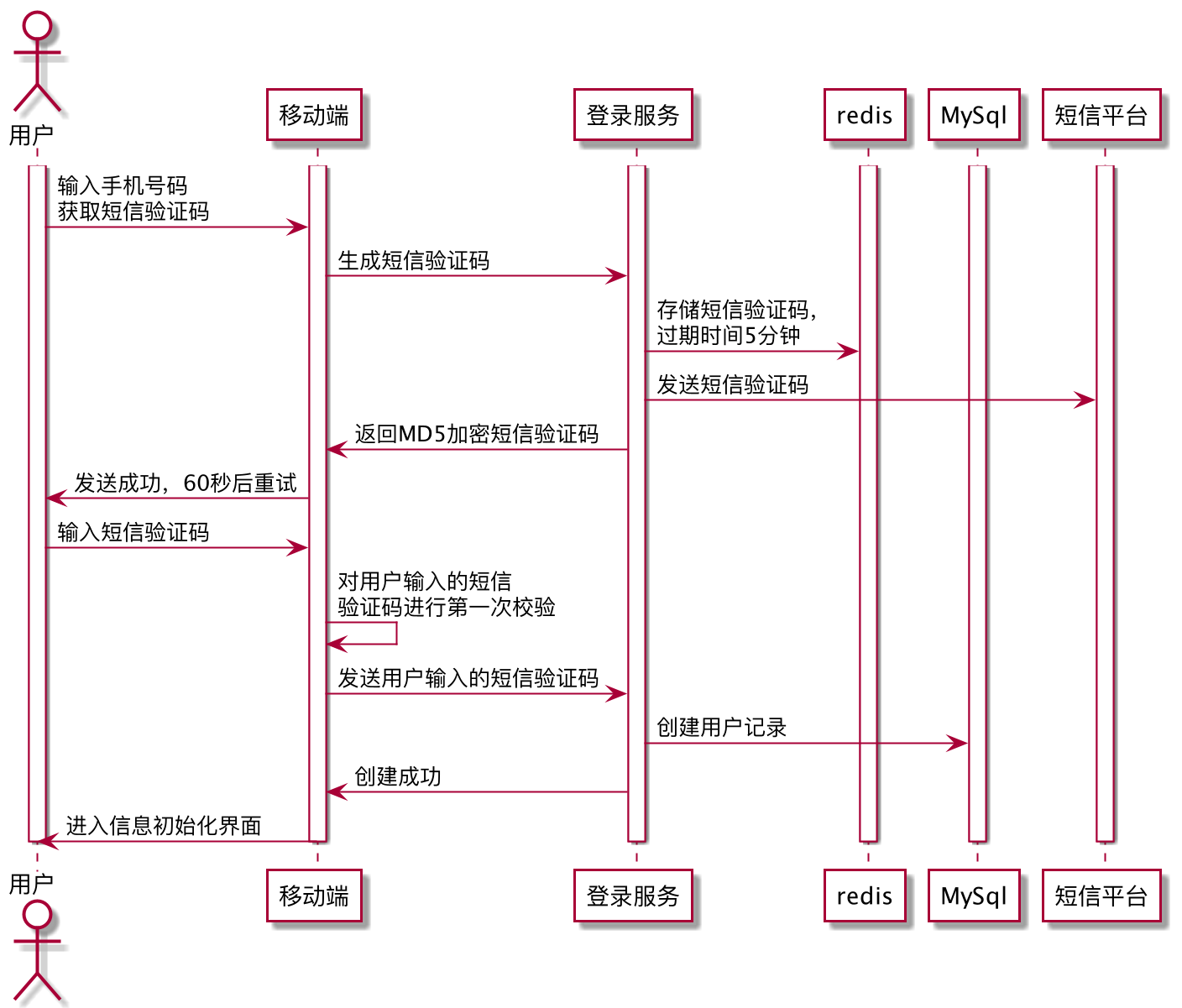


图 4-3 用户注册流程时序图

## 4.3.2用户登录流程

用户打开未登录的APP后，就进入了登录界面，默认使用密码登录，也可以点击密码框做下架的“短信登录”文本切换成短信登录。

1. 密码登录。用户在手机如输入框中填写手机号码，在密码框中填写注册时提供的登录密码，应用程序APP调用POST请求给后端进行校验，成功则进入主页界面，不成功前端提示错误。图4-4为用户密码登录流程时序图。

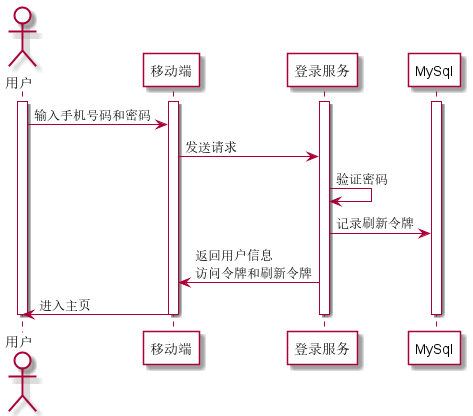


图 4-4 用户密码登录流程时序图

1. 短信登录。在切换成短信登录后，用户在手机如输入框中填写手机号码，点击验证码输入框右侧的获取短信验证码，后端返回一个BCrypt-5（强度为5）加密后的短信验证码，用于前端首次校验，用户输入短信中发送的验证码后，调用POST请求给后端进行校验，成功则进入信息初始化界面，不成功前端提示错误。图4-5为用户短信登录流程时序图。

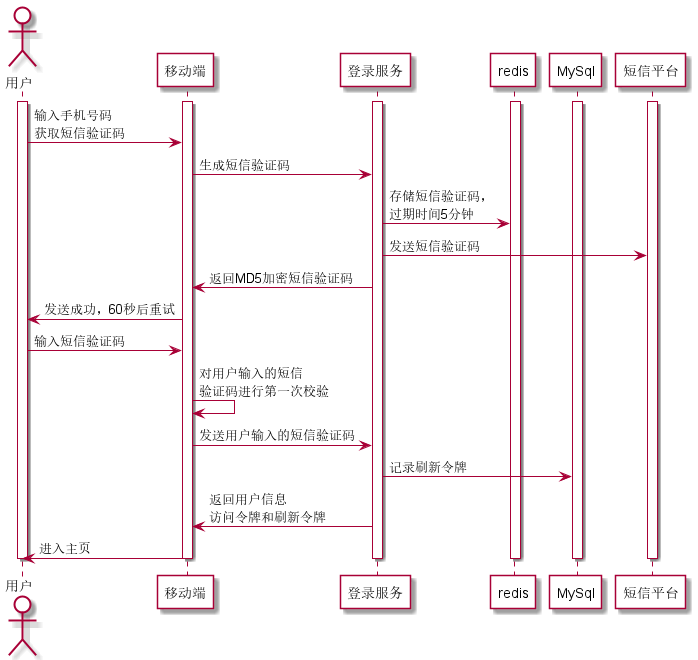


图 4-5 用户短信登录流程时序图

为了记录用户是否具有某些特权，可以记录客户维持登录状态的令牌（Token）。Token可以分为访问令牌（Access Token）和刷新令牌（Refresh Token）。

访问令牌除了可以用户接口的访问调用时，确定身份的依据，还可以用于存储一部分用户信息，用户权限等，访问令牌具有较短有效时间，当访问令牌过期后则失效，为了减少I/O的调用，本系统中的过期时间记录在访问令牌中，而不用从数据库或者Redis中校验访问令牌，所以令牌一旦发放，就不可撤回，直到令牌中存储的过期时间到期。

刷新令牌具有相对长的多的过期时间，用于在访问令牌已经过期或者即将过期的时候，获取新的访问令牌，本系统中，刷新令牌具有全局唯一性，如果在其他设备登录，原先的设备上存储的刷新令牌则会失效，原设备上将会在访问令牌过期后，因无法获取最新的访问令牌而退出登录。

## 4.3.3发布动态流程

用户在登录完成后就会进入动态广场页面，在页面底部中间的功能键，在弹出的功能列表中点击“发动态”按钮，应用跳转到动态发布界面，输入文本，选择图片，选择动态可见范围，可见范围包括：广场可见、关注人可见、仅自己可见，随后点击发布按钮，即可完成动态发布操作。

动态属于需要审核的功能，当动态发布完成后，动态处于仅自己可见状态，当审核通过审核之后转为原先设定的可见范围，若审核不通过，则删除处理，并予以警告。图4-6为用户发布动态流程时序图。

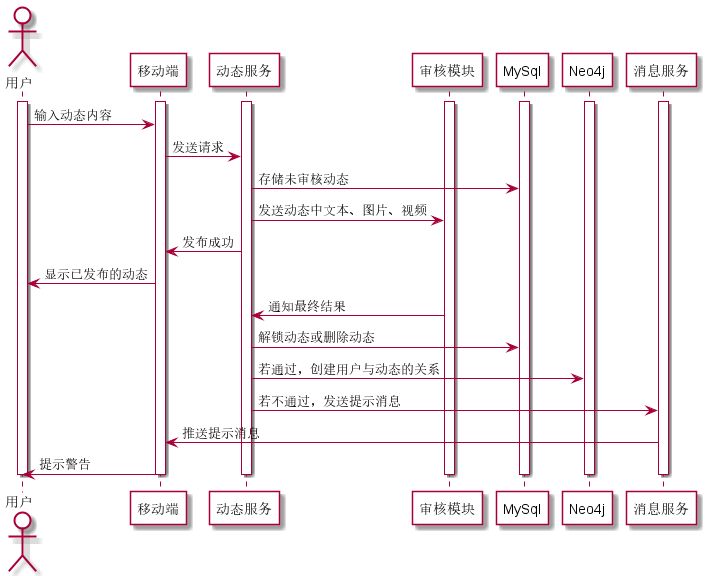


图 4-6 用户发布动态流程时序图

## 4.3.4关注好友流程

用户在广场上见到其他用户后，点击头像进入对方主页，如果未关注对方，则页面下方会出现两个按钮：“关注”和“私聊”，此时点击左下角的关注按钮即可完成关注。如果对方，则下方按钮只有私聊，点击右上角“更多操作”操作按钮，会弹出取消关注按钮，按下即可取消关注。

当点击关注后，当前用户的关注数和目标用户的粉丝数会同时加一，如果对方是已经是自己的粉丝时，双方的好友数会同时加一。相反的，取消关注时，当前用户的关注数和目标用户的粉丝数会同时减一，如果对方是已经是自己的粉丝时，双方的好友数会同时减一。图4-7为用户关注好友流程时序图，图4-5为用户取消关注流程时序图。

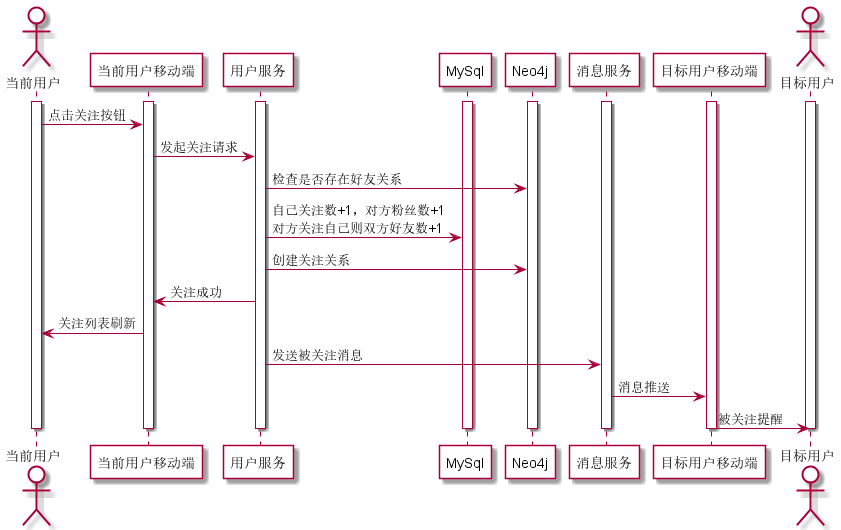


图 4-7 用户关注好友时序图

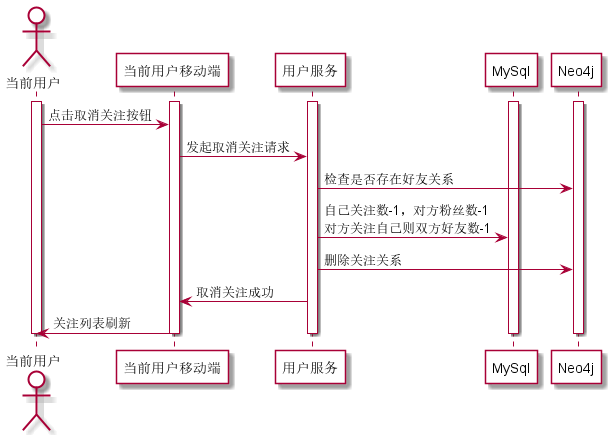


图 4-8 用户取消关注流程时序图

## 4.3.5接入即时通讯流程

当用户成功登录，随后与聊天服务建立TCP连接，使用访问令牌作为身份认证的依据，通过身份认证后，检查是否存在未读的历史消息，若存在，则以TCP下行的方式推送所有未读消息，随后进入监听消息发送消息阶段，期间每隔一段时间发送心跳包，并有不定期的推送消息提醒，一直到应用关闭时通讯终止。图4-9为用户即时通讯接入流程时序图。

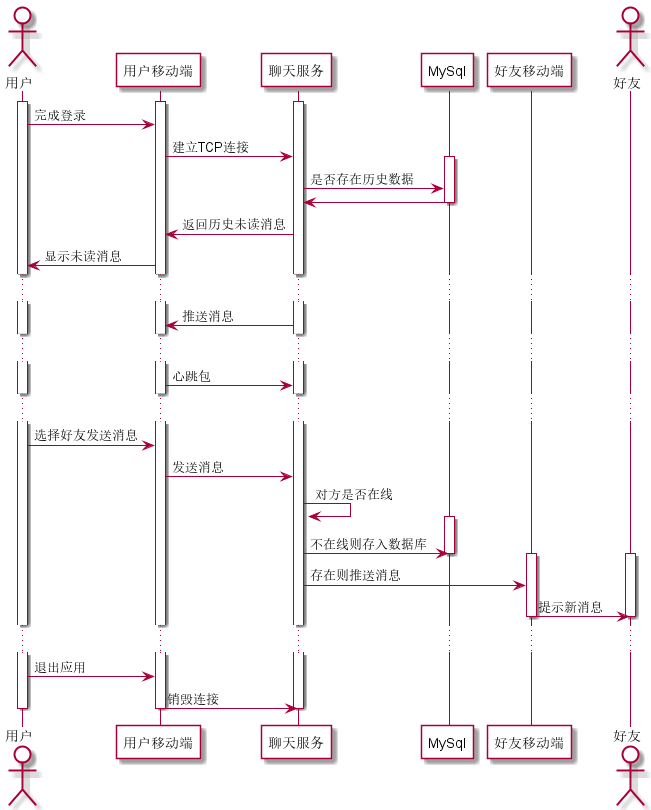


图 4-9 用户即时通讯接入流程时序图

用户所有的聊天信息均存储于客户端内部，用户一旦卸载应用并重装，就会丢失聊天记录。一方面，保护了用户隐私，另一方面，聊天数据是海量的，可以减少服务器数据存储压力。

## 4.3.6内容审核流程

内容审核模块为了与业务解耦，做到责任单一，那么就不能含有任何业务流程，本系统中，使用Rabbit MQ作为业务解耦中间件。

当业务模块需要进行模块审核时，需要提供业务id，回调所需的Rabbit MQ的queue关键字，文本内容，图片地址，视频地址，将如上信息发送给Rabbit MQ。审核模块在接收到消息后，先将所有任务暂时存储到Redis中，并依次对待审核内容发送给第三方审核。其中，文本可以同步完成审核，而图片和视频只能异步。过程中，如果出现任何一次审核不通过的情况，直接通过回调队列返回不通过信息，后续内容将自动忽略。如果全部内容均通过，则通过回调队列返回通过信息。图4-10为内容审核流程时序图。

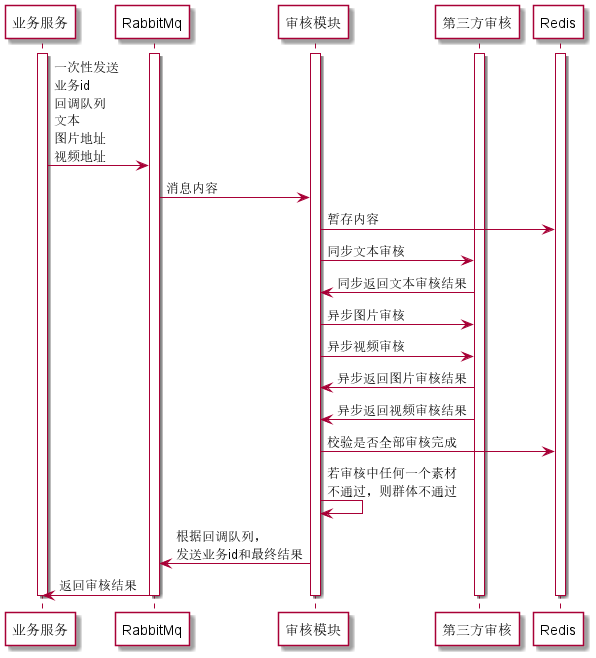


图 4-10 内容审核流程时序图

## 4.4接口设计

本节内容主要讲解系统中所用到的前端调用后端的Restful接口，以及几个关键接口的调用参数和返回值。

本系统中，不需要登录认证的接口有获取短信、注册用户、登录。其余接口需要将访问令牌以Bearer Token的形式传送给后端。Bearer Token指在接口调用时，以“Authorization”为请求头名，在访问令牌前拼上字符串“Bearer ”作为请求头值，后端读取到该头之后，便可获取到用户身份。

表 4-1 API接口总览

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 模块 | 名称 | 请求类型 | 请求URL |
| 1 | 认证模块 | 发送验证码 | POST | /auth/register/sendSms/{sense} |
| 2 | 手机短信登录 | POST | /auth/login/smsCode |
| 3 | 手机号绑定注册 | POST | /auth/register/smsRegister |
| 4 | 密码登录 | POST | /auth/login/password |
| 5 | 用户模块 | 关注/取关用户 | POST | /users/user/follow |
| 6 | 获取关注列表 | POST | /users/user/myFollows |
| 7 | 获取粉丝列表 | POST | /users/user/myFans |
| 8 | 设置好友别名 | POST | /users/user/friendAlias |
| 9 | 动态模块 | 动态广场 | POST | /blog/blog/recommend |
| 10 | 关注人动态 | POST | /blog/blog/follow |
| 11 | 发布动态 | POST | /blog/blog/publish |
| 12 | 删除动态 | POST | /blog/blog/delete |
| 13 | 动态点赞 | POST | /blog/blog/praise |
| 14 | 查看动态评论 | POST | /blog/comment/view |
| 15 | 添加动态评论 | POST | /blog/comment/add |
| 16 | 删除动态评论 | POST | /blog/comment/delete |

以下为本系统中核心接口的详细信息：

1. 短信登录接口

表 4-2 短信登录接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求URL | /auth/login/smsCode | | |
| 请求类型 | POST | | |
| 参数名 | 必填 | 类型 | 说明 |
| 请求参数 |  | | |
| account | 是 | string | 手机号 |
| smsCode | 是 | string | 短信验证码 |
| 返回参数 |  | | |
| code | 是 | int | 返回码 |
| success | 是 | bool | 是否成功 |
| message | 是 | string | 返回消息 |
| data | 否 | object | 返回数据，失败时为空 |
| data.token | 是 | object | 认证信息 |
| data.token.accessToken | 是 | string | 访问令牌 |
| data.token.expiresIn | 是 | int32 | 多久后过期，单位：秒 |
| data.token.refreshToken | 是 | string | 刷新令牌 |
| data.user | 是 | object | 用户基本信息 |
| data.user.id | 是 | int64 | 用户id |
| data.user.nickname | 是 | string | 用户昵称 |
| data.user.headUrl | 是 | string | 用户头像 |
| data.user.roles | 是 | string:int64 | 用户角色，key为角色名，value为过期时间戳，-1为永久 |

1. 关注/取关用户接口

表 4-3 关注/取消关注用户接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求URL | /users/user/follow | | |
| 请求类型 | POST | | |
| 参数名 | 必填 | 类型 | 说明 |
| 请求参数 |  | | |
| userId | 是 | int64 | 对方用户id |
| ops | 是 | int | 1关注 -1取关 |
| 返回参数 |  | | |
| code | 是 | int | 返回码 |
| success | 是 | bool | 是否成功 |
| message | 是 | string | 返回消息 |
| data | 否 | int | 成功时为1 |

1. 获取关注列表接口

表 4-4 获取关注列表接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求URL | /users/user/myFollows | | |
| 请求类型 | POST | | |
| 参数名 | 必填 | 类型 | 说明 |
| 请求参数 |  | | |
| pageNum | 是 | int | 页数 |
| pageSize | 是 | int | 每页长度 |
| 返回参数 |  | | |
| code | 是 | int | 返回码 |
| success | 是 | bool | 是否成功 |
| message | 是 | string | 返回消息 |
| data | 否 | array | 返回数据 |
| data[].id | 是 | int64 | 对方id |
| data[].nickname | 是 | string | 对方昵称 |
| data[].rel | 是 | int | 对对方的关系，1为有关系，0为无关系 |
| data[].revRel | 是 | int | 对方对我的关系，1为有关系，0为无关系 |
| data[].headUrl | 是 | string | 对方头像地址 |
| data[].alias | 否 | string | 给对方的备注 |

1. 动态广场接口

表 4-5 动态广场接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求URL | /blog/blog/recommend | | |
| 请求类型 | POST | | |
| 参数名 | 必填 | 类型 | 说明 |
| 请求参数 |  | | |
| pageNum | 是 | int | 页数 |
| pageSize | 是 | int | 每页长度 |
| 返回参数 |  | | |
| code | 是 | int | 返回码 |
| success | 是 | bool | 是否成功 |
| message | 是 | string | 返回消息 |
| data | 否 | array | 返回数据 |
| data[].id | 是 | int64 | 发布人id |
| data[].nickname | 是 | string | 发布人昵称 |
| data[].rel | 是 | int | 对发布人的关系，1为有关系，0为无关系 |
| data[].revRel | 是 | int | 发布人对我的关系，1为有关系，0为无关系 |
| data[].headUrl | 是 | string | 发布人头像地址 |
| data[].alias | 否 | string | 给发布人的备注 |
| data[].blog | 是 | object | 发布的动态 |
| data[].blog.id | 是 | int64 | 动态id |
| data[].blog.content | 是 | string | 动态文本部分 |
| data[].blog.resource | 否 | string | 动态图片视频资源部分 |
| data[].blog.genre | 是 | string | 动态类型，TEXT为纯文字，  IMAGE[空格][数字n]为n张图片，  VIDEO为视频 |
| data[].blog.praises | 是 | int | 动态点赞数 |
| data[].blog.comments | 是 | int | 动态评论数 |
| data[].blog.addTime | 是 | int64 | 动态创建时间，时间戳 |

1. 发布动态接口

表 4-6 发布动态接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求URL | /blog/blog/publish | | |
| 请求类型 | POST | | |
| 参数名 | 必填 | 类型 | 说明 |
| 请求参数 |  | | |
| content | 是 | string | 动态文本部分 |
| resource | 否 | string | 动态图片视频资源部分 |
| genre | 是 | string | 动态类型，TEXT为纯文字，  IMAGE[空格][数字n]为n张图片，  VIDEO为视频 |
| access | 是 | int | 0 自己可见 1关注可见 2广场可见 |
| 返回参数 |  | | |
| code | 是 | int | 返回码 |
| success | 是 | bool | 是否成功 |
| message | 是 | string | 返回消息 |
| data | 否 | int | 返回数据，成功时为1 |

1. 查看动态评论接口

表 4-7 查看动态评论接口

|  |  |  |  |
| --- | --- | --- | --- |
| 请求URL | /blog/comment/view | | |
| 请求类型 | POST | | |
| 参数名 | 必填 | 类型 | 说明 |
| 请求参数 |  | | |
| blogId | 是 | int64 | 动态id |
| pageNum | 是 | int | 页数 |
| pageSize | 是 | int | 每页长度 |
| 返回参数 |  | | |
| code | 是 | int | 返回码 |
| success | 是 | bool | 是否成功 |
| message | 是 | string | 返回消息 |
| data | 否 | array | 返回数据 |
| data[].id | 是 | int64 | 对方id |
| data[].nickname | 是 | string | 对方昵称 |
| data[].headUrl | 是 | string | 对方头像地址 |
| data[].praised | 是 | bool | 是否点赞 |
| data[].comment | 是 | object | 评论 |
| data[].comment.id | 是 | int64 | 评论id |
| data[].comment.userId | 是 | int64 | 评论者id |
| data[].comment.content | 是 | string | 评论内容 |
| data[].comment.praises | 是 | int | 评论点赞数 |
| data[].comment.addTime | 是 | int64 | 评论时间，时间戳 |

## 4.5数据库设计

数据库设计包括数据实体关系(Entity Relationship，简称E-R)模型，和数据库表结构的设计。

## 4.5.1数据库E-R模型

本系统中数据库有MySql和Neo4j。MySql用于主要的数据存储，Neo4j用于存储好友多对多的关系，同时存储用户动态，以便查找用户关注人的动态。图4-11为MySql的实体关系图，图4-12为Neo4j的实体关系图

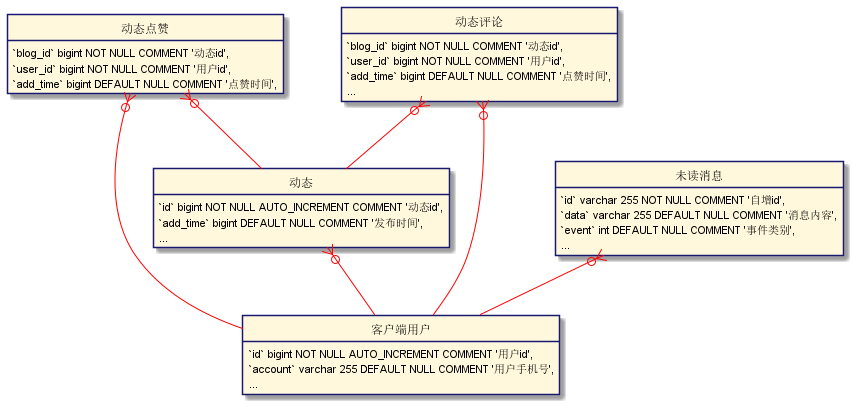


图 4-11 MySql E-R图

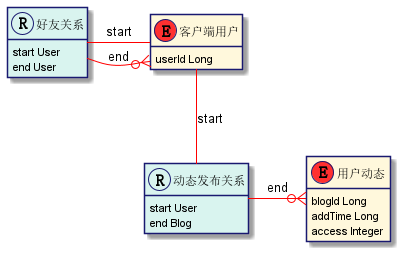


图 4-12 Neo4j E-R图

## 4.5.2数据库表设计

本系统中主要有五张数据表。主要用于存放用户信息、动态信息、动态点赞信息、动态评论信息、未读消息。

表client\_user用于存放用户信息；表blog用于存放动态信息，表中通过user\_id关联到client\_user的id字段；表blog\_praise用于存放点赞信息，表中通过blog\_id关联到blog表的id字段或关联到comment表的id字段；表comment用于存放评论信息，表中通过blog\_id关联到blog的id字段，user\_id关联到client\_user的id字段；表unread\_message用于存放用户因为不在线而错过的未读消息，表中通过user\_id关联到client\_user的id字段。

表 4-8 client\_user表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 允许空 | 说明 |
| id | bigint(0) | 否 | 自增id |
| account | varchar(255) | 是 | 手机号 |
| add\_time | bigint(0) | 是 | 创建时间戳 |
| fans | int | 是 | 粉丝数 |
| follows | int | 是 | 关注数 |
| friends | int | 是 | 好友数 |
| head\_url | varchar(255) | 是 | 头像地址 |
| Login\_time | bigint(0) | 是 | 登录时间戳 |
| nickname | varchar(255) | 是 | 昵称 |
| password | varchar(255) | 是 | 密码，BCrypt-5加密 |
| self\_state | varchar(255) | 是 | 个性签名 |
| status | int | 是 | 状态0无效用户1有效 |
| update\_time | bigint(0) | 是 | 更新时间戳 |
| uid | bigint(0) | 是 | 用户易记的号码 |
| active\_time | bigint(0) | 是 | 活跃时间戳 |
| roles | json | 是 | key是角色名，value是过期时间，-1永不过期 |

表 4-9 blog表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 允许空 | 说明 |
| id | bigint(0) | 否 | 自增id |
| add\_time | bigint(0) | 是 | 创建时间戳 |
| comments | int | 是 | 评论数 |
| content | varchar(255) | 是 | 内容 |
| genre | varchar(255) | 是 | 动态类型，TEXT为纯文字，  IMAGE[空格][数字n]为n张图片，  VIDEO为视频 |
| praises | int | 是 | 点赞数 |
| resource | varchar(255) | 是 | 资源地址 |
| user\_id | bigint(0) | 是 | 发布者id |
| access | int | 是 | 0 自己可见 1关注可见 2广场可见 |
| snapshot | json | 是 | 评论快照 |
| status | int | 是 | 状态码0删除1可用 |

表 4-10 blog\_praise表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 允许空 | 说明 |
| blog\_id | bigint(0) | 否 | 联合主键，动态或评论id |
| user\_id | bigint(0) | 否 | 联合主键，用户id |
| add\_time | bigint(0) | 否 | 创建时间戳 |

表 4-11 comment表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 允许空 | 说明 |
| id | bigint(0) | 否 | 自增id |
| blog\_id | bigint(0) | 是 | 动态id |
| add\_time | bigint(0) | 是 | 创建时间戳 |
| content | varchar(255) | 是 | 内容 |
| praises | int | 是 | 点赞数 |
| user\_id | bigint(0) | 是 | 发布人 |
| reply | bigint(0) | 是 | 回复人 |
| status | int | 是 | 状态0无效1有效 |

表 4-12 unread\_message表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 类型 | 允许空 | 说明 |
| id | varchar(255) | 否 | 自定义id |
| data | varchar(255) | 是 | 消息内容 |
| event | bigint(0) | 是 | 事件类型 |
| from\_id | bigint(0) | 是 | 创建用户 |
| time | bigint(0) | 是 | 创建时间 |
| user\_id | bigint(0) | 是 | 目标用户 |

## 4.6本章小结

本章按照第三章的需求分析的前提下，对功能模块做了详细设计，利用文字结合图片的形式，讲解了功能执行的过程，包括整体服务架构、前端设计、后端设计、前后端接口设计、数据库设计，为第五章提供指导骨架。

# 第五章 社交聊天系统的实现

## 5.1前端实现

## 5.1.1目录与文件结构

## 5.1.2核心功能实现

## 5.2后端实现

## 5.2.1概述

## 5.2.2用户注册的实现

## 5.2.3用户登录的实现

## 5.2.4内容审核的实现

## 5.2.5发布动态的实现

## 5.2.6关注好友的实现

## 5.2.7接入聊天的实现

## 5.3系统部署运行

## 5.4本章小结

……

# 第六章 社交聊天系统的实现

## 5.1测试环境

## 5.1接口测试

## 5.1功能测试

## 5.1性能测试

# 结 论

进行绿色生态建筑设计是当今时代的任务和要求，也是现代设计师们在未来工作中面临一项重要课题。本文在分析绿色建筑设计理论知识基础上，通过广州市番禺区长隆103地块重点开发项目的绿色设计进行了实证分析，总结如下：

（1）本文选择绿色建筑为研究主题，是响应世界和国家节能减排的号召，在继承相关学者理论研究基础上，结合自身专业实践开进行开展的研究方向，重点将高层住宅设计的难点和弱点转换为设计特色和优势，实现现代高层住宅宜居而节能目标。

……

本文创新之处是能够结合案例，能够识别并抓住绿色设计关键或核心约束并予以适当处置，通过采用动式设计方法进行绿色建筑项目的设计，确保绿色建筑设计绿色品质。

由于本人专业的限制和技术专业知识的匮乏，分析角度难免过于局促，本文未能详尽结合广州市番禺区长隆103地块重点开发项目案例进行细节观点分析，这些不足应成为我将来学习研究的动力和改善方向。

# 参考文献

1. 辛园园,钮俊,谢志军,张开乐,毛昕怡.微服务体系结构实现框架综述[J].计算机工程与应用,2018,54(19):10-17.
2. 李娜.基于Spring Cloud微服务架构的应用[J].电子技术与软件工程,2019(12):142.
3. 王萍.基于Spring Security和JWT实现无状态登录[J].电子技术,2021,50(12):114-115.
4. 童敏,张黎娜,梁伍七.基于JWT的分布式系统认证授权机制设计和实现[J].合肥师范学院学报,2022,40(03):7-10.
5. 魏井辉,吕明.基于Netty通信的消息推送系统的设计与实现[J].工业控制计算机,2020,33(12):57-59.
6. 曾超宇,李金香.Redis在高速缓存系统中的应用[J].微型机与应用,2013,32(12):11-13.DOI:10.19358/j.issn.1674-7720.2013.12.004.
7. 马豫星.Redis数据库特性分析[J].物联网技术,2015,5(03):105-106.DOI:10.16667/j.issn.2095-1302.2015.03.032.
8. 唐权,周蓉,张勇.Rabbit MQ消息中间件在Spring Boot教学中的应用[J].现代信息科技,2020,4(18):125-127.DOI:10.19850/j.cnki.2096-4706.2020.18.035.
9. 王余蓝.图形数据库NEO4J与关系据库的比较研究[J].现代电子技术,2012,35(20):77-79.DOI:10.16652/j.issn.1004-373x.2012.20.045.
10. 王红,张青青,蔡伟伟,姜洋.基于Neo4j的领域本体存储方法研究[J].计算机应用研究,2017,34(08):2404-2407.
11. 陈丽珊.如何从社交软件市场分一杯羹[J].通信企业管理,2018(04):50-51.
12. 梁剑豪.网络废话的基本特征、成因及其对策探究[J].新媒体研究,2021,7(16):107-109+119.DOI:10.16604/j.cnki.issn2096-0360.2021.16.026.

# 

# 致 谢

在[论文](http://www.so.com/s?q=%E8%AE%BA%E6%96%87&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)完成之际，我要特别感谢我的指导老师王亮明老师的热情关怀和悉心指导。在我撰写论文的过程中王亮明老师倾注了大量的心血和汗水，无论是在论文的[选题](http://www.so.com/s?q=%E9%80%89%E9%A2%98&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)、构思和[资料](http://www.so.com/s?q=%E8%B5%84%E6%96%99&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)的收集方面，还是在论文的[研究方法](http://www.so.com/s?q=%E7%A0%94%E7%A9%B6%E6%96%B9%E6%B3%95&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)以及成文定稿方面，我都得到了王亮明老师的悉心细致的教诲和无私的帮助，特别是他广博的[学识](http://www.so.com/s?q=%E5%AD%A6%E8%AF%86&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)、深厚的[学术](http://www.so.com/s?q=%E5%AD%A6%E6%9C%AF&ie=utf-8&src=internal_wenda_recommend_textn" \t "_blank)素养、严谨的治学精神和一丝不苟的工作作风使我终生受益，在此表示真诚地感谢。