摘要

千禧年后， Twitter 开始进入中国，引发了一代新流行。紧接着微博就凭借它的立即互动特性快速成为新兴的大众信息交流平台。

随着微博的热潮，人们由以前的传统媒体纯获取信息的身份转换为信息的发布者和获取者，在传统媒体时期，大多是人民群众只能是被动的获取信息，可以说微博的兴起是人民发声的一大转折点。

微博系统作为新媒体的代表，自然也需要先进的解决思路和方案，本文结合当今中小型企业中广泛使用的技术以及实际应用，设计并实现了一个基于 Node.js 的微博系统。

我的毕业设计主要做到了微博的基础和特色功能，比如常规的发布微博、上传图片、回复微博、关注感兴趣的用户、@他人来提醒别人看微博，等等。

在系统的后台主要采用了 koa2 的框架进行开发，在数据库上选择了目前中小型企业中使用最广的关系型数据库 mysql，使用 session 完成登录，在前端页面上使用 EJS 后端模板引擎结合 CSS，此外还使用了 Redis 作为缓存数据库。基于 node.js 的微博系统具有轻量型且易部署的特点，该系统适合于想要推出自己的微博网站的用户进行使用。

本文的主要工作包括：简单的介绍微博的背景和意义以及微博的发展现状以及所使用的相关技术，对系统进行需求分析并根据架构设计进行基于 Node.js 的微博系统的编码实现和测试。

关键词：微博系统 Node.js koa2 mysql session EJS Redis

Abstract

After the millennium, Twitter began to enter China, triggering a new generation of popularity.Then weibo quickly became an emerging public information exchange platform with its immediate interactive characteristics.

With the upsurge of micro-blog, people have changed from the previous status of pure information acquisition by traditional media to that of information publisher and acquirer. In the period of traditional media, most people can only passively obtain information. It can be said that the rise of micro-blog is a turning point for people to make their voices heard.

As a representative of new media, microblog system naturally also needs advanced solutions and solutions. In this paper, a microblog system based on node. js is designed and implemented based on the technologies widely used in today's small and medium-sized enterprises and practical applications.

My graduation project mainly accomplished the basic and characteristic functions of weibo, such as regular release of weibo, upload pictures, reply to weibo, follow interested users, @others to remind others to read weibo, and so on.

In the background of the system, koa2 framework is mainly used for development. In the database, mysql, the most widely used relational database in small and medium-sized enterprises, is selected. Session is used to complete login.The micro-blog system based on node.js is lightweight and easy to deploy. The system is suitable for users who want to launch their own micro-blog website.

The main work of this paper includes: briefly introducing the background and significance of weibo, the development status of weibo and the related technologies used, analyzing the requirements of the system, and implementing and testing the node.js -based weibo system according to the architectural design.

Keywords: weibo system Node.js koa2 mysql session EJS Redis

目录

第一章 绪论

本章主要通过从不同的角度分析微博系统的背景与意义，以及从历史的不同媒体的演变来展示研究现状，从而开启项目的绪论。让大家能够更好的了解微博系统的发展和意义，从而明确本项目的研究意义所在。

1.1 研究的背景与意义

1.1.1 研究的背景

数百年前，我们的祖祖辈辈通常忙于生计，并无多少休闲娱乐的时间，他们的快乐大多来源于田地或者大自然所给予的点点滴滴。所获取到的信息也仅仅局限于大小村落。

从报纸到杂志、小说，又发展到广播，在我们的生活中，他们都潜移默化的产生着巨大的影响。它们不仅是人们的娱乐通道，更是官方的舆论宣传工具，民国时期，各个党派、团体都尽量创建广播电台以利用其作为自己的宣传工具。因此广播事业在民国时期客观上具有一定发展[1]。它的优势十分明显，但劣势也不容忽略，所有的广播内容随着声音消散在空气之中，遇到不感兴趣的内容无法跳过，只能顺序收听，如果语言不通，那么收听会变得十分困难。

随着科技的发展，电视逐渐走入大众的视野。它凭借着动态的影像和声音以及丰富而真实的资料,运用手段高超的表现艺术,不仅能够吸引人的眼球,还能够激发人的学习情绪[2]。我们在不知不觉中已经从听觉转向了视觉，非常自然的开始从屏幕上获取众多的信息。但电视上的信息除非重播，否则也是播过就很难回顾查询的。

千禧年后，互联网技术蓬勃发展，日新月异的更新速度把微博客这一即时信息传递的媒体方式推向大众，曾经在报纸或者杂志上能得到的新闻或者优惠信息在微博上一样能获取到。而且人们可以通过进入个人空间浏览到往期的信息，发布在微博上的信息很好的避免了广播或者电视的时效性问题，极大的方便了人们进行信息的回顾与查阅。在传统媒体时期，大多数人民群众只能是信息的获取者，而微博的兴起使这一现象发生了历史性的改变，所有用户都可以成为信息的发布者、传播者以及获取者。

1.1.2 研究的意义

作为超新媒体的微博拥有其他传统的媒体身上没有的优势：

1. 微博散播消息的速度非常迅速。

2. 内容短小精悍，你可以在任何地方任何时间立刻将心中所想的写发布在微博上，甚至还可以根据自己的想法插入图片或者@自己想提醒的人来看微博。

3. 老人小孩上手快，便捷易理解的操作方式。

4. 可以及时互动，有别于其他传统媒体的无法互动，微博可以通过@迅速通知另一个用户或者回复对象，极大的提高了信息传播的及时性。

5. 影响力有时超乎想象，一条微博，有时候石沉大海，但只要被大量转发，会产生超乎想像的影响力，广泛收到社会的关注。可以帮助有需要的人群进行发声以解决许多实际的问题，方便了很多求助无门的人群。

6. 交友便利，用户可以极为简单的建立自己的听众群，形成自己的爱好圈子，和同好们一起收获快乐。

微博系统作为新媒体的代表，自然也需要先进的解决思路和方案，本文设计并实现了一个基于 node.js 的微博系统，主要实现发布微博、发布图片、回复微博、用户之间相互关注、@关注对象、主页微博内容发布即更新等功能。该系统主要采用了 koa2 的框架进行开发，在数据库上选择了关系型数据库 mysql，使用 session 完成登录，在前端页面上使用 ejs 后端模板引擎，使用 redis 作为缓存数据库。基于 node.js 的微博系统具有轻量型且易部署的特点，适合于想要推出自己的微博网站的用户进行使用。

1.2 研究现状

微博作为媒体传播快速发展的一个重要特征,正越来越受到关注[3]。

微博的兴起，改写了传统媒体对突发事件的报道方式，它既是信息的提供者、事件的围观者，又与突发事件的发展过程紧密相关，成为事态发展的影响者、参与者和推动者，在突发事件的报道和舆论传播中扮演着十分重要的角色[4]。

第二章 相关技术研究

本章主要介绍了基于 node 的微博系统所使用的相关技术，包括系统所用的技术、框架、数据库等。

2.1 Node.js 简介

Nodejs 是 Ryan Dahl 在 2009 年开发的基于 Chrome V8 引擎的 JavaScript 运行环境[5]，而 Chrome V8 引擎执行 JavaScript 具有异常快的速度以及非常优秀的性能。因为 Chrome V8 引擎使用的是非常先进的编译技术，他可以使 JavaScript 这样的脚本语言编写的程序与 C 等高级语言编写的程序拥有差不多的性能。

同时 Nodejs 使得 JavaScript 焕发新的活力，它可以使 JS 摇身一变成为服务端语言。Nodejs 的编写可以理解为使用 JavaScript 语言去利用 Nodejs 的 API 库进行服务器端开发[6]。

与 Python 等动态语言相比较，JavaScript 的性能突出，且它的匿名函数和闭包都特别适合 Nodejs 的事件驱动以及异步编程，这也在无形之中提高了 Nodejs 的性能[7]。

Node 的优点：

1．提供包含各类实用函数的模块。Node.js 使用 Module 模块（类似 C 语言的类库）划分出不同的功能，每个模块都提供了相关功能的各类实用函数。比如，最常使用的 HTTP 库，可以直接调用它的函数来快速创建 HTTP 服务器。

2. 支持高并发。传统的 Web 服务技术是每个请求到来都创建一个新的线程，系统为每个线程分配内存，最终会因为内存不够而透支，假设每创建一个线程系统为它分配 2M 内存，在一台 8G 内存的系统上它的最大并发数是 4000[8]，这是使用传统 Web 服务技术的场景。但反观 Node.js，就会发现其可扩展性远远超过传统认知。单线程工作的 Node.js 使用非阻塞 I/O 调用，这就让他能够承受上万的并发连接！

3. 运用事件循环来解决大规模的 HTTP 请求。Node.js 虽然是单线程、单进程的，但它采用了事件驱动机制和异步编程风格（提供的 API 基本都是异步风格且基于事件的），使用“事件循环”的架构来编写出可扩展性高的服务器[9]。当出现大规模 HTTP 请求时，Node.js 会产生事件循环队列，剔除掉那些多线程资源的占用以及上下文切换，Node.js 可以理解为单纯的为数据库和文件等的资源提供了接口，简化了对慢资源的访问。事件循环和异步提高了 Node.js 的性能，也降低了开发复杂度。

Node.js 从发布伊始就受到很多前端工程师的热爱，它是一个用于开发高性能并发程序的框架，这些程序不依赖于主流的多线程方法，而是使用异步 I/O 和事件驱动的编程模型[10]。作为前端开发者迈入全栈工程师的重要过程，其应用场景十分广泛，比如：网站的搭建、即时聊天软件、前端构建工具（比如我们熟知的 webpack）、操作系统（NodeOS）、跨平台打包工具（比如 NW.js）、命令行工具（比如 Cordova）、编辑器（VSCode）等。但同时由于其特殊的机制，Node.js 并不适用于解决大规模的计算问题（CPU 密集型操作）[11]。

2.2 koa2 框架

使用过 Express 的同学应该对 koa 框架有所耳闻，毕竟他们是“一母同生”，可以说 koa 是 Express 的升级版，很多语法上有共同之处，学习过 Express 再去接触 koa 将会发现极容易上手，非常友好。

而 koa 又有两个不同的版本，现在来看直接上手 koa2 是最好的，因为他是基于 ES7 规则诞生的产物，完全支持 Promise 的 async 来进行编码。这无疑是我们丢掉可怕的回调函数的好机会！

对比 express 框架后，最终我决定使用 koa2 框架。虽然 Express 的 API 很简单，但是它始终是“过时”了，毕竟是基于 ES5 的产物。想要通过它实现异步就只有一条路可走——回调。而回调对于编程来说是十分繁琐的，特别是对 js 的异步编程来说，如果异步嵌套层次过多，那么就会形成可怕的回调地狱。所以从 js 的异步编程的角度来看，express 这个框架就会被排除掉。

几年前，express 的团队基于 express 和他的中间件模型又重新开发出 koa2 这个框架。koa2 是原生支持 async awit 异步编程的。在目前看来（或者未来三五年内），它是支持 js 异步开发最好的方式。

2.3 mysql 关系型数据库

Mysql 应该是所有程序员的入门数据库了。

目前企业中应用最广泛也是成本最低的关系型数据库，对于一般的个人使用者和中小型企业来说，Mysql 提供的功能已经绰绰有余，而且由于 Mysql 是开放源码软件，因此可以大大降低使用总体拥有成本。

虽然 mongodb 也是常用于和 node 进行搭配的数据库，但 mongodb 在企业中（特别是中大型企业中）的应用绝对没有 mysql 这么广泛。而在中大型企业中中，数据库的部分是有专门的团队进行运维的，他们对 mysql 的熟悉程度和数据管理都远超过 mongodb。

因此我最终选择了 Mysql 作为帮我存储数据的好帮手。

2.4 session 登录技术

说到登录自然会联想到 cookie，而 cookie 的不安全自然让我联想到 session。

session 是目前使用最广泛的登录技术，它比较适合用于 “页面统一，有时候可能运用后端模板引擎”这样比较集中的 web sever 的项目。而本次开发的微博页面就比较统一，所以我们会在项目的 koa2 框架中使用 session 的第三方中间件 koa-session 来记录请求者的身份。

koa-session 使得我们可以很方便地做验证登录，它内部帮我们做了信息的加密和解密，在浏览器中看到的信息是一串类似 uuid 的乱码，安全性极好。

还有一个比较流行的登录技术——jwt，jwt 虽然也是我们日常会使用的登录技术，但论广泛程度还是不如 session，特别是在 web sever 系统中。

2.5 EJS 后端模板引擎

我刚开始学习后端的时候就听过一句话——“学习后端一定要掌握 EJS 模板。”使用下来确实发现了他的美妙之处。

“E” 代表“可嵌入（Embedded）”，也可以是“高效（Effective）”、“优雅（Elegant）”或者是“简单（Easy）”。

EJS 没有如何组织内容的教条，也没有再造一套迭代和控制流语法，可以理解为它只是我们熟知的 JavaScript 代码而已。

相比于 vue 或者 react 等前端框架来说，后端模板引擎更加方便快捷。分工合作时，我喜欢前后端分离。但倘若是个人开发，那么在后端开发时我更倾向一个简单的服务端渲染技术，很显然，ejs 就是本次系统设计中我所需要的。

2.6 Redis 缓存

用到 mysql 不禁联想到另一个数据存储的绝佳选手，她很好的弥补了 mysql 所做不到的事情，那就是 Redis。

Redis 可以对关系型数据库起到很不错的补充作用，且 Java、C、C++、Python 等很多语言都支持 Redis[12]。不能说 Redis 是一个纯粹的键值对存储数据库，他其实是一个数据结构类型的数据库，通过 Redis 存储你内容是二进制安全的，也就是说数据是使用二进制形式进行传输的，这保证了数据的安全性（比如加密）[13]。

Redis 支持存储的类型十分多，我们常见的字符串（string）、链表（list）、集合（set）、有序集合（sorted set）和哈希表（hash）都是支持的[14]。什么都接纳的 Redis 加上他的特性，就特别适合作为系统公共信息的存储处。

并且在实际的情况下，往往我们无法使用单台极其来保证系统一直安全的运行，因为硬件上的故障和软件上的问题都有可能使系统无法使用。因此 Redis 的另一个优势就显示出来了，它能够很好的支持数据库的集群搭建，而一个高性能、高可用的内存数据库集群正是生产中所必需要有的。正是如此，目前考虑到缓存数据库我们可以毫无疑问选择 Redis，他并没什么比较强的竞争对手。

第三章 系统需求分析

本章将对基于 node 的微博系统进行详细的分析，包括可行性分析、功能需求分析、非功能性需求分析、数据库需求分析四个部分。

3.1 可行性分析

可行性分析在项目中是非常必要的，如果不经过可行性分析，就有可能导致软件在开发过程中发现某些方面不可行而失败。所以我将在本节中通过技术、经济已经开发可行性三个角度来剖析改项目的可行性。

3.1.1 技术可行性

Node.js 是在 2009 年就已经完成的，多年以来都是 Github 上最受欢迎的语言，贡献度也占据着第一的位置。同时 Node.js 的技术栈一直在急速发展之中，我能接触到的相关项目十分丰富，所以技术上肯定没问题。

Mysql 和 Redis 都已经是相当成熟的数据库，从书本到实战都有丰富的参考资料。并且他们可以互为补充，所以使用 Mysql 和 Redis 来提供数据层面的保障是可行的。

Koa2 框架是几年前由 express 的团队基于 express 和他的中间件模型又重新开发的，它原生支持 async awit 异步编程的，在目前看来（或者未来三五年内），它是支持 js 异步开发最好的方式。而且几年下来她已经发展的相对完备，具有不少参考资料，所以使用 Koa2 进行开发是完全可行的。

session 是目前使用最广泛的登录技术，它比较适合用于 “页面统一，有时候可能运用后端模板引擎”这样比较集中的 web sever 的项目。而本次开发的微博页面就比较统一，所以我们在项目中使用 session 的第三方中间件 koa-session 来记录请求者的身份是完全可行的。

3.1.2 经济可行性

本人在开发该项目时使用的是普通的笔记本电脑，可以满足一般的要求。单纯用作学习和研究来说，一个普通的笔记本电脑就可以了。所以经济上是完全可行的。

3.1.2 开发可行性

Node.js 是基于 JavaScript 语言的，Redis 和 Mysql 操作都不难，所以综合来说只需要学习 koa2 框架的使用并且精通 JavaScript 即可，这些都是单人就可以完成的操作。由于本人在往期实际项目开发中对 JavaScript 的研究最为深入，所以开发基于 node 的微博系统是完全没有问题的。

综上所述，开发完全可行。

3.2 功能需求分析

本人在进行了大量国内外文档的阅读和实际的使用后，将功能需求进行分析，主要是发布微博、回复微博、删除微博、@好友、发布图片、关注与粉丝等。

由于基于 node 的微博系统的轻量化特性要求，本项目主要实现注册、登录、个人信息修改、微博的发布、删除、评论等功能，支持用户@好友进行微博的发布以及回复，支持关注别的用户，支持发布图片。

1. 注册

众所周知，一个系统逃不开的就是注册。本项目采用的是用户名和密码的方式进行注册，不允许相同用户名重复注册，对已注册的用户名会进行检测后给出相关提示。

2. 登录

用户可以拿用户名和密码登录本系统，系统对输入的用户名和密码进行校验后判断成功与否。登录后方可进行微博的发布、转发等操作。

3. 微博列表

用户发出的微博需要立即显示在列表中，而这个列表会根据不同的需求展示在首页、广场页、个人主页等不同页面上。

4. 发布微博

作为微博系统最重要的功能之一，发布微博内容至关重要，因为必须先有用户发布了微博，才会产生数量庞大的微博列表以供浏览。微博被发布之后会立即显示在微博列表顶端。

本系统不允许游客进行微博发布，游客必须通过注册登录后才能进行微博发布。如果游客并未登录，系统会提示先登录后再进行微博发布。

微博的发布有单帖字数限制。

5. 删除微博

对于不想要的微博，用户可以进行删除处理。

6. 插入图片

不同于传统的文字描述，用户可上传自己喜欢的图片来搭配展示自己的微博。

7. 关注

关注其他用户是微博系统的特点之一。

也就是说，用户可以通过关注和自己有共同爱好的用户群来最快的获取最大程度上的信息与快乐。

当然，也有“取消关注”的功能，用户可以随时随地的选择自己是否要继续关注另一个用户。

关注其他用户的前提条件是必须先进行登录，游客状态下是无法进行关注的。

1. @其他用户

在发布或者转发微博时可以使用@其他用户的用户名，这样其他用户就会即时收到提醒，以便及时准确的查看。

当然，用户只能@她所关注的用户，这样一来可以防止@错人，二来也可以减少恶意骚扰的可能性。

2. 转发微博

除了发布自己的微博，用户还可对别人的微博进行转发，达到快速传播的效果。

3. 回复

如果只能发布微博就显得过于自娱自乐了，长期以往肯定会打消用户的积极性，此时回复功能就显得至关重要了。

用户可以对微博的原始发布者进行回复，也可以对其他评论者进行回复，但本系统不允许游客进行微博回复，用户需要登录才能进行回复。如用户在游客状态下就去点击回复按钮，那么系统将自动跳转到登录页面。用户登录后会自动跳回刚刚访问的页面。单次回复有字数限制。

4. 设置

导航栏中的“设置”可以进行用户个人信息的修改，包括昵称、城市以及头像，还可以修改密码。

为提高用户使用感受，本系统支持通过点击用户头像来访问“设置”来进行个人信息的查看与修改。

5. 敏感词处理

用户在发布微博和回复微博之前，系统会进行敏感词处理，如果包含敏感词就会给出提示，不允许继续发布或者回复。

3.3 非功能性需求分析

虽然非功能性需求并不会影响系统的逻辑，但在一定情况下会影响系统的功能需求，所以本人将所有和业务功能无关的需求分为两个部分进行分析——性能需求和安全性需求。

3.3.1 性能需求分析

随着科技极速发展，各种技术也在不断的更新换代，所以我们必须考虑使得整个系统的复用性能，要求在设计系统时多使用能跟得上技术发展的工具或者框架。比如本次系统设计中我所使用的 koa2 框架，他是基于 ES7 开发的，良好的支持了 Promise+async 来实现异步。而 ES7 必定是未来的趋势，所以早一步下手一定有利于提高系统的可复用性。

另外还要考虑到多用户并发，微博系统的特性导致了在某些时间段可能会产生大量用户一起登录的情况，那么系统就必须保证维持稳定且不会过于卡顿的访问。本人在该项目中使用的 Node.js 就是支持高并发的，他的工作是单线程的，又使用的是非阻塞 I/O 调用，所以能够承受上万的并发连接。

综上所诉，本人通过各种途径很好的保证了性能上的需求。

3.3.2 安全性需求分析

作为 web 程序，系统一定会和另一端的服务器进行持续的信息交流，所以必须保证系统不受外部攻击，也就是说系统的安全性不可避免的成为重中之重。

在本系统中采用了 koa-session 来做验证登录，它内部帮我们做了信息的加密和解密，以二进制形式传递，在浏览器中看到的信息是一串类似 uuid 的乱码，保证了安全性。

3.4 业务流程分析

在充分结合系统的各个功能需求后，本人将普通用户的业务总流程图绘制如下图 3-1 所示，此图中涵盖了本系统普通用户的所有核心功能。

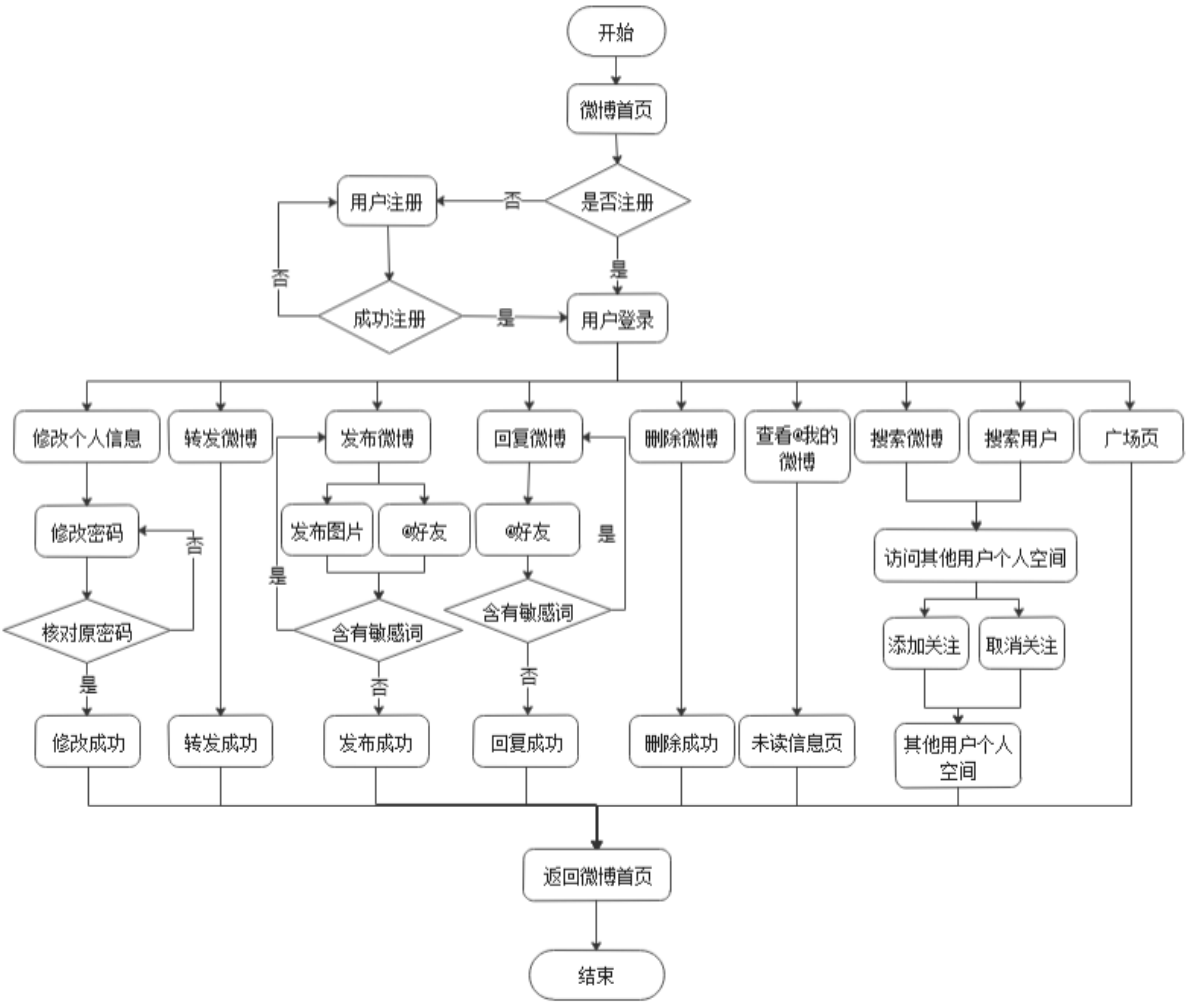

图 3-1 业务流程分析图

第四章 系统设计

4.1 系统架构设计

我将微博系统分为了三个不同的层次——用户界面层、业务逻辑层和数据访问层。架构设计模型如下图 4-1 所示。

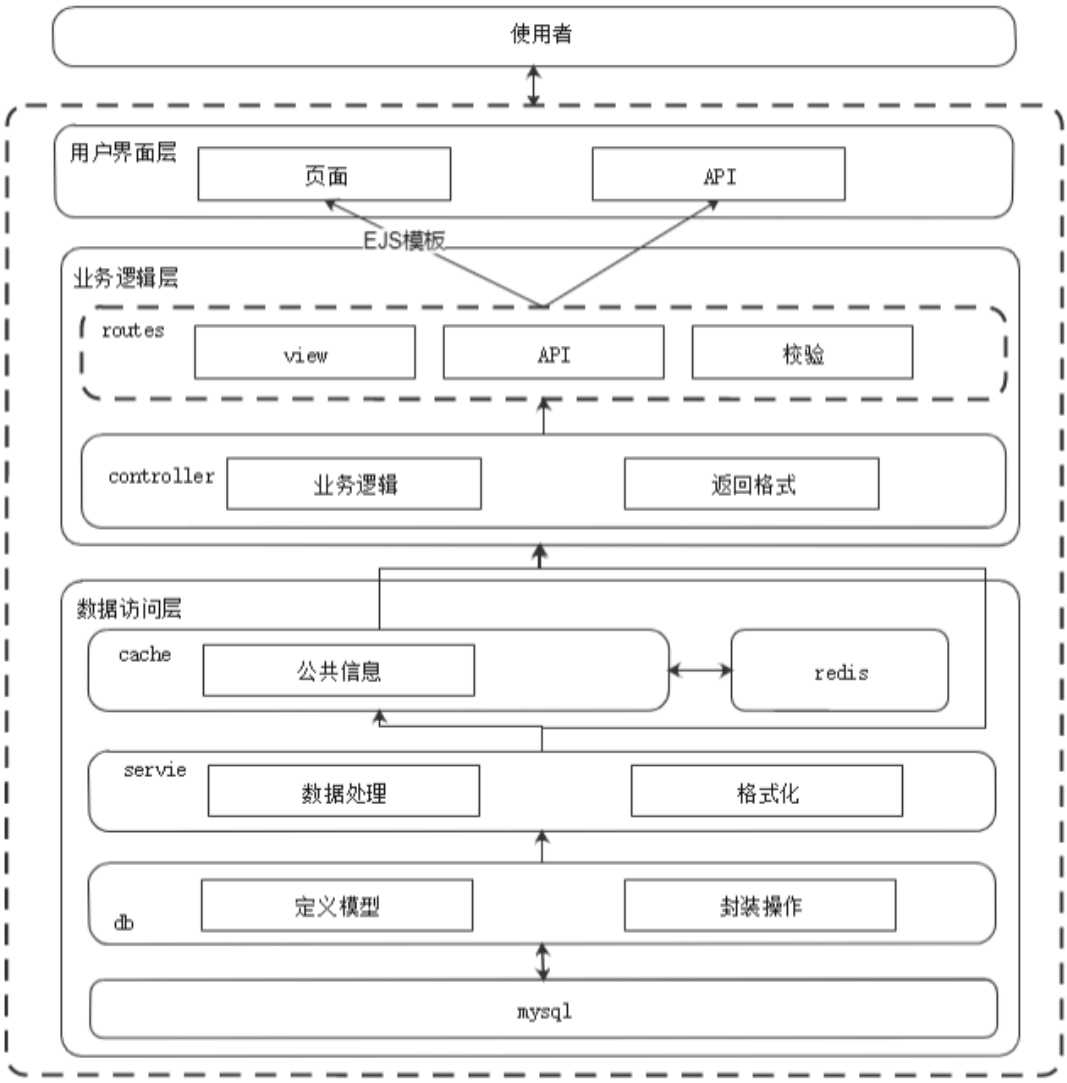

图 4-1 系统架构设计图

（1）用户界面层

用户界面层也就是系统用户所能看到的界面，本系统中采用 EJS 模板来使得普通的 JavaScript 代码生成最终的 HTML 页面展示给用户。

（2）业务逻辑层

业务逻辑层只搭理路由相关的数据处理，对数据库的数据操作是一概不管。处理好的数据被传入该层后会统一为一种格式。

往细了说，路由其实也不能分到业务逻辑层中，但我考虑到他是沟通用户界面层和业务逻辑层的桥梁，故我也将其放在此层中。

（3）数据访问层

数据访问层处于系统底部，它与数据库直接相邻，通过该层可以实现对数据库的增删查改并将数据通过接口提供给业务逻辑层。我主要用 redis+mysql 来支持数据访问层。

除设计模型外还从网络的角度出发绘制了网络架构图来阐述本系统的架构设计，如下图 4-2 所示。

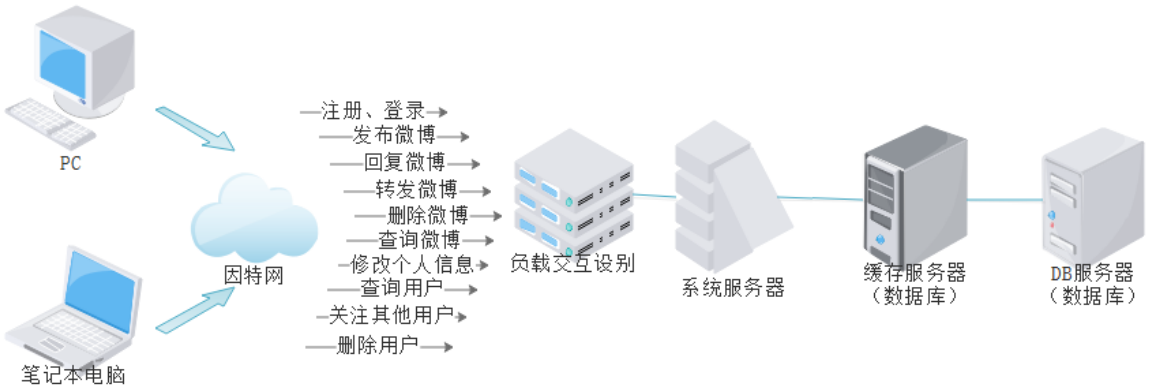

图 4-2 网络架构图

我希望通过系统模块包图更加清晰的展示各个包之间的关系，故通过该系统的架构设计做出了系统模块包图，如下图 4-3 所示。

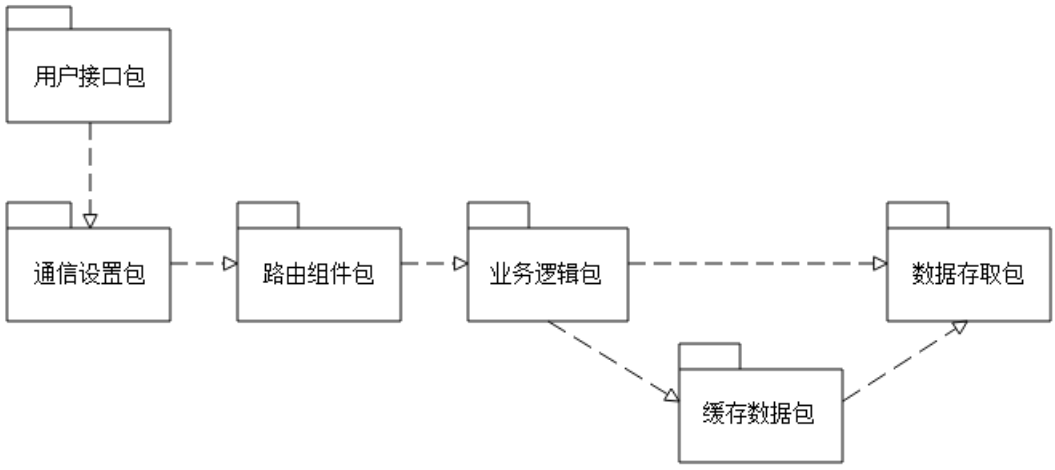

图 4-3 系统模块包图

由包图扩展出的系统架构类图，如下图 4-4 所示，可以通过该图观察到系统架构中的静态结构图，它体现了每个类相互的静态联系。大致表现出了每一个类所包含的功能要点。

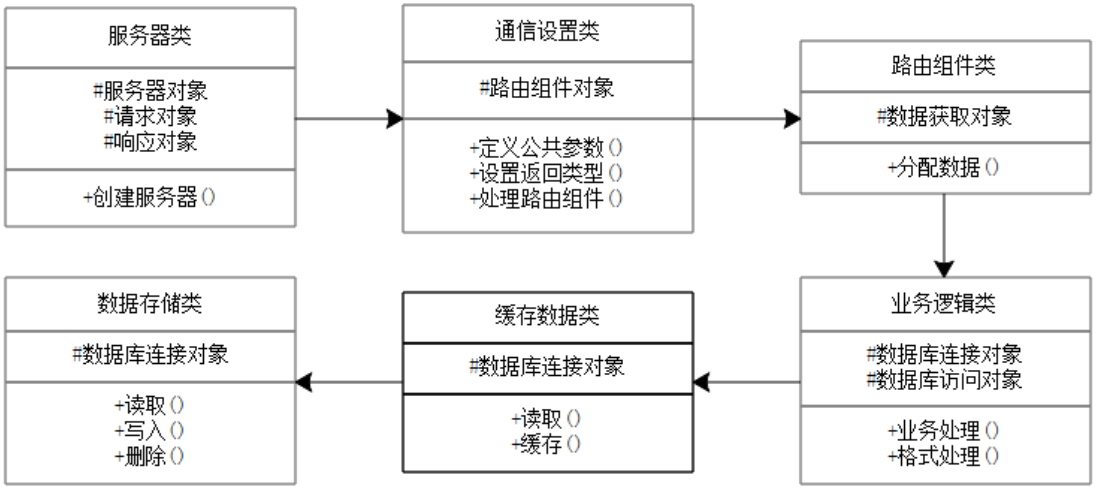

图 4-4 系统架构类图

归纳了系统架构类之后，本人将通过系统架构类的交互图来阐述系统的工作流程，系统架构类的交互图如图 4-5 所示。

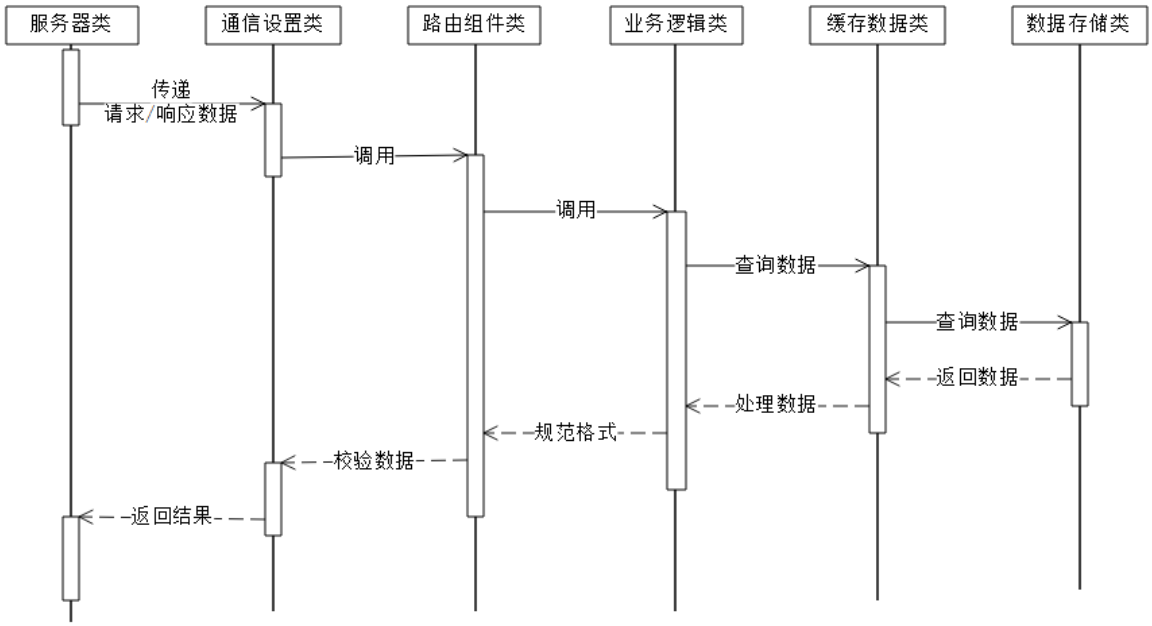

图 4-5 系统架构类的交互图

在此基础上进一步细化得出系统边界图如下图 4-6 所示，通过系统边界图可以帮我们进一步明确整个微博系统中对象与用例之间的关系。

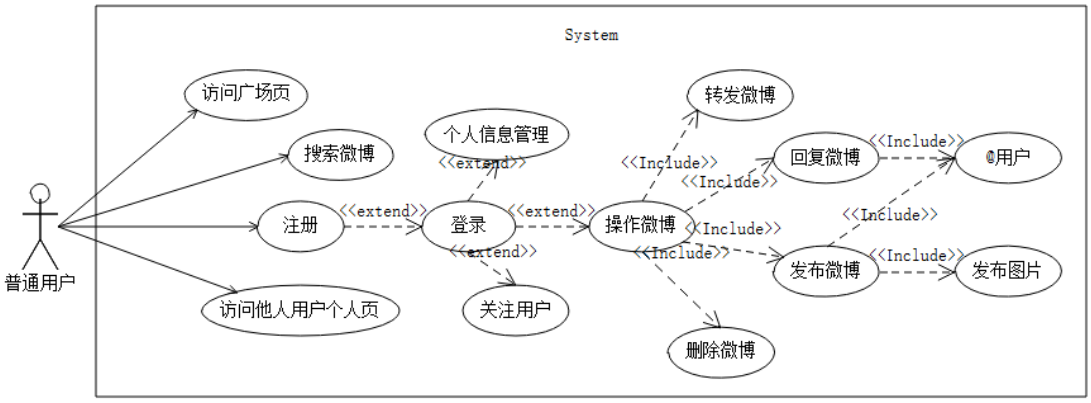

图 4-6 系统边界图

4.2 功能结构设计

本微博系统所实现的众多功能基本都通过功能结构图展示出来了，如下图 4-7 所示。

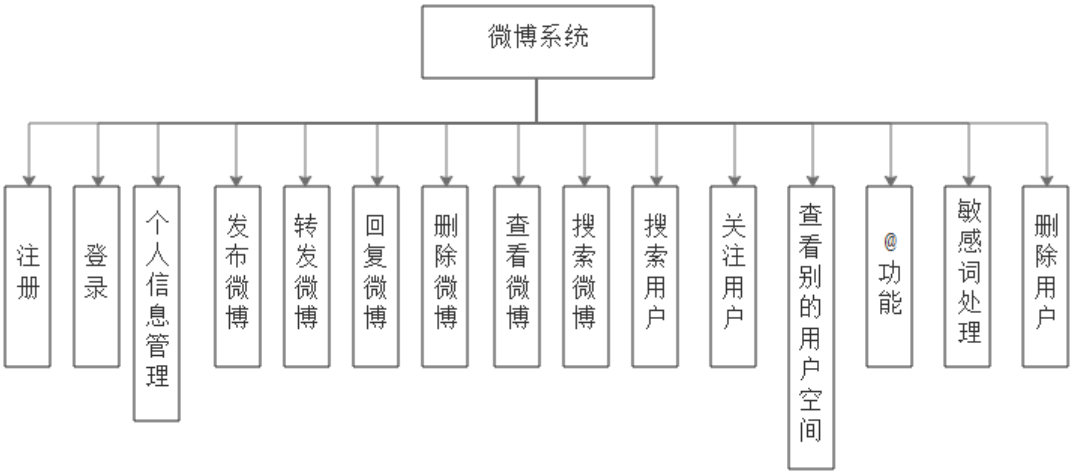

图 4-7 系统的功能结构图

4.3 系统详细设计

微博系统的用户可以进行注册、登录、个人信息的修改、微博的发布、转发、删除、回复。也支持在发布或者回复又或者转发时可以@其他用户。同时可以对自己感兴趣的用户可以进行关注，能看到关注自己的粉丝用户。

用户的用例如图 4-8 所示。

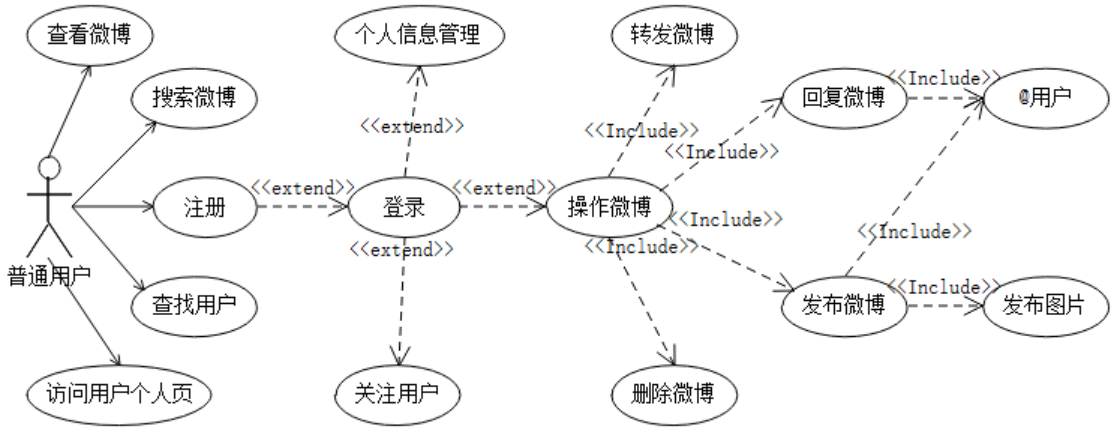

图 4-8 普通用户的用例图

4.4 功能设计

1. 注册

本系统采用的是用户名和密码的方式进行注册，不允许同一用户名进行重复注册。用户信息使用 mysql+redis 中 session 来进行存储，mysql 中的 users 表用于存储用户的相关信息。具体的注册流程如下图 4-10 所示。

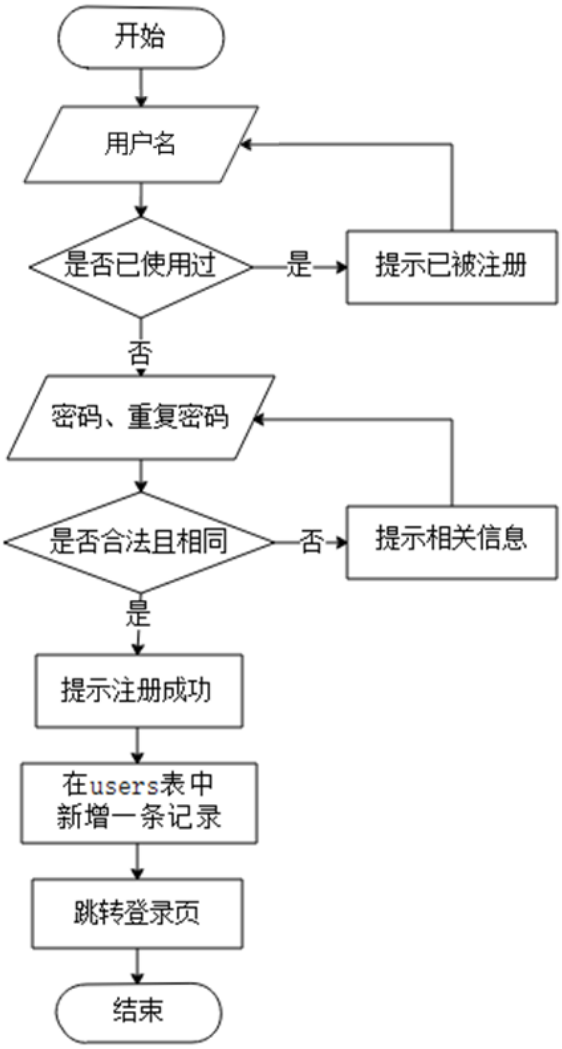

图 4-10 注册流程图

登录

用户可使用自己的用户名和密码登录本系统，系统会对输入的用户名和密码进行校验来判断是否登录成功。

如果 redis 中的 session 并未过期，可以从中获取相关信息，则可直接判断为已登录。

如过 redis 中拿不到相关数据，则到 mysql 中查询该用户是否存在，存在则加入到 redis 中并设置一个过期时间。具体流程如下图 4-11 所示。

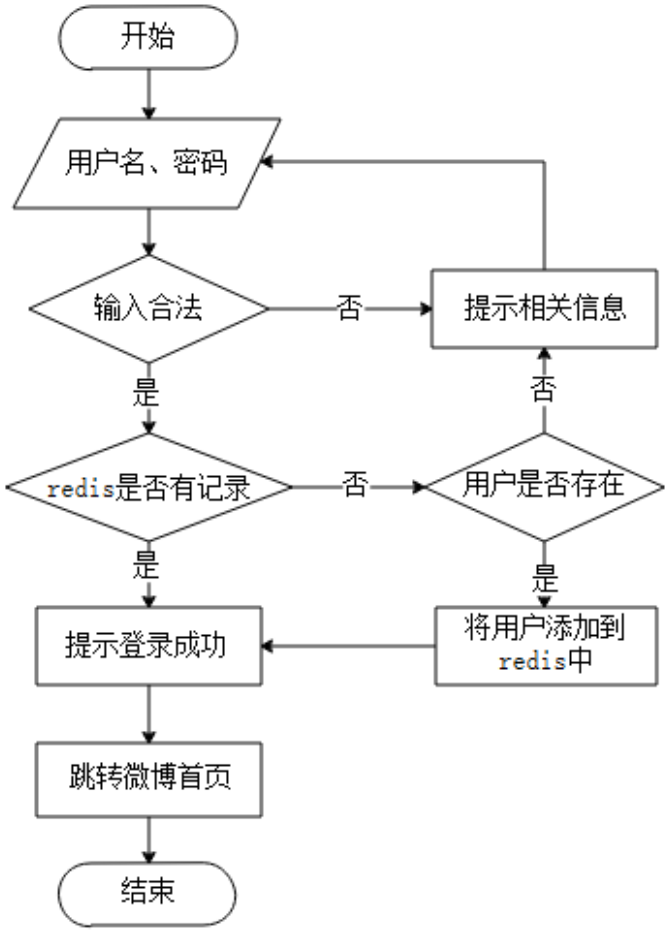

图 4-11 登录流程图

微博列表

我将微博列表通过 mysql 的“blogs”表进行保存。

当用户未登录时显示的是广场页所有人最新的微博组成的列表，用户登录后则显示用户自己的微博和他所关注的用户的微博组成的列表。

用户下滑到一定位置时，如果后面还有数据，则显示 “查看更多”按钮，点击可继续查看下一页微博列表。

具体流程如下图 4-12 所示。

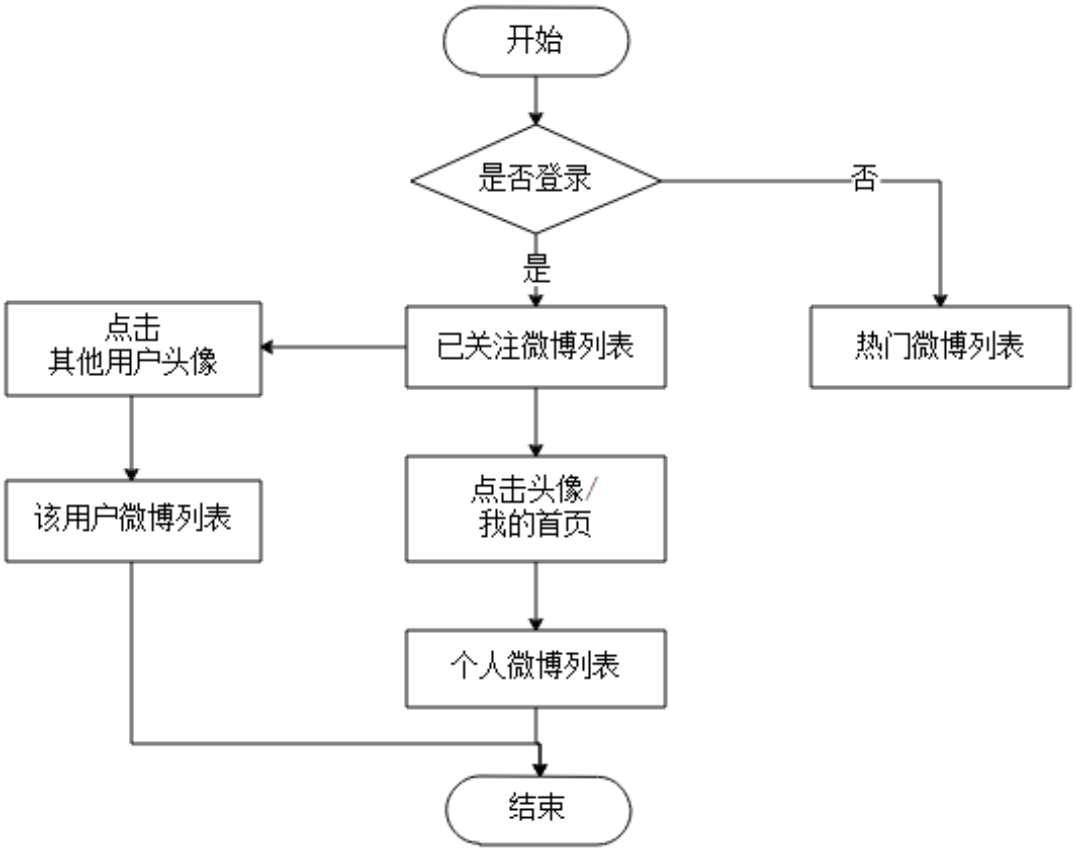

图 4-12 微博列表流程图

发布微博

发布的微博信息也存储在 mysql 的“blogs”表中。

操作之前需要先检测用户是否登录，未登录的提示并跳转登录页面，登录则允许输入微博内容，发布前检查是否合法（140 字以内并不含关键字/敏感词）。

具体流程如下图 4-13 所示。

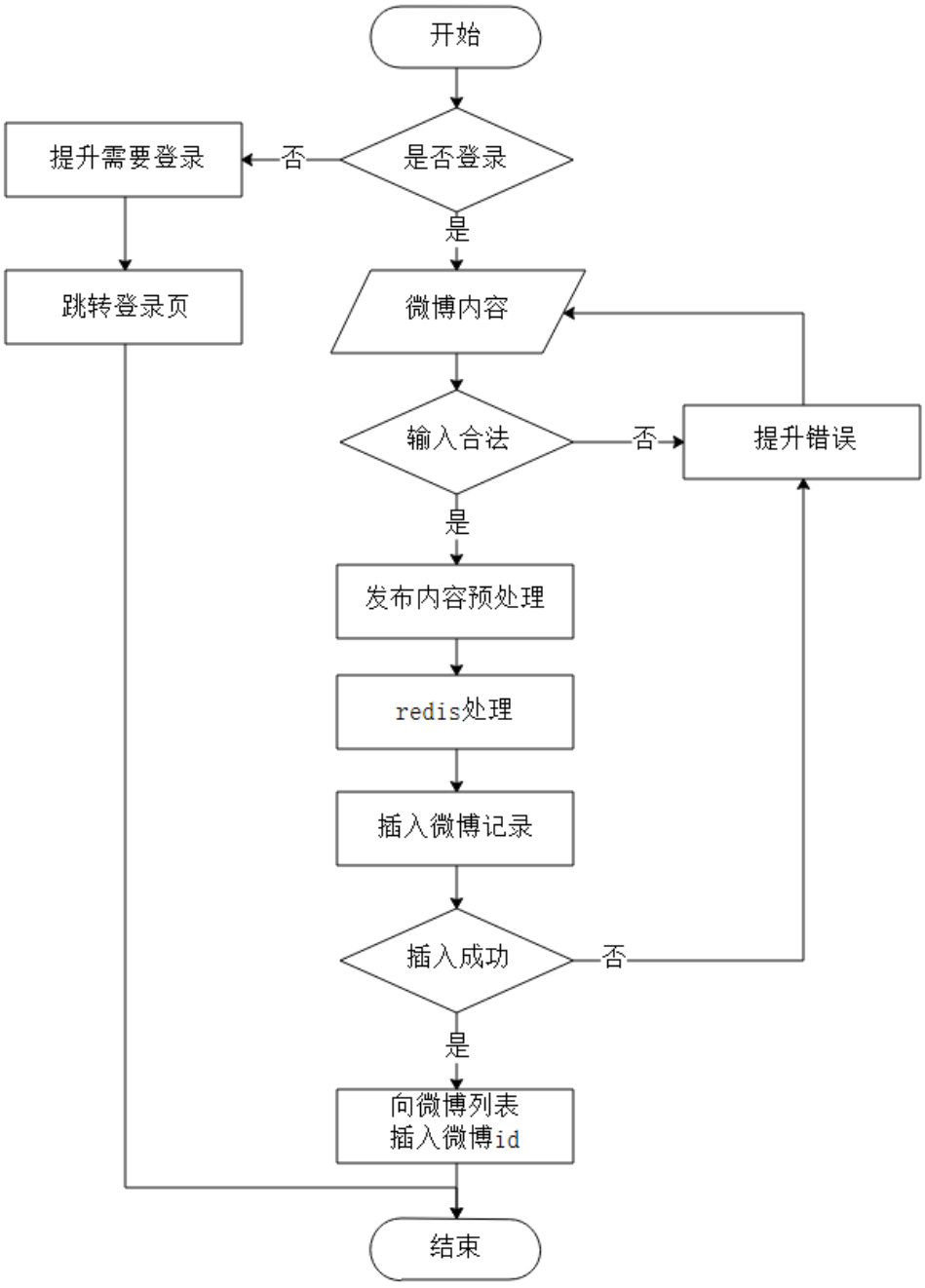

图 4-13 发布微博流程图

关注、取消关注

关注和取消关注的按钮只在他人用户界面显示，默认显示关注按钮，如果是已关注的用户则出现“取消关注”按钮。

点击按钮后我会先检测用户是否已登录，游客状态下是无法进行这两个行为的。

具体流程如下图 4-14 所示。

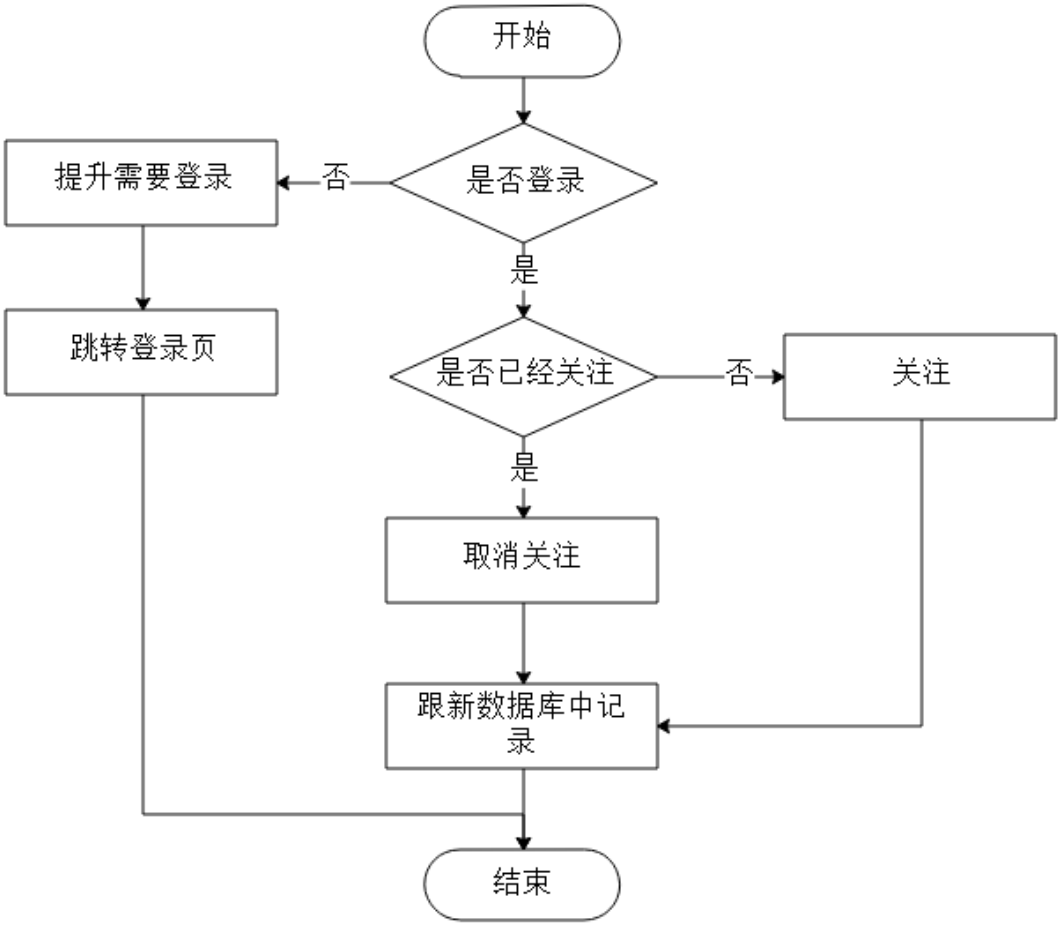

图 4-14 关注/取消关注流程图

转发微博

转发的微博存储在 mysql 的“blogs”表中，只是“isShared”字段做出修改，并提示分析成功，一样只有登录用户才可进行相关操作。

具体流程如下图 4-15 所示。

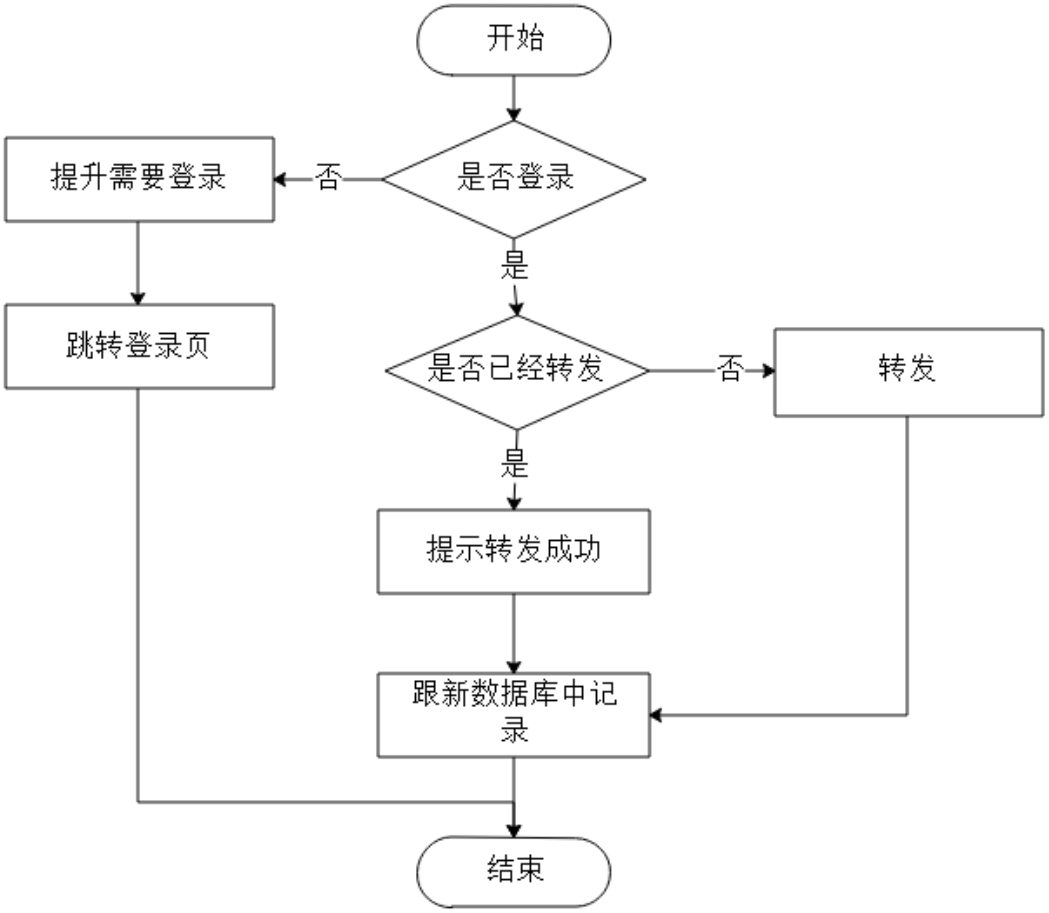

图 4-15 转发微博流程图

回复微博

回复前需检测用户是否登录，未登录的提示并跳转登录页面，登录则允许输入回复内容，回复前检查是否合法（140 字以内并不含关键字/敏感词），如合法则提示发布成功。

具体流程如下图 4-16 所示。

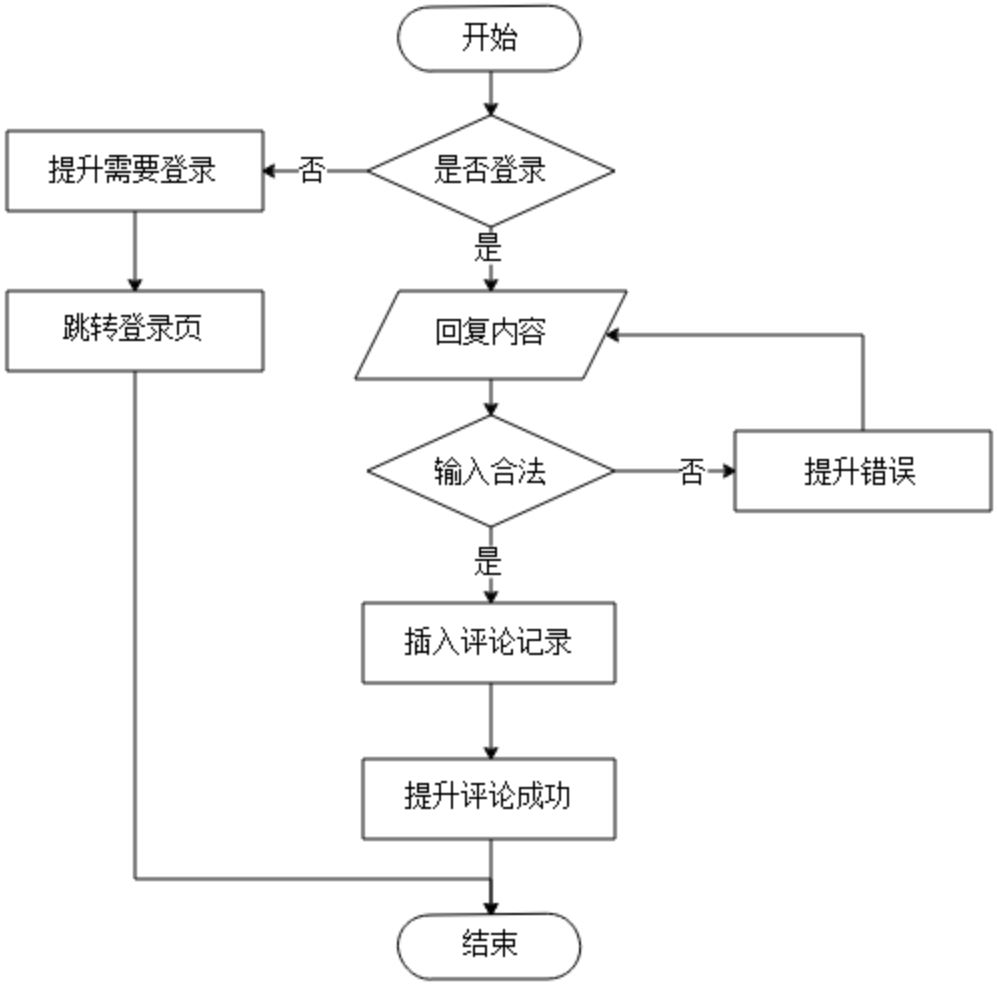

图 4-16 回复微博流程图

4.5 数据库设计

4.5.1 数据库的概念设计

在本系统中，我选择了 mysql 来进行数据存储。根据系统的设计所绘制出的 E-R 模型图如下图 4-17 所示：

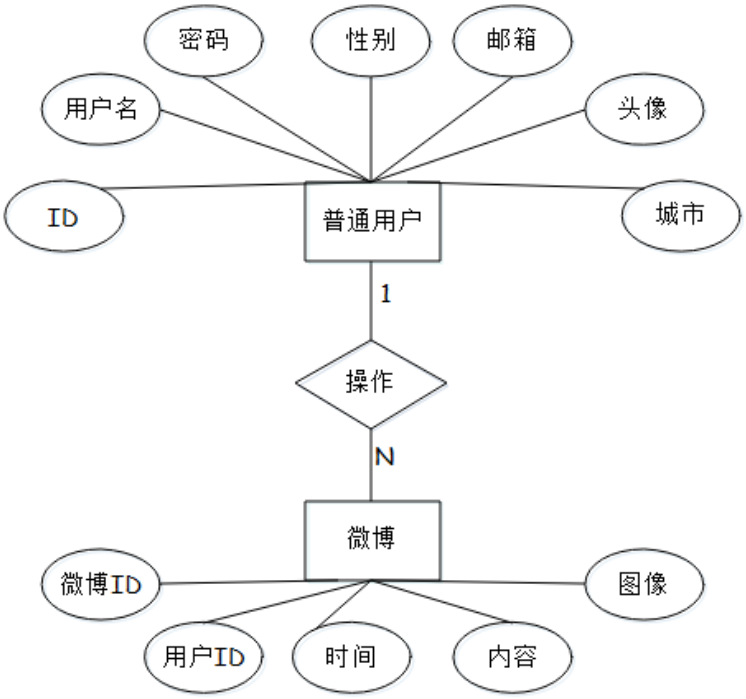

图 4-17 数据库 E-R 图

根据系统的功能需求，将数据库的 E-R 图进行细化后得到各个对象的实体属性图。

（1） 用户实体属性如下图 4-18 所示。

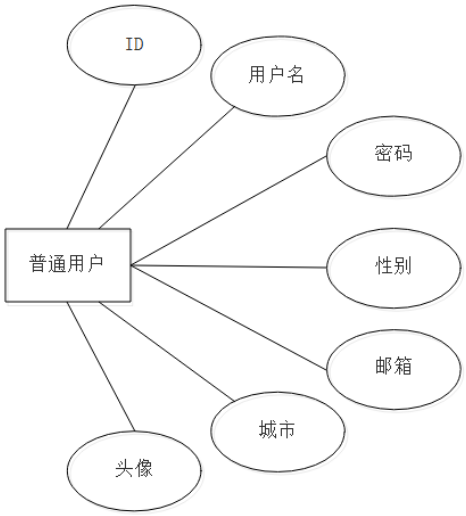

图 4-18 普通用户的实体属性图