# 博客系统网站的设计与实现

# 摘要

互联网时代的到来，使得记录人们生活的方式也发生了变化。为了能更好的记录生活中的点滴，博客应运而生。人们通过博客记录自己的学习心得，抒发自己的想法，记录社会热点。本系统的开发与实现，旨在满足人们对博客的需求，并提供高质量的性能保障，给用户一个舒适的写作环境。

本文详细描述了一个通过Java Web实现的博客管理系统，实现博客管理，用户管理，标签管理，首页搜索等为核心功能的博客系统网站。能实现用户书写博客，博客查询，博客管理，对于管理者能实现用户管理等特色功能。本系统用Spring boot、Elastic Search、MySQL等技术进行开发，前端采用Bootstrap等流行的前端框架实现动态部署，保证了优质的用户体验。

本论文从需求到开发阶段，充分考虑了博客系统网站的实际需求，详细分析了实现的每一个步骤。首先，系统前后台相辅相成，实现了前后端分离的博客系统网站；其次，对于博客的检索和排序进行了认真分析和处理；在系统开发完成后用了简单的黑盒测试和单元测试，均符合并达到预期效果。本系统的设计和开发符合软件的开发流程和相关法规，并且能够满足用户对于一个博客系统网站的需要，具有一定的现实意义。

关键词: 博客系统网站;JavaWeb;MySQL;IntellijIDEA

**Title:** The design and implementation of idle goods trading website

# Abstract

With the advent of the Internet era, the way of recording people's lives has also changed. In order to better record the little things in life, blog came into being. People record their learning experience, express their ideas and record social hot spots through blogs. The development and implementation of this system aims to meet people's needs for blogs, provide high-quality performance guarantee and provide users with a comfortable writing environment.

This paper describes in detail a blog management system implemented by Java Web, which realizes the core functions of blog management, user management, tag management, home page search and so on. Can realize the user write blog, blog query, blog management, for managers can achieve user management and other features. The system is developed with spring boot, elastic search, MySQL and other technologies. The front-end uses bootstrap and other popular front-end frameworks to realize dynamic deployment, ensuring high-quality user experience.

In this paper, from the requirements to the development stage, the actual needs of the blog system are fully considered, and each step of the implementation is analysed in detail. First of all, the front and back of the system complement each other to realize the blog system website with the front and back end separated; secondly, the retrieval and sorting of blogs are carefully analysed and processed; after the completion of the system development, simple black box test and unit test are used, which are in line with the expected results. The design and development of this system conforms to the software development process and relevant laws and regulations, and can meet the needs of users for a blog system, which has certain practical significance.

**Key words**：Blog system；JavaWeb; MySQL; IntellijIDEA

目录

[博客系统网站的设计与实现 1](#_Toc35790319)

[摘要 2](#_Toc35790320)

[Abstract 3](#_Toc35790321)

[第一章 绪论 5](#_Toc35790322)

[1.1课题的来源及意义 5](#_Toc35790323)

[1.2国内外发展状况 5](#_Toc35790324)

[1.3系统可行性 5](#_Toc35790325)

[1.3.1经济可行性 5](#_Toc35790326)

[1.3.2技术可行性 5](#_Toc35790327)

[1.3.3 社会可行性(法律可行性) 6](#_Toc35790328)

[1.4 本文的组织结构图 6](#_Toc35790329)

[1.5本章小结 6](#_Toc35790330)

[第二章 系统分析 8](#_Toc35790331)

[2.1系统业务需求分析 8](#_Toc35790332)

[2.2系统性能需求分析 8](#_Toc35790333)

[2.3系统运行环境需求分析 9](#_Toc35790334)

[2.4系统数据流图 9](#_Toc35790335)

[2.5数据字典 9](#_Toc35790336)

[2.6系统的用例图及用例描述 10](#_Toc35790337)

[2.6.1用例图 10](#_Toc35790338)

[2.6.2用例描述 11](#_Toc35790339)

[2.7本章小结 13](#_Toc35790340)

[第三章 系统设计 14](#_Toc35790341)

[3.1系统功能分析 14](#_Toc35790342)

[3.1.1系统功能概述 14](#_Toc35790343)

[3.1.2系统功能模块设计 14](#_Toc35790344)

[3.2系统模块结构图 15](#_Toc35790345)

[3.3系统主要功能模块流程图 18](#_Toc35790346)

[3.4系统类模型 21](#_Toc35790347)

[3.5数据库设计 22](#_Toc35790348)

[3.5.1数据库实体关系图（E-R图） 22](#_Toc35790349)

[3.5.2数据库总体关系设计 24](#_Toc35790350)

[3.5.3数据库表设计 24](#_Toc35790351)

[3.6 动态建模 25](#_Toc35790352)

[3.6.1 账号管理设计时序图 25](#_Toc35790353)

[3.6.2博客功能设计时序图 27](#_Toc35790354)

[（可以添加人员管理时序图） 29](#_Toc35790355)

[3.6.4 部分功能设计活动图 29](#_Toc35790356)

[3.7 本章小结 30](#_Toc35790357)

[第四章 系统实现 32](#_Toc35790358)

[4.1 本系统开发过程中技术特色说明 32](#_Toc35790359)

[4.2主界面的设计与实现 33](#_Toc35790360)

[4.3前台子系统的设计与实现 33](#_Toc35790361)

[4.4 管理员子系统的设计与实现 33](#_Toc35790362)

[4.5 特色功能的设计与实现 33](#_Toc35790363)

[4.6 本章小结 33](#_Toc35790364)

[第五章 系统测试 33](#_Toc35790365)

[5.1软件测试的目的 33](#_Toc35790366)

[5.2测试的方法 33](#_Toc35790367)

[5.3系统测试 33](#_Toc35790368)

[5.4测试结论 33](#_Toc35790369)

[5.5本章小结 33](#_Toc35790370)

[第六章 结论及展望 33](#_Toc35790371)

[6.1 工作总结 33](#_Toc35790372)

[6.2 工作展望 33](#_Toc35790373)

[致谢 34](#_Toc35790374)

[主要参考文献（资料）： 35](#_Toc35790375)

# 绪论

## 1.1课题的来源及意义

1997年12月17日Jorn Barger创造了术语“网络日志”（weblog），而1999年Peter Merholz，开玩笑的把“weblog”拆分成为“we blog”，从此blog的叫法就流传开来。在博客流行之前人们通过各种方式交流信息。如，在线的聊天室，邮件，BBS等方式。

随着博客技术发展的成熟和服务器商的陆续出现，越来越多的人们开始接触和维护自己的博客。人们通过博客记录生活的点点滴滴，传达政治、经济、科技等的个人思想，分享自己开心和不开心得事情，通过写技术性博客分享自己的所学心得，和网 友们讨论和解决问题，或者分享自己的文学或自己的艺术创作。

博客的出现一方面使得人们的生活发生了改变。人们交流思想一定程度上促进了技术的进步，使得政治更加透明化，丰富了人们的日常生活。

## 1.2国内外发展状况

## 1.3系统可行性

### 1.3.1经济可行性

首先，在本系统的开发过程中所使用的开发环境和开发工具如 IntellijIDEA、 MySQL 等均是开源软件，这在很大程度上节省了经济成本。其次，本系统对服务器 的硬件要求不是很高，因此，开发完成后的系统能够在一般的服务器上运行。由于 Java 语言有跨平台的特性，因此运行系统的服务器可以选择 Linux 操作系统，这将大大节 省系统的授权费用。

综上所述，本系统的开发与实现在经济上切实可行。

### 1.3.2技术可行性

首先，本系统使用集成开发工具 intellij IDEA 进行开发。IDEA 是 JAVA 语言集成开发环境，此工具对社区用户免费，是公认最好的 Java 集成开发工具。Java 语言是现如今计算机编程语言占有率第一位的开发语言，大多数的企业级应用都使用 Java 作为后台开发语言，虽然 Java 语言推出已经二十余载，但依旧如日中天，现如今的多种开发框架使得 Java 可以胜任各式各样的开发要求，无论是桌面级应用还是企业及应用都可以让用户和开发者满意，所以本系统使用 Java 和 Spring 框架进行发。

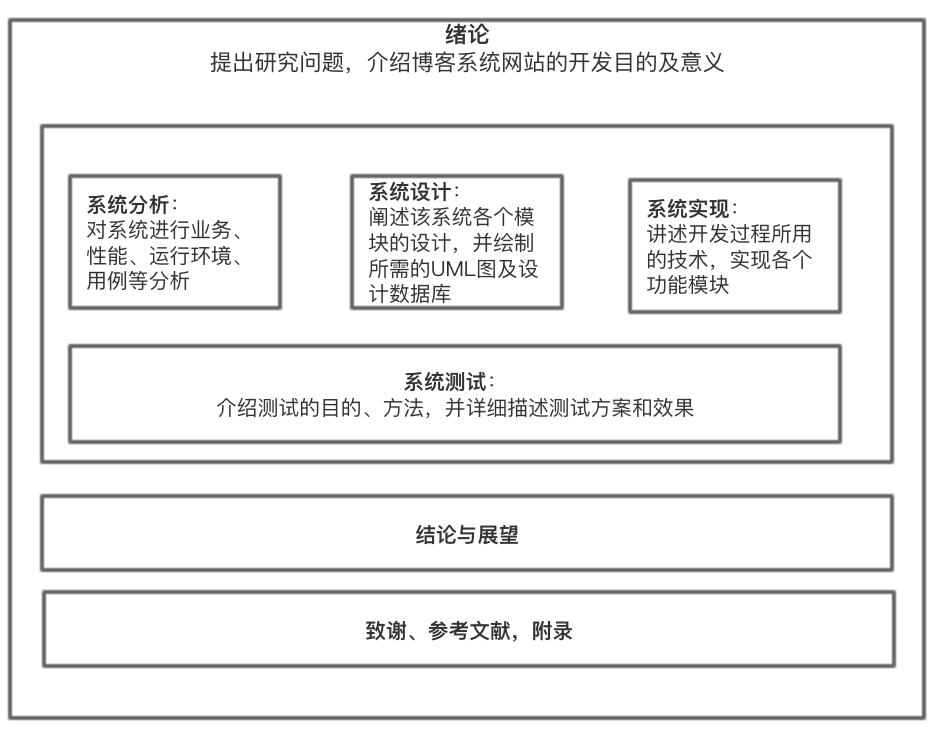
在数据库方面，本系统采用 MySQL 5.7 作为系统数据库。MySQL 是现如今最受欢迎的关系型数据库之一，其特点是开源，使用方便，功能强大。本系统对于数据的存取非常适合使用关系型数据库，使用 MySQL 可以很好的胜任系统对数据库的要求。

### 1.3.3 社会可行性(法律可行性)

博客系统网站的开发，会给一定的用户群体带来很大的便利，社会接受 基础强。网站在开发过程中未侵害任何企业和个人的版权或所有权，并且本系统在开 发过程中使用的开源软件符合开源许可证要求。

综上所述，本系统的开发与实现在法律上切实可行。

## 1.4 本文的组织结构图



## 1.5本章小结

本章节中，首先阐述了博客系统网站这个课题的来源及意义，接着又分别从系统的经济可行性、技术可行性、法律可行性三个方面对该系统进行可行性究。同时，在进行本章任务时，查阅了相关的文献，对该系统的国内外发展现状进行研究。本章作为基础部分，也为后续系统的设计和实现打下了理论基础。

# 系统分析

## 2.1系统业务需求分析

博客系统网站包含的基本业务有：

1. 博客系统一共有八个模块，用户管理模块，博客管理模块，点赞管理模块，分类管理模块，标签管理模块和首页搜索模块。
2. 用户管理模块除了简单的注册和登陆功能外，管理员还可以还有增加用户，修改用户，删除用户的功能。
3. 用户管理模块还应包含安全设置，也就是角色授权和权限设置，其中角色分为管理员和普通博主两类。
4. 博客管理是系统的主要核心功能，业务也最为重要，包含发表博客，编辑博客，删除博客，博客分类，设置博客的标签，模糊查询，最新排序，最热排序，阅读量统计等功能。是实现博客系统的关键环节。
5. 群众的眼睛是雪亮的，好的博客可以得到更多人的认可，所以在博客的阅读中还要有评论管理和点赞管理。评论管理包括发表评论，删除评论评论量统计等功能，点赞管理也应该包含点赞，取消点赞，点赞量统计的功能。
6. 对博客文章分类会使博客系统显得整洁有序，所以还应该具有分类管理和标签管理。分类管理包括创建分类，编辑分类，删除分类，和按分类查询的功能。标签管理包括创建标签，删除标签，按标签查询等功能。
7. 首页，是给用户第一印象和用户访问最频繁的一页，合理设计首页是十分必要的。首页的功能应该包括全文检索，最新文章，最热文章，热门标签，热门用户，热门文章和最新发布等功能。

## 2.2系统性能需求分析

根据上述的业务需求，并且结合实际情况可能出现的性能需求，现在对所需要实现的系统在性能上提出以下几个方面的要求：

1. 系统核心的功能响应时间不能超过200ms
2. 更新处理时间应小于500ms
3. 检索和查找时间应小于1000ms

## 2.3系统运行环境需求分析

系统运行时对于环境的要求有:

#### 1) 硬件环境最低需求

(1) 服务器:

a. 处理器(CPU):1.5GHz 或更快

b. 内存(RAM):2GB 以上

c. 硬盘空间:20GB 以上

d. 接入互联网

(2) 客户端:

a. 处理器(CPU):1.5GHz 或更快

b. 内存(RAM):2GB 以上

c. 分辨率在 800\*600 以上的显示器

d. 2M 以上宽带

e. 键盘及鼠标等输入设备

#### 2）服务器软件环境需求

JDK版本：JDK 8  
软件托管版本：Gradle 3.5  
Spring Boot版本：Spring Boot 1.5.2.RELEASE  
Thymeleaf 版本：Thymeleaf  3.0.3.RELEASE  
数据库版本：MySQL Community Server 5.7.17

## 2.4系统数据流图

## 2.5数据字典

表2-1数据结构

|  |  |  |
| --- | --- | --- |
| 数据结构名称 | 含义说明 | 组成 |
| 登陆用户 | 通过浏览器访问系统的已注册人 | 用户名，密码，角色，注册时间等可用用户信息 |
| 管理员 | 可使用系统后台用户管理的用户 | 用户名，密码，角色等 |
| 博客 | 登陆用户写的内容 | 标题，内容，作者，分类，标签等 |
| 评论 | 博客的评论 | 作者，时间，内容等 |

## 2.6系统的用例图及用例描述

### 2.6.1用例图

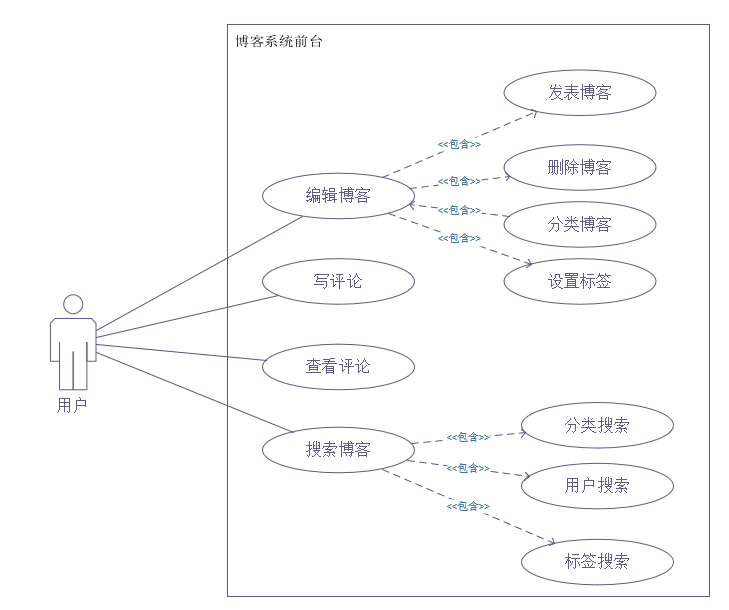


图2-1博客系统前台用例图

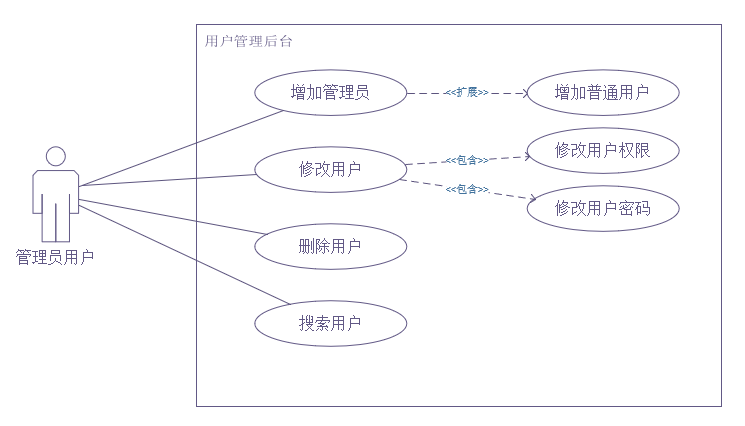


图2-2博客系统后台用户管理用例图

### 2.6.2用例描述

#### 1）用户注册

当用户访问本网站时可以注册，注册的用例描述如表2-2所示：

表2-2 注册用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户注册 | |
| 用例ID | UC1 | |
| 参与者 | 普通用户 | |
| 描述 | 普通用户注册系统过程 | |
| 优先级 | 高 | |
| 前置条件 | 用户未注册 | |
| 后置条件 | 将用户注册信息保持至数据库 | |
| 主事件流 | | |
| 用户 | | 系统 |
| 1.进入用户注册页面 | |  |
| 2.用户输入用户名，密码邮箱等信息 | |  |
|  | | 3.系统检查用户名是否存在 |
|  | | 4.系统检查邮箱格式是否合法 |
|  | | 5.系统检查两次密码输入是否一致 |
|  | | 6.提示用户注册成功，转向主页面 |
| 异常流 | | |
| 异常流 | | 该用户已在系统中存在 |
| 系统 | | 给出提示信息“该用户已在系统” |
| 异常流 | | 邮箱格式不合法 |
| 系统 | | 给出提示“请正确输入邮箱” |
| 异常流 | | 两次输入密码不一致 |
| 系统 | | 给出提示信息“两次输入的密码不相同” |

2）用户登陆

注册完的用户可以通过邮箱，账号密码登陆，登陆之后可以享受用户的基本权限，登陆的用例描述如表2-3所示：

表2-3登陆用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户登陆 | |
| 用例ID | UC2 | |
| 参与者 | 普通用户 | |
| 描述 | 普通用户登陆系统过程 | |
| 优先级 | 高 | |
| 前置条件 | 用户未登陆 | |
| 后置条件 | 将用户信息放入会话中 | |
| 主事件流 | | |
| 用户 | | 系统 |
| 1.进入用户登陆页面 | |  |
| 2.用户输入用户名，密码 | |  |
|  | | 3.系统检查用户名和密码是否存在 |
|  | | 4.将用户信息放到会话中 |
|  | | 5.返回主页面 |
| 异常流 | | |
| 异常流 | | 用户名与密码不匹配 |
| 系统 | | 给出提示信息“用户名或密码错误” |

1. 用户管理

用户管理包括用户查询、用户添加、用户删除等功能，用户查询可以通过邮箱、用户名等条件进行不同程度的查询，用户管理用例描述如2-4所示。

表2-4用户管理用例描述

|  |  |  |
| --- | --- | --- |
| 用例名称 | 用户管理 | |
| 用例ID | UC3 | |
| 参与者 | 普通用管理员 | |
| 描述 | 管理员进行用户管理 | |
| 优先级 | 高 | |
| 前置条件 | 管理员主动进入用户管理页面 | |
| 后置条件 | 成功管理用户 | |
| 主事件流 | | |
| 管理员 | | 系统 |
| 1.点击用户管理页面链接 | |  |
|  | | 2.显示用户管理主页面，包括用户增加，用户修改和用户删除功能 |
| 可选事件流 | | |
| 用户查询 | | |
| 管理员 | | 系统 |
| 输入用户名，点击查询 | | 显示符合条件的用户 |
| 用户删除 | | |
| 选中要删除的用户，点击后面的删除按钮 | | 删除目标用户 |

## 2.7本章小结

本章的主要内容是系统的需求分析，以上内容对系统的基本功能需求进行了分析， 介绍了系统完整体验所需的硬件及软件要求。本章展示了系统的基本数据流图和一层 数据流图，编写了数据字典，介绍了系统在开发和使用过程中涉及的部分易歧义的数 据项。在本章的最后，展示了系统的用例图，介绍系统的用例关系，详细进行了用例 描述，为后续步骤做好准备。

# 系统设计

## 3.1系统功能分析

### 3.1.1系统功能概述

根据系统的需求分析，结合实际情况具体细化，本网站要实现的功能主要有：

1. 用户可以通过网站前台实现登陆和注册的功能
2. 用户可以通过浏览器浏览已经发布的博客
3. 用户登陆后可以编辑和修改博客并进行发布
4. 用户可以对博客进行评论和点赞
5. 用户可以删除博客
6. 网站有搜索功能可以对用户和博客进行查找
7. 网站具有后台管理系统供管理员使用
8. 管理员可以通过后台管理系统，对用户进行查找，修改信息和删除操作

### 3.1.2系统功能模块设计

针对上述功能，设计了七个模块分别是用户管理模块，安全设置模块，博客管理模块，评论管理，分类管理，标签管理和首页搜索。其中核心模块是博客管理模块，博客管理模块包括，发布博客，编辑博客，删除博客，博客分类，设置标签和搜索博客等功能。用户管理除了常规的登陆功能和注册功能外，还应该有增加用户删除用户，搜索用户的功能。对数据库的操作要进行安全设置，对角色进行分类和权限管理。在分类上也需要做博客类型的区分，分为两种，博客文章分类和标签设置。

还有比较重要的就是主页的搜索功能，首页的搜索功能主要包括，全文检索，最新文章，最热文章，热门标签，热门用户，热门文章和最新发布。

## 3.2系统模块结构图

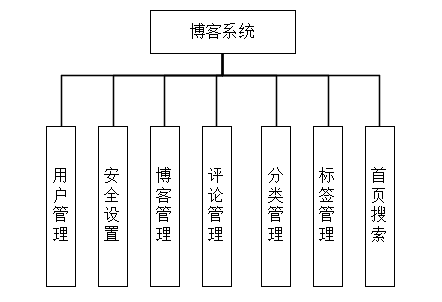


图3-1系统总体模块结构图

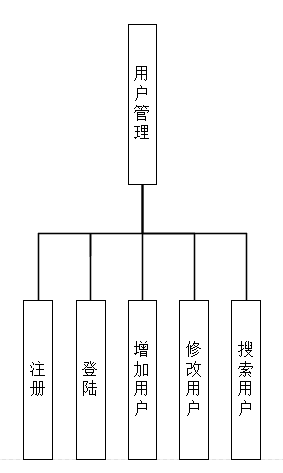
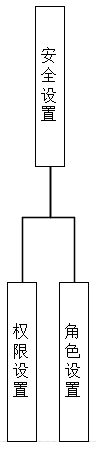
 

图3-2 用户管理功能结构模块图 图3-3安全设置功能结构模块图

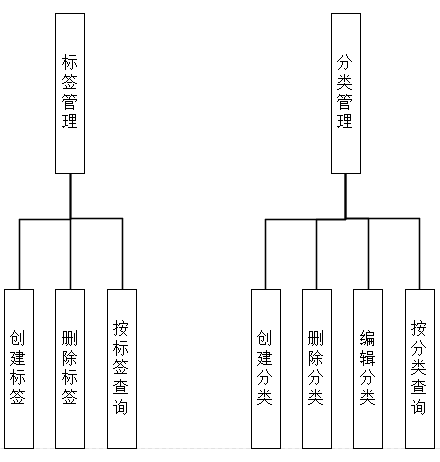


图3-4 标签管理功能模块结构图 图3-5分类管理功能模块结构图

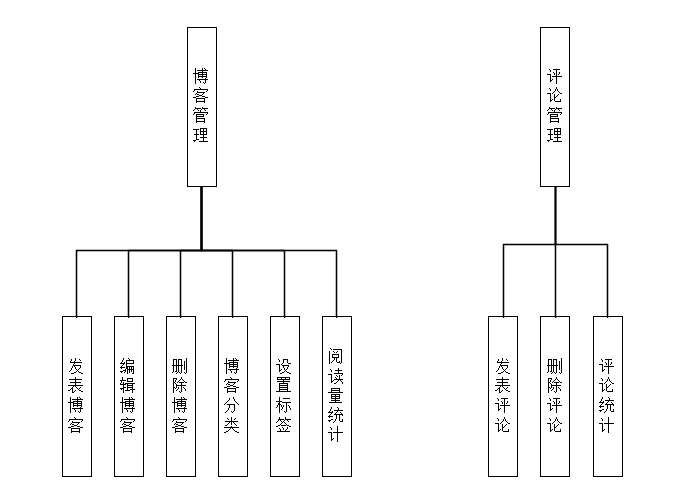


图3-6博客管理功能模块结构图 图3-7评论管理功能模块结构图

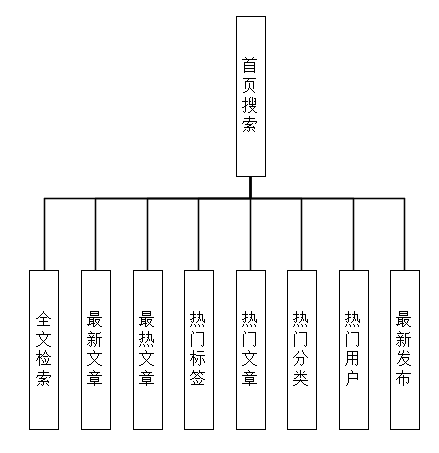


图3-8 首页搜索功能模块结构图

## 3.3系统主要功能模块流程图

1）博客管理模块流程图



图3-9博客管理模块流程图

2）博客评论管理模块流程图



图3-10博客评论管理模块流程图

3）用户管理模块流程图



图3-11用户管理模块流程图

## 3.4系统类模型

因系统关系调用均相似，故以user类为例子介绍关系之前的调用关系，如图3-12所示：

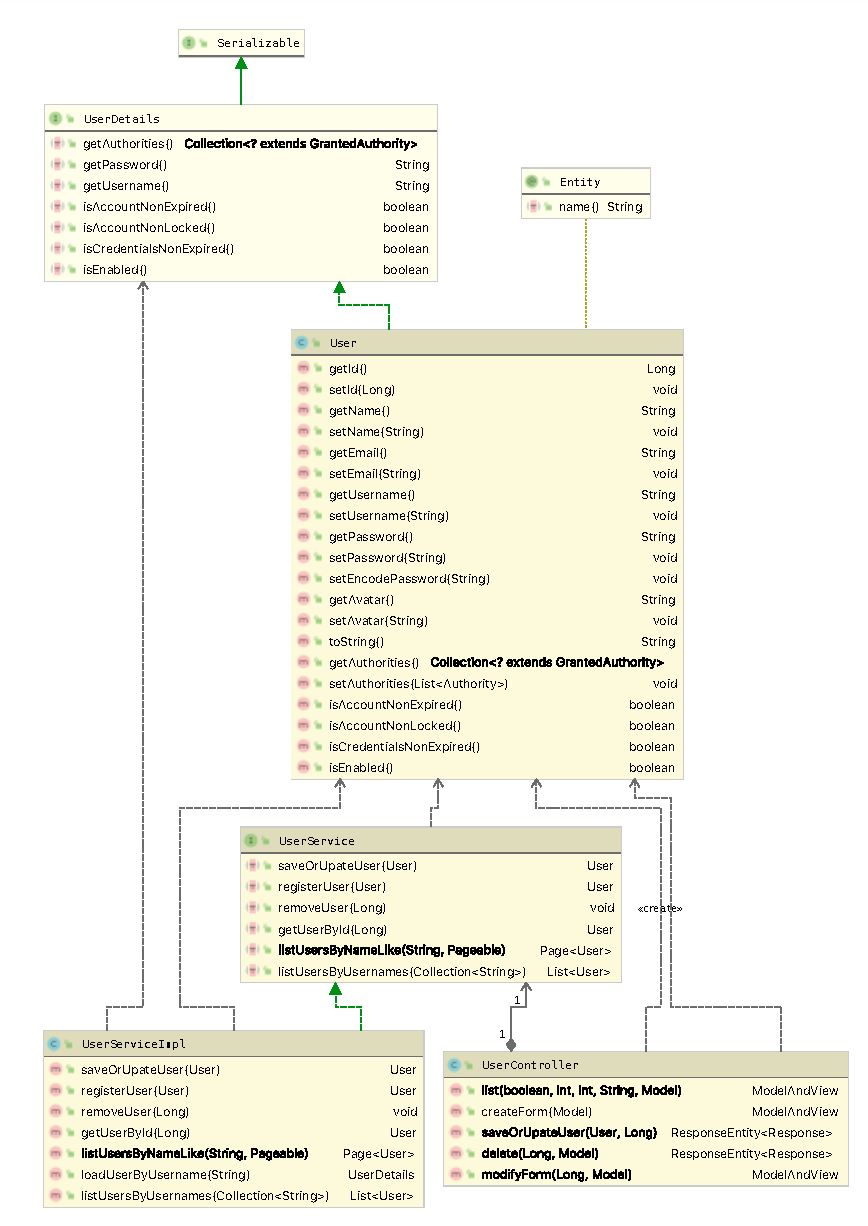


图3-12部分系统类模型

## 3.5数据库设计

### 3.5.1数据库实体关系图（E-R图）

博客系统网站主要的实体有用户、博客、点赞、评论等，每个实体都有相应的属性，实体之间的关系如下：

1）博客系统网站总体E-R图



图3-13 博客系统网站总体E-R图

以下分别展示每个实体与属性的E-R图

2）用户E-R图



图3-14 用户E-R图

设计该实体的理由如下：用户作为网站的根本，几乎所有的实体都以用户为核心，以用户ID作为实体之间的关联，用户实体包含用户的基本属性，认证状态等信息。

3）博客E-R图



图3-15博客E-R图

设计该实体的理由如下：博客作为网站的核心，是网站的重要实体，除了标题，作者和内容外，为了进行分类，还应该有标签和分类。每个博客都有属于自己唯一的博客ID，阅读量统计等。

3）评论E-R图



图3-16评论E-R图

设计该实体的理由如下：评论是用户对网站中博客的评论，所以评论必须包含创建时间，评论内容和被评论的博客。

4）点赞E-R图



图3-17 点赞E-R图

设计该实体的理由如下：相比起评论，点赞更能反应一篇博客的好坏，降低了读者的评论成本，而且实体简单，只需要有创建时间。

### 3.5.2数据库总体关系设计

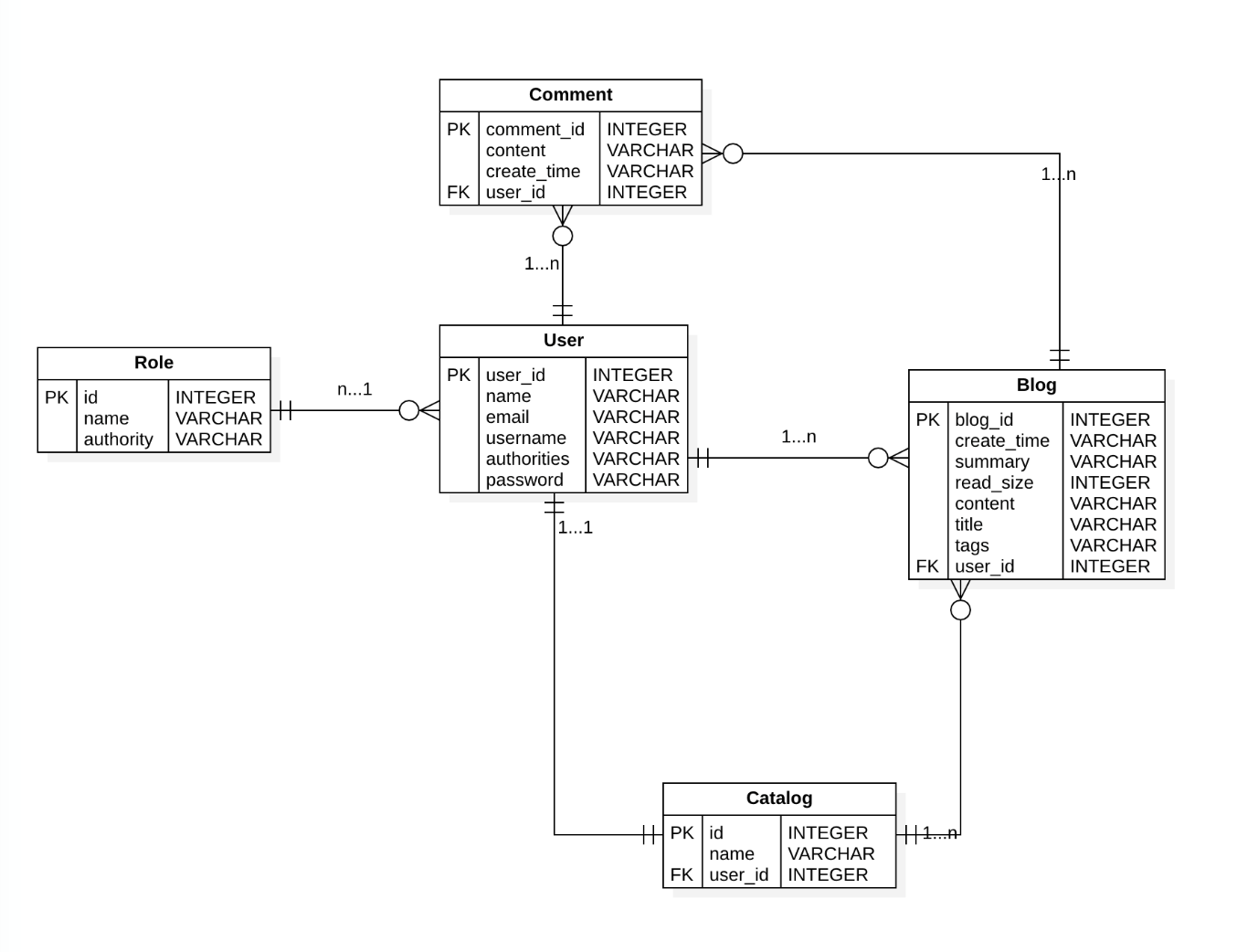


图3-18 数据库总体关系图

### 3.5.3数据库表设计

在设计程序之前需要设计好系统所涉及的数据的表结构并将系统用到的模拟数据存储在数据库中，用于的系统开发和测试。基于以上分析得出的 E-R 图，本网站的数据库表为以下详细表结构。

1）使用网站前台系统的用户在数据库存储时对应的是 users 表，表结构设计下。

表3-1 users

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| user\_id | int | 9 | not |  | yes |  | 用户编号 |
| role\_id | int | 9 | not |  |  | yes | 角色编号 |
| moblie | varchar | 11 |  |  |  |  | 用户电话 |
| email | varchar | 50 | not |  |  |  | 用户邮箱 |
| password | varchar | 50 | not |  |  |  | 用户密码 |
| sex | tinyint | 1 |  | 0 |  |  | 用户性别 |
| birthday | date |  |  |  |  |  | 用户出生日期 |
| create\_time | datetime |  |  |  |  |  | 用户创建时间 |
| is\_avalible | tinyint | 1 |  | 1 |  |  | 账号状态 |
| pic\_img | varchar | 255 |  |  |  |  | 用户头像 |
| last\_login\_time | datetime |  | not |  |  |  | 上次登陆时间 |

该表主键为 user\_id 字段，该字段为 int 类型，设置为自动递增，插入时数据库自 动完成赋值，保证主键的唯一性，其他的信息将会用于登陆等业务的数据存储操作。

2）在本系统中，另外一个重要的用户是管理员用户，管理员用户是，它包括的各个字段、类型等如下。

表 3-2 role表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 属性名 | 类型 | 长度 | 空值 | 缺省值 | 主键 | 外键 | 说明 |
| role\_id | int | 9 | not |  | yes |  | 角色编号 |
| role\_name | varchar | 20 | not |  |  |  | 角色名 |
| last\_login\_time | datetime |  | not |  |  |  | 上次登陆时间 |

role表中role\_id 作为主键，包含角色基本信息的字段，角色类型分为最高权限管理员、用户管理员、普通用户，分别以“0”， “1”，“2”表示。last\_login\_time 字段在管理员每登录一次后自动更新，保证账号安全。

**3）博客系统网站中的博客主要指用户发布的网络日志，es**

## 3.6 动态建模

在动态建模机制中，以消息来完成对象之间的交互，用状态图、时序图和活动图 来描述系统的行为。

时序图是一种交互图用于描述执行系统功能的各个角色之间相互传递消息的顺 序关系，显示跨越多个对象的系统控制流程，强调的是时间和消息的次序，用来说明 系统的动态情况。

博客系统网站的各个功能模块的动态建模如下。

### 3.6.1 账号管理设计时序图

账号管理模块中包括账号注册、账号登录、账号信息的修改。登录账号和修改账 号信息均在用户注册账号后，修改账号信息需要用户的登录。

1. 注册账号

用户想登录到本网站并使用网站所提供的服务，必须先注册一个账号，管理员账号不需要注册，是在系统完成时，由管理人员在后台用户管理系统手动添加到数据库中的。用户的账号 是通过账号注册页面提交申请成功后获得的。用户账号存在唯一性，即一个账户只能 属于一个用户，相同的账号在相同的角色下不能重复注册，并且账号密码符合网站的 要求，账号注册的时序图如图所示。

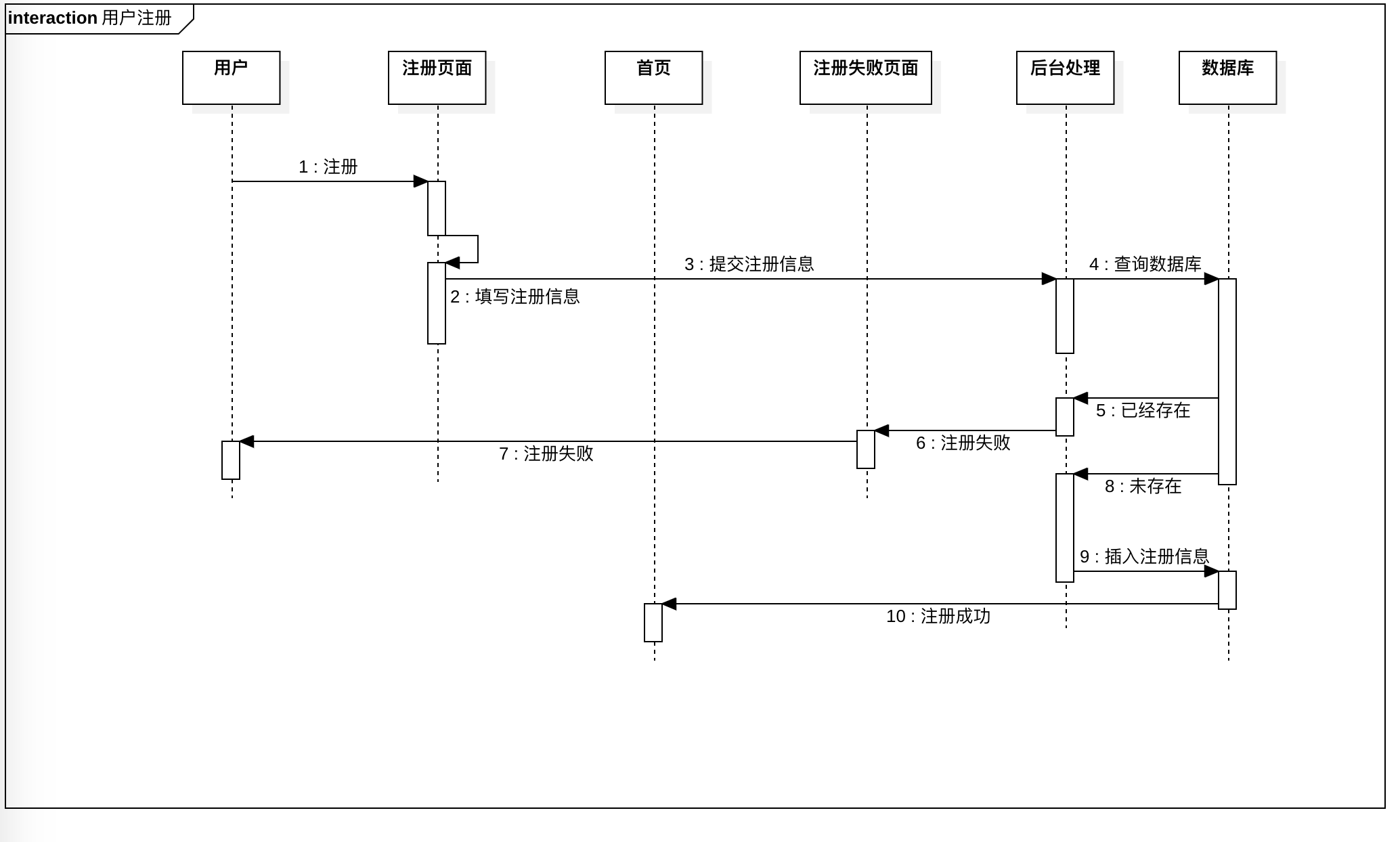


图3-21 注册账号时序图

2)账号登录

账号登录是将用户填写到登录页面表单中的账号、密码提交到业务逻辑层，业务逻辑层在获得登录信息后到数据库中相应的表中查询该用户信息是否存在，若存在该用户则允许用户登录，系统跳转至首页，若不存在该用户则不允许用户登录系统，跳转至登录页面中，重新登陆，登录过程的时序图如图 3-22所示:

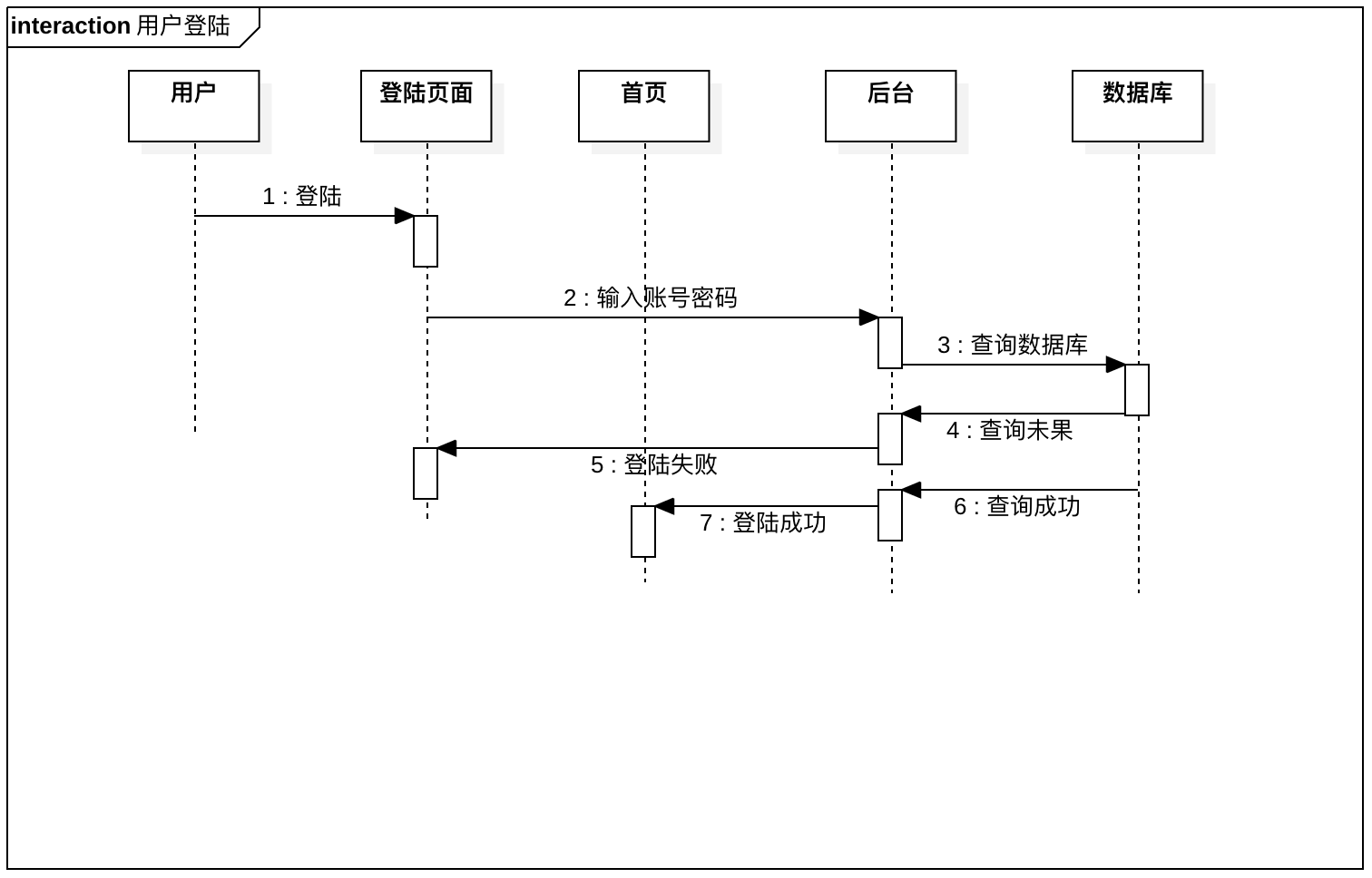


图3-21 用户登陆时序图

### 3.6.2博客功能设计时序图

用户完成登陆后可以进入博客编辑页面，进行博客的编写，修改和删除，以及博客阅读量的查看。对博客的操作是博客系统网站的主要功能

1. 编辑博客

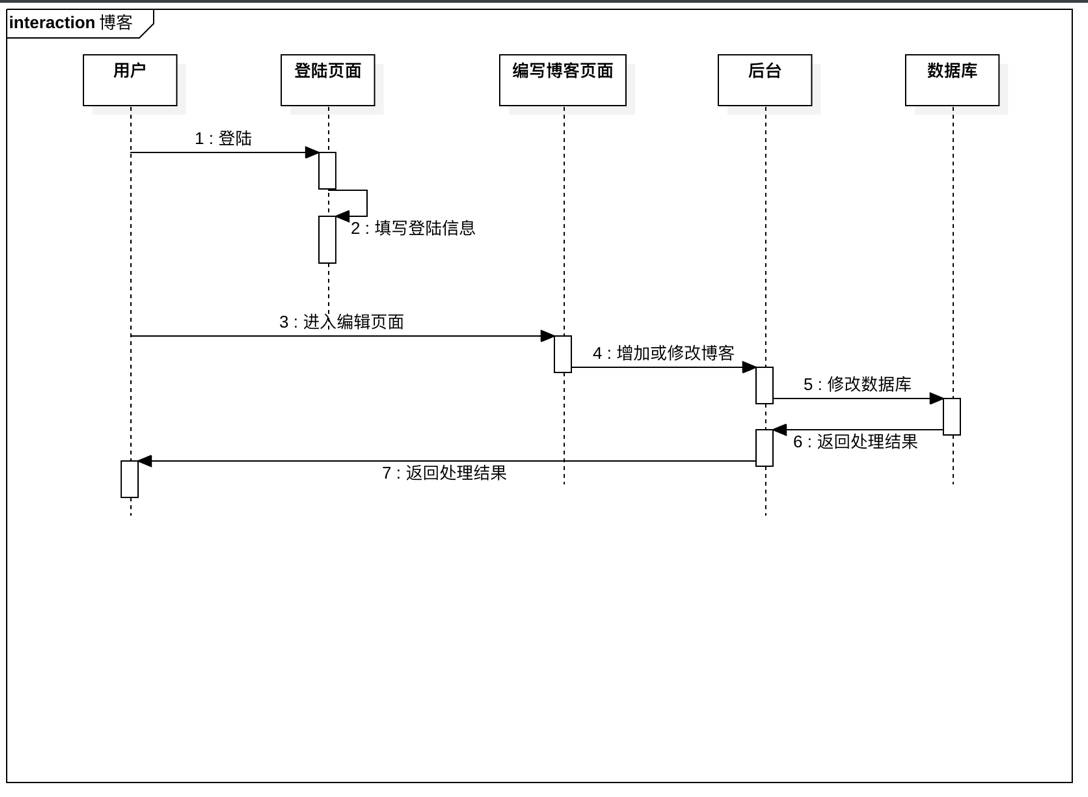


图3-22 博客编辑时序图

博客的编辑需要用户登陆账号信息，在登陆完成后进入博客编辑页面，对目标博客或者新建博客进行博客内容的编辑。

1. 删除博客

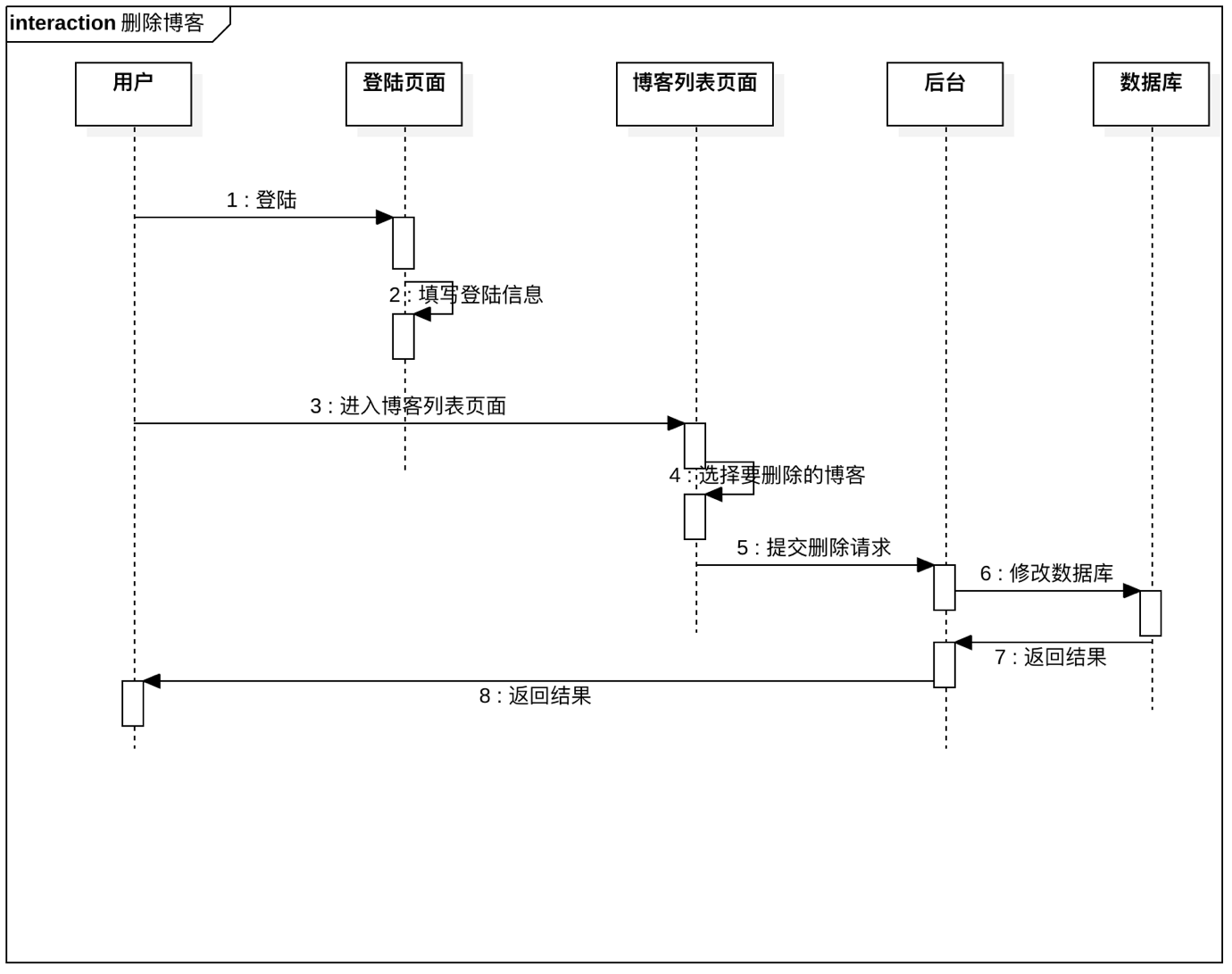


图3-23 博客删除时序图

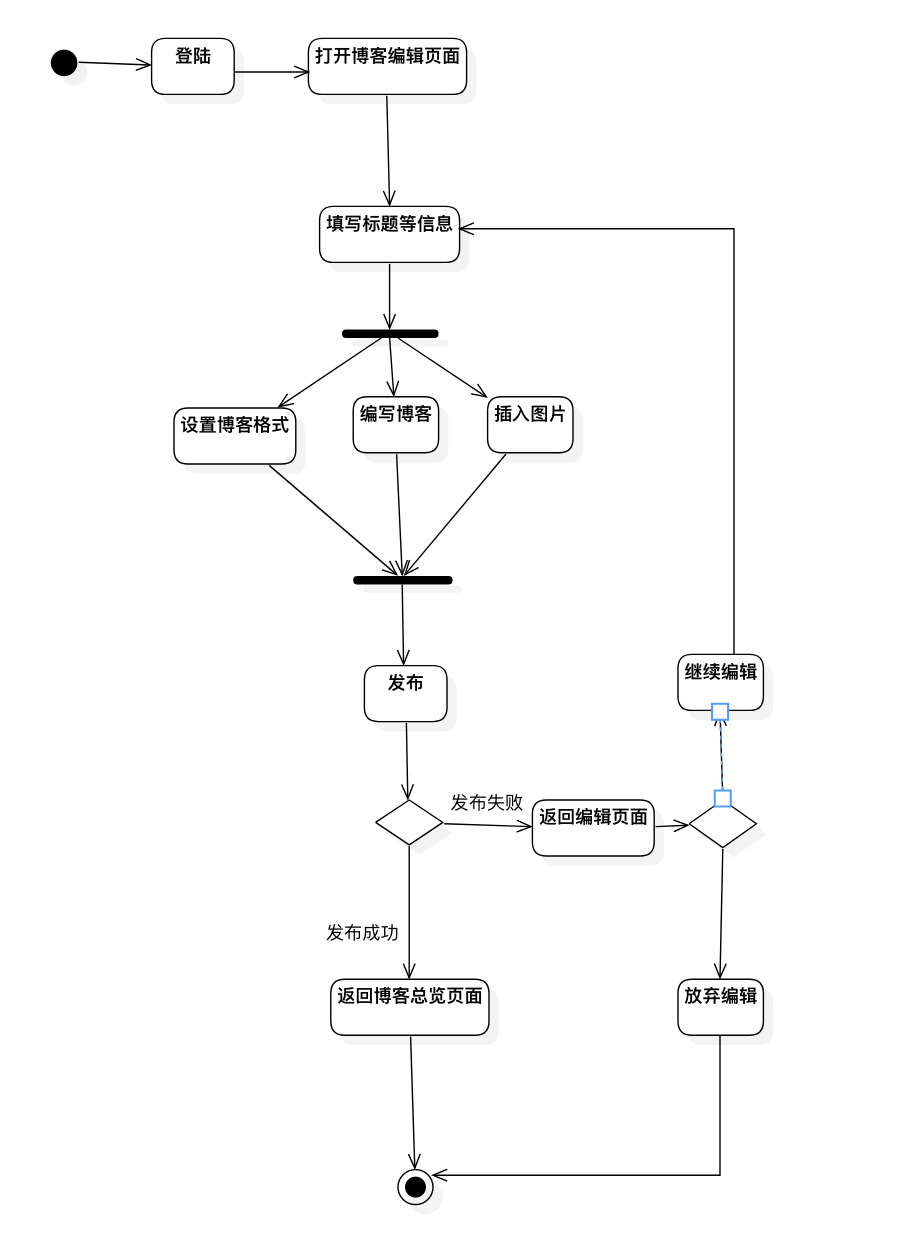
在拥有博客编辑的功能后，删除博客的功能当然必不可少，同样需要登陆账号，在登陆账号之后，进入往期编写的博客页面列表，选择目标博客，点击删除按钮。

（可以添加人员管理时序图）

3.6.4 部分功能设计活动图

活动图(activity diagram，动态图)是阐明了业务用例实现的工作流程。业务工作流程说明了业务为向所服务的业务主角提供其所需的价值而必须完成的工作。业务用例由一系列活动组成，它们共同为业务主角生成某些工件。工作流程通常包括一个 基本工作流程和一个或多个备选工作流程。工作流程的结构使用活动图来进行说明。

1)在用户编辑博客中，第一个动作是登陆，第二个动作是进入编辑页面并填写博客标题等信息，第三个动作是编写博客，从这个动作结 点出发，有三个流出边分别为“编辑博客”、“设置博客格式”、“插入图片”。这三个步骤 可以不分先后而分别进行，形成三个控制流最终在一个汇合点上同步，汇成一个控制流。第四个动作是点击发布，该动作由“发布成功”和“发布失败”两个判断条件产 生了两个分支，如果发布成功，返回博客总览页面;如果发布失败，则再次回到编辑页面，重新填写信息然后在一个合并点上，两个分支合并在一起，最后以活动结束符号表示活动 的终结。



编辑博客活动图

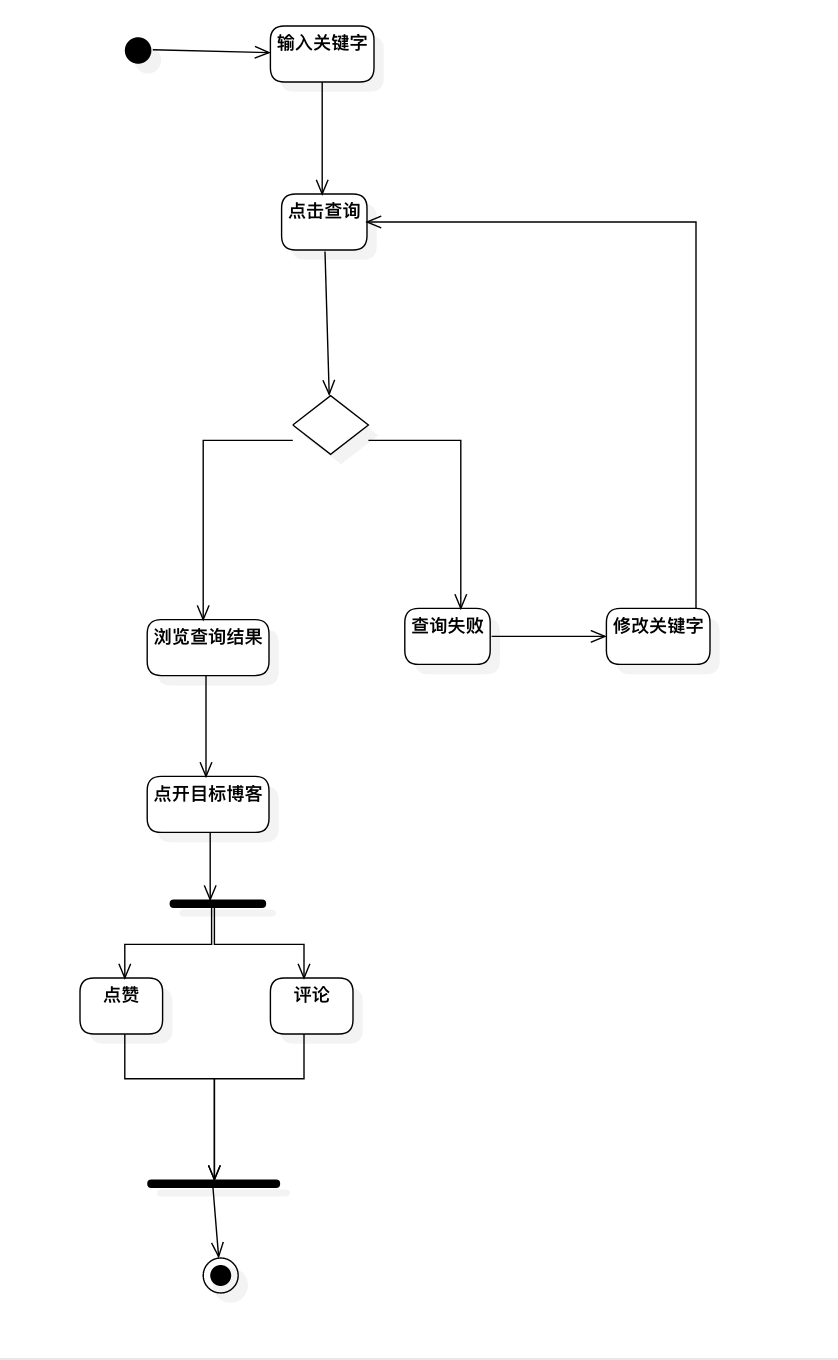


图 3- 博客查询浏览活动图

2)在博客查询浏览活动图中，第一个动作是输入关键字，第二个动作是点击查询，由“查询成功”和“查询失败”两个判断条件产生了两个分支，第一个分支:如果查询失败,则需要重新输入关键字查询,如果查询成功，那么就可以浏览查询到的博客，选中目标博客把它打开。打开后分为两个动作，“点赞”和“评论”这两个步骤可以不分先后而分别进行，形成两个控制流最终在一个汇合点上同步，汇成一个控制流。最后以活动结束符号表示活动的终结，- 博客查询浏览活动图如图 3-27 所示。

## 3.7 本章小结

本章阐述了闲置物品交易网站的总体设计，根据业务需求撰写了系统的功能分析， 并根据分析对系统进行了功能模块的划分并绘制了功能模块图，展示了系统主要功能 模块的操作流程图，并且绘制了多种介绍该系统的 UML 图。为系统的详细设计做好 了准备。还阐述了系统数据库的设计，首先展示了系统的总体 E-R 图，并介绍了各个 E-R 图之间的关系，其次分别展示了每个实体与属性的 E-R 图，并且根据系统 E-R 图 对数据库进行了分析和设计，并介绍了数据库表结构，完成了系统开发之前数据库的 准备工作，为系统的进一步开发做好了充分的准备。

# 第四章 系统实现

## 4.1 本系统开发过程中技术特色说明

本系统主要使用 JavaWeb 技术开发，所涉及的技术包括 HTML、CSS、JSP、 Bootstrap、Javascript、Ajax、Struts2 等。

HTML+CSS 是网页开发技术的基本技术，从动态网页技术为普及之前就作为网 页开发的基础技术，HTML 即超级文本标记语言，是网页显示内容的基本表现方式， 同时又是一种规范和标准，所有浏览器都可以解析其中的内容。HTML 语言简单易用 且可扩展，与平台无关，如今发展到 HTML5 标准。CSS 即层叠样式表，他可以定义 HTML 中的元素，实现样式与结构的分离，使 HTML 更加简洁，提高工作效率。

JSP 全名为 Java Server Pages，中文名叫 Java 服务器页面，其根本是一个简化的 Servlet 设计，而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完 成，JSP 具备了 Java 技术的简单易用，完全的面向对象，具有平台无关性且安全可 靠，主要面向因特网的所有特点。它里边有强大的标签库，可结合 Java 及网页两种 语言混合来写，其根本就是 Java Servlet 的抽象版。

Bootstrap，来自 Twitter，是目前最受欢迎的前端框架。Bootstrap 的简洁、直观、 强悍的前端开发框架，让 Web 开发更迅速、简单。它主要用于开发响应式布局、移动 设备优先的 Web 项目。

Javascript 是世界上最流行的 Web 脚本语言，由浏览器进行解析，可以用于 PC 设 备或移动端设备，在 Web 开发中，Javascript 语言用来实现向 HTML 页面增加交互 性，与 CSS 配合使用，是动态页面技术的核心。JavaScript 语言语法非常简单，并且 可以减少服务器的负担，被广大开发者青睐。

Ajax 即异步 JavaScript 和 XML，是一种用于创建快速动态网页的技术，Ajax 实 现了网页的异步更新，很大程度的减轻了服务器的负担，因为可以局部更新页面而不 需要重新加载整个页面，它使得 Web 应用更小，更快，更友好。

## 4.2主界面的设计与实现

## 4.3前台子系统的设计与实现

## 4.4 管理员子系统的设计与实现

## 4.5 特色功能的设计与实现

## 4.6 本章小结

# 第五章 系统测试

## 5.1软件测试的目的

## 5.2测试的方法

## 5.3系统测试

## 5.4测试结论

## 5.5本章小结

# 第六章 结论及展望

## 6.1 工作总结

## 6.2 工作展望

# 致谢

# 主要参考文献（资料）：