



**专业项目设计**

**基于Spring框架的团队博客系统**

**学 院 计算机学院**

**专 业 软件工程**

**年级班别 2016级(3)班**

**学 号 3116004810**

**学生姓名 陈伟霖**

**指导教师 杨易扬**

**2019年 10 月**

# 摘 要

随着信息化时代的来临，通过团队博客分享技术已经成为一种主流。博客是继Email、BBS和ICQ之后出现的第四种网络交流方式，早期的博客是论坛式的，分享的内容专业度不高，不利于优质内容的寻找、传播及获取，因此由专业团队执笔的团队博客成为一种需求。

本论文阐述了开发一个团队博客系统必经的几个阶段，包括该类系统的背景分析、可行性分析、开发模型分析、需求分析以及概要设计，解决了在这些阶段开发系统会遇到的问题。本系统后端开发使用JAVA语言、SSM框架，前端开发使用Layui库，通过本论文及上述技术，可以实现该系统。

**关键词：** 团队博客系统，SSM框架，Layui库

# Abstract

In the era of information explosion, sharing the technical information through team blogs has become popular. Blog is the fourth kind of network communication method after Email, BBS and ICQ. The early blogs are forum-style, and the sharing content is not professional, which is not conducive to the search, dissemination and acquisition of high-quality content. Therefore, team blogs written by professional teams become a requirement.

This paper describes the stages necessary to develop a team blog system, including background analysis, feasibility analysis, development model analysis, requirements analysis and outline design of the system, and solves the problems encountered in the development of the system at these stages. The back-end development of this system uses JAVA language and SSM framework, and the front-end development uses the Layui library. Through this paper and the technology mentioned above, the system can be implemented.

**Key words**：team blog system, SSM framework, Layui library

**目 录**

[1 绪论 1](#_Toc25071295)

[1.1 选题背景 1](#_Toc25071296)

[1.2 选题的目的和意义 1](#_Toc25071297)

[1.3 与当前系统的比较 2](#_Toc25071298)

[1.4 本论文主要工作 2](#_Toc25071299)

[2 技术介绍 3](#_Toc25071300)

[2.1 关系型数据库管理系统介绍 3](#_Toc25071301)

[2.2 SSM框架介绍 3](#_Toc25071302)

[2.3 Layui库介绍 3](#_Toc25071303)

[2.4 开发环境、运行环境、开发工具等介绍 4](#_Toc25071304)

[3 可行性分析 5](#_Toc25071305)

[3.1 技术可行性 5](#_Toc25071306)

[3.2 经济可行性 5](#_Toc25071307)

[3.3 操作可行性 6](#_Toc25071308)

[4 开发模型 7](#_Toc25071309)

[4.1 开发模型工作流程 7](#_Toc25071310)

[4.2 开发模型优缺点表述 7](#_Toc25071311)

[4.3 选取增量模型的依据 8](#_Toc25071312)

[5 需求分析 9](#_Toc25071313)

[5.1 信息需求 9](#_Toc25071314)

[5.2 功能需求 9](#_Toc25071315)

[5.3 数据库安全性与完整性需求 10](#_Toc25071316)

[5.4 流程建模 10](#_Toc25071317)

[5.5 用例建模 12](#_Toc25071318)

[5.6 数据字典 14](#_Toc25071319)

[6 概要设计 18](#_Toc25071320)

[6.1 概念模型E-R图 18](#_Toc25071321)

[6.2 系统架构设计 21](#_Toc25071322)

[6.2.1 架构采用 21](#_Toc25071323)

[6.2.2 架构内技术采用 22](#_Toc25071324)

[6.3 功能模块概要设计 22](#_Toc25071325)

[6.4 面向对象的设计方法 23](#_Toc25071326)

[6.4.1 软件类的设计 23](#_Toc25071327)

[6.4.2 交互模型 25](#_Toc25071328)

[7 总结与展望 27](#_Toc25071329)

[参考文献 28](#_Toc25071330)

# 

# 1 绪论

## 1.1 选题背景

随着信息技术的发展，通过博客分享技术成为当下的一种主流。目前市面上的博客有两种。

第一种是面向群众的大众博客、社区，比如CSDN[1]、博客园[2]，这类博客是属于论坛式的，每个人具有自己的小空间能够发表自己的博客，并且彼此之间能够互相浏览、评价、互动，具有大众性。

第二种是面向私人化的个人/团队博客，比如阿里中间件团队博客[3]、Tgideas腾讯互动娱乐创意设计团队博客[4]等，其文章的编写具有较高专业水平，由团队内部成员编写博客并进行相应的分类，不提供外人编写博客的权利，但保留其评论讨论的功能。



图1.1 阿里中间件团队博客首页

比如在阿里云中间件团队的博客网站上，发表的所有博客均由团队内部人员执笔，该系统提供展示、归档、分类、标签等功能，无游客登录功能。

## 1.2 选题的目的和意义

本次选题是面向专业团队的博客系统，即选题背景中的第二种博客。

本次选题的目的在于为专业团队设计出一种通用的博客系统，使技术人员能够专注于技术分享，因此该论文以实现一个具有权限管理、团队博客编辑、博客展示、游客评论、标签分类的系统为目标，同时后台系统要有相关统计功能。

## 1.3 与当前系统的比较

当前网络上普遍使用的系统，比如前面叙述的Tgideas腾讯互动娱乐创意设计团队博客、阿里中间件团队博客，都存在不同方面的问题。Tgideas腾讯互动娱乐创意设计团队博客界面设计复杂，对浏览者不够友好，评论需要使用QQ进行登录；阿里中间件团队博客采用Hexo模板定制，足够简洁，对浏览者足够友好，但没有评论功能，统计功能也仅有关于博客阅读量、站点访问人数等，缺乏对标签的热度的统计，同时使用hexo编写markdown[5]（以下均简称md）博客，md格式语法需要在线图片的地址（URL），而hexo并不支持图片上传功能。

与当前已有的团队博客比较，我们希望能够设计出一款对浏览者来说足够简洁的、对博客编写者来说功能足够的博客系统。在我们的设计中，博客系统的功能足够简单而强大，访客无需登录，可以自由讨论博文，博客编写者能够根据md语法编写博客，同时系统提供图片上传功能为博客编写提供便利，提供完善的统计功能。

## 1.4 本论文主要工作

本论文分为六大部分，在绪论中介绍选题背景，在技术介绍中简要介绍想要实现该系统应当用到的技术，在可行性分析中分析分析技术、经济、操作可行性，在需求分析中确定软件需要完成哪些功能，开发人员应完成哪些工作，在概要设计中通过需求分析建立E-R图，并作功能模块设计，最后一部分对已完成的工作做出了总结跟展望。

# 2 技术介绍

## 2.1 关系型数据库管理系统介绍

本次选用的关系型数据库管理系统（RDBMS）为MySQL[6]。

MySQL是当前最流行的关系型数据库管理系统之一，数据存储遵守关系型数据库的约束，将数据保存在不同的表中，提高了灵活性和增快了速度。MySQL使用标准化的数据库访问语言—Structure Query Language(SQL)语言，简单易学。同时由于MySQL的开源性，使得其受到大部分中小型企业的喜爱。

## 2.2 SSM框架介绍

SSM框架都是基于JAVA开发的开源框架，分为Spring[7]、SpringMVC[8]、MyBatis[9]三部分。

SpringMVC是基于Model And View(MVC)设计模式的、具有高扩展性的轻量级web框架，Spring通过IOC、AOP两大核心技术解耦，使得开发者能顾专注于应用层的业务逻辑，MyBatis是一款轻量级Object Relational Mapping(orm)框架，其灵活、半自动化的思想既大大简便了开发者的工作，又使得开发者能够为特定的应用场景配置SQL，增加了数据库操作的效率[10]。

## 2.3 Layui库介绍

Layui（谐音：类UI) 是一款采用自身模块规范编写的前端 UI 框架，遵循原生 HTML/CSS/JS 的书写与组织形式，门槛极低，拿来即用[11]。

通过使用Layui库，能够极大简化开发者的工作，节省大量用来调节前端样式的时间，大部分JS代码也不必重新开发，Layui库已经实现并封装好，仅在需要的时候查找对应的API即可。

## 2.4 开发环境、运行环境、开发工具等介绍

开发工具使用eclipse。

项目管理工具软件使用maven，主要用于包管理。

开发语言为JAVA，版本为1.8。

http服务器为Tomcat，版本为8.0。

MySQL开发环境版本为5.5.53，线上版本为5.7.16，理论上可以兼容5.5-5.7版本中的任意一个版本。

接口开发使用SpringMVC，Spring版本为4.1.7，Mybatis版本使用3.3.0。

开发环境配置为华硕笔记本，win10系统，线上环境为阿里云服务器，Linux系统（CentOS 7.2 64位）

# 3 可行性分析

## 3.1 技术可行性

对系统的技术可行性分析，是为了解决“当前技术能实现这个系统吗”的问题[12]。

对本系统进行技术可行性分析，主要分三部分：

1. 后端框架：在目前市面上常用的框架组合是SpringMVC+Spring+MyBatis（ssm）框架组合跟Strut+Spring+Hibernate（ssh）框架组合，通过对该系统的概览，当前技术完全可以实现这个系统，在技术选用方面，选择ssm框架主要由于：
   1. 开发者更加熟悉ssm框架组合；
   2. ssm框架中MyBatis能够使开发者更简单的、灵活的书写sql语句，使得某些查询更具效率。
2. 前端库：选用Layui库，简单易用，适合不熟悉前端的后端开发者使用。
3. 数据库：使用MySQL，MySQL开源易用，适合当前规模不大的系统[13]。

## 3.2 经济可行性

对软件的经济可行性分析，意在解决“这个系统的经济效益能超过它的开发成本吗”的问题[12]。

对本系统进行经济可行性分析，可以得到有关技术、人力、物力的经济支出：

1. 技术上分析，本系统的开发使用的均为开源免费框架，前端UI库也是免费的库，数据库MySQL也属于开源免费使用，因此，技术上的经济支出为0；
2. 人力支出上分析，可预见的支出为开发者的时间支出；
3. 物力支出上分析，系统开发前期无需支出，系统开发后部署在阿里云平台上，使用阿里云学生服务器，价格为120元/年，使用阿里云平台购买的xin域名，价格为88元/年，合计208元/年。

## 3.3 操作可行性

对系统的操作可行性分析，意在解决“系统的操作方式在这个用户组织行得通吗”的问题[12]。

当前软件的功能点不多，每个不同层级的操作人员都有对于不同的操作事项，比如超级管理员实际上无需关心博客相关的信息，只需要对管理员进行增伤改查操作，而博客信息、评论信息等的维护实际上由管理员进行操作。因此，对系统的分级，能够很大程度上降低操作人员的操作困难性。另外，该软件开发时也从操作者用户的角度出发，进行了合理的UI设计，因此，该系统的操作方式，在用户组织（即博客团队）上是必然行得通的。

# 4 开发模型

## 4.1 开发模型工作流程

选取增量模型作为软件开发模型。增量模型的工作流程可以表述为[14-15]：

①在系统开发的前期阶段，为了确保所建系统具有优良的结构，仍需要针对整个系统进行需求分析和概要设计，需要确定系统的基于增量构件的需求框架，并以需求框架中构件的组成及关系为依据，完成对软件系统的体系结构设计。

②在完成软件体系结构设计之后，可以进行增量构件的开发。这个时候，需要对构件进行需求细化，然后进行设计、编码测试和有效性验证。

③在完成了对某个增量构件的开发之后，需要将该构件集成到系统中去，并对已经发生了改变的系统重新进行有效性验证，然后再继续下一个增量构件的开发。

## 4.2 开发模型优缺点表述

（1）增量模型的最大特点就是将待开发的软件系统模块化和组件化。基于这个特点，增量模型具有以下优点：

①将待开发的软件系统模块化，可以分批次地提交软件产品，使用户可以及时了解软件项目的进展。

②以组件为单位进行开发降低了软件开发的风险。一个开发周期内的错误不会影响到整个软件系统。

③开发顺序灵活。开发人员可以对组件的实现顺序进行优先级排序，先完成需求稳定的核心组件。当组件的优先级发生变化时，还能及时地对实现顺序进行调整。

（2）增量模型的缺点是要求待开发的软件系统可以被模块化[14]。如果待开发的软件系统很难被模块化，那么将会给增量开发带来很多麻烦。

## 4.3 选取增量模型的依据

对于该系统而言，选取增量模型作为开发模型的依据有以下两点：

1. 软件开发人员一般是博客拥有者，对系统常常需要做二次开发，增加功能，即最终结果肯定是难以一次性开发完成；
2. 整个团队博客系统的功能点很容易模块化，比如按照面向的用户就有超级管理员、管理员以及有课，因此增量模型适合作为该项目的开发模型。

# 5 需求分析

## 5.1 信息需求

前台界面能够展示团队的博客、管理标签，游客可以通过前台界面进行留言讨论。后台界面能够实现权限管理，超级管理员具有一切权限，由开发者持有，其能够为其他管理员分配一定的权限使得管理员即团队成员能够设计标签、书写博客并直观感受博客热度、标签热度、总站浏览量、留言等统计结果。

## 5.2 功能需求

从系统功能出发，应该分为三种使用者：超级管理员、管理员、游客。

超级管理员的功能包括：

1、权限的管理：与开发息息相关，只有开发过过的权限才能添加。包括了①创建权限；②修改权限；③删除权限。

2、角色的管理：①创建角色，能够指定该角色拥有的权限集合。②修改角色，修改角色的名称、权限等；③删除角色。

3、管理员的管理：包括①创建管理员，指定其相关信息如密码、账号，同时为其分配相应的角色以规范权限；②修改管理员，修改指定的信息等；③删除管理员。

4、网站设置：包括设置网站标题、网站版权。

1. 其他功能：管理员所拥有的所有功能。

管理员的权限由超级管理员分配，但目前而言，其分配的功能包括：

1、标签的管理：每个博客都有一个或多个标签，管理员能够①创建标签；②修改标签；③删除标签。

2、博客的管理：①编写博客，规定使用MD语法书写博客，指定博客所属标签等；②编辑博客；③删除博客。

3、图片的管理：由于书写博客的MD语法所支持的图片需要首先上传到线上网络中并给出链接，因此提供图片管理功能，包括①上传图片；②删除图片。

4、统计模块浏览：①博客热度；②标签热度；③浏览总量；④留言统计，留言指游客的留言管理，包括禁用某个留言和恢复某个留言及所有留言的展示。

游客不具有登录注册的功能，目前仅仅能通过前台界面实现：

1. 浏览博客
2. 浏览标签
3. 搜索博客，包括根据标签搜索和根据内容搜索
4. 留言功能，如若留言时提供邮箱，留言成功时会发送邮件提示成功。

## 5.3 数据库安全性与完整性需求

安全性需求：

1. 数据库安全：要求实现一定的权限控制，比如通过授权使得应用程序端不得任意修改表的结构等，同时减低SQL注入的风险，对SQL注入能够有所防范，对非法操作和非法存取能够有良好的应变能力。同时要求按系统操作人员分类，高级DBA具有一切权限，普通DBA具有对某一数据库的所有权限，开发人员不具有更改表的结构等的权限，但具有基本的增删改查的权限。
2. 应用层安全：要求应用程序能够对用户的权限进行良好的管理，防止非法的越权操作，比如不具有分配权限的管理员试图分配权限时应当拒绝。

完整性需求：

要求保证数据的正确性和相容性，拒绝不正确、不和语义的数据，使它们无法存在于数据库中，该要求具体实现应在前端验证跟后端验证、数据库数据验证中体现。

## 5.4 流程建模

根据对系统信息、功能需求的分析，我们选取编辑博客、图片上传两个核心模块进行分析，

针对编辑博客模块，分析需求后，得到的该业务顶层数据流图为：

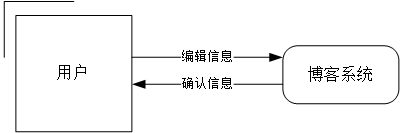


图5.1 顶层数据流图-编辑博客模块

对顶层数据流图进一步分解，对博客系统展开，得到DFD：

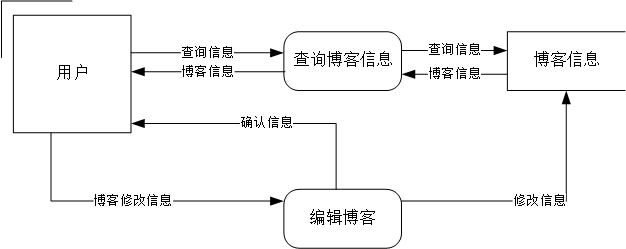


图5.2 一级数据流图-编辑博客模块展开的DFD

针对图片上传模块，分析需求后，得到的该业务顶层数据流图为：

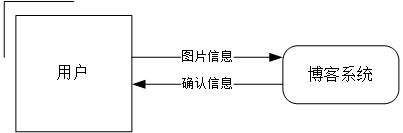


图5.3 顶层数据流图-上传图片模块

同样的，对顶层数据流图分解，展开得到DFD：

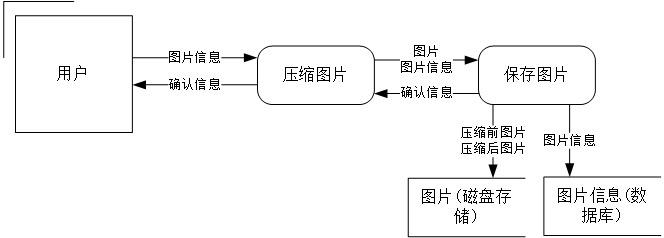


图5.4 一级数据流图-上传图片模块展开的DFD

## 5.5 用例建模

根据功能需求的分析，我们选取后台管理系统这个子系统，进行用例分析，得到用例图：

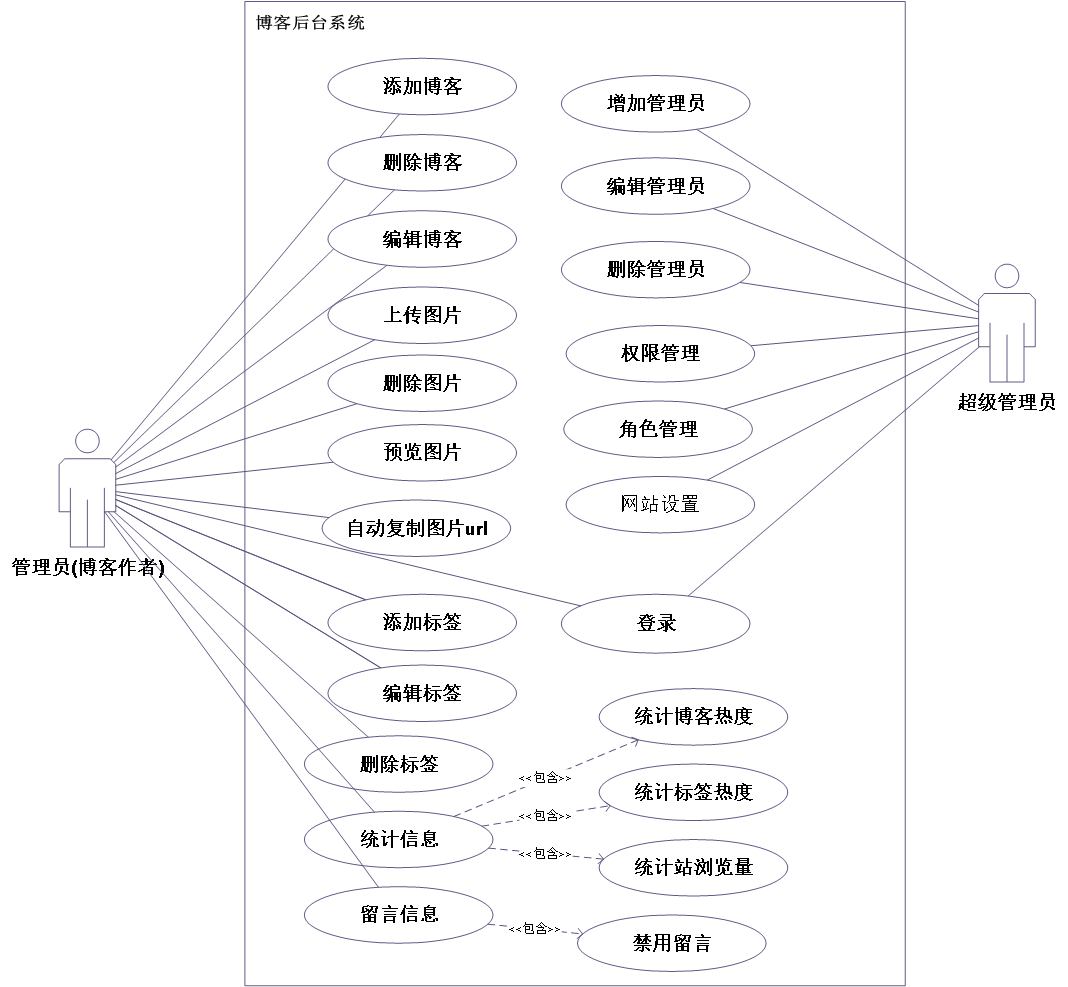


图4.5 后台系统用例图

同样的。选取编辑博客模块、上传图片模块进行用例分析，作用例规约如下：

表5.1 编辑博客用例规约

|  |  |  |
| --- | --- | --- |
| 用例名称 | 编辑博客 | |
| 参与者 | 管理员 | |
| 前置条件 | 管理员已经登录 | |
| 后置条件 | 无 | |
| 主事件流 | | |
| 参与者动作 | | 系统行为 |
| 1、点击“博客管理”进入该模块  2、填写博客查询信息  3、点击“查询”  5、点击“编辑”选中博客进行编辑  6、编辑博客信息  7、点击“立即提交”提交表单 | | 4、系统显示查询信息  8、系统修改博客信息  9、返回博客列表 |
| 备选事件流 | | |
| 7a. 表单填写不正确 | | |
| 业务规则 | | |
| 1、博客编辑的信息需要符合规范，必填项必须填写 | | |

表5.2 上传图片用例规约

|  |  |  |
| --- | --- | --- |
| 用例名称 | 上传图片 | |
| 参与者 | 管理员 | |
| 前置条件 | 管理员已经登录 | |
| 后置条件 | 无 | |
| 主事件流 | | |
| 参与者动作 | | 系统行为 |
| 1、点击“图片管理”进入该模块  2、点击“图片上传”  3、填写图片描述，上传图片  4、点击“立即提交” | | 5、生成压缩图，压缩图与原图片一并保存本地磁盘，图片信息入库  6、返回图片列表 |
| 备选事件流 | | |
| 5a. 图片描述未填写  5b. 上传文件不是图片格式 | | |
| 业务规则 | | |
| 1. 图片描述必须填写 2. 上传文件必须为图片格式 | | |

## 5.6 数据字典

[数据字典](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AD%97%E5%85%B8)是指对数据的[数据项](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%A1%B9)、数据结构、[数据流](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%B5%81)、[数据存储](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AD%98%E5%82%A8)、处理逻辑等进行定义和描述，其目的是对[数据流程图](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%B5%81%E7%A8%8B%E5%9B%BE)中的各个元素做出详细的说明，使用数据字典为简单的建模项目。简而言之，数据字典是描述数据的信息集合，是对系统中使用的所有数据元素的定义的集合。

该系统在需求分析阶段导出的数据字典为有：

表5.3 数据字典-admin(管理员):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| admin\_account | char(50) | 否 | 管理员帐号 |
| admin\_password | char(32) | 否 | 管理员密码 |
| admin\_fullname | char(255) | 是 | 管理员姓名 |
| admin\_phone | char(11) | 是 | 管理员手机号码 |
| status | tinyint(1) | 是 | 状态 1可用 2禁用 默认2 |

表5.4 数据字典-rule(权限):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| id | int(11) | 否 |  |
| rule\_name | char(30) | 是 | 权限名称 |
| rule | char(255) | 是 | 权限规则 |
| is\_menu | tinyint(1) | 是 | 是否菜单 1是 2否 默认2 |
| parent\_id | int(11) | 是 | 父级ID 0一级 非0子级 |
| icon | char(100) | 是 | 图标 |
| sort | int(11) | 是 | 排序 |
| status | tinyint(1) | 是 | 状态 1可用 2禁用 默认2 |

表5.5 数据字第-role(角色):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| role\_name | char(30) | 是 | 角色名称 |
| rule\_ids | char(2000) | 是 | 角色权限集合 如1,2,3... |
| status | tinyint(1) | 是 | 角色状态 1正常 2禁用 默认2 |
| remark | char(255) | 是 | 角色注释 |

表5.6 数据字典-blog(博客):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| title | char(255) | 否 | 标题 |
| description | char(255) | 是 | 简介描述 |
| content | text | 否 | 内容 |
| heat | int(11) unsigned | 否 | 热度 |

表5.7 数据字典-comment(评论):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| content | text | 否 |  |
| name | char(50) | 是 | 评论者名字。允许匿名 |
| mailbox | char(50) | 是 | 评论者邮箱，允许匿名 |
| ip | char(32) | 否 | 评论者ip |
| status | tinyint(3) unsigned | 是 | 1通过 2下架，默认1，即评论即见 |

表5.8 数据字典-img(图片):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| url | char(255) | 否 | 图片url |
| thumbnail\_url | char(255) | 否 | 压缩图url |
| description | text | 否 |  |

表5.9 数据字典=setting(网站设置):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| website\_title | char(255) | 是 | 网站标题 |
| copyright | char(255) | 是 | 网站版权 |

表5.10 数据字典-tag(标签);

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| tag\_name | char(30) | 否 |  |
| tag\_content | char(255) | 是 | 描述或内容 |

表5.11 数据字典-visits(总站浏览量):

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 允许空 | 备注 |
| number | int(11) unsigned | 否 | 网站浏览量 |

需求分析阶段的数据字典指出了该系统的基本数据及其数据项、数据结构、对该数据的描述等，但该数据字典会随着概念结构、物理结构设计而加以充分完善，甚至在开发维护阶段，如果有新的更改或需求，也会对数据字典加以充分完善。

# 6 概要设计

## 6.1 概念模型E-R图

在概念模型的设计中，需要清楚认识到，概念模型是不依赖于具体的计算机系统，它只是纯粹的反应信息需求的概念结构。E-R图的建立，是在需求分析的基础上展开，对数据进行抽象处理，先对局部E-R图逐个建立，最终再整合形成完整的全局E-R图。

以下根据需求分析建立的局部E-R图：

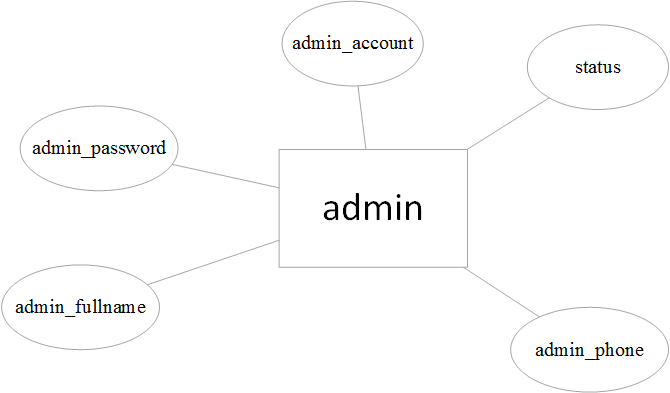


图6.1 admin E-R

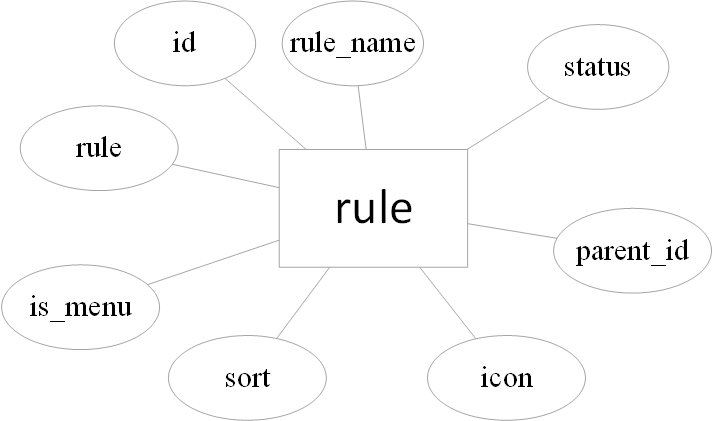


图6.2 rule E-R

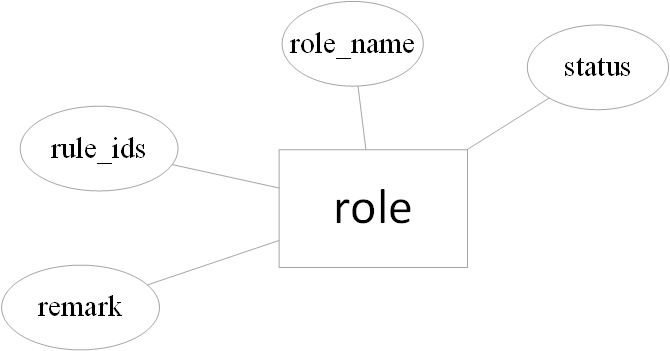


图6.3 role E-R

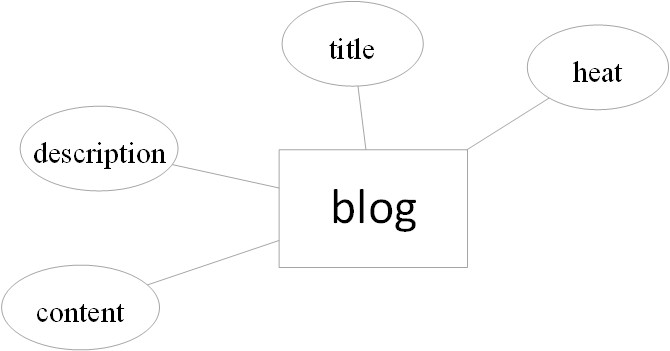


图6.4 blog E-R

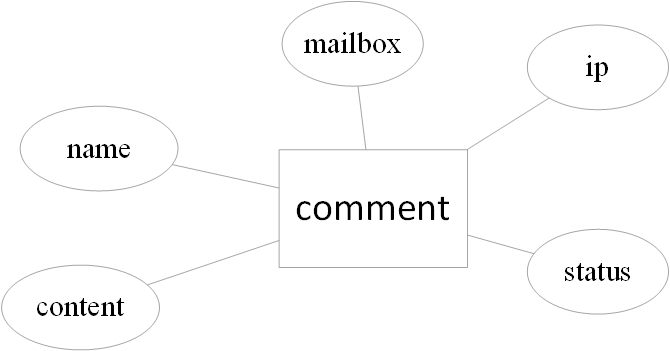


图6.5 comment E-R

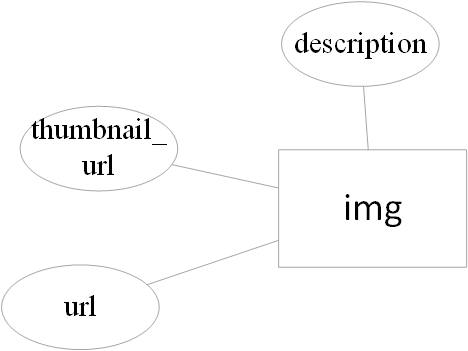


图6.6 img E-R

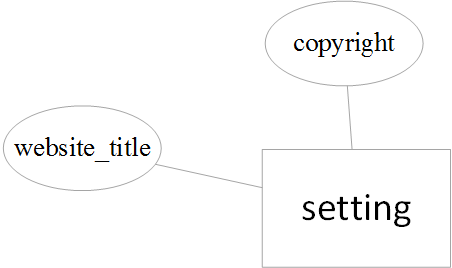


图6.7 setting E-R

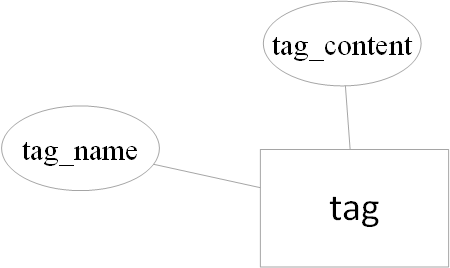


图6.8 tag E-R

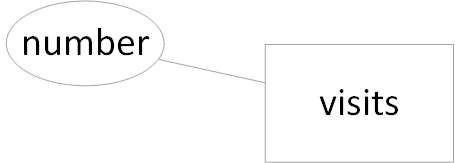


图6.9 visits E-R

确定各E-R实体之间的联系，得出全局E-R图如下：

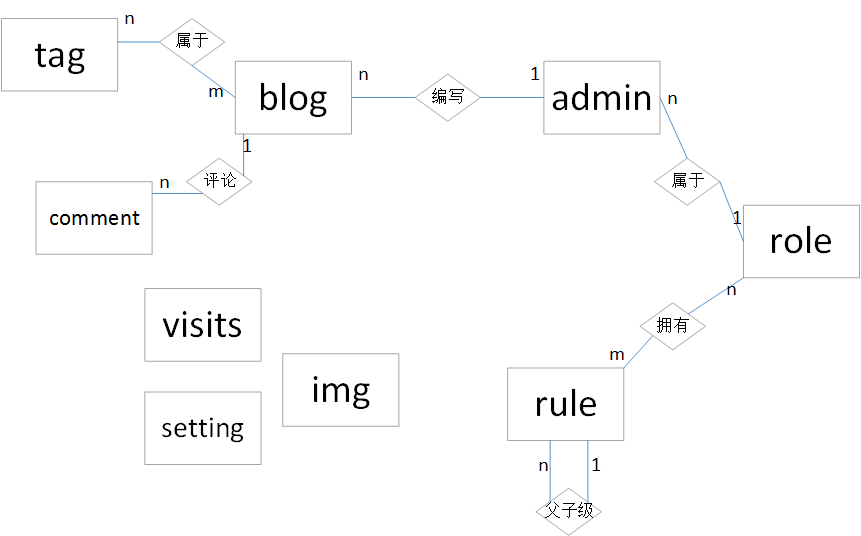


图6.10 全局E-R图

需要指出的是，visits、setting、img三者与其他实体之间并无联系。setting存放的是系统的配置如网站名称、网站版权，visits存放的是总站的访问人数，而img存放博客会用到的图片，但是由于博客使用md语法，不需要产生联系，只需要在博客编辑的时候给出url即可显示图片，因此没有直接联系。

## 6.2 系统架构设计

### 6.2.1 架构采用

采用多层次的应用架构设计[16]，具体如下：

1. 表现层：处理用户跟信息系统之间的交互。过滤层包含在表现层内；
2. 控制层：负责处理业务逻辑中的工作流、会话状态、数据的合成分解等事务；
3. 领域层：也称业务逻辑层，业务逻辑中的领域类的集合，不包含复杂工作流；
4. 数据映射层：负责将基于对象表达的领域层数据映射到数据库关系表中的记录；
5. 数据访问层：负责与数据库的交互，主要责任是存储持久数据。

### 6.2.2 架构内技术采用

根据多层次的应用架构，我们可以选用比较流行的技术或框架来简便开发，具体如下：

1. 表现层：使用JSP动态网页技术，比起传统html、js、css更加方便与控制层的交互；
2. 控制层：使用SpringMVC框架，简化控制器的开发流程；
3. 领域层：使用Spring框架，利用提供的aop跟ioc技术开发；
4. 数据映射层：传统的javabean技术即可；
5. 数据访问层：使用MyBatis框架，采用半自动化的框架能够在我们自由书写sql语句以达到效率优化的情况下，减少重复性工作比如连接数据库、释放数据库等等的开发。

## 6.3 功能模块概要设计

为完成功能模块的概要设计，分析系统功能需求，得出功能模块图：

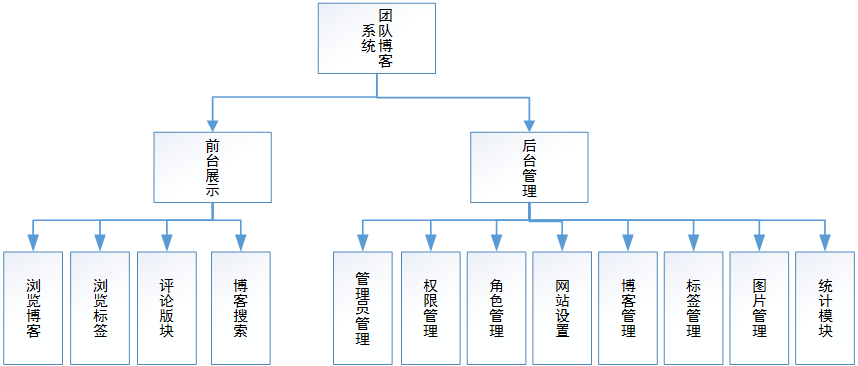


图6.11 系统功能模块图

根据分析，团队博客系统分为前台展示跟后台管理，前台展示面向游客，后台管理面向超管跟管理员，不同对象划分不同的子系统，不同的子系统划分不同模块，每个模块对应一组功能。

## 6.4 面向对象的设计方法

### 6.4.1 软件类的设计

在架构设计中，我们选用了五层架构模式，在此模式下，我们需要设计边界类（JSP）、控制类（Controller）、领域类（Service）、实体类（Entity）、数据访问类（Dao），在概要设计阶段，要设计类的属性跟方法而不谈具体实现，在java中不谈实现而设计方法，就是利用接口（interface）来设计，因此在Dao、Service中我们主要设计接口，其他直接设计类，没有必要设计接口，符合五层架构模式的规范。

根据五层架构模式，分别设计4个包结构：web包、service包、entity包、dao包，以及一个存放jsp的文件夹。这里以包图的形式给出设计好的类图：

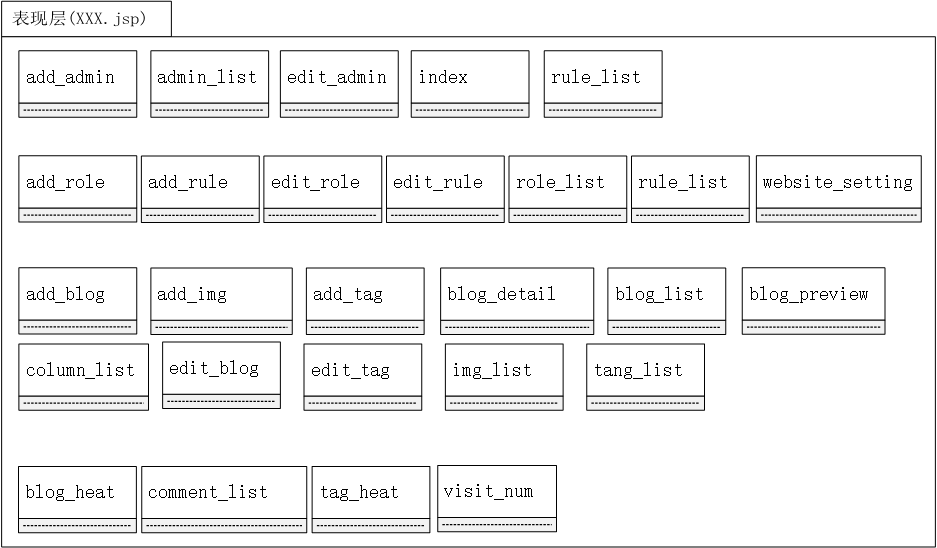


图6.12 包图-表现层

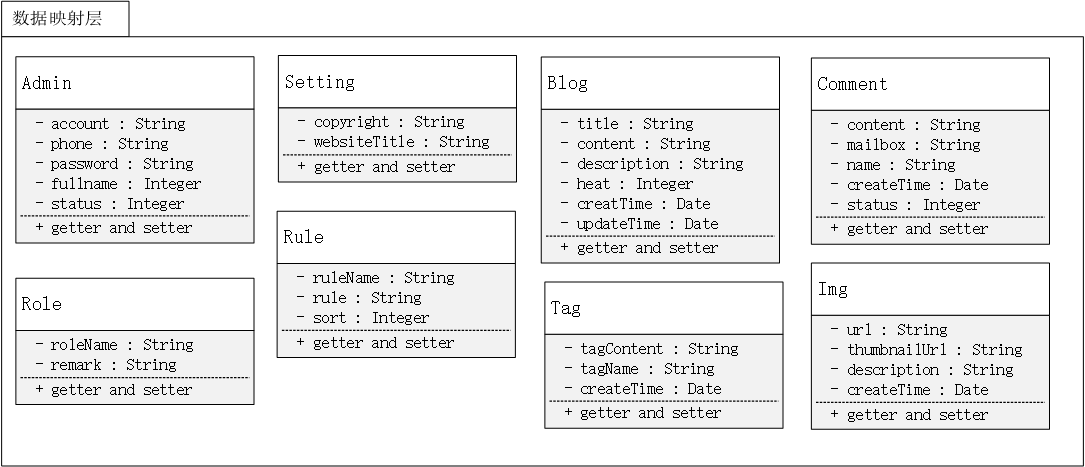


图6.13 包图-数据映射层

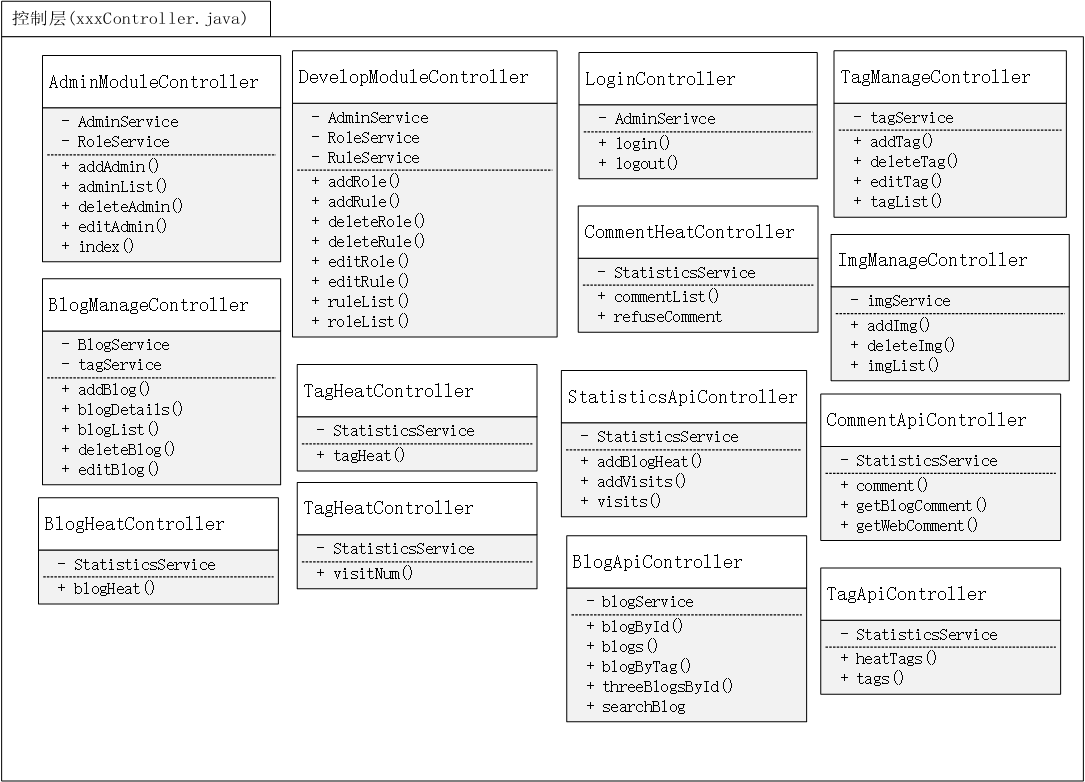


图6.14 包图-控制层

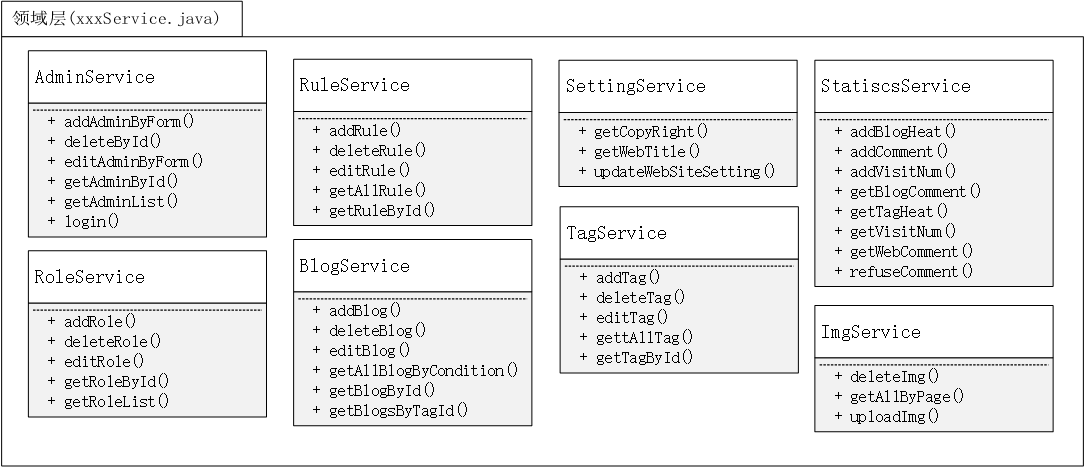


图6.15 包图-领域层



图6.16 包图-数据访问层

### 6.4.2 交互模型

需求分析中，我们通过对核心模块编辑博客、上传图片业务进行需求、用例、流程建模，在上一节中初步完成了软件类的设计，在此基础上，我们进一步细化，完成交互模型。

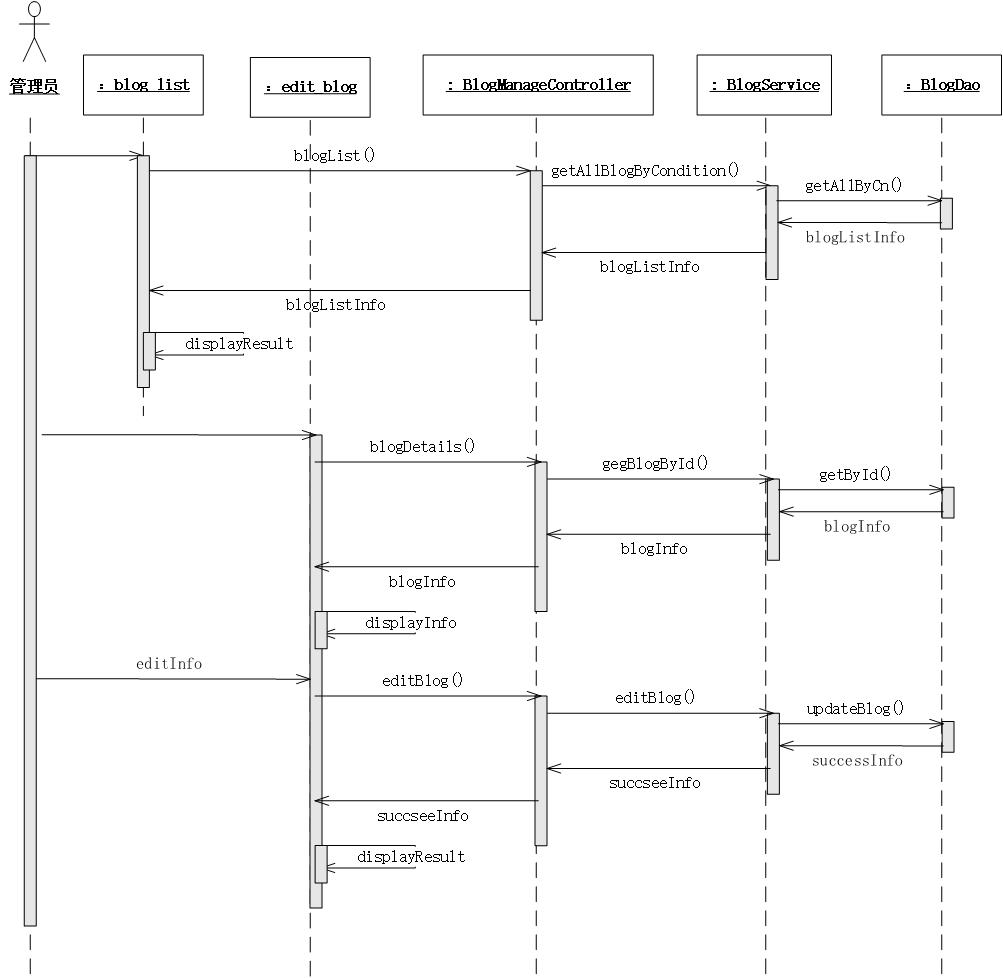


图6.17 顺序图-编辑博客模块

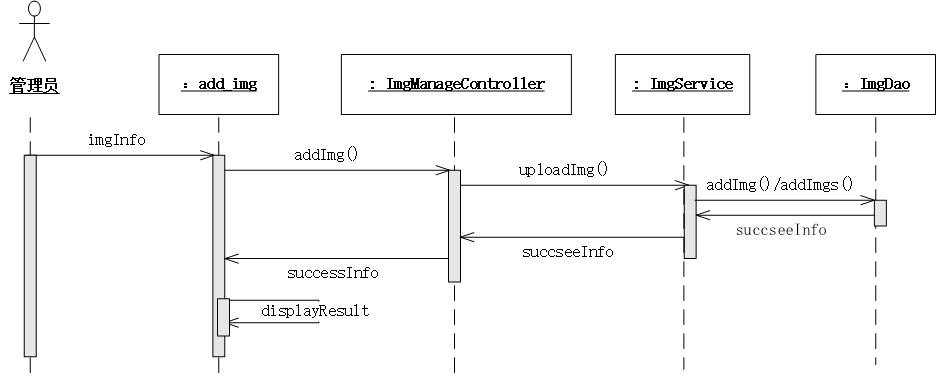


图6.18 顺序图-上传图片模块

交互模型体现的是用户完成一项业务，系统的各个类之间相互调用的关系。越复杂的业务，类之间的调用也会越复杂。

# 7 总结与展望

在本论文中完成了团队博客系统背景分析、可行性分析、开发模型分析、需求分析以及概要设计。在可行性分析中从经济、技术、操作三方面分析可行性；在开发模型分析中，选取了合适与本系统的开发模型；在需求分析中，完成了信息、功能、数据库安全性完整性需求的阐述，同时完成流程建模与用例建模，生成数据字典；在概要设计中，完成了概要模型建模，选取了开发架构，完成了功能模块概要设计，以面向对象的方法设计了软件类和交互模型。可以认为本论文已经完成了开发一个系统的前期工作。

在接下来的工作中，从上述已完成的分析、设计出发，从详细设计、系统实现、系统测试入手，彻底完成本系统的设计跟实现。

# 参考文献

[1] 世纪乐知. CSDN-专业IT社区[DB/OL] . https://www.csdn.net/,2019-10-27

[2] 博客园. 博客园-开发者的网上家园[DB/OL] . ttps://www.cnblogs.com/,2019-10-27

[3] 阿里中间件团队. 阿里中间件团队博客[DB/OL] . http://jm.taobao.org/,2019-10-27

[4] Tgideas. Tgideas-腾讯互动娱乐创意设计团队[DB/OL] . https://tgideas.qq.com/,2019-10-28

[5] HarryZhu. 文档定义应用：数据科学的文档革命[DB/OL] . https://segmentfault.com/a/1190000005125984,2019-10-28

[6] ORACKE. MySQL[DB/OL] . https://www.mysql.com/,2019-10-28

[7] Pivotal. Spring[DB/OL] . https://spring.io/,2019-10-28

[8] Pivotal. Web on Servlet Stack[DB/OL] . https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html,2019-10-28

[9] MyBatis. mybatis-Mybatis 3|Introduction[DB/OL] . https://mybatis.org/mybatis-3/index.html,2019-10-28

[10] 王天顺,程杰,张志文. MyBatis在《Web中间件技术》课程中的应用[J] . 教育教学论坛,2019(38):204-205.

[11] 贤心. 经典模块化框架[DB/OL] . https://www.layui.com/doc/,2019-10-28

[12] 王晓敏. 信息系统分析与设计[M] . 北京：清华大学出版社，2013： 65-180.

[13] J. Murach. murach’s MySQL[M] . WOW! eBook & Associates，2015(2nd edition)： 4-5

[14] 张海藩. 软件工程导论[M] . 北京：清华大学出版社，2003：17-18

[15] I. Sommerville. Software Engineering[M] . 北京：机械工业出版社，2015：29-51

[16] 相洁, 吕进来. 软件开发环境与工具[M] . 北京, 电子工业出版社, 2015：21-33.