

《JAVAEE》课程设计

****

**题 目 基于JAVAEE的博客后端管理平台**

学生姓名 黄敬旺

学 号 201983290498

学 院 计算机与软件学院

专 业 计算机科学与技术

指导教师 王玉祥

**二Ｏ二一 年 十二 月 十五 日**

目录

[1 引言 2](#_Toc15759)

[1.1课题背景和意义 2](#_Toc10728)

[1.2 课题内容 2](#_Toc23413)

[2 系统需求分析及相关技术介绍 2](#_Toc17842)

[2.1 功能需求分析 2](#_Toc8765)

[2.2 开发工具和技术 3](#_Toc13066)

[3 系统总体设计 4](#_Toc25797)

[3.1 系统功能结构设计 4](#_Toc23774)

[3.2 系统访问监控链 4](#_Toc24491)

[3.2 系统功能流程设计 6](#_Toc7467)

[3.2.1 登入功能 6](#_Toc15715)

[3.2.2 注册功能 6](#_Toc4930)

[3.2.3 邮箱验证功能 6](#_Toc27650)

[3.2.4 shiro安全认证 7](#_Toc6365)

[3.2.5 数据监控 7](#_Toc23142)

[3.3 数据库设计 8](#_Toc13612)

[3.3.1概念结构设计 8](#_Toc8101)

[4 系统实现 13](#_Toc9480)

[4.1 登入功能的实现 13](#_Toc26343)

[4.2 注册页面 14](#_Toc2532)

[4.3 邮箱登入检验的功能 15](#_Toc26885)

[4.4 登入的主面板 15](#_Toc582)

[15](#_Toc14380)

[16](#_Toc8596)

# 1 引言

1.1课题背景和意义

随着社会的发展，数据化成为一种趋势，很多公司和个人创建自己的网站。而大部分使用网站的人并不了解程序的底层，因此，中台应用在每个网站中就必不可少。中台应用，区别于呈现在普通用户面前的页面和程序员的后端代码和数据库。它将一些关键的统计信息展现在系统的超级管理员面前，例如所有用户的账号，各种资源的状态，访客记录。中台应用可以对网站的流量和安全进行全方位的监控，为网站的发展提供有效的参考数据。

在一个成熟的项目中，不同的用户存在着不同的角色，角色的不同使他们获取到不同的数据，如何有效的控制不同用户的访问权限也是一个至关重要的问题。

本项目的初衷是搭建一个具有权限管理，流量跟踪，访问限制，网站数据统计分析的中台应用，为普通的博客网站搭建一个安全高效后台。

## 1.2 课题内容

本次课题的主要内容是基于SSM的博客后台管理系统，其用于给网站的拥有者提供网站的访问信息和基本的客户信息。这个系统可以进一步完善，加入其他的模块，并对这个模块进行监控。系统的面向用户主体为博客园系统的超级管理员。

一个完善的系统需要注意安全和角色问题。具体来说，就是一个系统的使用用户可以分为一般无需任何登入的访客，发表内容的用户，vip用户，超级管理员等等，这些可以简单的称之为不同的角色，不同的角色对数据用户不同的获取和修改权力。为了很好的实现这个功能，本博客系统集成了shiro安全认证框架。框架下的角色初步分为三类: 一般的游客，可以浏览和搜索网上的博客；用户，可以发表博客。超级管理员，对所有用户的信息进行控制，同时能够跟踪检测网站的运行情况。本系统暂设设计好了后端系统。其他的有待实现。

为了实现对所有请求的跟踪，我们通过设计springmvc相应的拦截器实现。后端通过对url进行分类和标注，利用注解和反射，实现了访问url资源的识别。

同时本系统设计了最近出的防木马攻击的策略。本系统所有的数据存储在数据库。恶意频繁的访问会导致系统的拥挤，为此，本系统针对此类攻击设计了响应的预防策略。具有了初步的成效。在下一阶段，本系统将集成正常的blog应用和redis数据库，将一些需要高响应的数据用redis存储，并引入消息队列等技术，缓解服务器的运行压力。

此外本文还实现了邮箱认证的功能。项目集成了邮箱客户端，在注册和找回密码阶段实现了邮箱注册。

基于上述的功能分析，本课题做了一个blog后台系统。

# 2 系统需求分析及相关技术介绍

2.1 功能需求分析

根据上述的分析，我们将系统大致的分为三个模块: 1.基于shiro的用户权限认证模块；2. 浏览记录跟踪模块。3. 博文管理模块。当前系统的功能能笼统的概括为一下内容：

1. 登入和邮箱注册功能: 一般用户的注册，超级管理员的查询和修改功能。

2. IP地址拦截功能: IP地址拦截添加黑名单和移除黑名单的功能。

3. 访问资源识别功能: url请求资源分析功能。

4. 待做事项功能: 添加待做时间功能，消息提醒功能。

2.2 开发工具和技术

该博客后端系统基于中台系统Admintle进行二次开发，采用SSM架构，前端的显示方面运用了Themeleaf模板引擎来实现数据的渲染和展示。其中，spring创建全局ioc容器，集成mabytis和shiro等框架的必要bean,SpringMvc创建子容器，主要负责controller层前后端的交互。Mybatis作为持久层的框架，对数据库进行读写。

Mybatis是apache的一个[开源项目](https://baike.baidu.com/item/%E5%BC%80%E6%BA%90%E9%A1%B9%E7%9B%AE/3406069" \t "https://baike.baidu.com/item/MyBatis/_blank)iBatis, 2010年这个[项目](https://baike.baidu.com/item/%E9%A1%B9%E7%9B%AE/477803" \t "https://baike.baidu.com/item/MyBatis/_blank)由apache software foundation 迁移到了[google code](https://baike.baidu.com/item/google code/2346604" \t "https://baike.baidu.com/item/MyBatis/_blank)，并且改名为MyBatis 。MyBatis 是一款优秀的持久层框架，它支持定制化 SQL、存储过程以及高级映射。MyBatis 可以使用简单的 XML 或注解来配置和映射原生信息，将接口和 Java 的 POJOs(Plain Ordinary Java Object,普通的 Java对象)映射成数据库中的记录。每个MyBatis应用程序主要都是使用Sql[Session](https://baike.baidu.com/item/Session" \t "https://baike.baidu.com/item/MyBatis/_blank)Factory实例的，一个SqlSessionFactory实例可以通过SqlSessionFactoryBuilder获得。SqlSessionFactoryBuilder可以从一个xml配置文件或者一个预定义的配置类的实例获得。MyBatis 最强大的特性之一就是它的动态语句功能。通过OGNL表达式和标签的使用创建的动态语句能很好的满足开发时遇到的各种情况，减少sql语句的书写。

Spring的核心思想是IoC(控制反转)，即程序员不用显式的new对象，而是采用工厂模式，用户提前注入一些类进入ioc容器，需要时可以直接取出。因为Spring即拿即用的特性，很适合作为黏合整合框架，可能很好的将需要的类自动注入到正确的位置。用户只需要定义需要被用到的类。在需要的地方ioc自动注入，从何实现了复杂功能的整合。本项目中，spring用来整合shiro，springmvc,mybatis等框架。

SpringMvc在项目中拦截用户请求，它的核心servlet即DspatchServlet承担着中介或者前端的职责，负责将用户所有的请求拦截下来，然后根据处理器映射器HandlerMapping将不同的url请求映射到响应的Controller去处理。于此同时执行链直接的结果会打包成ModalAndViews交给相应的模板引擎去渲染。其中Controller就是请求的具体处理过程。

Shiro是一个强大且易用的Java安全框架,执行身份验证、授权、密码和会话管理。三个核心组件：Subject, SecurityManager 和 Realms.其中Subject表示当前的操作用户，在应用中则表示第三方进程，账户等等。当需要权限认证的时候，Subject会提交验证信息给SecurityManager进行认证。SecurityManager根据不同的认证策略或者认证信息进行认证。其中Realms则是认证信息定义类，为SucurityManager提供认证的标准，包括了用户的角色，每种角色的访问和修改权限。每种请求需要的权限。Shiro本质还是一窜安全过滤器链。

邮箱认证技术。本系统采用消息队列，映射表，集成邮箱客户端的技术事先了邮箱认证技术。采用java程序获取设计团队的邮箱授权，模拟邮箱客户端给用户发送认证信息。为了保证消息队列的有效性。本系统采用了消息队列技术，每次的认证信息会有一个生命周期，放入认证信息队列中，同时写入映射表中。保证在一个生命周期内认证信息有效可认证。生命周期结束，认证信息过期自动出栈。

此外，为了项目的便捷的测试，本系统集成了Swagger2组件，可以为之后的前后端应用分离做铺垫。Swagger2是一个用于前后端数据交互的url接口指定组件。它可以用可视化的形式指定当前的前后端交互接口。

# 3 系统总体设计

针对整个系统，我们主要设计了一条执行链事先权限认证，访客记录跟踪和资源识别等功能，此外还设计了一个后台进行相关数据的统计功能。我们主要设计了三个拦截器和一个shiro安全认证器实现了上述的功能。此外本文还通过多张数据表格实现了登入注册和管理员后端查询跟踪等功能。

3.1 系统功能结构设计

我们以模块的数据表为基础对功能进行划分。商品的功能如2.1叙述，如下图:

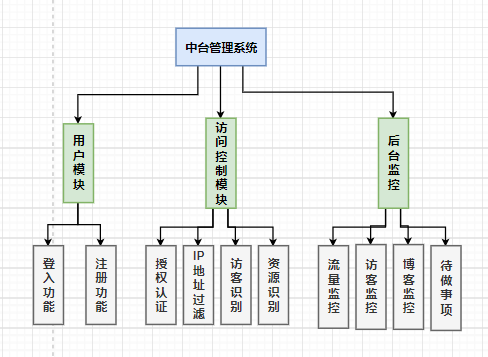


图3-1 系统功能结构图

本功能按照功能对系统进行划分，其中前端显示的是用户模块和后台监控模块。其中访问控制模块主要起到资源保护和监控的安全措施，它为后端监控提供初始数据的产生支持。在用户模块，暂时只设计了登入和注册模块，后期会相应的开发博客和个人信息中心模块。下面先介绍文本的访问控制链，问更好的了解这个后台监控系统做铺垫。

3.2 系统访问监控链

访问控制执行链就是本监控系统的核心，他是个基于注解和反射以及springmvc的拦截器的访问控制框架，所有的url都会经过他的过滤，其中在防攻击方面有待进一步的扩充，具体的访问控制链如下:

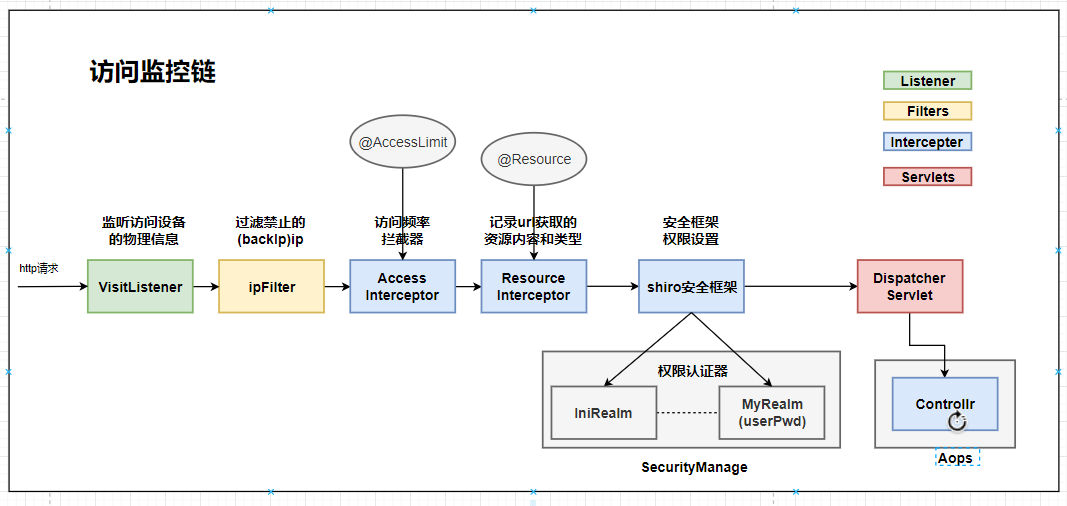


图3-2 系统功能结构图

访问控制链由VisitListener(访客监听器), Ipfilter(ip地址拦截器)，AccessInterceptor(访问频率控制器),ResourceInterceptor(资源识别拦截器)，shiro(安全框架)，DispatchServlet等组成。

其中VisitLenter用来获取用户登入的物理信息，包括了登入地址，浏览器平台，网络服务供应商，访问时间，IP地址的所属地点。访客监听器目的是为了记录访客的登入记录，为访客跟踪提供必要的物理信息，为进一步的安全分析和网站的使用情况分析做铺垫。IpFilter是一个IP拦截器，里面存储了被列入黑名单的IP地址，可以用来过滤一些恶意访问的IP地址。 AceessInterceptor核心功能是访问频率控制，本网站根据不同资源的类型对网站的资源进行访问频率限制，例如对于一些修改数据库和大量读取页面的操作，我们对其进行访问频率限制，避免出现恶意攻击，导致服务器阻塞的情况。他的核心是根据自定义的@AccessLimit注解来实现和判别。

ResourceInterceptor资源识别器主要是通过@Resource对Controller进行标注实现对url进行识别，本应用大致将url的类型分为以下几类:HTML(网页)，IMAGE(图片)，BLOG(博客),RECORD(数据资源)，CHECK(数据校验)，MODIFY(资源修改)。和运用平台等等。该拦截器可以实现对url资源访问的进一步监控，和VisitListener形成一个完整的游客访问监控体系。

shiro安全认证框架，我们自定义了两个认证信息模板Reamls，其中MyReaml是基于数据库的数据库的用户密码登入和权限设置，于此同时，支持居于配置的认证模板类IniRealm支持对数据认证的灵活性。所有的访问资源只有相应的权限才可以进行访问，本系统主要是基于角色的访问系统，角色大致分为3类，游客，一般账户和超级管理员，实现不同用户的授权认证。

通过这些执行链之后，最终交给SpringMVC的dispatcherServlet进行正常的数据处理。

3.2 系统功能流程设计

3.2.1 登入功能

首先在前端页面输入对应的账号，进行初步的账户验证，如果存在则显示用户的头像，等待用户进一步输入密码进行认证。和后端接收到用户名密码首先被环绕AOP方法包裹，首先进行shiro的权限认证在进入用户的登入。shiro先对输入的密码进行加密，然后和数据库的信息进行对比。

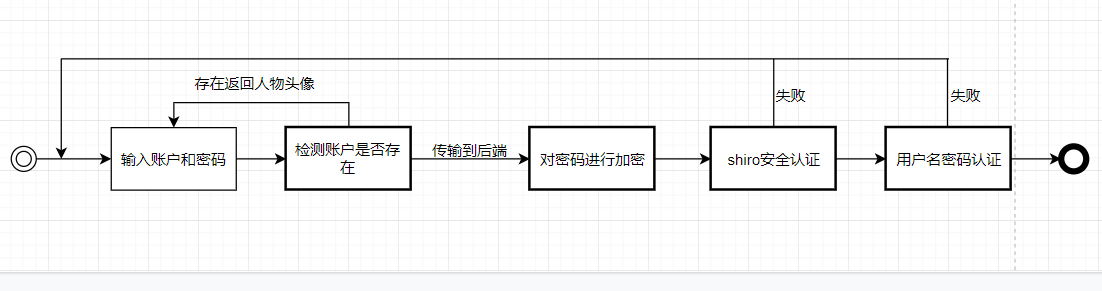


图3-2-1 用户登入流程图

3.2.2 注册功能

用户首先进入用户注册界面，输入用户名，用户输入用户名之后，会提交给后端验证用户名是否已经存在。然后输入密码和重复密码验证是否存在，然后输入邮箱，通过邮箱向相应的邮箱输送验证码同时可以传入相应的图片。

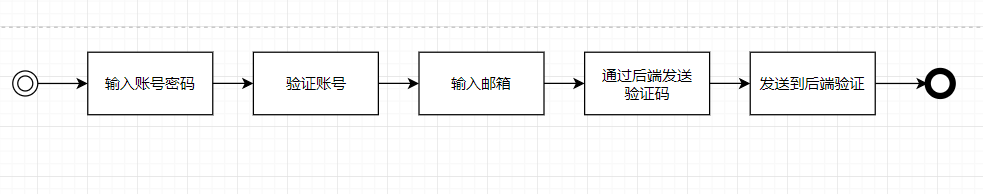


图3-2-2 注册功能流程图

3.2.3 邮箱验证功能

注册用户输入发送验证码的请求，后端接收到请求和前端发送的相应的邮箱地址，然后取映射表中查找是否已经存在有效的验证消息，不存在则重新生成，生成之后发送到相应的邮箱进行。当用户提交注册的时候，也许先经过验证码映射表观察验证码是否合法和过期，满足一起一切条件则完成邮箱验证功能。

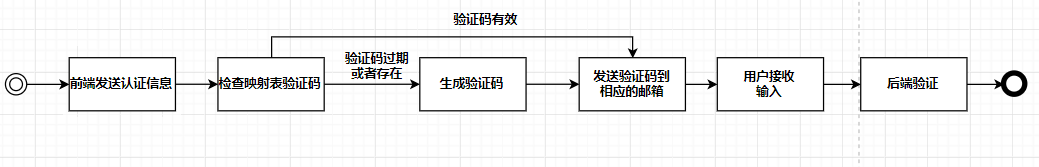


图3-2-3 邮箱验证功能流程图

3.2.4 shiro安全认证

Shiro安全认证，首先检查url是否需要被验证。如果需要，则先生成一个subkect对象，然后用生成的账号密码生成token，给交securityManager进行认证，securityManager先查看缓存，看该token是否已经认证，如果认证，则修改获取角色和权限校验角色和权限是否满足条件。如果该token不存在，则由Realms提供验证信息，比对token和验证信息是否满足条件。认证通过，获取相应的认证信息:角色和权限，认证url资源的权限。

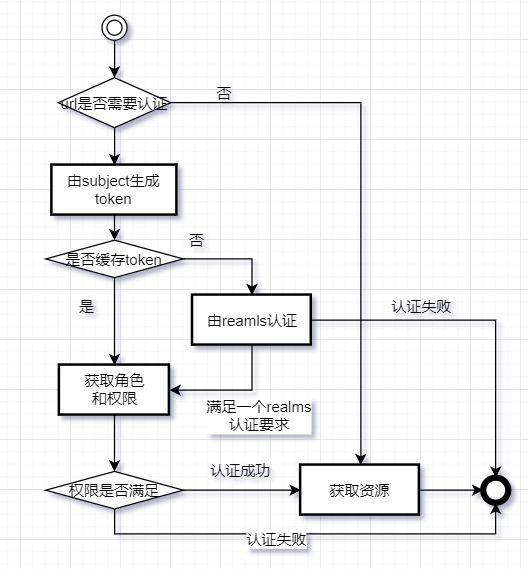


图3-2-4 shiro安全认证流程图

3.2.5 数据监控

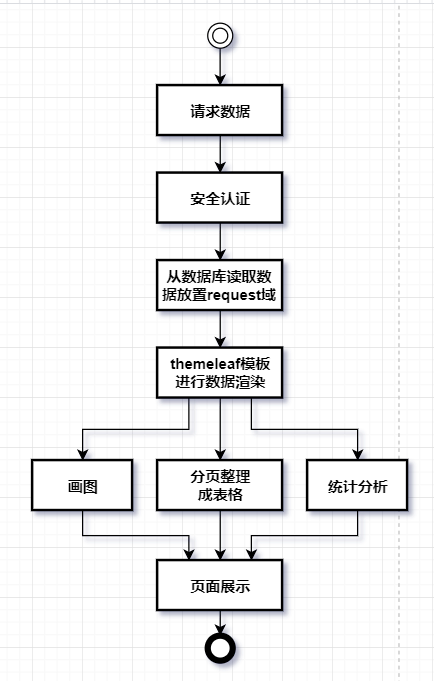
从后端获取到相应的数据，然后交给themeleaf模板引擎进行数据渲染。传递给前端，前端进行分页以表格的形式呈现，或者前端通过画折线，饼图等统计图展示相应的数据

图3-2-5 数据监控流程图

3.3 数据库设计

数据库根据模块设计了三类数据，按照功能划分主要分为博客相关的类，访问相关的类，用户极其相应权限的类。在下面的概念设计中，我们加按照着三个模块进行讲述。

3.3.1概念结构设计

**博客类数据概念结构设计**

博客类数据主要有博客表，博客类型，博客图片映射表和图片表。在设计的时候考虑到博客在加载的时候时间上的限制，对于一些有很多图片的博客来说，如果一起加载，会导致加载的速度变慢，基于这个考虑，我们的图片存储管理和其他功能模块进行分类，其他模块只能获取到到图像的在线地址，然后通过在线地址去获取图片。其中博客类中username字段是user的外键，blogimage表中的blodid是博客类的id字段的外键，imageid是image表id字段的外键，考虑到博客分类中所有的属性和blogType的id存在完全依赖，blog和blogtype存在多对一的关于，于是我们将blogType单独的分离出来，便于后期的统计。

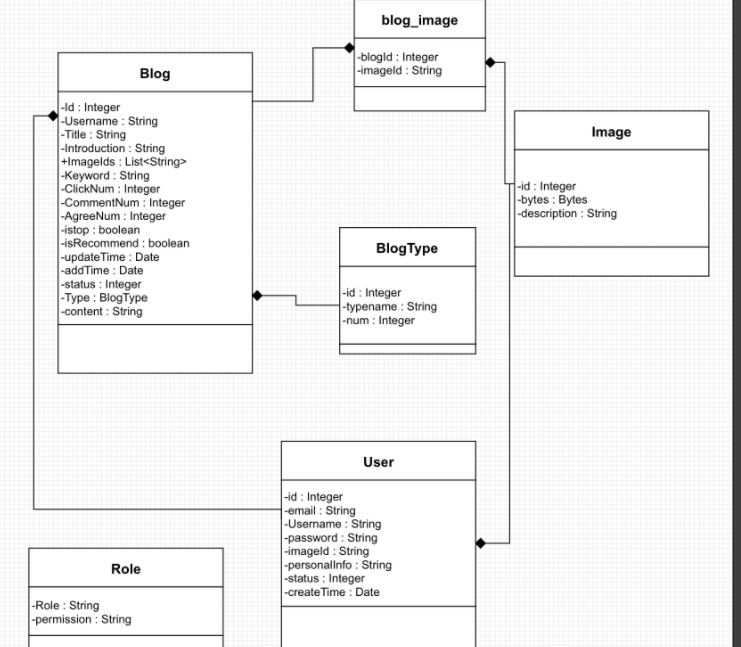


图3-3-1 博客相关类图

1. Blog类

博客实体包括的属性可以拆解成博文ID、创建者、文章标题、简介、关键字、点击量、评论量、点赞量，是否置顶，是否推荐、更新时间、添加时间、文章状态、文章内容, 其中各个元素的完整性约束和逻辑设计表格如下：

表3-1 Blog表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 博文ID | Id | bigint | Yes | Yes | No | 自增主键 |
| 文章标题 | title | varchar(100) | Yes | No | No |  |
| 简介 | introduction | varchar(200) | No | No | No |  |
| 关键字 | keyword | varchar(100) | No | No | No |  |
| 点击量 | clickNum | Int | Yes | No | No |  |
| 评论量 | CommentNum | Int | Yes | No | No |  |
| 点赞量 | AgreeNum | int | Yes | No | No |  |
| 是否置顶 | Istop | bool | Yes | No | No |  |
| 是否推荐 | isRecommand | bool | Yes | No | No |  |
| 更新时间 | update\_time | Datetime | Yes | No | No |  |
| 添加时间 | add\_time | datetime | Yes | No | No |  |
| 文章状态 | status | int | Yes | No | No | -1为草稿，0:回收,1正文 |
| 文章内容 | content | LongText | Yes | No | No |  |
| 类别id | blogType | BigInt | Yes | No | No | 博客类型，外键指向blogType |

1. 博客类型类

博客类别表主要包括BlogType的标识符id,类型名,该类博文的数量，约束和逻辑表格如下。

表3-2 BlogType表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 列别id | Id | bigint | Yes | Yes | Yes | 标识符 |
| 类别名 | typename | Varchar(50) | Yes | No | No |  |
| 博文数量 | num | bigInt | Yes | No | No |  |

1. 博客图片表

博客图片表是一个连接工具表，用来索引博客以及相关的图片id.

表3-3 BolgImage表结构说明

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | | 备注 |
| 博文ID | Id | bigint | Yes | No | | No | 外键,指向blog表 |
| 图片ID | title | varchar(100) | Yes | No | | No | 外键,指向image表 |

1. 图片列表

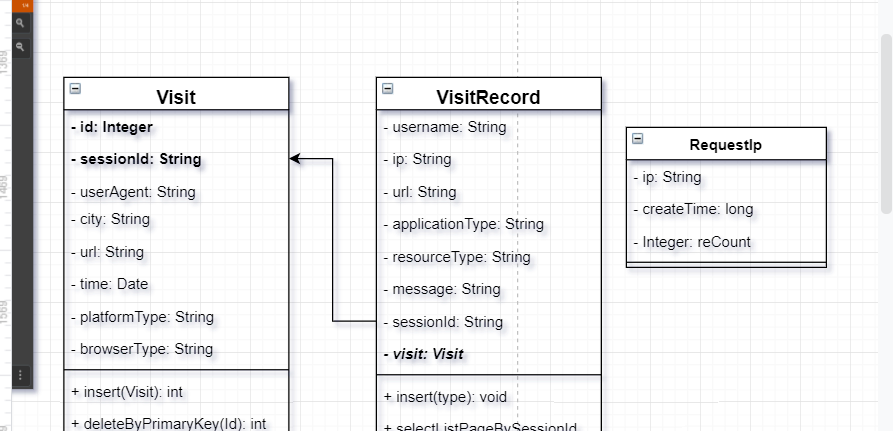
图片列表包含的属性有:图片的id,图片的二进制文件和图片的描述。

表3-4 Image表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 图片ID | Id | bigint | Yes | Yes | No | 自增主键 |
| 二进制文件 | bytes | MiddleBlod | Yes | No | No | 二进制文件 |
| 描述 | description | middleText | No | No | No | 图片描述 |

**访问控制类表：**

访问控制类的表格主要有三个，分别是记录物理信息的Visit表，记录资源访问的VisitRecord类和记录拦击ip的RequestIp类。

图3-3-1 访问控制相关类图

1. 访客表(Visit)

访客表主要记录的是访客的物理信息，区别域访客记录表，它侧重于记录每个访客的物理信息，以sessionId区分每一条记录。而VisitRecord类则是比较全面，记录了同一个sessionId前后访问所有资源的记录。访客表格主要包括了访客id,sessionId，代理商，所在城市，访问时间，访问使用的系统，浏览器类型，其设计和表格如下:

表3-5 Visit表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 访客ID | Id | bigint | Yes | Yes | No | 访客标识符 |
| SessionId | SessionId | Varchar(50) | Yes | No | No |  |
| 代理商 | userAgent | Varchar(50) | No | No | No | IP地址的代理  运行商 |
| 所在城市 | city | Varchar(50) | Yes | No | No | IP地址所在城市 |
| 访问时间 | time | DateTime | Yes | No | No |  |
| 使用系统 | platformType | Varchar(50) | Yes | No | No | 操作系统 |
| 浏览器类型 | browserType | Varchar(50) | Yes | No | No | 浏览器类型 |

1. 访客表记录(VisitRecord)

访客记录表主要包含了用户浏览每一条资源的记录，主要的信息包括了访问者，访问IP，访问url,访问的应用类型，访问的资源类型，访问信息备注，访问时间，访问sessionID，其中的主键是时间和IP的组合。

Username可以为空，当用户登入之后，自动填充入用户名，为以后跟踪每个用户的访问记录做铺垫。

表3-6 Visit表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 访客 | Username | Varchar(50) | No | Yes | No | 可以为空，代表游客 |
| SessionId | SessionId | Varchar(50) | Yes | No | No | 外键,指向Visit  中SessionID |
| 代理商 | userAgent | Varchar(50) | Yes | No | No | IP地址的代理  运行商 |
| 所在城市 | city | Varchar(50) | Yes | No | No | IP地址所在城市 |
| 访问时间 | time | DateTime | Yes | No | No |  |
| 使用系统 | platformType | Varchar(50) | Yes | No | No | 操作系统 |
| 浏览器类型 | browserType | Varchar(50) | Yes | No | No | 浏览器类型 |

1. 拦截IP记录(requestIp)

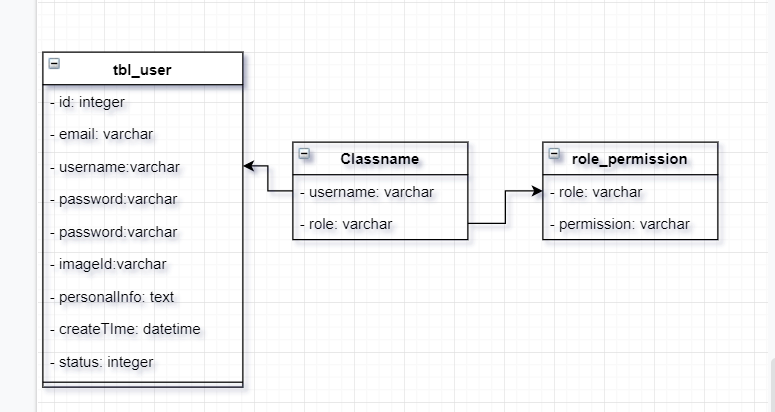
拦截IP，记录了所有被网站拦截的ip地址，为ip拦截过滤器提供数据支持。主要包含的字段有:拒绝访问的IP地址，创建的时间，被拒绝的次数。

表3-7 RequestIp表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 拦击IP | RejectIP | Varchar(50) | Yes | Yes | No |  |
| 创建时间 | CreateTime | dateTime | Yes | No | No |  |
| 拒绝次数 | reCount | Integer | Yes | No | No |  |

**用户模块类:**

用户模块类主要指的是用户的安全类，用来记录用户的角色和权限。包含了用户表，权限表，用户权限映射表.



1. 用户表(tbl\_user)

用户表用来收集用户的基础信息.包括了字段:用户名的id,用户的邮箱,用户的昵称,用户的密码,用户的头像图片.用户的个人信息简介,账号的创建时间,用户的状态.其中用户的状态包含了冻结和活跃等等.

表3-8 tbl\_user表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 用户Id | Id | Integer | Yes | Yes | No | 主键 |
| 邮箱 | CreateTime | Varchar(50) | Yes | No | No | 用户可用邮箱 |
| 用户名 | reCount | Varchar(50) | Yes | No | No |  |
| 密码 | Password | Varchar(50) | Yes | No | No |  |
| 头像ID | imageId | Varchar(50) | No | No | No | 头像的连接 |
| 个人信息 | Personalinfo | middleText | No | No | No | 个人简介 |
| 创建时间 | CreateTime | Datetime | Yes | No | No | 账号创建的时间 |
| 状态 | Status | Integer | Yes | No | No | 1:激活,2:冻结 |

1. 角色表(role\_permission)

角色表主要是规定每种角色所具有的权限的数据结构，它包含了角色和权限字段，其中角色和权限的组合是数据表的主键。

表3-9 role\_permission表结构说明

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 角色 | Role | Varchar(50) | Yes | Yes | No | 角色名 |
| 权限 | permission | Varchar(50) | Yes | No | No | 权限 |

1. 用户角色关联表(user\_role)

用户角色关联表是用来描述用户和角色之间的关系的。是一张辅助性质的角色和权限。

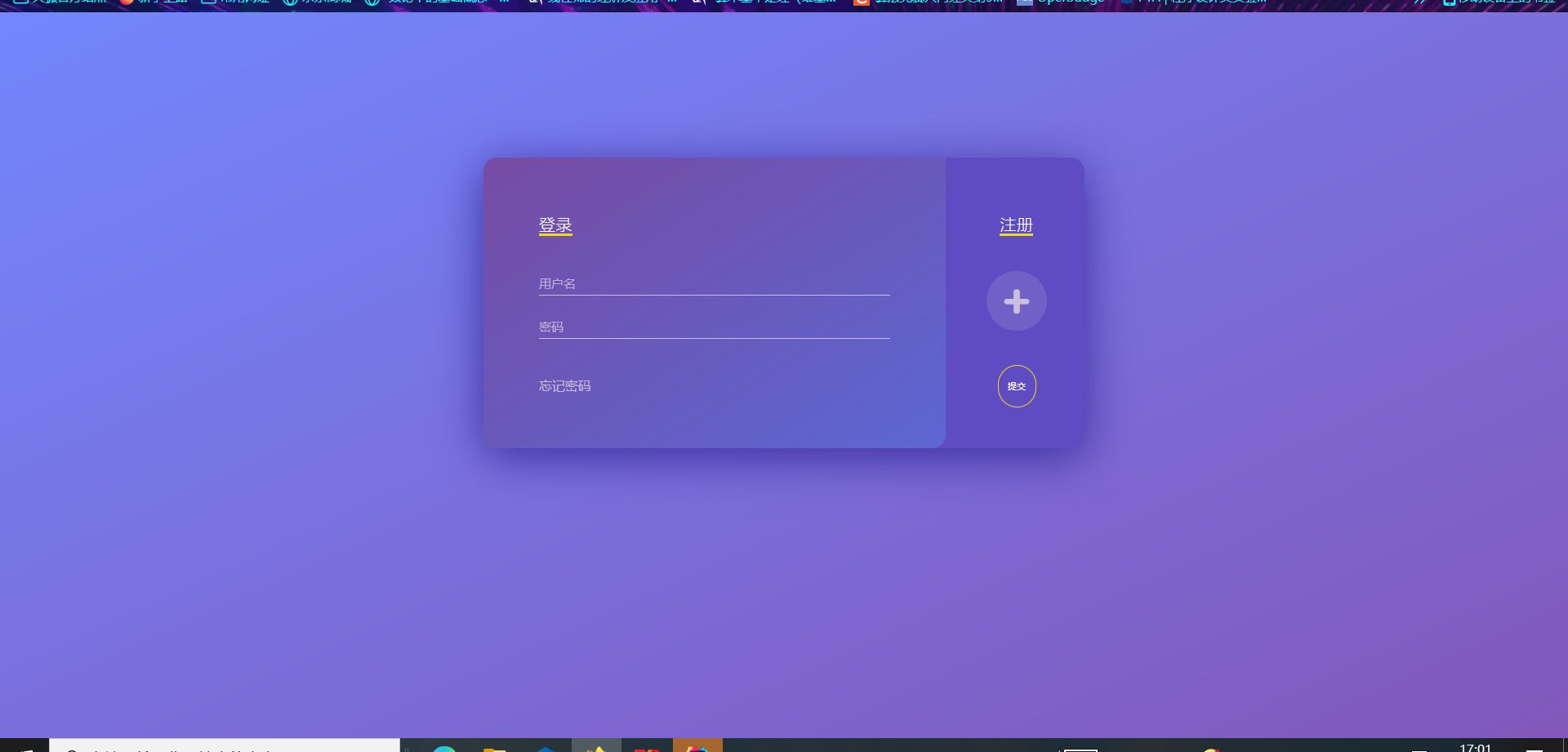
表3-9 role\_permission表结构说明

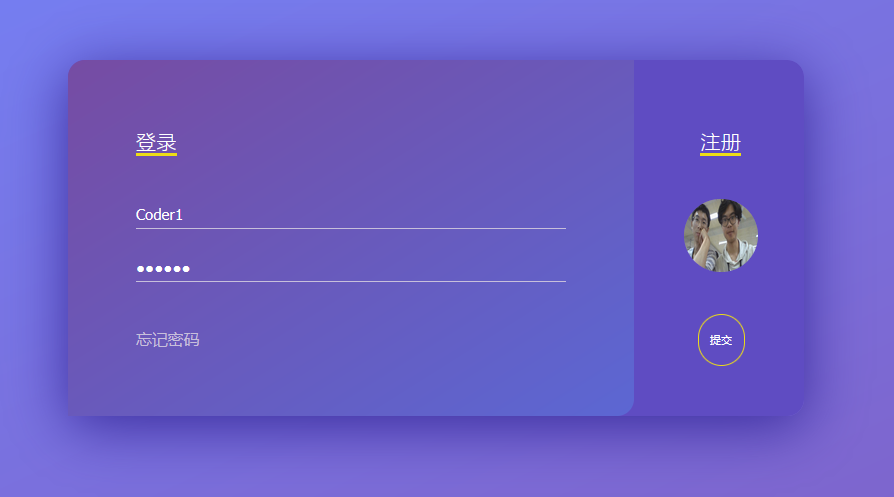
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 列名 | 字段 | 数据类型 | 非空 | 主键 | 自增 | 备注 |
| 用户名 | Username | Varchar(50) | Yes | Yes | No |  |
| 角色 | role | Varchar(50) | Yes | No | No |  |

# 4 系统实现

4.1 登入功能的实现

登入页面的设计主要有一个登入框，账号框、密码框和人物头像框。当用户输入密码账号框时对账号是否存在进行检查。然后返回头像。





4.2 注册页面

注册页面需要需要输入各种用户信息。并附带了初步的自动校验的过程。

|  |  |
| --- | --- |
|  |  |

4.3 邮箱登入检验的功能

输入正确的邮箱，然后后台将验证信息发送到对应的邮箱去，然后注册用户需要进行及时的注册。

|  |  |
| --- | --- |
|  |  |

4.4 登入的主面板

|  |  |
| --- | --- |
|  | |
|  | |
|  |  |
|  |  |

**4.5 表格渲染**

|  |  |
| --- | --- |
|  |  |
| **全局搜索** | **按照各个指标排序** |