个人博客网站

项目详细设计说明书

**编写人员：**

**编写时间：**

**审核人员：**

**版本记录：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **版本号** | **时间** | **说明** | **更改人** | **审批人** |
| 1.0 | 2017/9/8 | 详细设计文档初步完成 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

[1引言 1](#_Toc493687860)

[1.1编写目的 1](#_Toc493687861)

[1.2背景 1](#_Toc493687862)

[1.3定义 1](#_Toc493687863)

[1.4参考资料 1](#_Toc493687864)

[2程序系统的结构 2](#_Toc493687865)

[2.1系统模块 2](#_Toc493687866)

[2.2总体体系结构 2](#_Toc493687867)

[2.3处理流程 3](#_Toc493687868)

[3个人中心模块设计 3](#_Toc493687869)

[3.1模块描述 3](#_Toc493687870)

[3.2功能 3](#_Toc493687871)

[3.2.1注册 3](#_Toc493687872)

[3.2.2登陆 4](#_Toc493687873)

[3.2.3个人信息管理 4](#_Toc493687874)

[3.3输入项 5](#_Toc493687875)

[3.4输出项 5](#_Toc493687876)

[3.5算法 5](#_Toc493687877)

[3.6性能 6](#_Toc493687878)

[3.6.1精度要求 6](#_Toc493687879)

[3.6.2时间特性要求 6](#_Toc493687880)

[3.7流程逻辑 7](#_Toc493687881)

[3.8接口 8](#_Toc493687882)

[3.9模块内部设计 9](#_Toc493687883)

[3.9.1 UsersRegister类 9](#_Toc493687884)

[3.9.2 UserSignIn类 10](#_Toc493687885)

[3.9.3 InforChange类 10](#_Toc493687886)

[3.10 时序图 10](#_Toc493687887)

[3.10.1 登陆 10](#_Toc493687888)

[3.10.2注册 11](#_Toc493687889)

[3.10.3 个人信息管理 11](#_Toc493687890)

[3.11限制条件 11](#_Toc493687891)

[3.12测试计划 11](#_Toc493687892)

[4博文管理模块设计 12](#_Toc493687893)

[4.1模块描述 12](#_Toc493687894)

[4.2功能 12](#_Toc493687895)

[4.2.1发表博文 12](#_Toc493687896)

[4.2.1修改博文 12](#_Toc493687897)

[4.2.1删除博文 13](#_Toc493687898)

[4.3性能 13](#_Toc493687899)

[4.3.1精度要求 13](#_Toc493687900)

[4.3.1时间特性要求 13](#_Toc493687901)

[4.4输入项 13](#_Toc493687902)

[4.5输出项 14](#_Toc493687903)

[4.6算法 14](#_Toc493687904)

[4.7流程逻辑 15](#_Toc493687905)

[4.8接口 15](#_Toc493687906)

[4.9模块内部设计 16](#_Toc493687907)

[4.9.1 PublishArticle类 16](#_Toc493687908)

[4.9.2 updataArticle类 16](#_Toc493687909)

[4.10限制条件 16](#_Toc493687910)

[4.11测试计划 17](#_Toc493687911)

[5管理员模块设计 17](#_Toc493687912)

[5.1模块描述 17](#_Toc493687913)

[5.2功能 17](#_Toc493687914)

[5.2.1 审核博文 17](#_Toc493687915)

[5.2.1 受理举报 18](#_Toc493687916)

[5.3性能 18](#_Toc493687917)

[5.4输入项 18](#_Toc493687918)

[5.5输出项 18](#_Toc493687919)

[5.6算法 18](#_Toc493687920)

[5.7流程逻辑 19](#_Toc493687921)

[5.8接口 19](#_Toc493687922)

[5.9模块内部设计 20](#_Toc493687923)

[5.9.1 ArticleExamine类 20](#_Toc493687924)

[5.9.2 ReportExamine类 20](#_Toc493687925)

[5.10限制条件 20](#_Toc493687926)

[5.11测试计划 20](#_Toc493687927)

[6博文浏览模块设计 21](#_Toc493687928)

[6.1模块描述 21](#_Toc493687929)

[6.2功能 21](#_Toc493687930)

[6.2.1搜索博文 21](#_Toc493687931)

[6.2.2查看博文 21](#_Toc493687932)

[6.2.3排行榜 22](#_Toc493687933)

[6.2.4收藏博文 22](#_Toc493687934)

[6.2.5发表评论 22](#_Toc493687935)

[6.2.6评分 23](#_Toc493687936)

[6.2.7举报 23](#_Toc493687937)

[6.2.8分享 23](#_Toc493687938)

[6.3性能 24](#_Toc493687939)

[6.3.1精度要求 24](#_Toc493687940)

[6.3.2时间特性要求 24](#_Toc493687941)

[6.4输入项 24](#_Toc493687942)

[6.5输出项 25](#_Toc493687943)

[6.6算法 25](#_Toc493687944)

[6.7流程逻辑 26](#_Toc493687945)

[6.8接口 26](#_Toc493687946)

[6.9模块内部设计 27](#_Toc493687947)

[6.9.1 UserInfoService类 27](#_Toc493687948)

[6.9.2 BlogService类 28](#_Toc493687949)

[6.9.3 BlogListServlet类 28](#_Toc493687950)

[6.10限制条件 28](#_Toc493687951)

[6.11测试计划 28](#_Toc493687952)

# 1引言

## 1.1编写目的

本文档是建立在概要设计文档之上，也参考了需求文档及数据库设计文档，为明确个人博客网站的结构与功能的详细划分，对各模块的功能调用细节，安排项目规划与进度、制定详细测试计划、组织软件开发与测试，特撰写此文档。

在上一阶段概要设计中，已经明确系统的模块功能的划分，大致划分了各个模块的接口、接口的设计，同时也对数据结构作出定义。在以下的详细设计报告中将对在本阶段中对系统所做的详细设计进行说明。

本文档的预期读者为与个人博客网站的开发有利益关系的决策人员，项目开发人员，支持本项目的领导和公司人员以及软件的验收者

## 1.2背景

该项目是运行在浏览器上面的个人博客网站项目。开发者为2014级软件工程专业第三小组。

本网站的架构为B/S，致力于为用户提供便捷高效的博客体验，在各主流网页浏览器上通过简便的操作即可实现博客功能。方便技术，经验，总结等知识的分享与传播，使更多人获益。

本系统是一套完整独立的系统

## 1.3定义

B/S：浏览器/服务器端的软件架构。

Chrome：是一款由Google公司开发的网页浏览器。

Firefox：是一个自由及开放源代码网页浏览器。

SQL：结构化查询语言，关系数据库查询的标准语言。

## 1.4参考资料

1. 《详细设计说明书》（GB8567——88）国标
2. 《个人博客网站 软件需求说明书》第三小组内部资料
3. 《个人博客网站 概要设计说明书》第三小组内部资料

# 2程序系统的结构

## 2.1系统模块

|  |  |  |
| --- | --- | --- |
| 系统名称 | 模块 | 功能 |
| 个人博客网站系统 | 博文浏览 | 分享博文、收藏博文、发表评论、举报、评分、搜索博文、查看博文、排行榜 |
| 博文管理 | 发表博文、删除博文、修改博文 |
| 个人中心 | 登录、注册、个人信息管理 |
| 管理员 | 审核博文，受理举报 |

表2-1 功能-模块规定表

## 2.2总体体系结构

用框图的形式说明本系统的系统元素的划分。

图2-1 网站模块功能结构图

图2-2页面跳转关系

## 2.3处理流程

Tomcat 服务器提供 HTTP 服务给用户前端。

1. 用户前端在向 Tomcat 服务器发请求时，以 URI 地址标识要进行的操作，如有附加数据，转成 JSON 放至 HTTP body 内，一同发送给 Tomcat。

2. Tomcat 服务器在接受到 HTTP 请求后，解析 JSON 数据，转成内部使用对象，进行后续操作。

3. 而后，Tomcat 发送 HTTP response 给用户前端，如有附加数据，转成 JSON 放至 HTTP body 内。

# 3个人中心模块设计

## 3.1模块描述

“个人中心模块”包括注册、登录和个人信息管理。只有注册并成功登录之后，才可以管理个人信息。

## 3.2功能

### 3.2.1注册

新用户打开网站后，点击新用户注册，获得一个新账户。

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 注册 | | |
| **编号：** | GRZX\_ZC | | |
| **被调用：** | 用户登录 | **调用：** | 无 |
| **输入：** | 注册信息 | **输出：** | 注册提示信息 |
| **处理：** | 检测输入是否合法，上传并保存用户信息，创建新账户 | | |
| **局部数据元素：** | 注册账户，密码，身份信息 | **注释：无** | |

### 3.2.2登陆

此功能用于用户登录，用户输入用户账号，密码登录系统。登录成功后跳转至主页，并刷新列表。登陆后可以使用更多博客功能。

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 登陆 | | |
| **编号：** | GRZX\_DL | | |
| **被调用：** | 注册 | **调用：** | 无 |
| **输入：** | 登陆信息 | **输出：** | 登陆提示信息 |
| **处理：** | 检测输入是否合法，验证信息是否正确，并返回结果 | | |
| **局部数据元素：** | 用户账户、密码 | **注释：无** | |

### 3.2.3个人信息管理

在用户的个人中心，点击编辑个人资料，进入编辑个人信息界面。用户可以修改个人相应的信息。

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 个人信息管理 | | |
| **编号：** | GRZX\_GRXXGL | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 修改项信息 | **输出：** | 修改提示信息 |
| **处理：** | 检测输入是否合法，是否有未填写项目，操作数据库数据 | | |
| **局部数据元素：** | 用户个人信息 | **注释：无** | |

## 3.3输入项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 标识 | 数据类型格式 | 有效范围 | 输入来源 |
| 用户名 | Username | 字符型 | 不超过16字符 | 用户 |
| 密码 | Password | 字符型 | 不超过20位的字母或数字 | 用户 |
| 电话号码 | Phone | 整型 | 存在的电话号码 | 用户 |
| 性别 | Gender | 字符型 | “男”，“女” | 用户 |
| 自我介绍 | Introduce | 字符型 | 不超过200字符 | 用户 |
| 城市 | City | 字符型 | 不超过9字符 | 用户 |

## 3.4输出项

对于用户单击按钮的行为，软件会做出相应的反应，将信息反馈给用户。以下是输出表格。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 标识 | 数据类型格式 | 有效范围 | 输入来源 |
| 用户名 | Username | 字符型 | 存在的手机号 | 用户 |
| 密码 | Password | 字符型 | 不超过18位的字母或数字 | 用户 |
| 姓名 | Name | 字符型 | 2-24个字符 | 用户 |
| 性别 | Gender | 字符型 | 男/女 | 用户 |
| 自我介绍 | Introduce | 字符型 | 不超过200字符 | 用户 |
| 电话号码 | Phone | 整型 | 存在的电话号码 | 用户 |
| 城市 | City | 字符型 | 不超过9字符 | 用户 |

## 3.5算法

本模块不负责对数据进行复杂的处理，主要通过和数据库的交互来提供用户所需的信息。

密码需要加密后进行传输，在服务器端对密文进行验证。需要使用数据的加密传输算法。

部分方法概述：

public boolean signup(UserInfo userInfo) throws Exception {

//用户注册

}

public UserInfo signin(UserInfo userInfo) throws Exception {

//用户登录

}

## 3.6性能

在网络畅通的情况下，用户注册与登陆的操作响应时间应该在可接受的范围以内，最好在1秒以内。

### 3.6.1精度要求

1. 用户账号：不超过16字符
2. 昵称：不超过16位
3. 用户密码：不超过20字符
4. 上传图片大小：不超过25MB

### 3.6.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于2s
3. 数据的转换和传送时间；小于4s

## 3.7流程逻辑



图3-1 用户注册流程图



图3-2 用户登陆流程图



图3-3 个人信息管理流程图

## 3.8接口

用户接口：本系统采用图形用户接口，以鼠标和键盘为用户接口，方便用户对博客系统的有效操作。更好地在博客里进行交流。

内部接口：内部是页面和数据库连接，对应的页面输入框和显示框的内容在数据库得到显示。

## 屏幕截图(1)3.9模块内部设计

图3-4 类图

### 3.9.1 UsersRegister类

此类负责处理用户注册，需要对用户的注册信息进行检查。

|  |  |
| --- | --- |
| 说明 | 检查用户身份是否合法 |
| 函数原型 | bool RegisterCheck (varchar Username,varchar Password) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | TURE表示注册信息无误，FALSE表示注册信息有误 |
| 参数描述 | Count:用户填写的账号 Password:用户填写的密码 |

|  |  |
| --- | --- |
| 说明 | 更新用户注册信息 |
| 函数原型 | void UpdateUsers (varchar Username,varchar Password) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | TURE更新成功，FALSE拒绝更新 |
| 参数描述 | Count:用户填写的账户 Password:用户填写的密码 |

### 3.9.2 UserSignIn类

此类负责处理用户登陆。

|  |  |
| --- | --- |
| 说明 | 处理用户登陆请求 |
| 函数原型 | bool SignInCheck (varchar Count,varchar Password) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | TRUE表示用户登陆成功，FALSE表示用户登陆失败 |
| 参数描述 | Count:用户填写的账户 Password:用户填写的密码 |

### 3.9.3 InforChange类

此类处理用户信息变更。

|  |  |
| --- | --- |
| 说明 | 变更用户信息 |
| 函数原型 | void ChangeInfor (UsersInforList\* list) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | TRUE成功更改，FALSE拒绝更改 |
| 参数描述 | 用户信息表 |

|  |  |
| --- | --- |
| 说明 | 检查信息变更是否合法 |
| 函数原型 | bool LegalChange(UsersInforList\* list) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | TRUE表示信息变更合法，FALSE表示不合法 |
| 参数描述 | 用户信息表 |

## 3.10 时序图

### 3.10.1 登陆



图3-5登陆时序图

### 3.10.2注册

图3-6注册时序图

### 3.10.3 个人信息管理

图3-7 个人信息管理时序图

## 3.11限制条件

软件必须连接到服务器才可以进行相关操作，不支持离线操作。

## 3.12测试计划

模块编写完毕后，进行注册、登录测试，验证数据的准确性。进行用户信息的增、删、改测试，确保操作的准确性。确保数据准确无误即可。

# 4博文管理模块设计

## 4.1模块描述

用户登录个人博客系统，审核成功后可进入博文管理界面。该模块非常驻内存，是子程序，无特殊覆盖要求，是并发处理的。

## 4.2功能

### 4.2.1发表博文

用户选择发表新的博文，输入博文内容，点击发表按钮或选择中途退出，显示发表成功或失败。

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 发表博文 | | |
| **编号：** | BWGL\_FB | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 博文信息 | **输出：** | 发表提示信息 |
| **处理：** | 前端向服务端发送博文，服务端接收到博文信息后，将博文信息置于审核队列中。 | | |
| **局部数据元素：** | 无 | **注释：无** | |

### 4.2.1修改博文

用户选择修改博文，输入新的博文内容，点击修改按钮或选择中途退出，显示修改成功或失败。

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 修改博文 | | |
| **编号：** | BWGL\_XG | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 博文信息 | **输出：** | 结果提示信息 |
| **处理：** | 服务端接收到信息后，更新相应博文的信息。 | | |
| **局部数据元素：** | 无 | **注释：无** | |

### 4.2.1删除博文

用户选择删除博文，选择个人的博文，点击删除按钮，显示删除成功。

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 删除博文 | | |
| **编号：** | BWGL\_SC | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 博文编号 | **输出：** | 删除结果 |
| **处理：** | 用户选择删除指定博文，服务端接收到数据后，将博文信息移入回收站。 | | |
| **局部数据元素：** | 无 | **注释：无** | |

## 4.3性能

### 4.3.1精度要求

（1） 博文内容不能为空或超出10000个字符

（2） 博文标题不能为空或找出200个字符

（3） 上传图片大小不能超过25MB

### 4.3.1时间特性要求

（1）操作响应时间：小于0.5s

（2）更新处理时间：小于2s

（3）数据转换和传送时间：小于4s

## 4.4输入项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 标识 | 数据类型 | 有效范围 | 输入来源 |
| 博文ID | Article\_id | 字符串 | 不超过11个字符 | 用户键入 |
| 博文内容 | Content | 字符串 | 不超过10000个字符 | 用户键入 |
| 发表时间 | Date | 日期 | 不超过20个字符 | 用户键入 |
| 文章标题 | Article\_name | 字符串 | 不超过200个字符 | 用户键入 |
| 文章摘要 | Digest | 字符串 | 不超过1000个字符 | 用户键入 |
| 用户名字 | Username | 字符串 | 不超过10个字符 | 用户键入 |
| 密码 | Password | 字符串 | 不超过20个字符 | 用户键入 |
| 用户注册时间 | UserRegistrationTime | 日期 | 不超过30个字符 | 系统自动生成 |

## 4.5输出项

此模块的输入项同时也作为输出项。

## 4.6算法

此模块主要进行数据的查找和修改操作，不涉及特殊的算法。

部分方法概述：

public Article getDeletedArticleInfoByArticleId(int article\_id){

//根据博文ID查找博文

}

public List<Article> getAllArticleInfo(String article\_name){

//根据博文题目查找博文列表

}

public List<Article> getAllArticleOrderByDate(){

//把博文按照日期排序

}

public void updateArticle(Article article){

//修改博文信息

}

## 4.7流程逻辑

****图4-1 博文管理流程图

## 4.8接口



图4-2 博文管理接口

## 4.9模块内部设计

### 4.9.1 PublishArticle类

负责提供发表文章的接口。

|  |  |
| --- | --- |
| 说明 | 从表单中获得文章内容 |
| 函数原型 | Article fetchArticle(Form article) |
| 返回值 | Article对象 |
| 返回值描述 | 返回的对象 |
| 参数描述 | 表单 |

|  |  |
| --- | --- |
| 说明 | 添加新的博文 |
| 函数原型 | Bool addArticle(Article newArticle) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | 布尔值 |
| 参数描述 | 新博文 |

### 4.9.2 updataArticle类

此类用于更新数据库中的博文信息，包括更新和删除。

|  |  |
| --- | --- |
| 说明 | 更新文章内容 |
| 函数原型 | Bool updataArticle(Form article) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | 布尔值 |
| 参数描述 | 表单 |

|  |  |
| --- | --- |
| 说明 | 删除指定博文，并移入回收站 |
| 函数原型 | Bool deleteArticle(int articleId) |
| 返回值 | TRUE/FALSE |
| 返回值描述 | 布尔值 |
| 参数描述 | 博文编号 |

## 4.10限制条件

1. 输入的博文内容不能超过10000个字符
2. 图片大小不能超过25MB
3. 输入博文的标题不能超过200个字符
4. 生成的博文摘要不能超过500个字符
5. 用户已登录

## 4.11测试计划

本模块编写完毕后，进行博文的发表，修改和删除等测试，确保操作的正确性。

A.若用户未登录，则提示“未登录”，若已登录，则满足要求

B.若用户输入博文的标题超过100个字符或者为空，则提示“字数超出”或“不能为空”，若在有效范围之内，则满足要求

C.若用户输入博文的内容超过20000个字符或者为空，则提示“字数超出”或“不能为空”，若在有效范围之内，则满足要求

# 5管理员模块设计

## 5.1模块描述

管理员登录单独管理员页面，之后进入审核和受理举报页面，审核通过，则博文发表，审核不通过，管理员将把该文章删除。

## 5.2功能

### 5.2.1 审核博文

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 审核博文 | | |
| **编号：** | GLY-SH | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 待审核博文 | **输出：** | 审核结果 |
| **处理：** | 管理员登陆系统，进入审核博文页面，查阅待审核博文，给出审核结果。 | | |
| **局部数据元素：** | 无 | **注释：审核不通过的文章直接从数据库删除** | |

### 5.2.1 受理举报

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 改签模块 | | |
| **编号：** | GLY-SLJB | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 举报信息 | **输出：** | 处理结果 |
| **处理：** | 管理员登陆系统，进入受理举报页面，查阅举报信息，给出处理结果。 | | |
| **局部数据元素：** | 无 | **注释：无** | |

## 5.3性能

本模块对程序性能没有特殊要求。

## 5.4输入项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 管理员 | 标识 | 数据类型 | 有效范围 | 输入来源 |
| 管理员编号 | admin\_id | 字符串 | 不超过11个字符 | 管理员 |
| 登录密码 | Admin\_passwd | 字符串 | 不超过20个字符 | 管理员 |

## 5.5输出项

并不直接输出任何东西，由管理员直接在后台进行相应操作。

## 5.6算法

部分方法概述：

public boolean adminLogin(String adminId, String passwd){

//管理员登录

}

public Article ArrayList<Article> pendingArticleList(){

//显示所有未审核的博文

}

public boolean checkArticle(int article\_id){

//根据博文ID检查与修改博文审核状态

}

## 5.7流程逻辑



图5-1 管理员业务流程图

## 5.8接口

管理员模块是一个独立的模块，直接与数据库进行交互，不需要提供任何外部接口。

## 5.9模块内部设计

### 5.9.1 ArticleExamine类

此类用于显示待审核的博文列表，以及接受处理结果。

|  |  |
| --- | --- |
| 说明 | 显示当前待审核的用户文章 |
| 函数原型 | void showPendingArticles() |
| 返回值 | 无 |
| 返回值描述 | 无 |
| 参数描述 | 无 |

|  |  |
| --- | --- |
| 说明 | 审核通过，允许发表，将调用checkArticle()修改审核状态 |
| 函数原型 | void examineSuccess(int articleId) |
| 返回值 | 无 |
| 返回值描述 | 无 |
| 参数描述 | 待审核的博文ID |

|  |  |
| --- | --- |
| 说明 | 审核失败，拒绝发表，并删除 |
| 函数原型 | void examineFail(int articleId) |
| 返回值 | 无 |
| 返回值描述 | 无 |
| 参数描述 | 待审核的博文ID |

### 5.9.2 ReportExamine类

此类用户只用来显示用户的举报信息。需要管理员自行操纵数据库进行后续举报处理工作。

|  |  |
| --- | --- |
| 说明 | 显示当前待审核的用户文章 |
| 函数原型 | void showPendingReport() |
| 返回值 | 无 |
| 返回值描述 | 无 |
| 参数描述 | 无 |

## 5.10限制条件

管理员必须按照规章制度进行文章审核与举报处理工作。

## 5.11测试计划

本模块编写完毕后，进行管理员相关操作的测试，编写测试用例，确保数据准确无误。

# 6博文浏览模块设计

## 6.1模块描述

用户在网站主页可以查看博文，搜索博文，在博文详情页中可以举报博文、评论，还可以分享博文。在用户举报、分享时通过点击相应的按钮触发。

评论功能仅限注册用户使用，以下IPO表假设用户已经登录，并进入浏览博文界面中的评论版块。用户已登录，在浏览博文或评论后，点击发表评论按钮或回复按钮，输入评论内容，出现提示信息，软件返回到评论界面，显示新增评论或回复。

## 6.2功能

### 6.2.1搜索博文

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 搜索 | | |
| **编号：** | LL\_SS | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 关键字 | **输出：** | 博文列表 |
| **处理：** | 服务端接受关键字，从数据库中搜索并返回博文列表 | | |
| **局部数据元素：** | 关键字 | **注释：无** | |

### 6.2.2查看博文

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 查看博文 | | |
| **编号：** | LL\_CK | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 博文编号 | **输出：** | 博文页面 |
| **处理：** | 前端传入博文编号，从数据库中取得博文内容，传回前端。 | | |
| **局部数据元素：** | 博文编号 | **注释：**无 | |

### 6.2.3排行榜

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 排行榜 | | |
| **编号：** | LL\_PHB | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 无 | **输出：** | 博文列表 |
| **处理：** | 加载指定页面时自动加载博文列表 | | |
| **局部数据元素：** | 无 | **注释：**无 | |

### 6.2.4收藏博文

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 收藏博文 | | |
| **编号：** | LL\_SC | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 博文编号，用户编号 | **输出：** | 无 |
| **处理：** | 将指定博文编号存入指定用户的收藏夹中 | | |
| **局部数据元素：** | 博文编号，用户编号 | **注释：无** | |

### 6.2.5发表评论

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 发表评论 | | |
| **编号：** | LL\_FBPL | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 用户编号，评论内容，博文编号 | **输出：** | 无 |
| **处理：** | 将评论内容，用户编号和博文编号存入数据库中 | | |
| **局部数据元素：** | 无 | **注释：**无 | |

### 6.2.6评分

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 评分 | | |
| **编号：** | LL\_PF | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 博文编号，用户编号 | **输出：** | 无 |
| **处理：** | 指定项目点击数加1 | | |
| **局部数据元素：** | 无 | **注释：**无 | |

### 6.2.7举报

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 举报 | | |
| **编号：** | LL\_JB | | |
| **被调用：** | 无 | **调用：** | 无 |
| **输入：** | 用户编号，博文编号 | **输出：** | 无 |
| **处理：** | 服务端将举报信息存入举报队列 | | |
| **局部数据元素：** | 无 | **注释：**无 | |

### 6.2.8分享

|  |  |  |  |
| --- | --- | --- | --- |
| **IPO表** | | | |
| **模块：** | 分享 | | |
| **编号：** | CKXX\_CX | | |
| **被调用：** | 无 | **调用：** | 操作系统剪切板 |
| **输入：** | 无 | **输出：** | 无 |
| **处理：** | 点击分享后，自动将博文链接复制到系统剪贴板 | | |
| **局部数据元素：** | 无 | **注释：需要系统允许浏览器操纵剪切板** | |

## 6.3性能

### 6.3.1精度要求

1. 搜索关键词：50字符
2. 举报信息：不超过1000字符
3. 分享信息：不超过100字符
4. 上传图片大小：25MB以内
5. 评论字数限制：不超过200个字符

### 6.3.2时间特性要求

1. 操作响应时间：小于0.5s
2. 更新处理时间：小于2s
3. 数据的转换和传送时间；小于4s

## 6.4输入项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 博文 | 标识 | 数据类型 | 有效范围 | 输入来源 |
| 博文ID | articleId | 整形 | 不超过11个字符 | 用户 |
| 博文内容 | content | 字符串 | 不超过10000个字符 | 用户 |
| 发表时间 | date | 日期 | 不超过20个字符 |  |
| 文章标题 | article\_name | 字符串 | 不超过20个字符 | 用户 |
| 文章摘要 | digest | 字符串 | 不超过100个字符 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评论 | 标识 | 数据类型 | 有效范围 | 输入来源 |
| 评论ID | comment\_id | 整形 | 不超过10个字符 | 系统 |
| 评论内容 | content | 字符串 | 不超过200个字符 | 用户 |
| 回复内容 | Replycontent | 字符串 | 不超过200个字符 | 用户 |
| 博文id | article\_id | 整形 | 不超过10个字符 | 系统 |
| 用户id | user\_id | 整形 | 不超过10个字符 | 系统 |

## 6.5输出项

对于用户单击按钮的行为，软件会做出相应的反应，将信息反馈给用户。

## 6.6算法

本模块不涉及复杂的算法，主要进行和数据库的相关读写操作。

在显示排行榜博文列表时，需要使用搜索算法来取得数据库中的符合要求的博文。

部分方法：

public List<Map<String, Object>> queryCommentsByBlogId(int arti-cle\_id) throws Exception {

//根据博文id查询评论内容

}

public void addComment(Comment comment) throws Exception {

//添加评论

}

public void addNewArticle(Article\_Delete article\_delete){

//往回收站表添加博文

}

## 6.7流程逻辑

大部分流程逻辑已在功能IPO表中描述完毕，以下举例。



图6-1 发表评论流程图

## 6.8接口

用户在浏览博文模块中可以查看博文和评论，然后用户可以选择进入用户评论模块对选中的博文或评论进行评论或回复。输入参数为评论输入项（上面已经描述过）的数据结构。

## 6.9模块内部设计

图6-5 UML类图

### 6.9.1 UserInfoService类

此类为用户服务接口，提供注册，登陆等功能。

|  |  |
| --- | --- |
| 说明 | 查询个人博文 |
| 函数原型 | Public static List<Article> queryMyBlog(string userId) |
| 返回值 | List<Article> |
| 返回值描述 | 博文列表 |
| 参数描述 | ID是用户编号 |

|  |  |
| --- | --- |
| 说明 | 访问作者页面 |
| 函数原型 | Public static List<Article>getAuthorInfo(string userId) |
| 返回值 | List<Article> |
| 返回值描述 | 博文列表 |
| 参数描述 | ID是其他博主的用户编号 |

### 6.9.2 BlogService类

此类为博客服务接口， 提供评分，评论，访问等功能。

|  |  |
| --- | --- |
| 说明 | 查看博文 |
| 函数原型 | public static Atricle quaryArticleById(String ID) |
| 返回值 | Atricle |
| 返回值描述 | 指定博文内容 |
| 参数描述 | ID是博文编号 |

|  |  |
| --- | --- |
| 说明 | 评分-like |
| 函数原型 | Public static void like(String ID) |
| 返回值 | 无 |
| 返回值描述 | 无 |
| 参数描述 | ID是博文编号 |

### 6.9.3 BlogListServlet类

此类为servlet实现类，负责接口的具体实现。

|  |  |
| --- | --- |
| 说明 | 发表评论 |
| 函数原型 | public static void addComment(String articleId,String userId,String comment) |
| 返回值 | 无 |
| 返回值描述 | 无 |
| 参数描述 | 博文编号，用户ID与评论内容 |

## 6.10限制条件

a.输入的评论内容不超过200个字符

b.回复的内容不超过200个字符

c.上传的图片大小在25MB以内

d.评论的内容不能为空，将视为无效的评论，不能发表

e.输入的举报信息不超过1000字符。

f.输入项的搜索关键字不超过50字符。

## 6.11测试计划

由模块编写人员组织测试，编写测试用例。