

多
功
能
博
客
系
统
设
计
报
告

Author: 15779604138@163.com

目录

- 1. 简要需求分析.....3
 - 1.1 项目背景.....3
 - 1.2 功能需求.....3
 - 1.3 性能需求.....3
- 2. 数据库设计.....4
 - 2.1 数据流图.....4
 - 2.1.1 博客模块.....4
 - 2.1.2 相册模块.....4
 - 2.1.3 留言模块.....5
 - 2.1.4 推荐模块.....6
 - 2.1.5 个人信息模块.....6
 - 2.2 概念结构设计.....7
 - 2.3 逻辑结构设计.....8
 - 2.4 物理结构设计.....9
- 3.系统设计.....9
 - 3.1 系统概要设计.....9
 - 3.1.1 系统构架设计.....9
 - 3.1.2 系统结构设计.....11
 - 3.1.3 系统开发环境.....12
 - 3.1.4 系统运行环境.....12
 - 3.1.5 系统接口设计及实现.....12
 - 3.2 系统功能结构图.....13
 - 3.2.1 游客用例图.....13
 - 3.2.2 用户用例图.....14
 - 3.2.3 管理员用例图.....15
- 4.系统界面截图.....15
 - 4.1 系统首页.....15
 - 4.2 博客管理.....16
 - 4.3 相册管理.....21
 - 4.4 留言管理.....24
 - 4.5 推荐管理.....25
 - 4.6 信息管理.....26
- 5.附数据库脚本.....30

1. 简要需求分析

1.1 项目背景

过去很多人都喜欢写文章写日记以及交流自己的文章和作品，以求实现相互间的沟通、展现自己的才华和让别人了解自己的想法观点。现在的网络已经成为人们生活中不可或缺的一个元素，所以自然而然诞生了博客这样一个新兴事物，它不仅仅能取代前面所说的功能，还能加入图片，而且使得作者更能无所拘束地生动地写出自己想写的，旁人也能非常便捷地阅读并且加以评论，并且它还能作为展示个人个性的窗户。个人博客现在已经成为很多人生活中必不可少的一部分，方便了人与人之间的沟通和交流。

1.2 功能需求

多功能博客系统，系统为四大模块：分别是“博主信息”、“最新留言”、“博主推荐”、“我的文章”及“我的相册”。

博客管理员拥有以下功能：

- a) 登录博客系统的后台管理平台。
- b) 更改博客账号的基本信息。
- c) 撰写博客。
- d) 博客管理。
- e) 评论管理。
- f) 留言管理。
- g) 照片上传管理

博客访问者具有以下功能：

- a) 浏览博客系统中的博客、留言、评论。
- b) 发表评论及留言。
- c) 查看相册及浏览照片。

1.3 性能需求

电脑性能要求

CPU 赛扬及以上

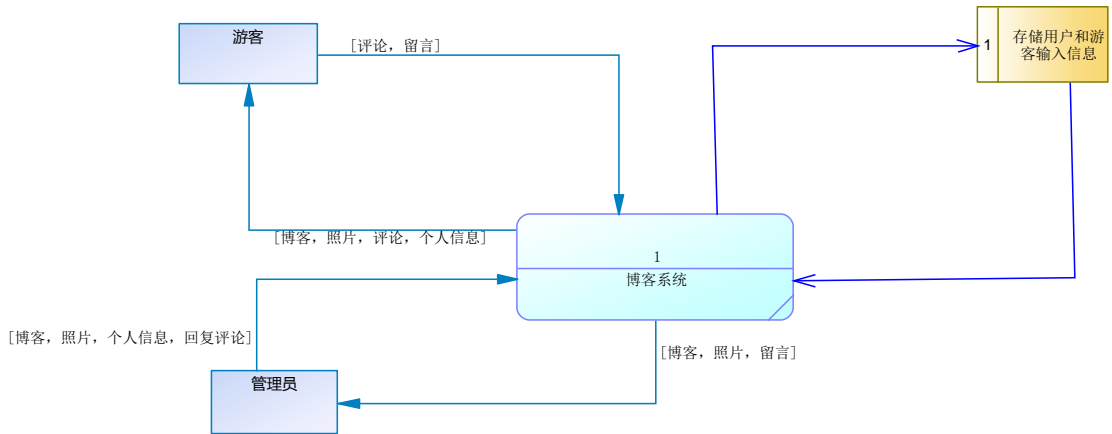
浏览器支持

谷歌浏览器

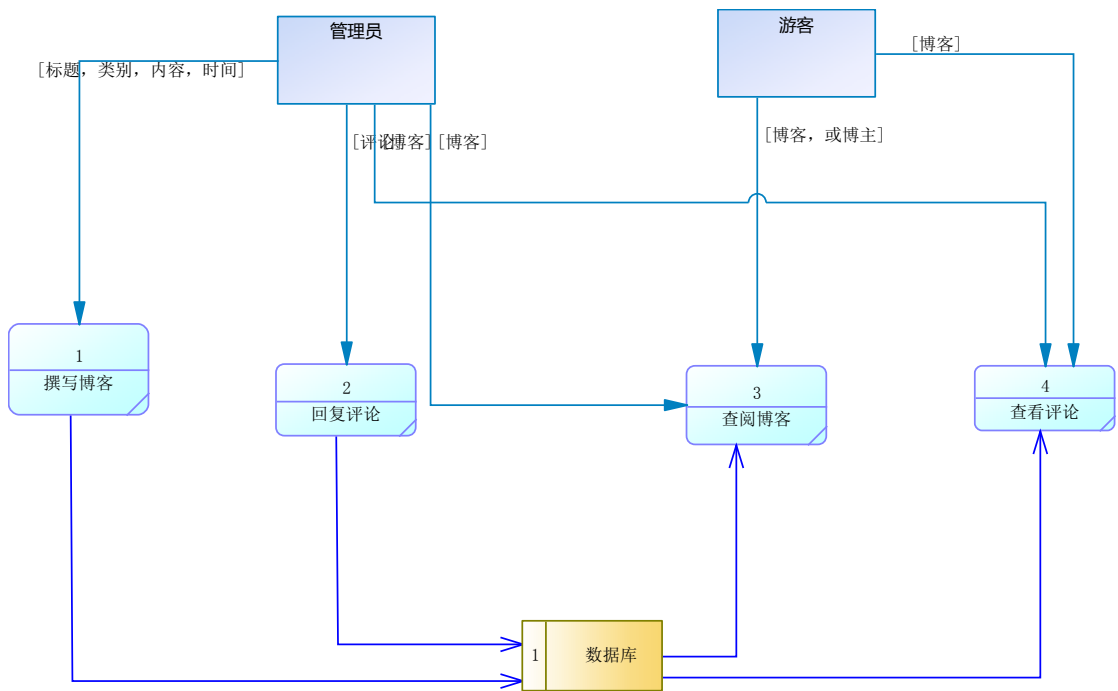
2.数据库设计

2.1 数据流图

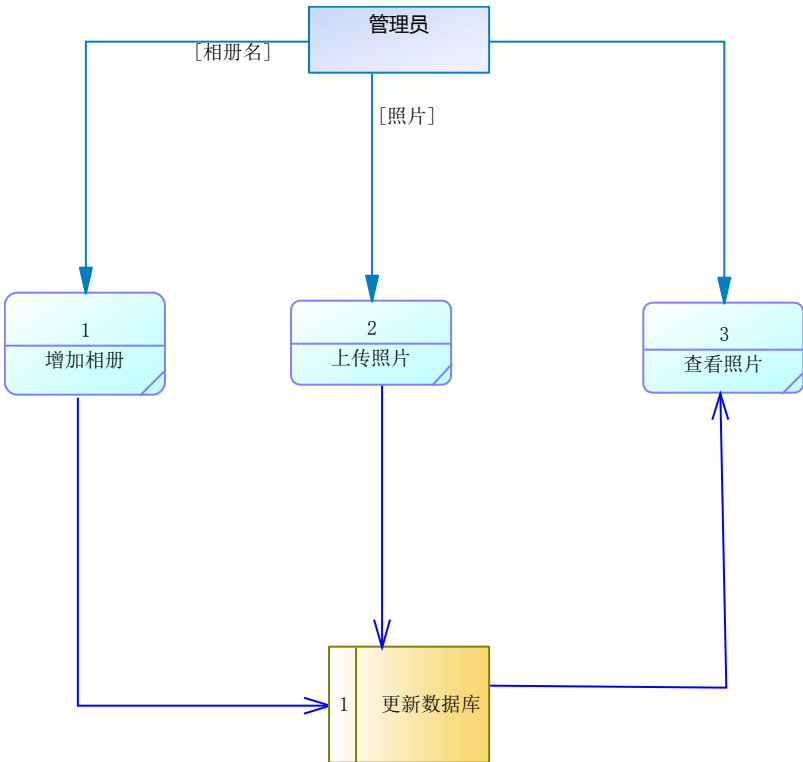
顶层数据流图



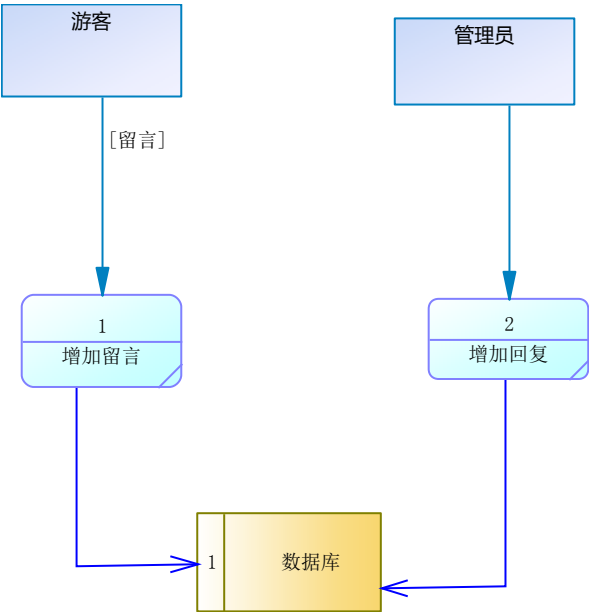
2.1.1 博客模块



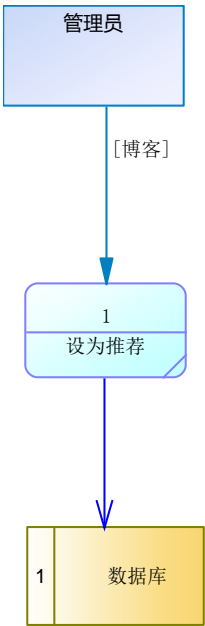
2.1.2 相册模块



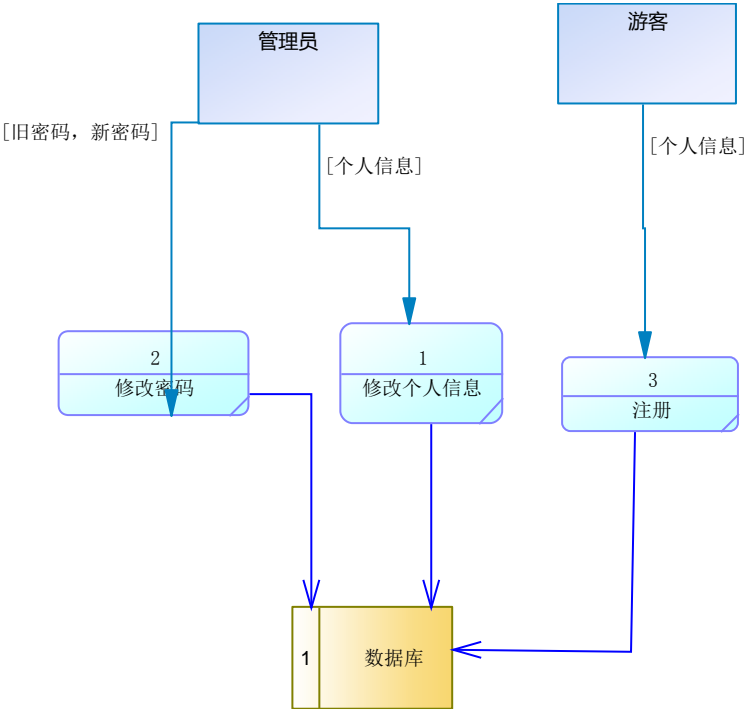
2.1.3 留言模块



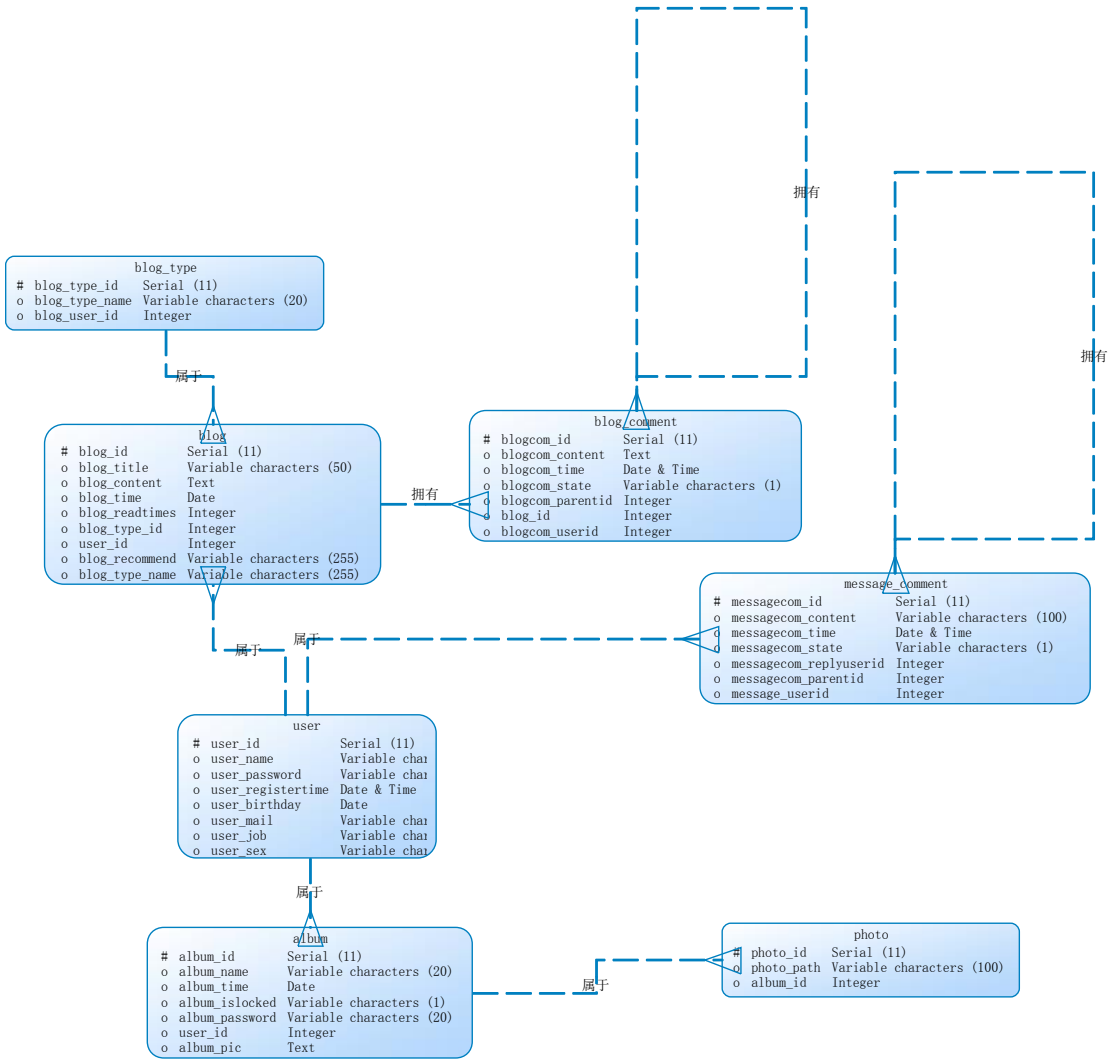
2.1.4 推荐模块



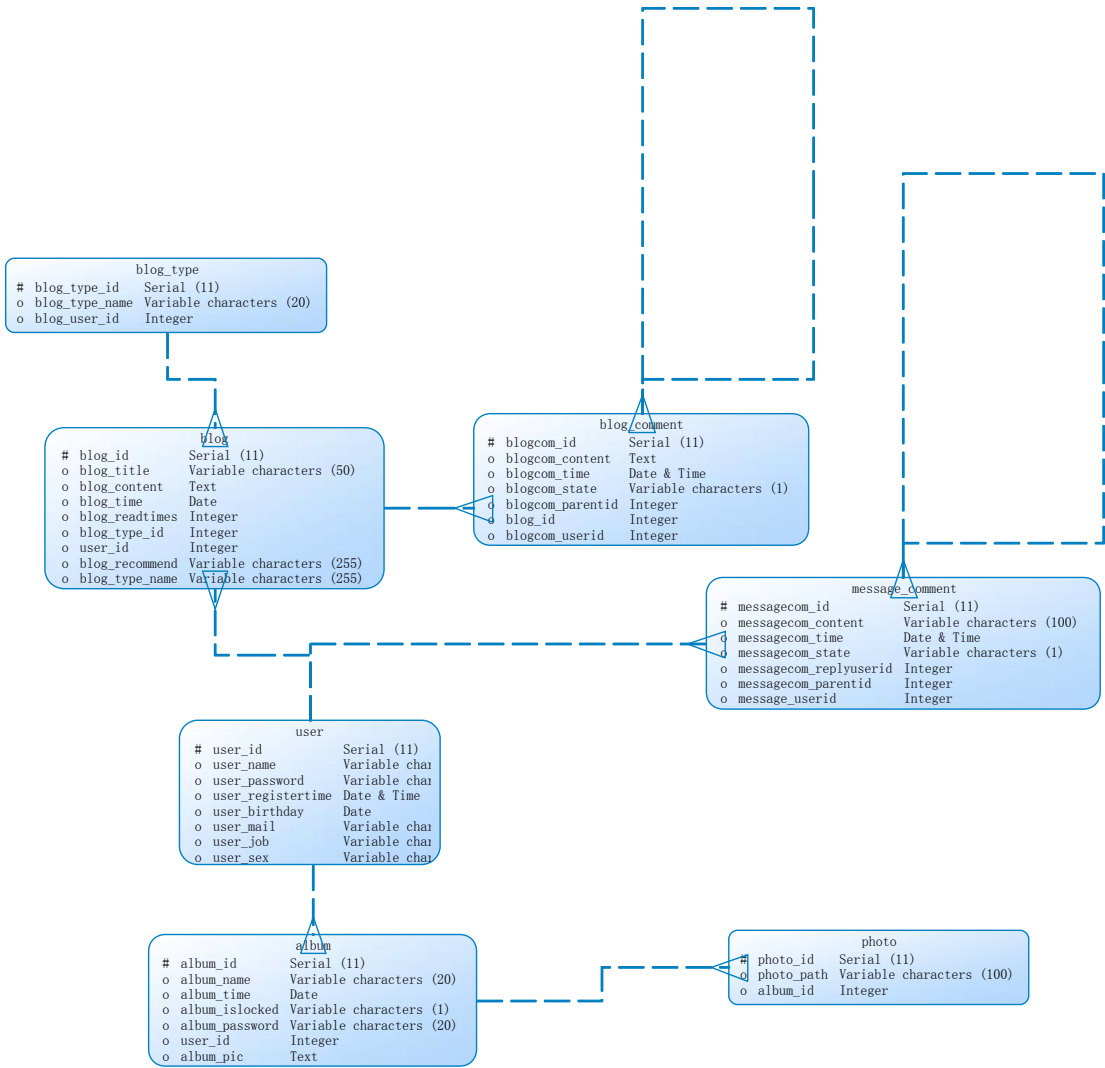
2.1.5 个人信息模块



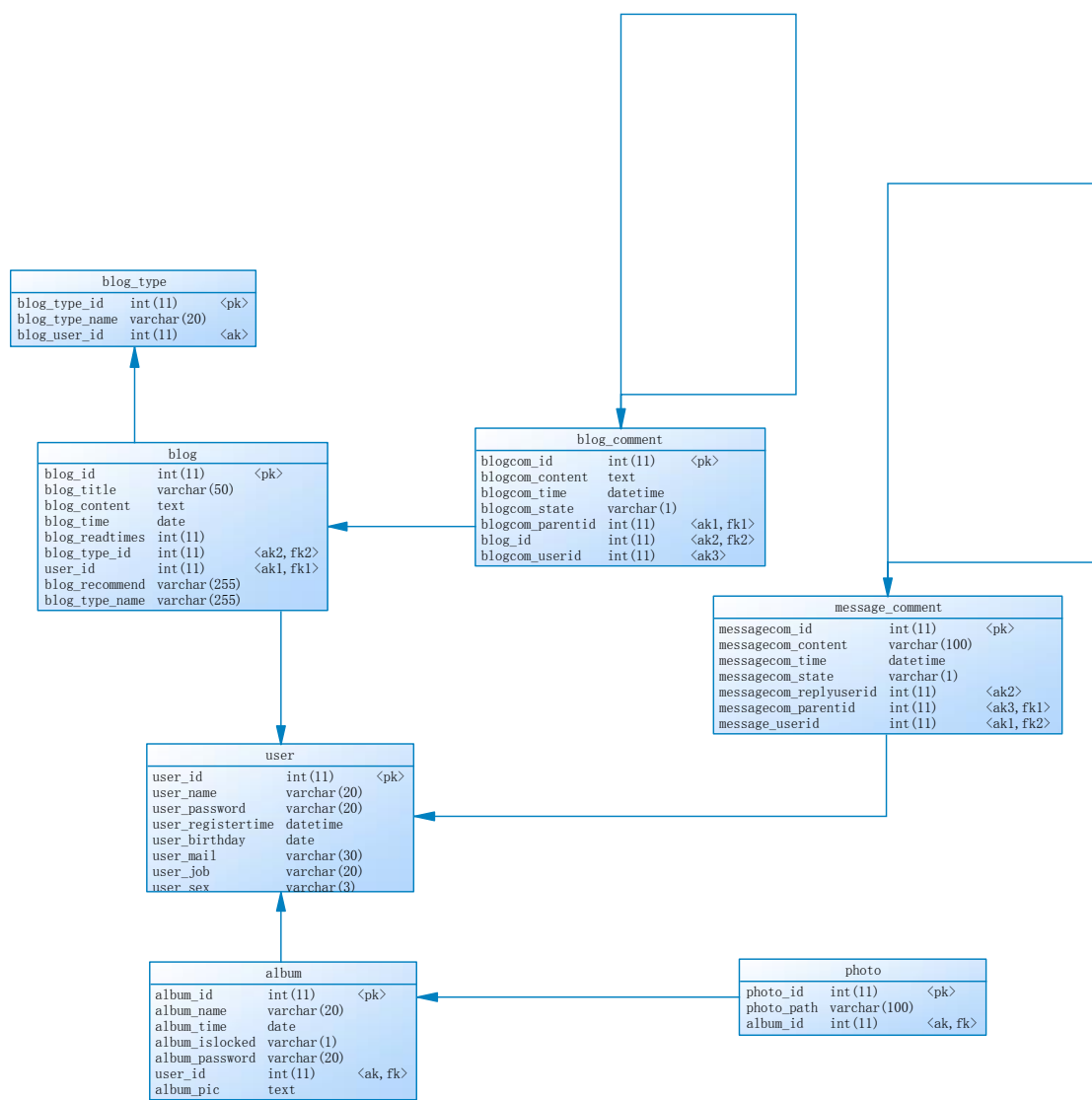
2.2 概念结构设计



2.3 逻辑结构设计



2.4 物理结构设计



3.系统设计

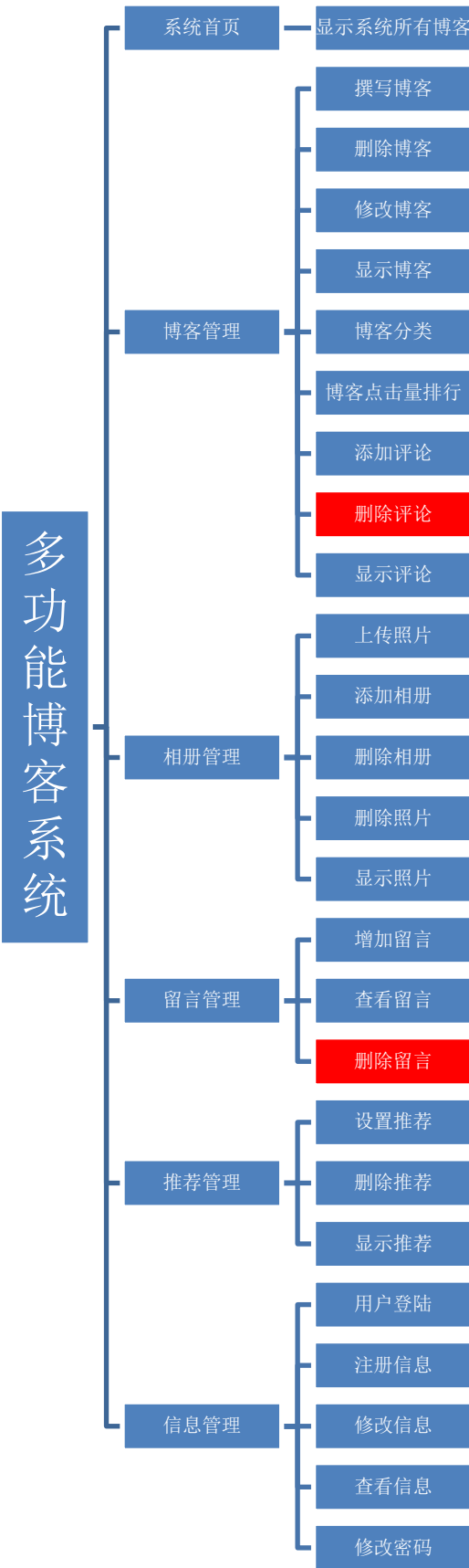
3.1 系统概要设计

3.1.1 系统构架设计

系统主要采用 MVC 的架构模式。系统分成五层：控制层(Controller)、业务逻辑层(Service)、 数据访问层(DAO)、数据模型层 (Model)、展现层(View)。数据访问层主要对数

数据库数据进行操作,包括增加数据,修改数据和删除数据。业务逻辑层基于数据访问层之上,它主要针对业务流程来实现系统的功能。控制层又基于业务逻辑层之上,它主要控制请求流转,并处理业务层通过数据访问层得到的数据在 **View** 表现层进行展现,以完成博客系统所要完成的功能。

3.1.2 系统结构设计



注：其中红色表示功能未实现，其余功能均已完美实现。

3.1.3 系统开发环境

数据库设计工具：Powerdesigner
开发工具：MyEclipse 8.5 + Mysql 5.7 + Tomcat 7 + JDK 1.7+Spring 框架

3.1.4 系统运行环境

客户端软件环境：
操作系统：Windows 7 或更高版本
服务端软件环境：
操作系统：Windows 7 或更高版本
数据库产品：Mysql 5.7
Java 开发工具：JDK1.7 或以上
服务器：Tomcat 7

3.1.5 系统接口设计及实现

数据访问层(DAO)
接口包名：cn.jxufe.dao
实现包名：cn.jxufe.dao

编号	模块接口	模块实现
1	AlbumDao	AlbumDaoImpl
2	BlogDao	BlogDaoImpl
3	Blog_CommentDao	Blog_CommentDaoImpl
4	Message_CommentDao	Message_CommentDaoImpl
5	Photo_CommentDao	Photo_CommentDaoImpl
6	PhotoDao	PhotoDaoImpl
7	UserDao	UserDaoImpl

业务逻辑层(Service)
接口包名：cn.jxufe.service
实现包名：cn.jxufe.service

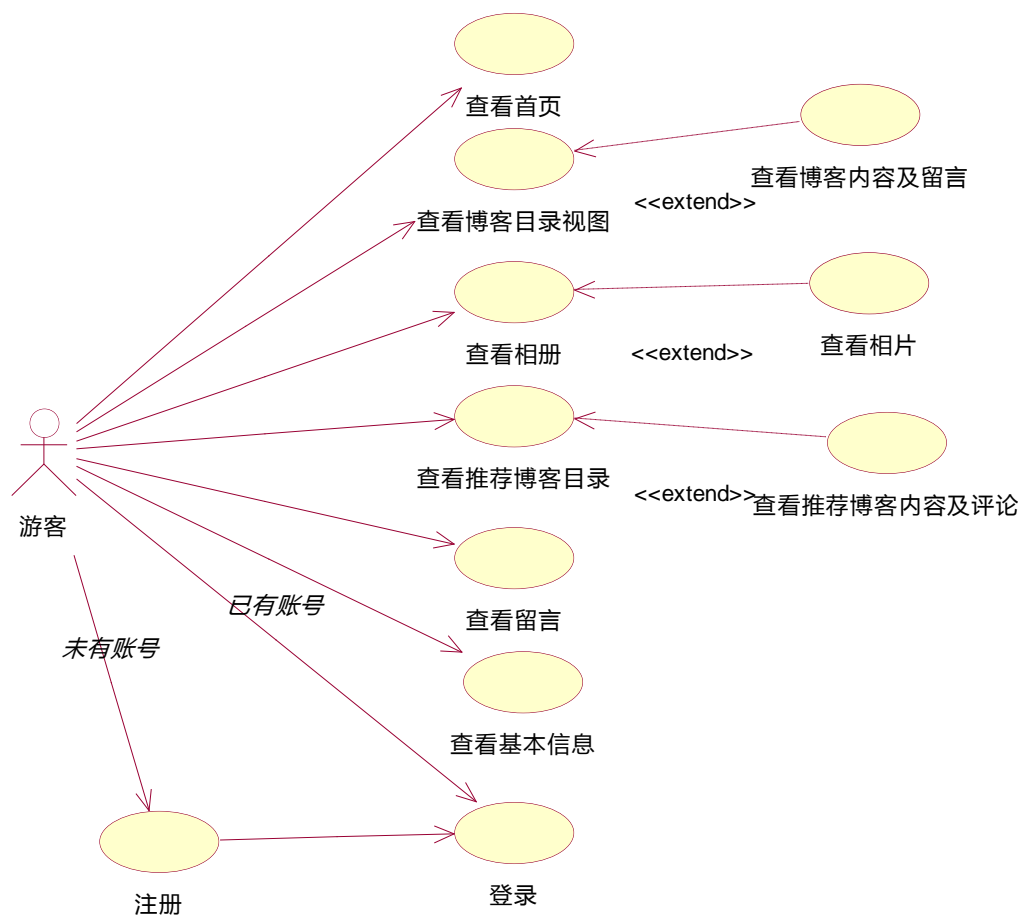
编号	模块接口	模块实现
1	AlbumService	AlbumServiceImpl
2	Blog_CommentService	Blog_CommentServiceImpl
3	BlogService	BlogServiceImpl
4	MessageService	MessageServiceImpl
5	UserService	UserServiceImpl

常用工具类的实现
日期转换，楼中楼回复的实现，有效值验证

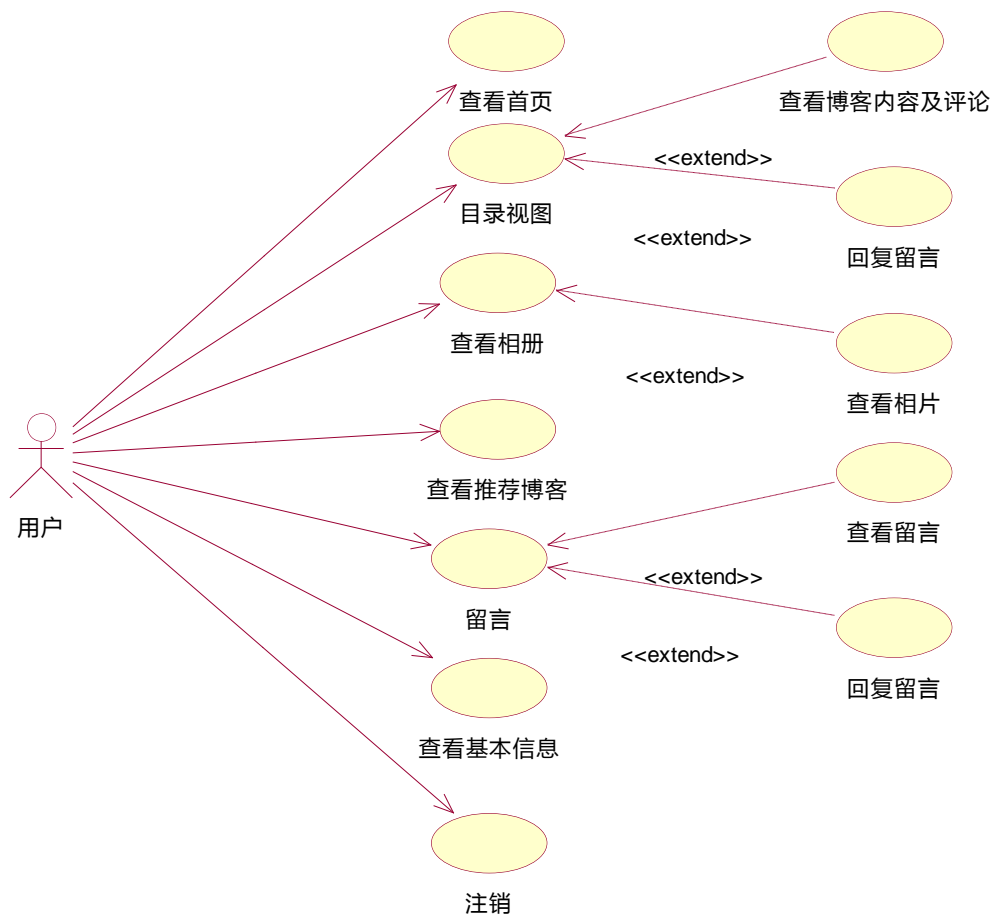
3.2 系统功能结构图

系统各模块用例图如下

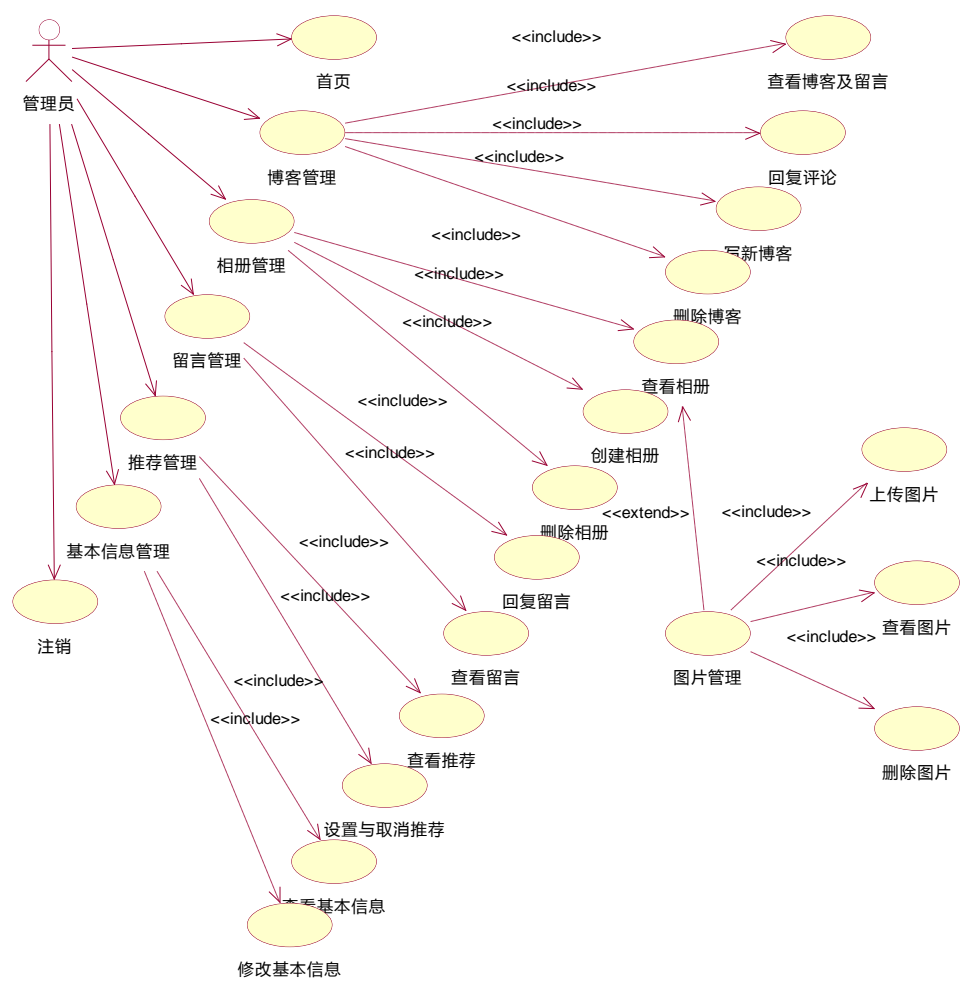
3.2.1 游客用例图



3.2.2 用户用例图



3.2.3 管理员用例图



4.系统界面截图

4.1 系统首页

我们非注册人员大家都可以进，可以看到不同博客的分类，以及各个博客的评论。

个人多功能博客系统

登录

首页

我的博客

全部分类

移动开发

Web前端

架构设计

编程语言

互联网

数据库

系统运维

云计算

研发管理

综合管理

推荐专家



牛华鹏



钱春





无图

系统全部博客

说说js闭包

js闭包涉及到作用域，js的作用域有两种，全局变量和局部变量，全局变量就是在函数外声明的，局部变量是在函数内声明的，函数内部可以直接读取全局变量，但是函数外部无法读取函数内的局部变量（函数内部声明变量的时候，要用var不然就是个全局变量）有时候需要得到函数内的局部变量
滥用闭包会导致内存泄露（由于闭包会使得函数中的变量都被保存在内存中，内存消耗
2016-07-07 00:00:00 阅读(3)

《Linux系统编程》笔记 第四章(一)

前言
第二章介绍了称为文件I/O的一系列系统调用，这些系统调用是文件操作的基础。本章介绍Linux提供的更高级的文件I/O方式，包括新的I/O方式、更高级的I/O多路复用方式以及内存映射等等。通过本章的学习，可以更加灵活高效的使用系统调用进行文件操作。
4.1 分散-聚合I/O
分散-聚合I/O(scatter-gather I/O)
2016-07-07 00:00:00 阅读(2)

《Linux系统编程》笔记 第四章(二)

4.2 Event Poll接口
在学习第二章-I/O多路复用技术时提到了select()和poll()两个系统调用，这两个系统调用都能阻塞在多个文件描述符上，让出CPU资源，阻塞等待对应的文件就绪、就绪或者出现异常。但是上述两个系统调用都有一些潜在的问题(具体参看对应章节)，内核2.6以后引入了epoll()机制，解决了之前提到的问题，并引入了边沿触发机制。在I/O多路
2016-07-07 00:00:00 阅读(0)

4.2 博客管理

主要包括了博客的增删改查，以及阅读排行，博客分类等。

个人资料

姓名: zcd

职业: 学生

邮箱: 635727914@qq.com

性别: 男

访问: 156

积分: 155

阅读排行

说说js闭包 (3)

《Linux系统编程》笔记 第四章(一) (2)

MySQL之分页优化 (2)

《Linux系统编程》笔记 第四章(二) (0)

分类

web前端精华

Linux编程

MySQL

Home / 博主博客

管理博客写新博客

博主全部文章

说说js闭包

js闭包涉及到作用域，js的作用域有两种，全局变量和局部变量，全局变量就是在函数外声明的，局部变量是在函数内声明的，函数内部可以直接读取全局变量，但是函数外部自然无法读取函数内的局部变量（函数内部声明变量的时候，要用var不然就是个全局变量）有时候需要得到函数内的局部变量

滥用闭包会可能导致内存泄露（由于闭包会使得函数中的变量都被保存在内存中，内存消耗

2016-07-07 00:00:00.0 阅读 (3)

《Linux系统编程》笔记 第四章(一)

前言

第二章介绍了称为文件I/O的一系列系统调用，这些系统调用是文件操作的基础。本章介绍Linux提供的更高级的文件I/O方式，包括新的I/O方式、更高级的I/O多路复用方式以及内存映射等等。通过本章的学习，可以更加灵活高效的使用系统调用进行文件操作。

4.1 分散-聚合I/O

分散-聚合I/O(scatter-gather I/O)

2016-07-07 00:00:00.0 阅读 (2)

《Linux系统编程》笔记 第四章(二)

博客博主界面：可以管理博主所有文章，查看留言

个人资料
姓名: zcd
职业: 学生
邮箱: 635727914@qq.com
性别: 男
访问: 156
积分: 155
阅读排行
说说js闭包 (3)
《Linux系统编程》笔记 第四章(一) (2)
MySQL之分页优化 (2)
《Linux系统编程》笔记 第四章(二) (1)
分类
web前端精华
Linux编程
MySQL

Home / 博主博客 / 博客详情

目录视图 写新博客 编辑博文

《Linux系统编程》笔记 第四章(二)

时间: 2016-07-07 00:00:00 点击量: 0

分类: Linux编程

4.2 Event Poll接口

在学习第二章-IO多路复用技术时提到了select()和poll()两个系统调用，这两个系统调用都能阻塞在多个文件描述符上，让出CPU资源，阻塞等待对应的文件就绪、就绪或者出现异常。但是上述两个系统调用都有一些潜在的问题(具体参看对应章节)，内核2.6以后引入了epoll()机制，解决了之前提到的问题，并引入了边沿触发机制。在IO多路复用的模型上，epoll()是目前最优的方案。

4.2.1 创建新的epoll实例

创建epoll实例是使用epoll模型的第一步。

#include <sys/epoll.h>int epoll_create(int size);12

该函数打开并返回一个新的epoll文件描述符，后续的epoll操作都要用这个文件描述符来进行。参数size原本是用来指定返回的epoll文件描述符初始能监听多少个文件描述符，但是Linux内核会动态增加可监听的文件描述符数量，因此该参数在较新的内核版本上会被忽略。在2.6.27的内核版本中引入了新的系统调用 int epoll_create1(int flags); 其作用与epoll_create()一样，参数flags表明新建的epoll文件描述符的属性，目前只支持 EPOLL_CLOEXEC (与之前介绍open(时的FD_CLOEXEC标志作用一致，在exec()系列函数被调用时该文件描述符被自动关闭)。

该调用在成功时返回epoll文件描述符的值，失败时返回-1并设置errno。

需要注意的是epoll文件描述符也占用系统资源，因此不再使用时需要调用close()来关闭。与文件描述符一样，当一个epoll资源对应的所有文件描述符被关闭时，内核会把对应的资源返回给系统。

4.2.2 控制epoll

在创建epoll实例后，我们可以使用返回的文件描述符来控制对哪些文件描述符以及哪些事件做监听。

发表评论

1楼 用户:1 在2016-06-20 05:13:10.0发表

回复 删除

评论: dasd

2楼 用户:1 在2016-06-20 05:14:21.0发表

回复 删除

评论: sasas

1 2016-06-20 07:21:00.0发表

回复 删除

回复1:啊啊

1 2016-06-20 07:21:08.0发表

回复 删除

回复1:三大殿

1 2016-06-20 07:21:13.0发表

回复 删除

回复1:等等

评论内容:

个人多功能博客系统

zcd的博客退出登录

博主博客博主相册博主留言博主推荐博主信息

Home / 博主博客 / 管理博客

目录视图写新博客

博客管理

标题	阅读	时间	分类	操作
说说js闭包	3	2016-07-07 00:00:00.0	web前端精华	编辑 删除 设为推荐
《Linux系统编程》笔记 第四章(一)	2	2016-07-07 00:00:00.0	Linux编程	编辑 删除 设为推荐
《Linux系统编程》笔记 第四章(二)	1	2016-07-07 00:00:00.0	Linux编程	编辑 删除 设为推荐
MySQL之分页优化	2	2016-07-07 00:00:00.0	Mysql	编辑 删除 设为推荐

共 1 页 第 1 页 首页 上一页 下一页 尾页

添加博客分类

提交

博客分类

web前端精华

Linux编程

Mysql

项目简介 / 广告服务 / 银行汇款帐号 / 联系方式 / 版权声明 / 法律顾问 / 问题报告 / 合作伙伴

☎400-004-444 / 江西财经大学软件软件工程 zcd / 版权所有 / 联系邮箱:635727914@qq.com

可以进行上传图片，删除图片，以及不同相册分类

个人多功能博客系统

zcd的博客 退出登录

博主博客

博主相册

博主留言

博主推荐

博主信息

个人资料

姓名: zcd

职业: 学生

邮箱: 635727914@qq.com

性别: 男

访问: 156

积分: 155

阅读排行

说说js闭包 (3)

《Linux系统编程》笔记 第四章(一) (2)

MySQL之分页优化 (2)

《Linux系统编程》笔记 第四章(二) (1)

分类

web前端精华

Linux编程

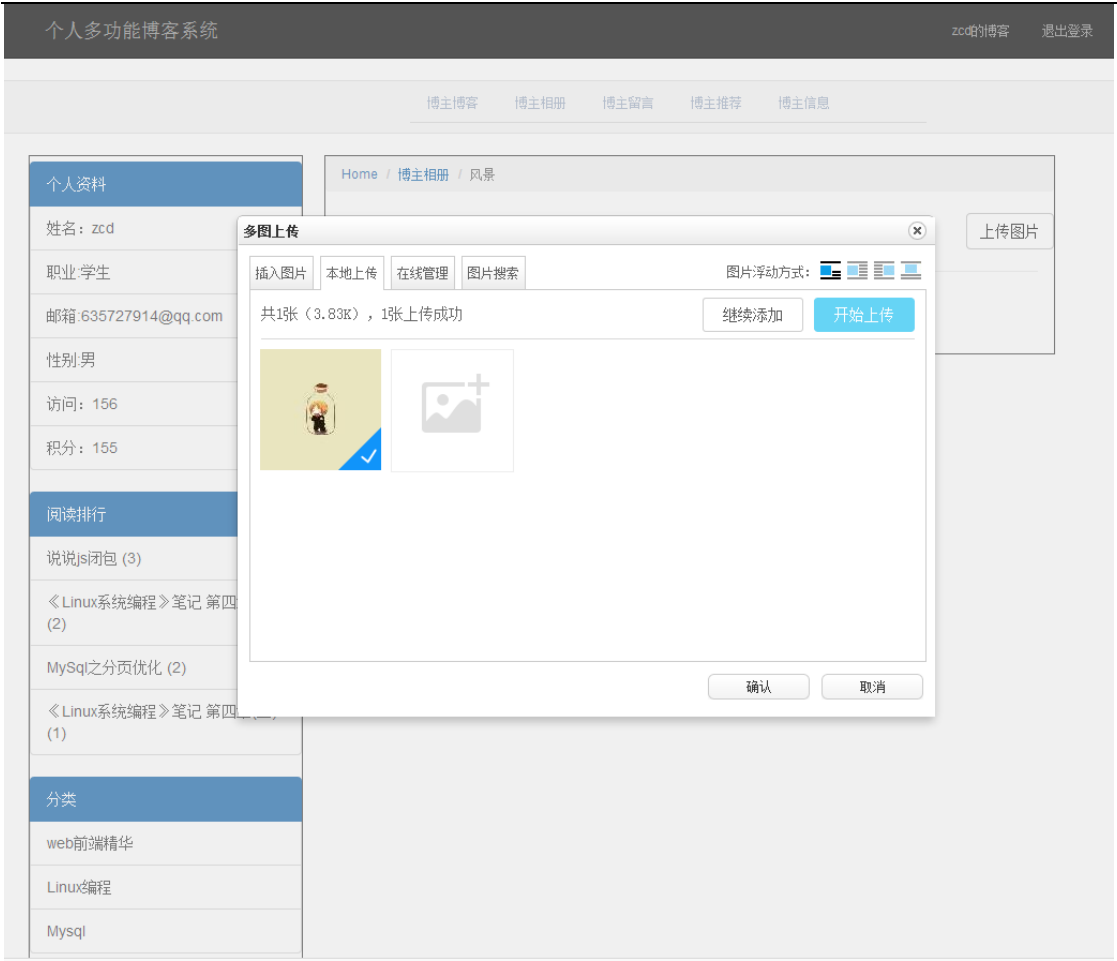
Mysql

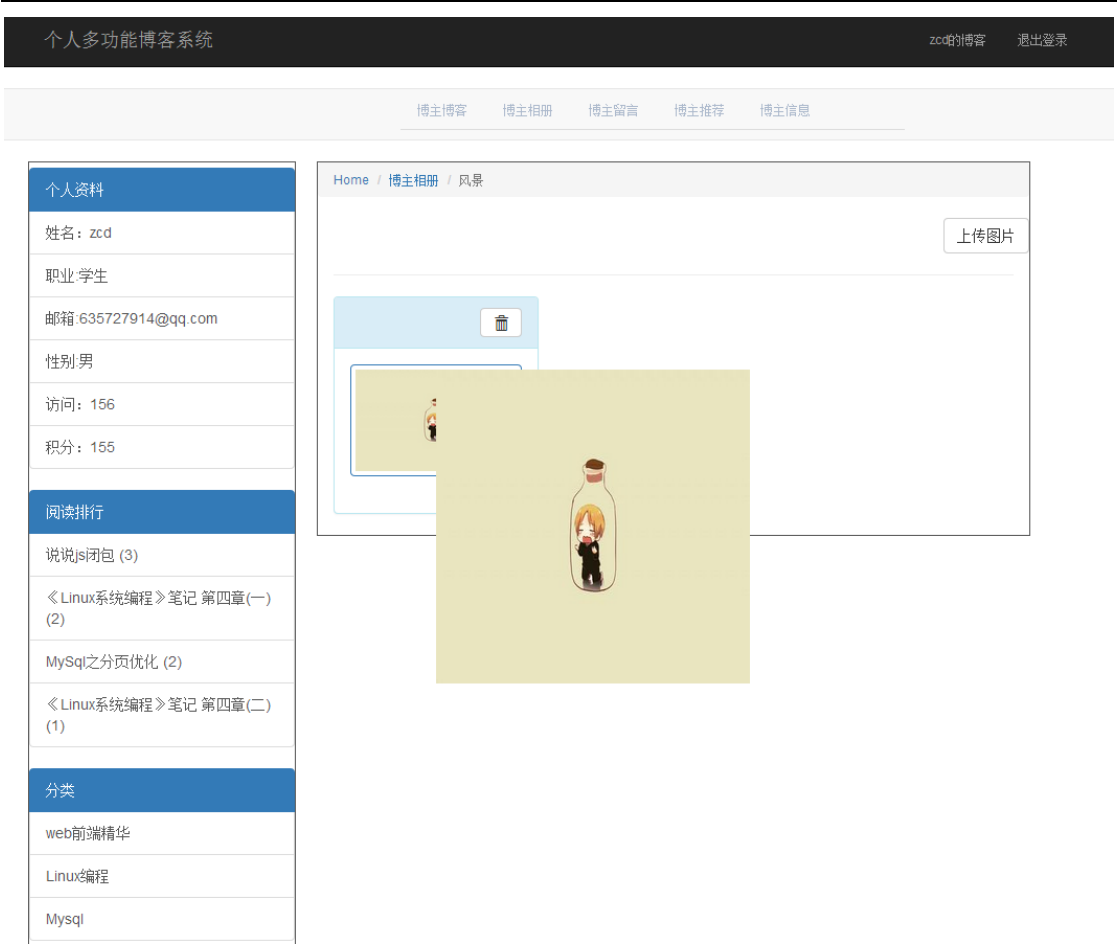
Home / 博主相册

添加相册

风景

萌宠





4.4 留言管理

可以进行留言和回复留言

个人多功能博客系统

zcd的博客退出登录

博主博客

博主相册

博主留言

博主推荐

博主信息

个人资料

姓名: zcd

职业: 学生

邮箱: 635727914@qq.com

性别: 男

访问: 156

积分: 155

阅读排行

说说js闭包 (3)

《Linux系统编程》笔记 第四章(一) (2)

MySQL之分页优化 (2)

《Linux系统编程》笔记 第四章(二) (1)

分类

web前端精华

Linux编程

Mysql

Home / 博主留言

发表留言

1楼 用户:10 在2016-07-09 11:54:54.0发表

回复

评论: 啊啊啊

2楼 用户:10 在2016-07-09 11:55:03.0发表

回复

评论: 留言啦

10 2016-07-09 11:55:15.0发表

回复

回复 好运

10 2016-07-09 11:55:25.0发表

回复

回复 zb

评论内容:

4.5 推荐管理

博主可以推荐他认为比较好的文章

个人多功能博客系统

zcd的博客 退出登录

博主博客 博主相册 博主留言 博主推荐 博主信息

个人资料

姓名: zcd

职业: 学生

邮箱: 635727914@qq.com

性别: 男

访问: 156

积分: 155

阅读排行

说说js闭包 (3)

《Linux系统编程》笔记 第四章(一) (2)

MySql之分页优化 (2)

《Linux系统编程》笔记 第四章(二) (1)

分类

Home / 博主推荐

管理博客 写新博客

博主推荐文章

《Linux系统编程》笔记 第四章(一)

前言

第二章介绍了称为文件I/O的一系列系统调用，这些系统调用是文件操作的基础。本章介绍Linux提供的更高级的文件I/O方式，包括新的I/O方式、更高级的I/O多路复用方式以及内存映射等等。通过本章的学习，可以更加灵活高效的使用系统调用进行文件操作。

4.1 分散-聚合I/O

分散-聚合I/O(scatter-gather I/O)还有很多别的叫法，例如分散输入、集中输出，向量I/O、散布聚合I/O等等，这里指的是同一个I/O方式，都是从一个文件描述符中读取数据并拷贝到不同的用户空间缓冲区或从不同的用户空间缓冲区将数据写入到同

2016-07-07 00:00:00.0

《Linux系统编程》笔记 第四章(二)

4.2 Event Poll接口

在学习第二章-I/O多路复用技术时提到了select()和poll()两个系统调用，这两个系统调用都能阻塞在多个文件描述符上，让出CPU资源，阻塞等待对应的文件读就绪、写就绪或者出现异常。但是上述两个系统调用都有一些潜在的问题(具体参看对应章节)，内核2.6以后引入了epoll()机制，解决了之前提到的问题，并引入了边沿触发机制。在I/O多路复用的模型上，epoll()是目前最优的方案。

4.2.1 创建新的epoll实例

创建epoll实例是使用epoll模型的第一步。

4.6 信息管理

注册界面：

软通 个人博客系统注册

Step 1 / 2

user_name

☒男 ☐女

you job...

Next

软通 个人博客系统注册

Step 2 / 2

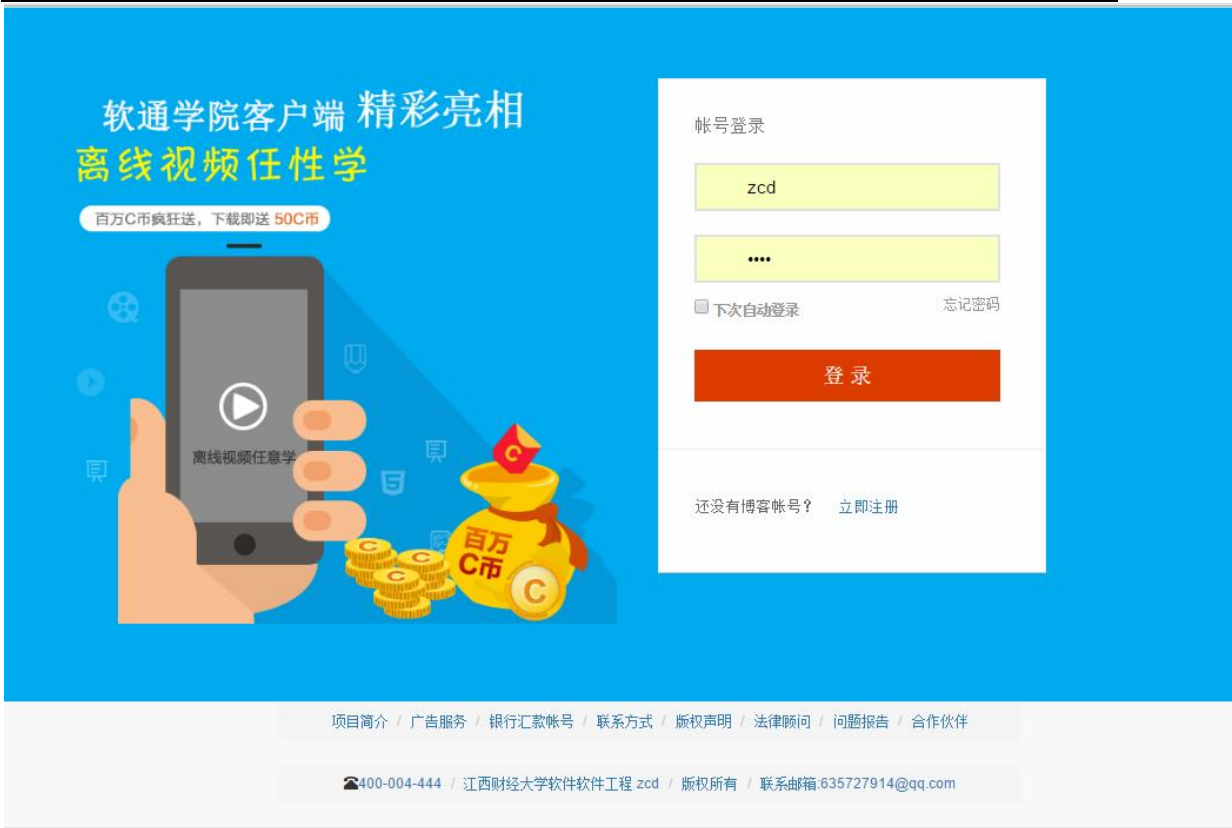
Email...

Password...

Repeat password...

Previous Sign me up!

登录界面:



博主可以进行修改信息，和修改密码

个人资料
姓名: zcd
职业: 学生
邮箱: 635727914@qq.com
性别: 男
访问: 156
积分: 155
阅读排行
说说js闭包 (3)
《Linux系统编程》笔记 第四章(一) (2)
MySql之分页优化 (2)
《Linux系统编程》笔记 第四章(二) (1)
分类
web前端精华
Linux编程
Mysql

Home / 博主信息	
修改密码	
个人基本信息	
姓名	zcd
职业	学生
邮箱	635727914@qq.com
性别	男
出生日期	1996-07-12

个人多功能博客系统

zcd的博客退出登录

博主博客

博主注册

博主留言

博主推荐

博主信息

个人资料

姓名: zcd

职业: 学生

邮箱: 635727914@qq.com

性别: 男

访问: 156

积分: 155

阅读排行

说说js闭包 (3)

《Linux系统编程》笔记 第四章(一) (2)

MySQL之分页优化 (2)

《Linux系统编程》笔记 第四章(二) (1)

分类

web前端精华

Linux编程

Mysql

Home / 博主信息

修改密码

旧密码

请输入旧密码

新密码

请输入新密码

确认新密码

请确认新密码

修改

5.附数据库脚本

```
/*
Navicat MySQL Data Transfer

Source Server          : localhost_3306
Source Server Version  : 50711
Source Host            : localhost:3306
Source Database        : blog

Target Server Type     : MYSQL
Target Server Version  : 50711
File Encoding          : 65001

Date: 2016-07-09 12:09:57
*/
```

```
SET FOREIGN_KEY_CHECKS=0;
```

```
-- -----  
-- Table structure for album  
-- -----
```

```
DROP TABLE IF EXISTS `album`;  
CREATE TABLE `album` (  
  `album_id` int(11) NOT NULL AUTO_INCREMENT,  
  `album_name` varchar(20) DEFAULT NULL,  
  `album_time` date DEFAULT NULL,  
  `album_islocked` varchar(1) DEFAULT NULL,  
  `album_password` varchar(20) DEFAULT NULL,  
  `user_id` int(11) DEFAULT NULL,  
  `album_pic` text,  
  PRIMARY KEY (`album_id`),  
  KEY `album_ibfk_1` (`user_id`),  
  CONSTRAINT `album_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`) ON  
DELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=28 DEFAULT CHARSET=utf8;
```

```
-- -----  
-- Table structure for blog  
-- -----
```

```
DROP TABLE IF EXISTS `blog`;  
CREATE TABLE `blog` (  
  `blog_id` int(11) NOT NULL AUTO_INCREMENT,  
  `blog_title` varchar(50) DEFAULT NULL,  
  `blog_content` text,  
  `blog_time` date DEFAULT NULL,  
  `blog_readtimes` int(11) unsigned zerofill DEFAULT '000000000000',  
  `blog_type_id` int(11) DEFAULT NULL,  
  `user_id` int(11) DEFAULT NULL,  
  `blog_recommend` varchar(255) DEFAULT '否' COMMENT '是否为推荐博客',  
  `blog_type_name` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`blog_id`),  
  KEY `FK_Reference_14` (`user_id`),  
  KEY `FK_Reference_15` (`blog_type_id`),  
  CONSTRAINT `blog_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`) ON DELETE  
CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB AUTO_INCREMENT=59 DEFAULT CHARSET=utf8;
```

```
-- -----  
-- Table structure for blog_comment
```

```

-----
DROP TABLE IF EXISTS `blog_comment`;
CREATE TABLE `blog_comment` (
  `blogcom_id` int(11) NOT NULL AUTO_INCREMENT,
  `blogcom_content` text,
  `blogcom_time` datetime DEFAULT NULL,
  `blogcom_state` varchar(1) DEFAULT NULL,
  `blogcom_parentid` int(11) DEFAULT NULL,
  `blog_id` int(11) DEFAULT NULL,
  `blogcom_userid` int(11) DEFAULT NULL,
  PRIMARY KEY (`blogcom_id`),
  KEY `FK_Reference_16` (`blogcom_parentid`),
  KEY `FK_Reference_6` (`blog_id`),
  KEY `blogcom_userid` (`blogcom_userid`),
  CONSTRAINT `blog_comment_ibfk_1` FOREIGN KEY (`blogcom_userid`) REFERENCES `user`
(`user_id`),
  CONSTRAINT `blog_comment_ibfk_2` FOREIGN KEY (`blogcom_parentid`) REFERENCES
`blog_comment` (`blogcom_id`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=58 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for blog_type
-----

```

```

DROP TABLE IF EXISTS `blog_type`;
CREATE TABLE `blog_type` (
  `blog_type_id` int(11) NOT NULL AUTO_INCREMENT,
  `blog_type_name` varchar(20) DEFAULT NULL,
  `blog_user_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`blog_type_id`),
  KEY `blog_user_id` (`blog_user_id`),
  CONSTRAINT `blog_type_ibfk_1` FOREIGN KEY (`blog_user_id`) REFERENCES `user` (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;

```

```

-----
-- Table structure for message_comment
-----

```

```

DROP TABLE IF EXISTS `message_comment`;
CREATE TABLE `message_comment` (
  `messagecom_id` int(11) NOT NULL AUTO_INCREMENT,
  `messagecom_content` varchar(100) DEFAULT NULL,
  `messagecom_time` datetime DEFAULT NULL,
  `messagecom_state` varchar(1) DEFAULT '否',
  `messagecom_replyuserid` int(11) DEFAULT NULL,
  `messagecom_parentid` int(11) DEFAULT NULL,

```



```
`message_userid` int(11) DEFAULT NULL,
PRIMARY KEY (`messagecom_id`),
KEY `message_userid` (`message_userid`),
KEY `message_comment_ibfk_2` (`messagecom_replyuserid`),
KEY `messagecom_parentid` (`messagecom_parentid`),
CONSTRAINT `message_comment_ibfk_1` FOREIGN KEY (`message_userid`) REFERENCES `user`
(`user_id`),
CONSTRAINT `message_comment_ibfk_2` FOREIGN KEY (`messagecom_replyuserid`)
REFERENCES `user` (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=25 DEFAULT CHARSET=utf8;
```

-- Table structure for photo

```
DROP TABLE IF EXISTS `photo`;
CREATE TABLE `photo` (
  `photo_id` int(11) NOT NULL AUTO_INCREMENT,
  `photo_path` varchar(100) DEFAULT NULL,
  `album_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`photo_id`),
  KEY `FK_Reference_10` (`album_id`),
  CONSTRAINT `photo_ibfk_1` FOREIGN KEY (`album_id`) REFERENCES `album` (`album_id`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;
```

-- Table structure for user

```
DROP TABLE IF EXISTS `user`;
CREATE TABLE `user` (
  `user_id` int(11) NOT NULL AUTO_INCREMENT,
  `user_name` varchar(20) DEFAULT NULL,
  `user_password` varchar(20) DEFAULT NULL,
  `user_registertime` datetime DEFAULT NULL,
  `user_birthday` date DEFAULT NULL,
  `user_mail` varchar(30) DEFAULT NULL,
  `user_job` varchar(20) DEFAULT NULL,
  `user_sex` varchar(3) DEFAULT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8;
```

存储过程:

```
CREATE DEFINER = CURRENT_USER PROCEDURE `NewProc`(IN `in_blog_type_name` varchar,OUT
`out_blog_type_id` int)
```

```
BEGIN
    #Routine body goes here...
    select blog_type_id INTO out_blog_type_id
    from blog_type
    where blog_type_name= in_blog_type_name
END;;

CREATE DEFINER = CURRENT_USER PROCEDURE `NewProc`(IN `in_blog_id` int,OUT
`out_read_times` int)
BEGIN
    #Routine body goes here...
    select blog_readtimes INTO out_read_times
    from blog
    where blog_id=in_blog_id
END;;
```

触发器:

将 blog 表的 blog_type_id 自动填充

delimiter \$

create trigger tg1

after insert on blog

for each row

begin

declare id int;

select blog_type.blog_type_id into id

from blog_type

where blog_type.blog_type_name=new.blog_type_name;

set new.blog_type_id=id;

end \$

delimiter ;

将相册的初始密码设为 aaaa

delimiter \$

create trigger tg2

BEFORE insert on album

for each row

begin

set new.album_password='aaaa';

end \$

delimiter ;