

个人博客需求分析

含有 Oracle 数据库主要存储过程语句

一 . 需求分析

该系统为个人博客，具有一般博客系统的功能。

博主，即管理员，拥有最高权限，能对文章，分类，评论，用户等进行管理。

游客只能浏览前台页面， 及浏览文章。 只有游客注册成为注册用户后， 才能对文章进行评论。

根据以上需求分析，该系统要实现的主要功能为：

前台显示文章部分

- 显示最新的前几篇文章

- 显示所有文章分类

- 按分类显示其下的所有文章

- 文章详细信息浏览阅读

- 游客注册以及注册用户的登录

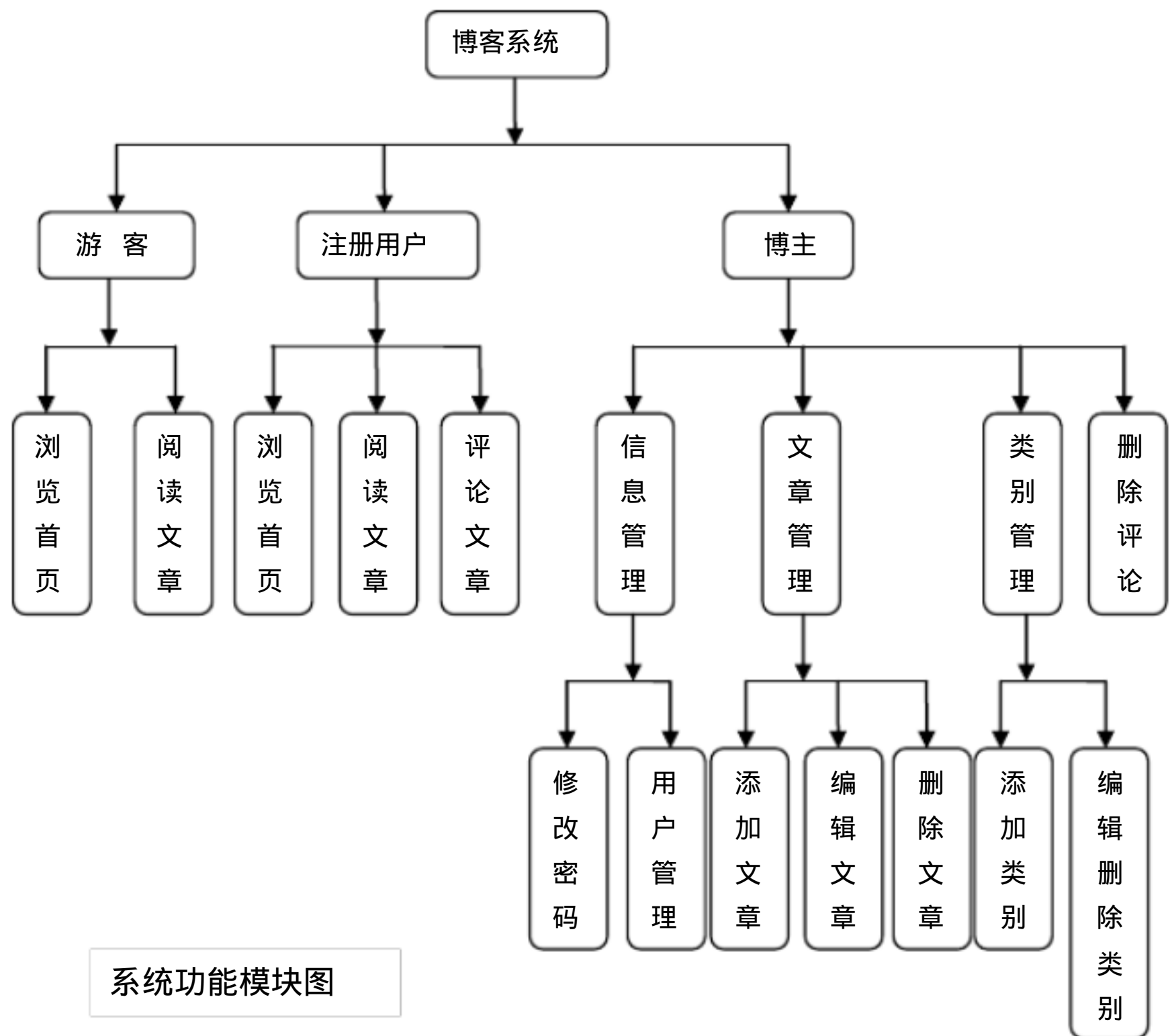
- 评论文章功能

后台管理部分

- 登录验证，只有博主能登录

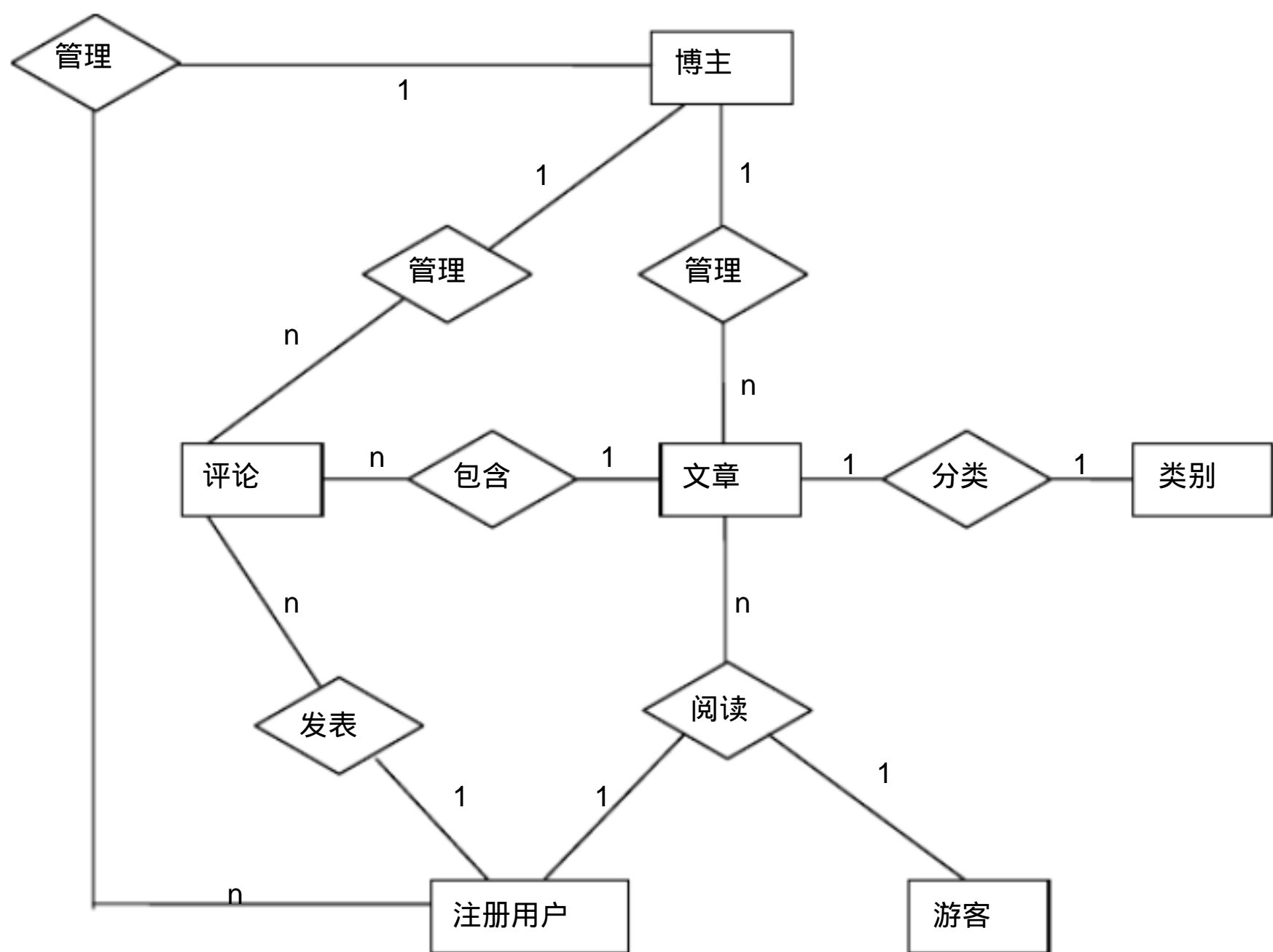
- 添加、编辑、删除文章功能

- 用户、类别、评论的管理功能



二、数据库设计

2.1、E-R 图



总 E-R 图

2.2、表结构设计

用户信息表 **userinfo**

列名	数据类型
userid	Int
username	Varchar2(50)
userpassword	Varchar2(50)
usertype	Int

文章信息表 **Articles**

列名	数据类型
A_id	Number
A_title	Varchar2(50)

A_content	Varchar2(4000)
A_publishedtime	Date
T_name	Varchar2(50)
a_commentnum评论数量	number
A_viewnum 阅读数量	number

文章分类表 **Tags**

列名	数据类型
T_id	number
T_name	Varchar2(50)

评论信息表 **Comments**

列名	数据类型
C_id	number
A_id	number
C_username	Varchat2(50)
C_comment	Varchar2(50)
C_leavetime	date

三．博客系统的性能需求

该系统在性能功能上应达到如下需求：

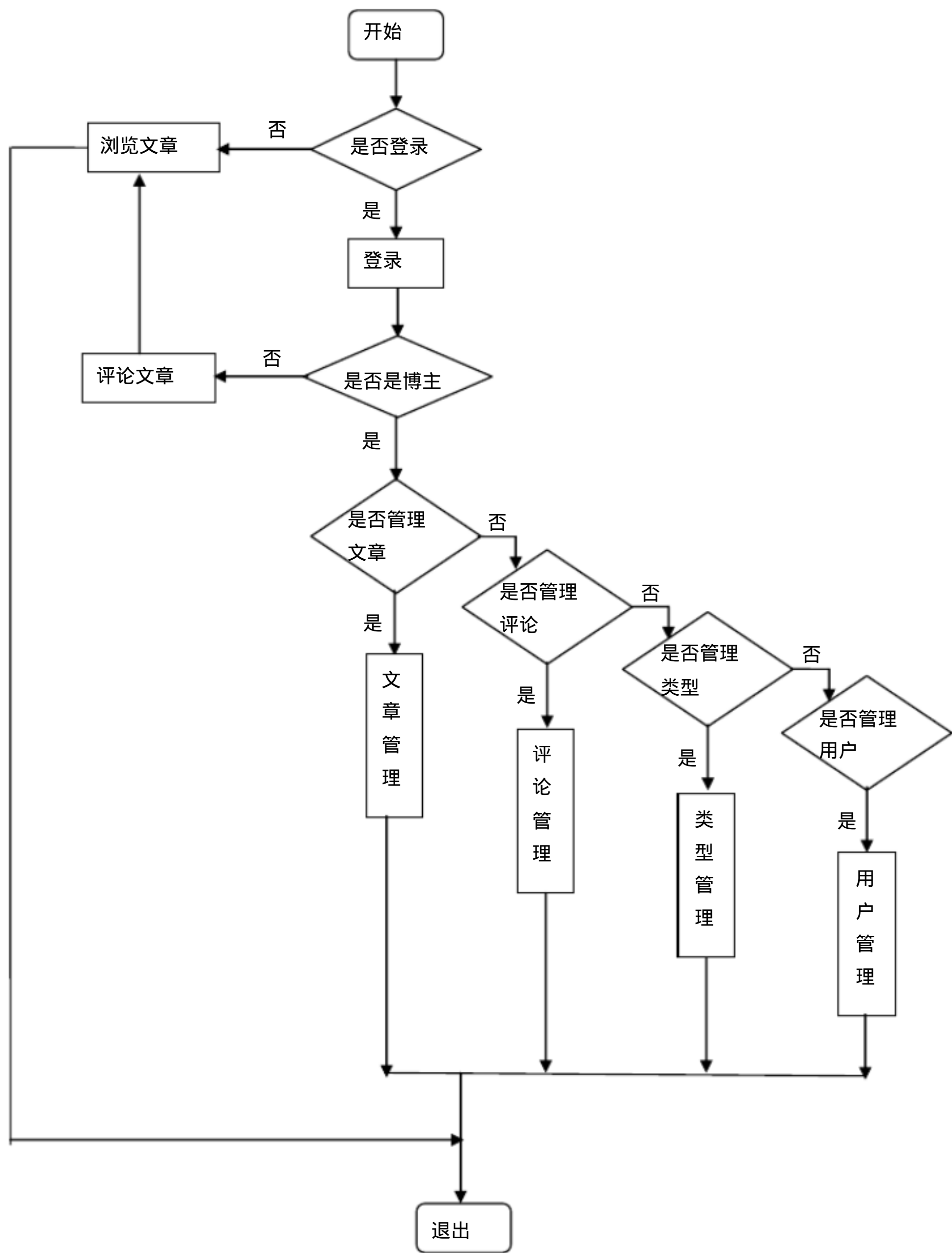
操作简单、界面友好：完全控件式的页面布局，使得文章的录入工作更简便，许多选项只需要点击鼠标就可以完成；另外，跟踪出现的提示信息也让用户随时清楚自己的操作情况。对常见网站的博客管理的各个方面：上传博客、浏览、删除、修改等方面都大体实现，实现了网站对即时文章的管理要求；

即时可见：对文章的处理（包括添加、编辑、删除）将立即在主页的对应栏目显示出来，达到“即时发布、即时见效”的功能；

系统运行应该快速、稳定、高效和可靠；

在结构上应具有很好的可扩展性，便于将来的功能扩展和维护。

四． 系统结构实现过程



系统流程图

整个系统的工作流程为：

打开该系统网站，游客能浏览、阅读文章，能进行注册，注册后成为注册用户，注册用户登录后能够对文章进行评论。 博主从后台入口登录， 可以添加文章，包括文章标题、类别、内容；还能对用户、文章、类别、评论进行管理。

数据库中主要 **Oracle** 操作语句

--ADDARTICLE

create or replace procedure addarticle

(

title in articles.a_title%type,

content in articles.a_content%type,

tagname in articles.t_name%type,

commentnum in articles.a_commentnum%type default 0,

viewnum in articles.a_viewnum%type default 0

)

as

begin

insert into articles (a_title, a_content, t_name,
a_publishedtime, a_commentnum, a_viewnum)

values(title, content, tagname, sysdate, commentnum,
viewnum);

end addarticle;

```
--ADDCOMMENT

create or replace procedure addcomment
(
aid in comments.a_id%type,
c_content in comments.c_comment%type,
cname in comments.c_username%type
)
as begin

insert into comments (a_id, c_comment, c_username,
c_leavetime)

values (aid, c_content, cname, sysdate);

end addcomment;
```

--ADDTAG

create or replace procedure addtag

```
(
tagname in tags.t_name%type
)as
```

begin

insert into tags(t_name) values (tagname);

end addtag;

--ADDUSERS

create or replace procedure addusers

(

v_name in varchar2,

v_password in varchar2,

v_type in userinfo.usertype%type default 0

)

as

begin

insert into userinfo(username,userpassword,usertype)

values (v_name,v_password,v_type);

end addusers;

--CHECKADMIN

create or replace procedure checkadmin

(

v_username in varchar2,

v_password in varchar2,

result out number

)

is

```
l_count number;

begin

select count(*) into l_count from userinfo

where USERNAME = v_username and USERPASSWORD

= v_password and usertype = 1;

if l_count=0 then

result:=-1;

else

result:=1;

end if;

end checkadmin;
```

--CHECKUSERS

create or replace procedure checkusers

(v_username in varchar2,

v_password in varchar2,

result out number)

is

l_count number;

begin

select count(*) into l_count from userinfo

where USERNAME = v_username and USERPASSWORD

```
= v_password and usertype=0;
```

```
if l_count=0 then
```

```
result:=-1;
```

```
else
```

```
result:=1;
```

```
end if;
```

```
end checkusers;
```

```
--DELETEARTICLE
```

```
create or replace procedure deletearticle
```

```
(
```

```
id_in in number
```

```
)as
```

```
begin
```

```
delete from articles where a_id=id_in;
```

```
end deletearticle;
```

```
--DELETEDCOMMENT
```

```
create or replace procedure deletecomment
```

```
(
```

```
id_in in number
```

```
)as
```

```
begin  
  
delete from comments where c_id=id_in;  
  
end deletecomment;
```

--DELETETAG

```
create or replace procedure deletetag  
  
(  
  
tagname in tags.t_name%type  
  
)as  
  
begin  
  
delete from tags where t_name=tagname;  
  
end deletetag;
```

--DELETEUSER

```
create or replace procedure deleteuser(id_in in number) as  
  
begin  
  
delete from userinfo where userid=id_in;  
  
end deleteuser;
```

--selectarticle

```
create or replace procedure selectarticle  
  
(
```

```
cur out article_type.cur_type
```

```
)
```

```
is
```

```
begin
```

```
open cur for
```

```
select * from articles;
```

```
end;
```

```
--selectcomment
```

```
create or replace procedure selectcomment
```

```
(
```

```
aid in comments.a_id%type,
```

```
cur out comment_type.cur_type
```

```
)
```

```
is
```

```
begin
```

```
open cur for
```

```
select * from comments where a_id=aid;
```

```
end;
```

```
--selecttag
```

```

create or replace procedure selecttag
(
cur out tag_type.cur_type
)
is
begin
open cur for
select * from tags;
end;

--updatearticle
create or replace procedure updatearticle
(
id_in number,
title in articles.a_title%type,
content in articles.a_content%type,
tagname in articles.t_name%type
)as
begin
update articles set a_title = title, a_content = content ,
t_name = tagname where a_id = id_in;
end updatearticle;

```

```
--updatepwd

create or replace procedure updatepwd

(

u_password in userinfo.userpassword%type

)as

begin

    update  userinfo  set  userpassword=u_password  where

userid=1;

end updatepwd;
```

```
--updatetag

create or replace procedure updatetag

(

id_in in number,

tagname in varchar2

)as

begin

    update tags set t_name=tagname where t_id=id_in;

end updatetag;
```