
《软件开发实践》课程报告

课题名称： RelaxBlog 个人博客系统

课题负责人名（学号）：

董建文(2015141463029)

同组成员名单（角色）：

黄伟(2015141463060, 开发人员)

指导教师： _____

评阅成绩： _____

评阅意见： _____

提交报告时间：2017 年 6 月 10 日

RelaxBlog 个人博客系统

软件工程 专业

学生 董建文 黄伟 指导老师 洪玫

【摘要】 随着自媒体时代的到来，博客与微博在网络中的影响力日益突出。而微博的篇幅限制和传播方式，导致技术类文章不适合以此形式发布，而博客在这方面则有不可替代的优势。旧版博客外观千篇一律，界面冗杂，不符合简洁清爽的流行风格。因此我们用 Spring MVC 开发了 web 应用 RelaxBlog 以满足那些期望拥有个人博客的用户的需求。最终项目通过验收测试，外观清爽，验错功能完备，可以作为通用型个人博客推广使用。

关键词：个人博客系统 RelaxBlog SpringMVC web 开发

1. 引言（项目背景）

如今自媒体在互联网领域流行开来，娱乐、技术、新闻等各式各样的信息通过博客、微博、直播等平台被广泛传播。微博的流行度虽高，但篇幅限制和实时性的特征使其无法支撑复杂知识、技术的分析和讲解，难以作为专业性文章的发布平台。直播则有着用户主播之间交互时间的统一性要求，且主题单一，不适合自学或检索。

在这种前提下，博客自始至终有着不可替代的地位。而大量的个人博客仍使用着陈旧的模板，缺乏清晰度，也不符合新的审美要求，如简洁化、清爽、动态化等。因此本项目将开发一款满足新界面、新元素要求的博客系统，以提供一个简洁易用、供网友交流沟通的网络平台。

本项目采用 SpringMVC+Hibernate 作为基本架构，采用 Java+JavaScript+Html+sql 作为主要开发语言，使用 IDEA、Mysql 和 chrome 作为主要开发工具，通过实体类和数据库的交互实现了文章、友情链接等的增删查改，通过动态加载页面实现了页面内容的变化。

项目最终通过了单元测试、功能测试和系统验收测试，达到了预期目的，可作为个人用户搭建博客系统的框架使用。

2. 相关工作（对现有的系统进行分析）^[1]

目前国内外有着许多出名的博客系统，支撑着大量用户的访问和管理。

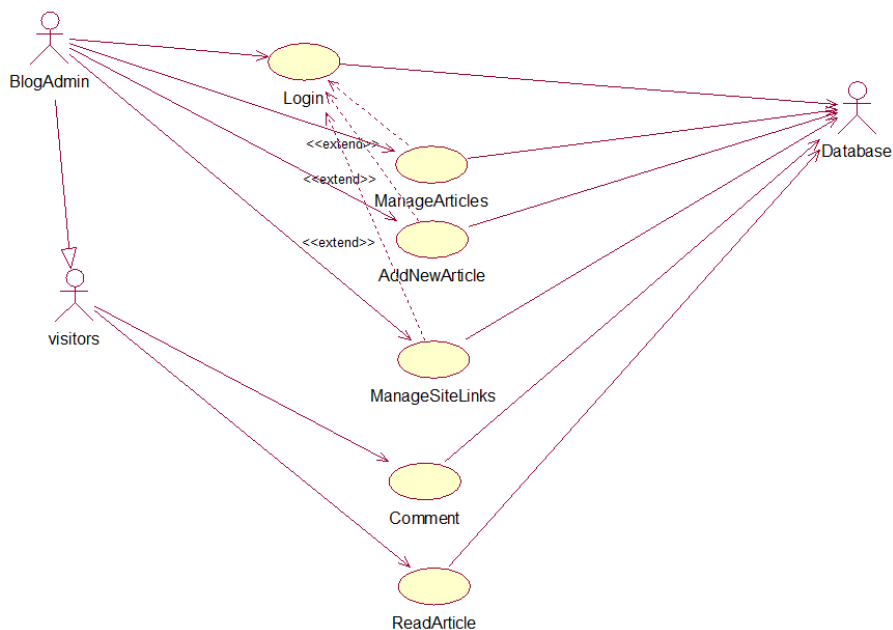
i.多用户大型博客系统，如新浪博客、CSDN，它们功能较全，性能强大，系统的可用性和稳定性相当高，但不符合个人博客的需求。

首先等级系统、首页等元素导致用户难以感受到自己博客主的存在感，而是以普通用户的身份投稿发表文章。其次，界面的内容复杂而拥挤，访问时浏览者会被各种推荐内容、数量众多的排行榜和新闻扰乱视线。

ii.开源的个人博客系统，以 DUZHI_BLOG 为例,它提供了大量的附加功能,如”历史上的今天”、打赏等，但是文章界面不够友好，用户评论需要登录或注册账号，不符合我们的“自由交流”的初衷。

3. 问题描述（需求分析）

用例图：



软件功能：

网站登录	网站登录模块	输入检查管理员账号
		输入检查管理员密码
网站后台页面	后台主页面	显示文章总数量
		显示附件总数量
		显示友情链接数量
		显示最新文章
	发表文章	输入检查文章标题
		输入检查文章标签=
文章编辑		

		文章预览
		存为草稿
	文章管理	显示文章信息列表
		修改文章
		删除文章
		文章预览
	友情链接管理	添加新连接
		链接编辑
		连接删除
游客访问页面	访问主页	显示文章列表
		显示文章标签
		查看友情链接
		文章搜索
	浏览文章	显示文章内容
		显示最后编辑时间
		添加留言
		显示留言

3.1 网站登录模块

输入检查管理员账号：

键盘输入字符串,长度不超过 30 字节,不为空。检查字符串长度,记录是否超长,是否为空,若不超长不空则记录名称。符合格式则不显示输出,不符合格式则确认时报错,显示相应错误

输入检查管理员密码：

键盘输入字符串,长度不超过 10 字节,不为空,不以数字开头。检查字符串长度,记录是否超长,是否为空,若不超长不空则记录账户。符合格式则不显示输出,不符合格式则确认时报错,显示相应错误

3.2 后台主页面

登录并访问主页面：

数据库计算总文章数量,显示在页面上,格式“发表了 X 篇文章”。

数据库计算友情链接总数量,显示在页面上,格式“共有 X 个友情链接”。

数据库中文章标题按发表时间排序后,取前五条显示在页面上,格式:标题-发布时期。

获取服务器时间并显示,格式 XXXX/XX/XX。

3.3 发表文章

输入文章标题：

键盘输入字符串,长度不超过 50 字节,不为空。检查字符串长度,记录是否超长,是否为空,若不超长不空则记录标题。符合格式则不显示输出,不符合格式则确认时报错,显示相应错误。

输入文章标签：

键盘输入字符串,长度不超过 10 字节,可为空。检查字符串长度,记录是否超长,是否为空,若不超长不空则记录路径,若为空则默认为无标签。符合格式则不显示输出,不符合格式则确认时报错,显示相应错误。

文章编辑

键盘输入字符串,通过相关编辑按钮修改字体格式等属性。最长不超过 10KB,可为空。检查文章长度,符合长度要求,点上传后记录入数据库。符合要求后上传至数据库,页面提示文章发表成功

图片/表情插入

通过 html 编辑插入图片链接,切换至 html 格式编辑,使用格式引用

文章预览

浏览文章发布后的情况,单击相应按钮,将把当前内容发布后的结果展示在页面上,显示发布后的文章阅读页面情况。

存为草稿

保存当前文章内容,单击相应按钮,将目前文章编辑框内的内容上传至服务器,但在游客的文章界面中不显示。用户登录后可以在文章管理内继续编辑,将编辑框内内容上传至服务器,上传成功则提示“已保存至草稿箱”,因网络等原因上传失败则提示“保存失败”。

3.4 文章管理

显示文章信息列表

将文章简要信息显示在页面上,登录并访问“文章管理”页面则自动显示,从数据库中读取文章信息并列表显示。显示表格,列信息为:文章标题-发布时间-标签-发布状态-操作。

修改文章

修改文章正文,点击对应文章的“修改”按钮,点击按钮后进入正文编辑界面,从服务器读取文章内容,编辑过程与文章编辑功能相同,将服务器上的文章信息显示在编辑框中,等待编辑,完成后检查长度等格式,符合要求后更新到服务器上。

删除文章

点击对应文章的“删除”按钮,删除指定的一篇文章。点击按钮后则将服务器端文章信息删除,页面上此文章信息消失。

文章阅读

阅读指定的一篇文章,点击按钮后从服务器端获取文章信息,以游客浏览页面显示文章内容。

3.5 友情链接管理

添加新连接

登录并访问“友情链接管理”，点击“添加”按钮，输入新连接名称、连接地址，均为字符串，名称不超过 20 字节，地址不超过 128 字节，均不能为空将新的链接添加到友情链接栏中，更新到服务器上，更新完成后返回成功信息，接收到成功信息后页面显示“添加成功”。若接收到成功信息，页面显示“添加成功”，否则超时时显示“添加失败，请刷新后重试”

链接编辑

登录并访问“友情链接管理”，点击对应的友情链接后的“编辑”按钮，格式同上。输出内容同上。

连接删除

登录并访问“友情链接管理”，点击对应的友情链接后的“删除”按钮。将已有的友情链接信息从服务器端删除，删除成功后返回成功信息，接收到成功信息后页面显示“删除成功”。

3.6 访问主页

显示文章列表

访问博客主页，文章列表自动显示。

显示文章摘要

访问博客主页，鼠标指向该文章

显示文章标签

访问博客主页，文章标签自动显示

查看友情链接

访问博客主页,点击"友情链接"按钮，从服务器获取友情链接信息,显示在页面上。

文章搜索

通过关键字搜索文章，访问博客主页,在搜索栏输入内容,点击搜索。

从数据库中搜索相应文章,将搜索结果返回并显示在页面上，显示搜索到的相关文章,若没有搜索结果则显示空白。

3.7 浏览文章

显示文章内容

访问博客主页,选定文章鼠标单击查看，从数据库中获取文章正文内容及页面，在浏览器中显示。

显示最后编辑时间

查看指定文章时,文章的最后编辑时间自动显示在文章标题下方。从数据库中获取文章最后编辑时间并在页面中显示。

添加留言

在指定文章后添加留言，输入邮箱地址，名字和留言内容，点击留言。

显示留言

浏览文章时可以看到当前文章的留言，查看指定文章时，若存在留言，则留言自动显示在文章下方。显示留言信息，若无留言则显示空白。

系统运行环境：

该系统为 B/S 三层结构，它的运行环境分客户端和数据库服务器端三部分。

以下是系统的软件环境。

（1）客户端

操作系统：Windows8 或更新版本。

浏览器：IE6 以上，其它常见浏览器如 FireFox，Chrome。

（2）数据库服务器端

操作系统：Windows2000 Server 或更新版本。

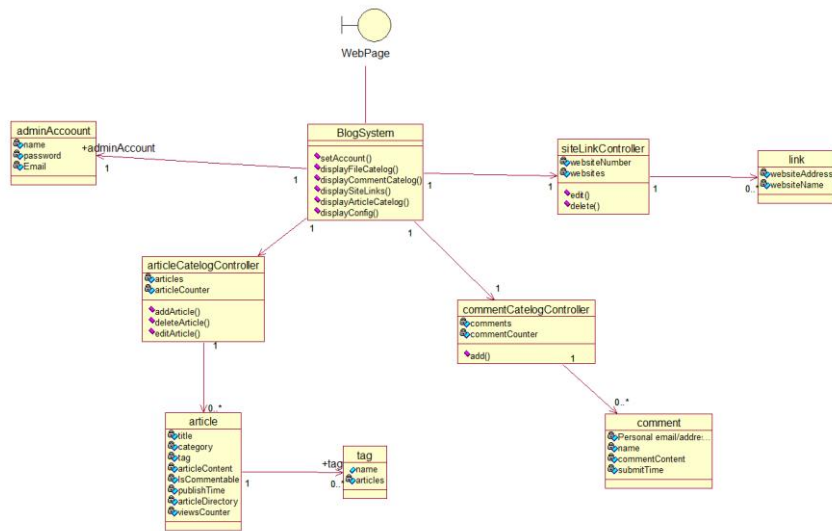
数据库系统：Mysql。

对系统进行需求分析和描述，包括系统的功能及其描述，系统处理的数据及其描述，系统的运行环境和运行方式（行为）描述。

4. 解决方案（概要设计）

系统开发时采用了瀑布模型进行开发，从需求、设计、编码到测试、部署。系统是基于 Web 的应用，采用了 B/S 架构。软件开发时采用了 SpringMVC 和 hibernate 框架。

设计类图：










数据库设计如下：


articles 表






▼	articles
	art_id INT(11) (auto increment)
	title VARCHAR(255)
	created_date DATE
	modified_date DATE
	content VARCHAR(10000)
	categories VARCHAR(50)
	comments_num INT(11)
	status INT(11)
	PRIMARY (art_id)

comments 表

▼  **comments**

	cmt_id INT(11) (auto increment)
	created_date DATE
	art_id INT(11)
	user_id INT(11)
	content VARCHAR(200)
	PRIMARY (cmt_id)

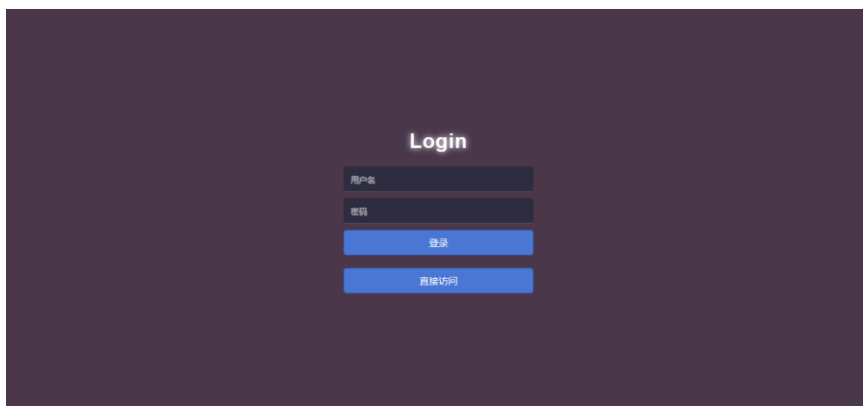
▼  **links**

	id INT(11) (auto increment)
	url VARCHAR(200)
	name VARCHAR(100)
	description VARCHAR(300)
	PRIMARY (id)

Links 表

GUI 设计如下：

登录界面：

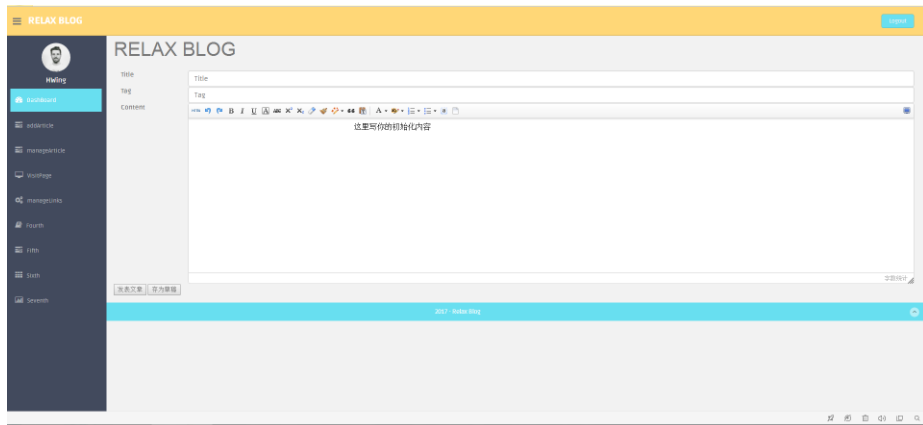


The image shows a login interface with a dark purple background. In the center, there is a white box containing the title "Login". Below the title, there are two input fields: "用户名" (Username) and "密码" (Password). Below these fields are two buttons: "登录" (Login) and "直接访问" (Direct Access).

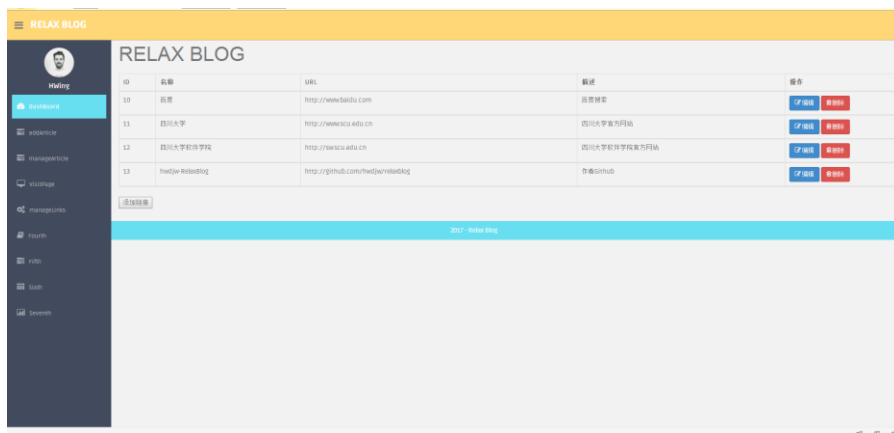
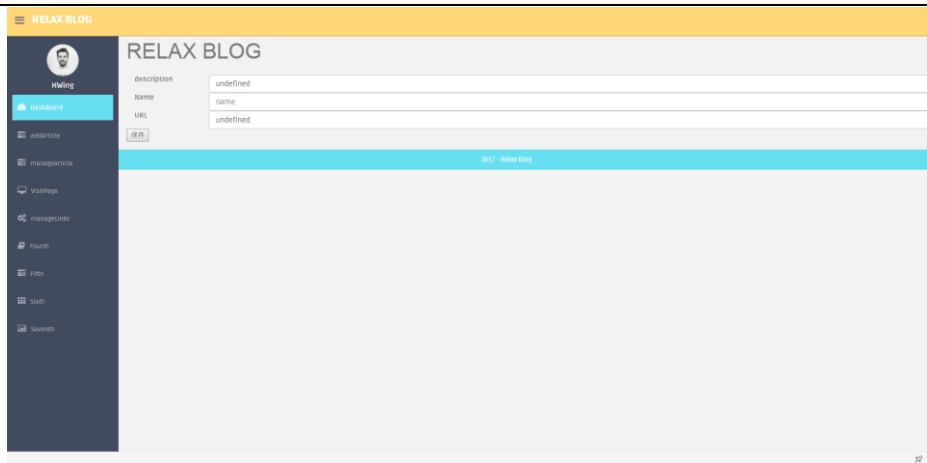
后台管理主页：



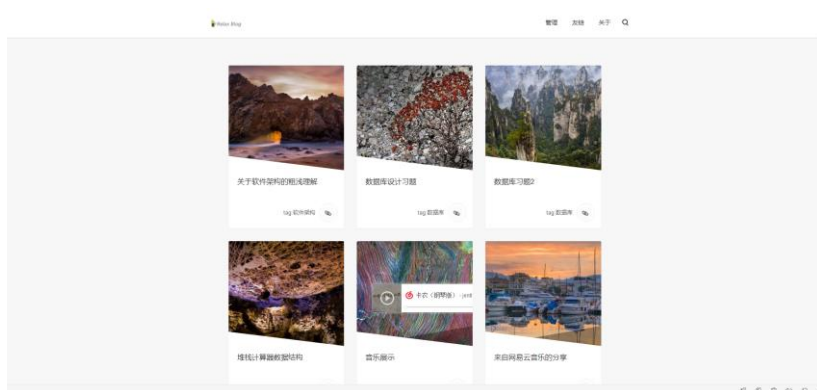
添加、管理文章：



添加、管理链接：



浏览主页：



浏览文章：

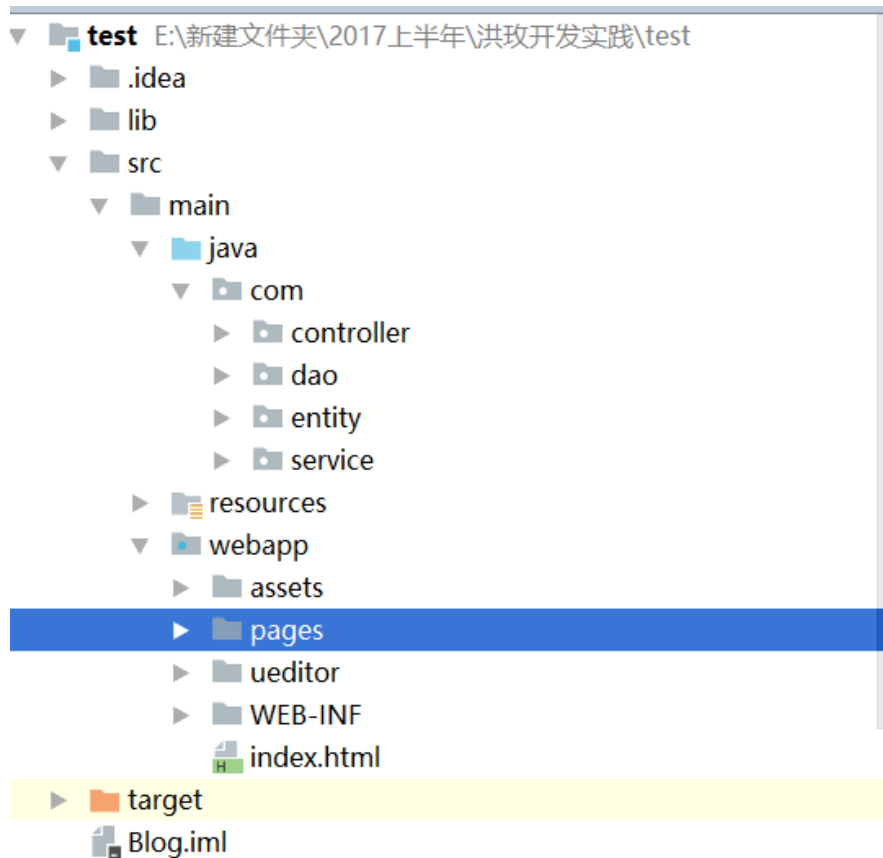


评论：



5. 实现细节（详细设计、实现）

框架：SpringMVC 和 hibernate，将前后段相互分离，以 xml 和 java 注解的形式进行映射及配置信息设定。



资源文件和代码文件的管理如上图，lib 文件夹内放置需要调用的库文件，java 文件夹内放有 controller（控制类），dao 层（实体信息获取），entity 实体类（对应数据库表项）和 service 类（用于支撑某项系统功能）。

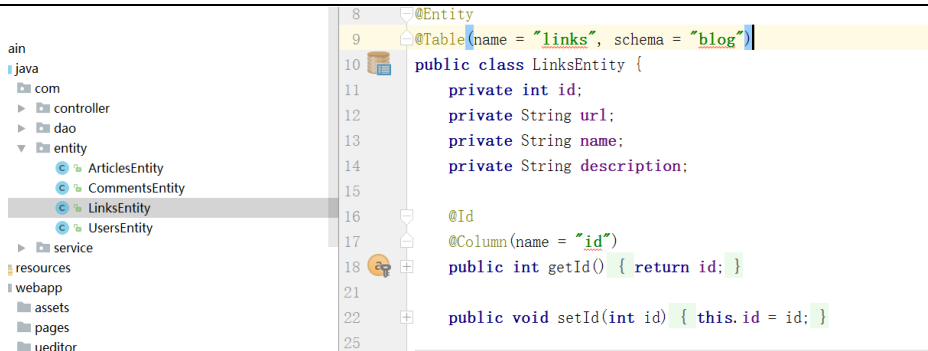
各种页面文件（.html）、页面样式文件（.css）、脚本文件（.js）放在 webapp 下的 pages 和 assets 内，ueditor 放编辑器的实现文件，web-inf 放 web 配置文件。

功能实现过程：

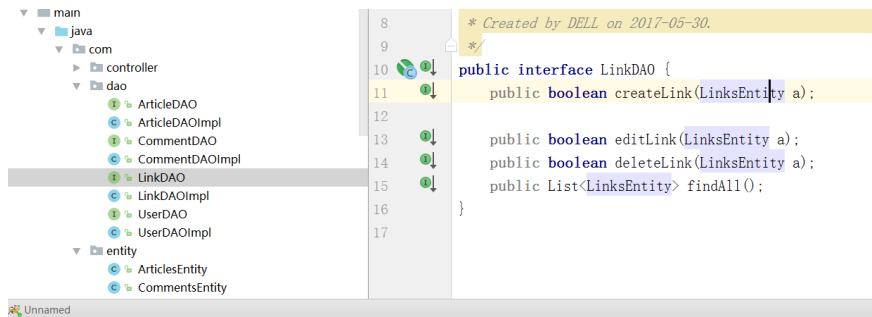
首先，在数据库建立对应的表来存放数据，如管理友情链接，则建立 links 表。代码如下：

```
CREATE TABLE `links` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `url` varchar(200) DEFAULT NULL,  
  `name` varchar(100) DEFAULT NULL,  
  `description` varchar(300) DEFAULT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8;
```

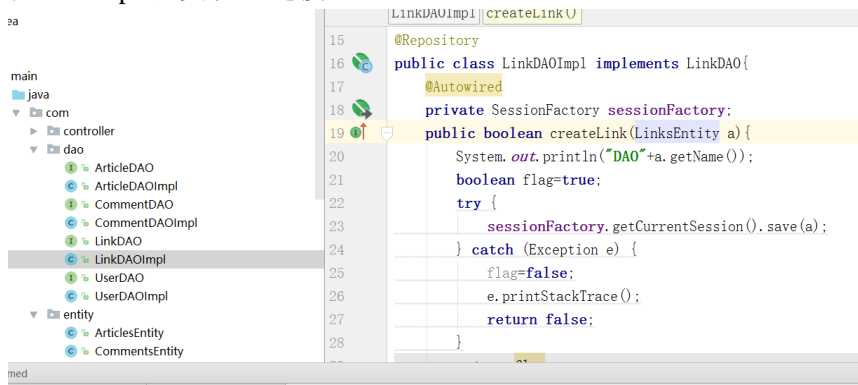
通过 links 表中的列，建立相应的 java 实体类，用于服务器信息的处理和传输。形式如下：



接下来，通过 dao 层建立数据库信息的管理（增删查改的接口）



在 daoimpl 里实现上述接口：



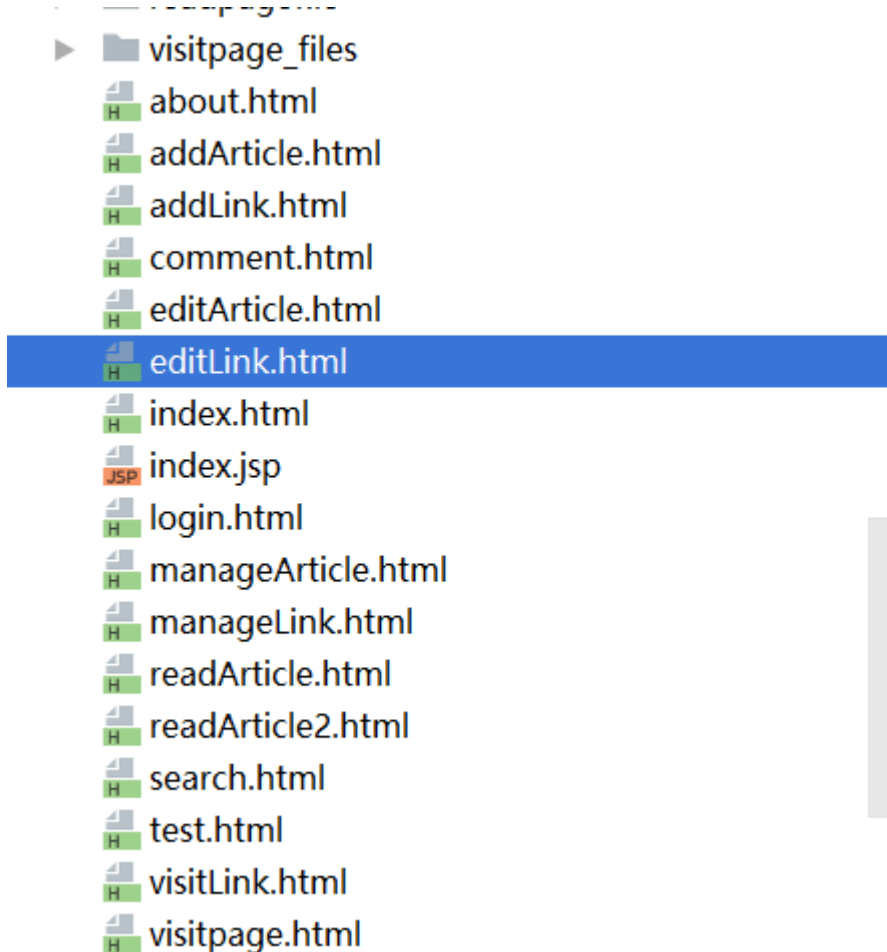
之后，在 service 层建立服务接口并实现



注意到@service 的注解,用于 web 应用中的前端函数调用的映射,前端只能访问服务层,服务层才能访问 dao 层,而 dao 层拥有修改数据库实体信息的能力。

如此分层后，后端功能可以独立实现，而不用将前后端杂糅在一起，提高了内聚性，降低了耦合性。

在实现前端时，需要利用设计阶段制作好的 GUI，仿照实现 html 和 css 文件，达到前段界面的完成。html 包含侧边栏，主内容，右上角按钮栏，搜索栏，下边栏几大元素。而 css 则负责调整颜色、布局以美化界面。



我们为不同的页面制作了不同的页面文件，具体如上。在页面中，功能的调用和展示通过 JavaScript 脚本文件进行动态获取和加载。

在主页获取文章列表时，使用了上述映射中的 all.action，调用获取文章服务，并将每篇文章的信息以 html 格式输出到页面，包装成一个前端“卡片”


```
result += '<div class=\'' + 'post-list-item\'' +  
'<div class=\'' + 'post-list-item-container\'' +  
'<div class=\'' + 'item-thumb bg-deepgrey\'' + ' style=\'' + 'background-image:url(visitp  
' + 'a href=\'' + 'readArticle2.html?page=' + data.artId + '\'' + ' onclick=\'' + 'turnto(d  
' + '<div class=\'' + 'item-desc\'' +  
'<p class=\'' + 'text-overflow\'' + ' style=\'' + 'width:200px;\'' + ' data.content + '</p>'+  
'</div> </a>'+  
'<div class=\'' + 'item-slant reverse-slant bg-deepgrey\'' + '</div>'+  
'<div class=\'' + 'item-slant\'' + '</div>'+  
'<div class=\'' + 'item-label\'' + '</div>'+  
'<div class=\'' + 'item-title\'' + '<a href=\'' +  
' + 'readArticle2.html?page=' +  
' + data.artId + '\'' +  
' + data.title + '</div>'+  
'<div class=\'' + 'item-meta clearfix\'' + '</div>'+  
'<div class=\'' + 'item-meta-ico bg-ico-lock\'' + ' style=\'' + 'background: url(visitpag  
' + '<div class=\'' + 'item-meta-cat\'' + '<a>'+ 'tag:' + data.categories + '</a></div>'+  
'</div> </div> </div> </div>'
```

效果如图



同理，文章管理页面也为 html 动态加载，实现代码如下：

```
status='草稿';
}

result += "<tr><td>" + data.title + "</td>" +
"<td>" + data.artId + "</td>" +
"<td>" + data.createdDate + "</td>" +
"<td>" + data.categories + "</td>" +
"<td>" + status + "</td>" +
//commentnum
"<td>" +
"<a href='\"javascript:void(0)\"' onclick='\"editArticle(this)\"'>\"<i class='\"fa fa-edit\">\"</i> <span>编辑</span></a> \" +
"<a href='\"javascript:void(0)\"' onclick='\"deleteArticle(this)\">\"<i class='\"fa fa-trash-o\">\"</i> <span>删除</span></a> \" +
"<a href='\"javascript:void(0)\"' onclick='\"readArticle(this)\">\"<i class='\"fa fa-rocket\">\"</i> <span>预览</span></a> \" +
"</td>" +
"</tr>"
)
result += "</tbody></table>";
$('#mcg').html(result);
```

效果如下：

RELAX BLOG					
名称	ID	创建日期	标签	状态	
关于软件架构的粗浅理解	43	2017-05-30	软件架构	已发布	编辑 删除 预览
关于软件架构的一个问题	44	2017-05-05	软件架构	草稿	编辑 删除 预览
数据库设计习题	45	2017-05-05	数据库	已发布	编辑 删除 预览
数据库习题2	46	2017-05-05	数据库	已发布	编辑 删除 预览
线性计算数据库结构	47	2017-05-05	数据库结构与算法	已发布	编辑 删除 预览
关于计算机网络的思考	50	2017-05-05	计算机网络	草稿	编辑 删除 预览
音乐赏析	52	2017-05-07	音乐	已发布	编辑 删除 预览
来自网易云音乐的分享	51	2017-05-05	音乐	已发布	编辑 删除 预览

2017 - Relax Blog

对于所有的编辑框和提交按钮，我们都完成了格式校验，以保证数据在传输向后端时符合服务器的数据库表格式。除此之外，后台操作的成功与否也会返回给前端调用处，我们在 JavaScript 脚本中也会检验后台传来的信息，如果信息为“fail”，则会弹出提示，提醒用户相应操作失败。而“success”则会提示操作成功。

因此，我们的操作过程就有了双重保证，以增加系统的稳定性。

检验范例代码如下（编辑文章前端）：

```
tcontent=ue.getContent();
var pageid=getQueryVariable("page");
tcate=$('#article-categories').val();

if(tttitle.length>50|| tttitle.length==0)
{
    alert("标题长度不合法");
    return;
}
if(tcontent.length>10000)
{
    alert("文章过长");
    return;
}
if(tcate.length>10)
{
    alert("标签过长");
    return;
}

function (data)
{
    //var a=data.equals('success')
    if(data=="success")
    {
        alert("保存成功");
        window.location.href="/blog/pages/manageArticle.html"
    }

    else{
        alert("保存失败! 格式不合法");
    }
}

);
```

系统开发过程使用 IDEA 和 mysql 作为开发工具，chrome 作为调试工具，Junit 作为测试工具。开发平台为 windows10 64 位。

6. 实验验证（测试、分析、评价）

进行系统测试时，使用的是等价类划分法设计测试用例。测试工具为 Junit 及人工测试，过程包括单元测试、系统功能测试、验收测试、回归测试。^[2]

测试用例及结果如下表：

用例序号	功能序号	功能	设计输入	预计输出	实际输出	是否通过
1	1	登录功能	空白	提示用户名为空	提示用户名长度错误	通过
2			正确用户名 admin, 正确密码 123456	登录成功	登录成功	通过
3			合理长度用户名 admin, 错误密码 -11111	提示登录信息错误	提示登录信息错误	通过
4			超长长度用户名 a123456789101112	提示用户名长度错误	提示用户名长度错误	通过
5			正确用户名 admin, 超长密码 123456789012345678901	提示密码超长	提示密码长度错误	通过
6			正确用户名 admin, 空密码	提示密码为空, 重新输入	提示密码长度错误	通过
7	2	仪表盘显示文章总数	登录并访问管理主页	显示文章总数量	显示文章总数量	通过
8			未登陆	不显示文章总数量, 跳转登录	不显示文章总数量, 跳转登录	通过
9	3	仪表盘日期	登录并访问管理主页	显示日期	显示日期	通过
10			未登陆	不显示日期, 跳转登录	不显示日期, 跳转登录	通过
11	4	仪表盘显示友情链接总数	登录并访问管理主页	显示友情链接数量	显示友情链接数量	通过
12			未登陆	不显示友情链接数量, 跳转登录	不显示友情链接数量, 跳转登录	通过
13	5	仪表盘显示最新文章	登录并访问管理主页	显示最新文章	显示 5 篇文章概况	通过
14			未登陆	不显示最新文章, 跳转登录	不显示最新文章, 跳转登录	通过
15	6	创建文章	空白标题+合法分类及内容, 发表文章	提示标题长度错误	提示标题长度错误	通过

16			标题“软件架构”+标签“软件架构”+内容“abcd”发表文章	提示发表成功	提示发表成功	通过
17			标题“abcd”+超长标签+内容“abcd”发表文章	提示标签超长	提示标签超长	通过
18			超长标题+标签“1234”+内容“你好”发表文章	提示标题长度错误	提示标题长度错误	通过
19			标题“abcd”+标签“abcd”+超长内容，发表文章	提示内容超长	提示内容超长	通过
20			未登陆	跳转登录界面	跳转登录界面	通过
21			标题“abcd”+标签“abcd”+内容“abcd”，发表文章	提示发表成功	提示发表成功	通过
22			标题“abcd”+标签“abcd”+内容粗体，发表文章	提示发表成功	提示发表成功	通过
23			标题“abcd”+标签“abcd”+内容下划线，发表文章	提示发表成功	提示发表成功	通过
24			复制图片代码到内容，发表文章	提示发表成功	提示发表成功	通过
25	7	存为草稿	空白标题+合法分类及内容，存为草稿	提示标题长度错误	提示标题长度错误	通过
26			合法标题+标签“软件架构”+内容“abcd”存为草稿	提示发表成功	提示发表成功	通过
27			标题“abcd”+超长分类+内容“abcd”存为草稿	提示标签超长	提示标签超长	通过
28			超长标题+分类“1234”+内容“你好”存为草稿	提示标题长度错误	提示标题长度错误	通过
29			标题“abcd”+标签“abcd”+超长内	提示内容超长	提示内容超长	通过

			容，存为草稿			
30			未登陆	跳转登录界面	跳转登录界面	未通过
31			标题“abcd”+标签“abcd”+内容“abcd”，存为草稿	提示发表成功	提示发表成功	通过
32			标题“abcd”+标签“abcd”+内容粗体，存为草稿	提示发表成功	提示发表成功	通过
33			标题“abcd”+标签“abcd”+内容下划线，存为草稿	提示发表成功	提示发表成功	通过
34			复制图片代码到内容，存为草稿	提示发表成功	提示发表成功	通过
35	8	管理页面显示文章信息	已登录，2篇文章	正确显示两篇文章信息	正确显示两篇文章信息	通过
36			已登录，一篇文章一篇草稿	正确显示文章及草稿信息	正确显示文章及草稿信息	通过
37			未登陆	跳转登录界面	跳转登录界面	通过
38	9	修改文章	原文章标题“abcd”，标签‘abcd’，内容‘abcd’，点击修改此文章	进入编辑页面，正确显示文章标题、标签、内容并等待修改	进入编辑页面，正确显示文章标题、标签、内容并等待修改	通过
39			原文章标题“abcd”，标签为空，内容‘abcd’，点击修改此文章	进入编辑页面，正确显示文章标题、标签、内容并等待修改	进入编辑页面，正确显示文章标题、标签、内容并等待修改	通过
40			空白标题+合法分类及内容，发表文章	提示标题长度错误	提示标题长度错误	通过
41			标题“软件架构”+标签“软件架构”+内容“abcd”发表文章	提示发表成功	提示发表成功	通过
42			标题“abcd”+超长标签+内容“abcd”发表文章	提示标签超长	提示标签超长	通过
43			超长标题+标签“1234”+内容“你	提示标题长度错误	提示标题长度错误	通过

			好”发表文章			
44			标题“abcd”+标签“abcd”+超长内容，发表文章	提示内容超长	提示内容超长	通过
45			未登陆	跳转登录界面	跳转登录界面	通过
46			标题“abcd”+标签“abcd”+内容“abcd”，发表文章	提示发表成功	提示发表成功	通过
47			标题“abcd”+标签“abcd”+内容粗体，发表文章	提示发表成功	提示发表成功	通过
48			标题“abcd”+标签“abcd”+内容下划线，发表文章	提示发表成功	提示发表成功	通过
49			复制图片代码到内容，发表文章	提示发表成功	提示发表成功	通过
50			空白标题+合法分类及内容，存为草稿	提示标题长度错误	提示标题长度错误	通过
51			合法标题+标签“软件架构”+内容“abcd”存为草稿	提示发表成功	提示发表成功	通过
52			标题“abcd”+超长分类+内容“abcd”存为草稿	提示标签超长	提示标签超长	通过
53			超长标题+分类“1234”+内容“你好”存为草稿	提示标题长度错误	提示标题长度错误	通过
54			标题“abcd”+标签“abcd”+超长内容，存为草稿	提示内容超长	提示内容超长	通过
55			未登陆	跳转登录界面	跳转登录界面	通过
56			标题“abcd”+标签“abcd”+内容“abcd”，存为草稿	提示发表成功	提示发表成功	通过
57			标题“abcd”+标签“abcd”+内容粗体，存为草稿	提示发表成功	提示发表成功	通过

58			标题“abcd”+标签“abcd”+内容下划线，存为草稿	提示发表成功	提示发表成功	通过
59			复制图片代码到内容，存为草稿	提示发表成功	提示发表成功	通过
60	10	删除文章	登录，删除第一篇文章	提示删除成功，更新页面	提示删除成功，更新页面	通过
61			登录，删除第三篇文章	提示删除成功，更新页面	提示删除成功，更新页面	通过
62			登录，删除第一草稿	提示删除成功，更新页面	提示删除成功，更新页面	通过
63			登录，删除第三草稿	提示删除成功，更新页面	提示删除成功，更新页面	通过
64			未登陆	跳转登录界面	跳转登录界面	通过
65	11	文章预览	登录，预览第一篇文章	打开预览页面，正确显示内容	打开预览页面，正确显示内容	通过
66			登录，删除第三篇文章	打开预览页面，正确显示内容	打开预览页面，正确显示内容	通过
67			登录，预览第一草稿	打开预览页面，正确显示内容	打开预览页面，正确显示内容	通过
68			登录，预览第三草稿	打开预览页面，正确显示内容	打开预览页面，正确显示内容	通过
69			未登陆	跳转登录界面	跳转登录界面	通过
70	12	添加链接	Name 为空	提示名称长度错误	提示名称长度错误	通过
71			Name= 'github' , URL 为空	提示 URL 长度错误	提示 URL 长度错误	通过
72			Name= 'github' , URL= 'github.com' ,描述为空	提交成功	未提示	通过
73			都为空	报告任意长度错误	报告任意长度错误	通过
74			Name 超长	提示名称超长	提示名称超长	通过
75			Name= 'github' , URL 超长,描述为空	提示 URL 超长	提示 URL 超长	通过
76			Name= 'github' , URL= 'github.com' ,描	提示描述超长	提示描述超长	通过

			述超长			
77			未登陆	提示登录信息错误	未提示	通过
78	13	编辑链接	原 Name='github', URL='github.com', 描述为空, 进入编辑	正确显示原内容	未显示	通过
79			Name 为空	提示名称长度错误	提示名称长度错误	通过
80			Name= 'github' , URL 为空	提示 URL 长度错误	提示 URL 长度错误	通过
81			Name= 'github' , URL= 'github.com' , 描述为空	提交成功	未提示	通过
82			都为空	报告任意长度错误	报告任意长度错误	通过
83			Name 超长	提示名称超长	提示名称超长	通过
84			Name= 'github' , URL 超长, 描述为空	提示 URL 超长	提示 URL 超长	通过
85			Name= 'github' , URL= 'github.com' , 描述超长	提示描述超长	提示描述超长	通过
86			未登陆	提示登录信息错误	未提示	通过
87	14	删除链接	删除第一个链接	提示删除成功	提示删除成功, 未更新页面	通过
88			删除第三个连接	提示删除成功	提示删除成功, 未更新页面	通过
89			未登陆	跳转登录界面	未跳转	通过
90	15	浏览主页时显示文章列表	登录	文章列表显示	文章列表显示	通过
91			未登陆	文章列表显示	文章列表显示	通过
92	16	浏览主页, 鼠标指向时显示文章摘要	登录, 文章为 'abcd'	摘要显示全篇文章	摘要显示全篇文章	通过
93			登录, 文章超过摘要大小	显示局部文章	显示局部文章	通过
94			未登录, 文章为 'abcd'	摘要显示全篇文章	摘要显示全篇文章	通过
95			未登录, 文章超过	显示局部文章	显示局部文章	通过

			摘要大小			
96	17	浏览主页时显示文章标签	未登陆	显示文章标签	显示文章标签	通过
97			登录	显示文章标签	显示文章标签	通过
98	18	浏览文章时正确显示内容	未登陆	显示文章内容	显示文章内容	通过
99			登录	显示文章内容	显示文章内容	通过
100	19	浏览文章时正确显示编辑时间	未登陆	显示文章编辑时间	显示文章编辑时间	通过
101			登录	显示文章编辑时间	显示文章编辑时间	通过
102	20	浏览文章时正确显示标题	未登陆	显示文章标题	显示文章标题	通过
103			登录	显示文章标题	显示文章标题	通过

经过 3 次系统功能测试（包括自测、分组测试和验收测试）及修复，所有测试用例已可顺利通过，满足功能需求的要求，完成 RelaxBlog 所有功能实现。

性能测试（本机环境）：

编辑文章≤100ms

添加文章≤100ms

删除文章≤10ms

浏览文章≤1s

编辑友情链接≤50ms

添加友情链接≤50ms

删除友情链接≤10s

登录≤100ms

页面跳转≤2s

总体性能符合预期,无大范围抖动。

7. 结束语（结论、进一步的工作）

RelaxBlog 是一款由我们小组全程开发的 web 应用,从基础的架构搭建到功能的实现, 面临了诸多的困难与挑战。我们使用了 SpringMVC 和 hibernate 框架,使前后端分离,从可修改性、安全性、稳定性等方面提升了软件的质量属性。^[3]

但从实用的角度来看,RelaxBlog 的功能还是比较单一。成熟的大型博客系统有着复杂的交互功能和储存功能,不仅局限于几张数据表。同时这些系统还利用分布式架构提升了软件性能,成熟的软件工程手段也使它们的错误率降到极低,而我们的系统显然是远远不够成熟的。

但作为一款个人博客系统,提供给少量用户浏览,本系统还是能够胜任各项功能的。前端动态界面和清爽风格使人机交互变得友好,没有冗杂的功能影响用户的观感,新元素的融入也使网站紧随潮流,不再呆板。此外,简单的爬虫实现的文章爬取的附加功能也为用户提供了极大的便利。

随着软件开发实践项目进入尾声,RelaxBlog 并不会停止进步的脚步。我们将在未来对软件的架构进行优化,对代码进行重构,实现诸如大量文章导入、多系统多客户端、安全加密等功能,提升软件的质量和便捷度,最终使 RelaxBlog 一步步接近于真正的具有可用性的成

熟系统。

参考文献

- [1] 中华人民共和国国家标准. 计算机软件产品开发文件编制指南. GB/T8567-1988
- [2] Roger S • Pressman. 软件工程——实践者的研究方法. 机械工业出版社, 2011
- [3] Len Bass, Paul Clements, Rick Kazman. 软件构架实践. 清华大学出版社, 2013

附录：

1. [项目开发计划](#)；
2. [软件需求规格说明书](#)；
3. [软件设计文档](#)；
4. [软件测试文档](#)；
5. [用户手册](#)；