

第 31 章 BBS 系统

BBS（电子公告牌系统），国内常称为“论坛”。BBS 系统所涉及的领域几乎无所不包，不论用户兴趣是什么，都会有特定的 BBS 系统。BBS 上的讨论有着极大的自由，不管身份如何，只要具备一定的写作能力与上网条件，都可以在上面发表话题和交流。每个用户不但是信息的接受者，同时也可以成为信息的提供者。本章将利用 JSP 技术构建一个功能相对完备的 BBS 系统。

本章要点包括以下内容：

- ☐ 管理员删贴、锁定、置顶、添加论坛
- ☐ 重新编辑帖子内容
- ☐ 多种方式投票
- ☐ 搜索论坛内容
- ☐ 版面发贴

31.1 系统总体设计

BBS 系统的设计也是按照使用者和 BBS 管理者两类用户功能和需求来设计的。其中包括了最常规的所有用户对 BBS 话题的浏览，发表对某个话题的评论、回复等。管理员可以进行论坛发贴的管理、论坛后台管理、用户管理等操作。本节将具体介绍系统的功能与数据库设计。

31.1.1 系统结构

本节所构建的 BBS 系统主要采用 JSP 以及 Servlet、Struts 等技术。其中显示论坛分类、话题、话题内容等采用 JSP 与 Servlet 相结合的模式，主要处理由 Servlet 来完成。用户提交数据、请求、信息验证等的处理由 Struts 的 Action 和 ActionForm 处理。另外 Struts 处理器和 Servlet 在处理或组织数据时也调用 Bean 中的处理逻辑或方法，如图 31.1 所示。

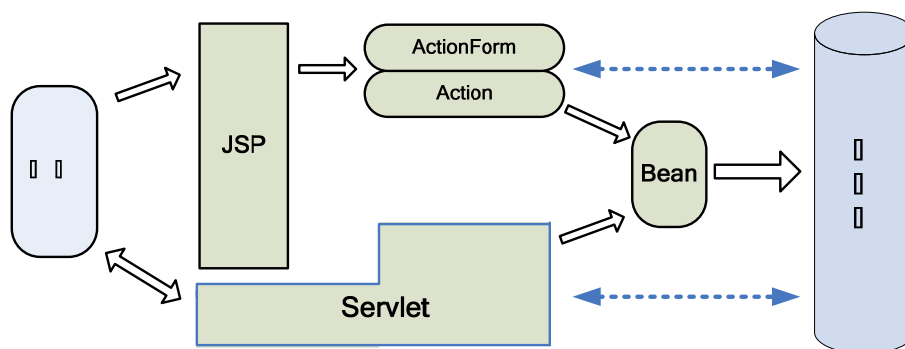


图 31.1 系统结构图

图 31.1 体现了本节所构建系统的总体处理结构，对数据库的操作由 Action、JavaBean 和 Servlet 来

完成。Action 和 Servlet 主要通过 JavaBean 访问数据库，JSP 中只用于构建用户界面视图。采用 Servlet 来组织响应数据，主要为了减少数据传递，可以实现由数据库直接输出到视图中，提高处理和响应效率。本系统的一大特点是视图页面中很少有 Java 脚本代码出现。

31.1.2 系统功能简介

在 BBS 系统中，所有的未锁定的用户可以登录并浏览论坛中的发帖，对话题进行回复，也可以搜索自己想要浏览内容；网站的管理员可以添加论坛分类、分论坛；版主可以进行发帖的管理、删贴等操作。BBS 系统可以分为以下几个功能模块，图 31.2 是系统功能图。

31.1.2.1 论坛浏览与搜索

此功能模块主要是从数据库中读取信息，并格式化输出。所有用户都可以浏览论坛里的发帖以及回复内容。通过首页等位置所列的论坛标题，进入分论坛的话题列表，再进一步进入阅读话题及相亲的回复内容的页面。用户可以利用模糊搜索的方式，查找或检索自己相要阅读的内容。

31.1.2.2 发表话题

这个模块实际是对数据库进行写入操作，将所发表的内容写入数据库中。所有未锁定用户都可以在登录的情况下发表新贴、重新编辑发帖内容、回复话题以及发起投票。用户还可以在发帖时在发帖内容中插入图片并上传到服务器。

31.1.2.3 用户信息维护

用户可以注册新的论坛账号，在登录的情况下可以对自己的注册信息进行修改，也可以对登录密码进行修改。本模块也包括了用户在登录与注销，在用户登陆时，会向用户 session 中写入用户名、昵称、权限等信息。用户在论坛中的行为通过用户 session 中的内容特别是权限来分别构建不同的视图。如管理员在登陆后，根据其权限会有导航条上显示“后台管理”的链接，其他用户没有此项。用户在登录成功后，会通过将数据库中用户在线状态更新为 1，在浏览发帖时，会将该在线用户所发贴是的作者“昵称”显示为红色。表示该用户在线。

31.1.2.4 论坛前台管理

管理员其实是在用户信息表中的一个权限值大于 10 的用户，系统会根据其权限值，判断其可以进行一切论坛提供的管理操作。另外版主是根据其已被设定的可以管理的分论坛，来判断是否可以对某一分论坛进行前台管理。管理员拥有有版主的一切权限，论坛管理内容主要有删贴、锁贴、置顶、加为精华贴、设置为论坛公告以及它们所对应的反向操作。其中设置为论坛公告是将普通贴子放在各个分论坛话题列表的顶端，只有管理员才有此权限。

31.1.2.5 论坛后台管理

论坛后台管理只有管于员才能有此权限进行操作，其他用户无权进入管理界面与进行相关功能操作。本系统后台项目较少，主要为添加分论、添加分论坛，编辑分论坛的属性。其中在编辑分论坛是对已添加的分论坛进行属性设定，可以设定论坛名、描述信息、图示以及论坛版主等。论坛后台管理还有一个功能就是用户管理，本系统只提供了一种操作，也就是锁定用户，使其不能发帖。

除上面这些用户、版主或管理员可以操作的功能模块外，论坛系统还要实现一些由系统自动操作的功能，这些操作是伴随用户操作而自动进行的，主要包括以下几种：

- ❑ 用户发帖后对用户总发帖数，以及分论坛总话题或全部贴子数量进行更新进行更新。这些内容将显示在用户信息和论坛信息中。

- ❑ 用户进入某一话题进行阅读时，会自动更新该话题的点击次数。发表对某一话题的回复或后，会更新该话题的回复次数以及最后回复作者与时间，这些内容也将显示在话题列表中。
- ❑ 用户在不进行任何操作超过一定时间后，对应的 `session` 自动过期而被销毁，由监听器自动将数据库中用户在线状态值置为 0。

下面是以各种用户权限为基础的系统功能图，如图 31.2 所示。

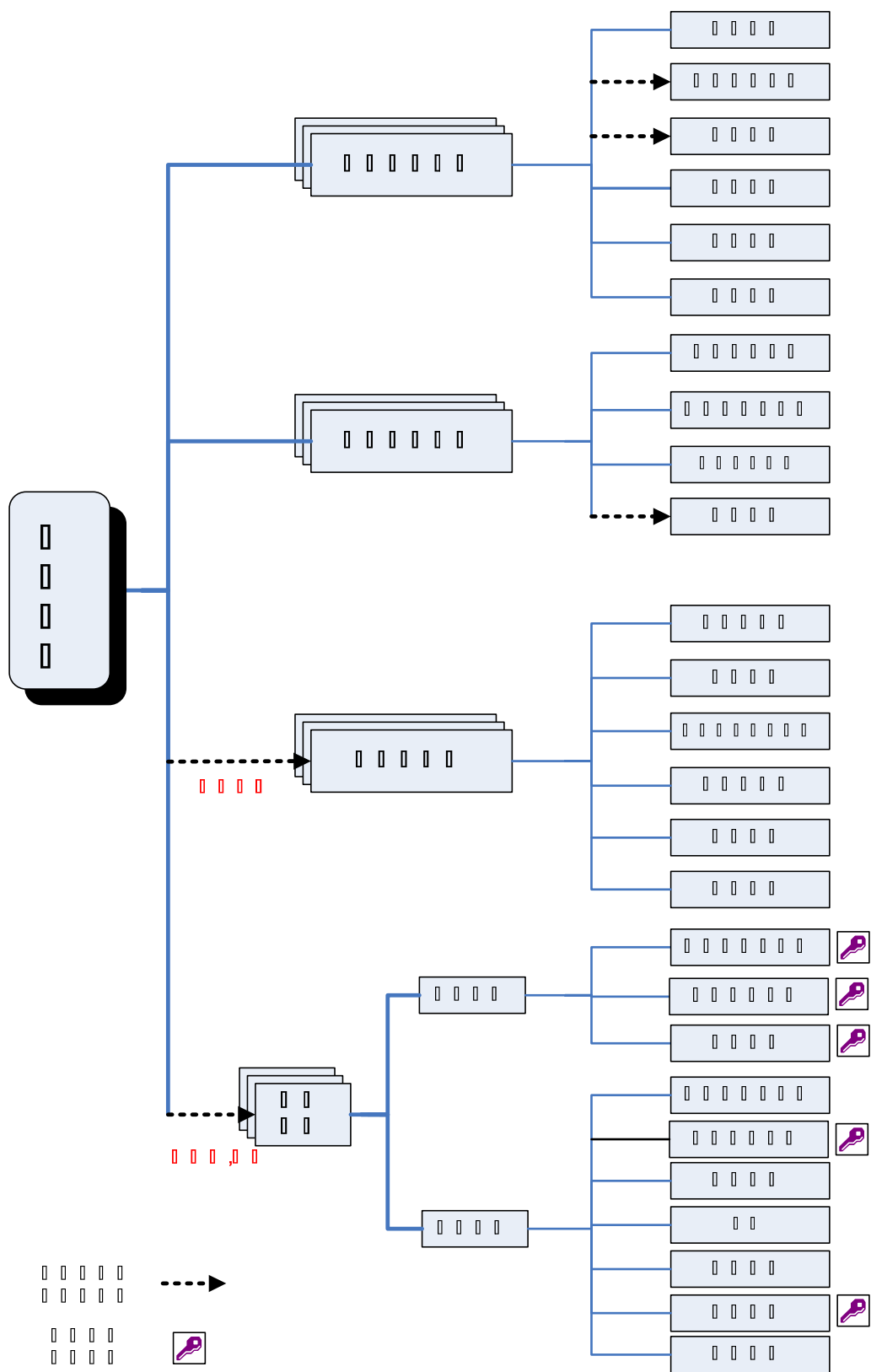


图 31.2 系统功能与权限

本系统中有一个用于设置常量值的数据类，其中定义了上传图片保存文件夹、文件的大小的上限、

论坛内容分页时每页显示的数量等系统常量。这些值是可以灵活修改的常量，将其放在一个类中是为了统一而且便于以后的维护。

31.1.3 数据库设计

根据 BBS 系统功能、需求以及数据的特点，在数据库中应实现对以下几类数据进行存储。

- ❑ 发帖内容：有关话题、回复的一些内容，如标题、详细内容、发帖时间等。
- ❑ 投票信息：用户发起投票和投票结果信息。
- ❑ 用户信息：BBS 系统所有用户的信息，包括管理员。
- ❑ 论坛分类信息：BBS 系统论坛的分类与分论坛信息。

根据上面的数据库信息需求，综合本系统的特点。在 Mysql 数据库服务器中建数据库 mybbs，其中的数据库表以及其存放数据描述如表 16.1 所示

表 16.1 mybbs数据库包含的数据表及其功能

数据表名	功能描述
posts	存放主题与回复的相关组成信息
vote	存放用户发起投票与投票结果信息
forums	存放论坛他类与论坛属性相关信息
bbsuser	存放所有BBS系统用户的信息

上面表中所列的系统所用到的数据库表，可以通过 MySQL 的图形管理工具 Mysql-Front 来建立，也可以通过批处理 sql 文件一次性建立。下面逐一介绍这些表的结构与字段功能描述。

31.1.3.1 主题与回复信息表 posts

主题与回复信息表 posts 中混合存放了主题与回复的信息，它是一个基础表也是本系统的核心数据表。其他表如投票信息表中的 postid 同 posts 表的 id 关联，表示该条投票项目对应 posts 表中的对应主题。posts 表包括记录编号 (id)、对应的分论坛编号 (forumid)、主题类型 (groupid)、主题与回复识别 (istopic)、主题编号 (topic_id)、主题或回复的标题 (title)、贴子的内容 (content)、发表的作者 (author)、作者编号 (author_id)、主题的点击次数 (click_times)、主题的回复次数 (reply_times)、发表者的 IP 地址 (post_ip)、发表时间 (post_time)、主题最后的回复者 (last_postor)、主题最后回复时间 (last_time)、主题排序控制参数 (istop)、主题是否被锁定 (islock)，主题或回复是否被删除 (isdel)。其中记录编号为主索引、自动增量。主题编号也为索引字段。表 posts 的结构如表 31.2 所示。

表 31.2 posts表的结构

字段名	列类型与长度	描述	备注
id	int(11)	记录编号	自动增量
forumid	int(2)	所属论坛编号	不允许为空
groupid	int(1)	话题种类	默认值为0
istopic	int(1)	是否为主题，1为主题，2为投票主题	默认值为0
topic_id	int(8)	主题编号	不允许为空
title	varchar(100)	标题	
content	text	内容	
author	varchar(20)	作者	不允许为空
author_id	int(8)	作者编号	不允许为空
click_times	int(7)	点击次数	默认值为0
reply_times	int(5)	回复次数	默认值为0
post_ip	varchar(16)	发表时的IP	
post_time	datetime	发表时间	

last_postor	varchar(20)	最后回复者	
last_time	datetime	最后回复时间	
istop	int(3)	主题排序控制值	默认值为0
islock	int(1)	是否锁定, 1为锁定	默认值为0
isdel	int(1)	是否删除, 1为删除	默认值为0

上表所列的 posts 表的字段中, 一些只有主题才有内容的字段描述如下。

- ❑ groupid: 值为 0 时为普通主题, 值为 1 时为精华主题, 值为 2 时为论坛公告。
- ❑ click_times: 每回复一次或点击一次自动加 1。
- ❑ reply_time: 每回复一次自动加 1。
- ❑ last_postor: 最后一个回复者的昵称。
- ❑ last_time: 最后一次回复的时间。
- ❑ istop: 用于控制置顶排序的值。
- ❑ islock: 主题是否被锁定, 0 为没有锁定, 可以继续发表回复, 1 为锁定, 不能回复。

建立表 posts 的 SQL 批处理代码如下。

```
CREATE TABLE `posts` (
  `id` int(11) NOT NULL auto_increment,
  `forumid` int(2) NOT NULL default '0',
  `groupid` int(1) default '0',
  `istopic` int(1) NOT NULL default '0',
  `topic_id` int(8) NOT NULL default '0',
  `title` varchar(100) default NULL,
  `content` text,
  `author` varchar(20) NOT NULL default '',
  `author_id` int(8) NOT NULL default '0',
  `click_times` int(7) default '0',
  `reply_times` int(5) default '0',
  `post_ip` varchar(16) default NULL,
  `post_time` datetime default '0000-00-00 00:00:00',
  `last_postor` varchar(20) default NULL,
  `last_time` datetime default NULL,
  `istop` int(3) NOT NULL default '0',
  `islock` int(1) NOT NULL default '0',
  `isdel` int(1) default '0',
  PRIMARY KEY (`id`),
  KEY `topicid` (`topic_id`)
) ENGINE=MyISAM DEFAULT CHARSET=gbk;
```

为使系统能够正常演示运行并显示, 表 posts 中至少应有一条主题与记录, 否则进入系统首页时将出现只有顶端 Logo。图 31.3 是表 posts 的示例数据。

id	forumid	groupid	istopic	topic_id	title	content	author	author_id	click_times	reply_times	post_ip	post_time	last_postor	last_time	istop	islock	isdel
1	2	0	1	1	第一个主题?	<MEMO>	caxton	1	4	1	10.4.25.18	2006-08-09	caxton	2006-08-	0	0	0
2	2	0	0	1	第一个回复	<MEMO>	caxton	1	0	0	127.0.0.1	2006-08-10		2006-08-	0	0	0
3	2	0	2	2	气候最适宜的地区	<NULL>	richard	2	3	1	10.4.25.16	2006-08-09	richard	2006-08-	0	0	0
4	2	0	0	2	第一个投票	<NULL>	caxton	1	0	0	127.0.0.1	2006-08-09		2006-08-	0	0	0

图 31.3 表 posts 的记录示例

31.1.3.2 投票信息表 vote

用户投票信息表主要存放用户发起的投票内容及投票结果。vote 表的字段包括记录编号 (id)、对

应主题的记录编号 (postid)、是否是投票主题以及投票的方式 (iskey)、投票主题中的条目编号 (keyid)、投票标题或条目描述 (item)、投票者昵称 (voter)、投票结果 (result)、投票者昵称 (voter)、是否锁定 (islock)。其中评论编号为主索引、自动增量。表 commentary 的结构如表 31.3 所示。

表 31.3 commentary 表的结构

字段名	列类型与长度	描述	备注
id	int(11)	记录编号	自动增量
postid	int(9)	对应主题编号	
iskey	int(1)	是否为投票主题的标题	默认值为0
keyid	int(2)	投票条目编号	默认值为0
item	varchar(160)	投票主题与分条目内容	
result	int(8)	投票结果	
voter	text	投票者名单	默认值为0
islock	int(1)	是否被锁定	默认值为0

上表中所列的字段，一些有特定的特殊用途，具体用法如下。

- ❑ iskey: 大于 0 的为投票的主题。其中 1、2、3、4 分别对应四种投票方式。
- ❑ keyid: 投票的分条目编号，主题记录的 keyid 为 0。分条目编号从 1 开始。
- ❑ item: 如果 iskey 大于 0 则存储投票标题，否则为分项目描述。
- ❑ result: 如果 iskey 大于 0，则存储总投票数量。否则为该条目的得票数。
- ❑ voter: 记录投该条目票的用户昵称，如果 iskey 大于 0，则为空值。

建立表 vote 的 SQL 批处理代码如下。

```
CREATE TABLE `vote` (
  `id` int(11) NOT NULL auto_increment,
  `postid` int(9) NOT NULL default '0',
  `iskey` int(1) NOT NULL default '0',
  `keyid` int(2) default '0',
  `item` varchar(160) default NULL,
  `voter` text,
  `result` int(8) default '0',
  `islock` int(1) default '0',
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=gbk;
```

图 31.4 是本系统 vote 表中的用户评论信息示例。

id	postid	iskey	keyid	item	voter	result	islock
1	3	1	0	气候最适宜的地区	<NULL>	25	0
2	3	0	1	河南省	<MEMO>	2	0
3	3	0	2	四川省	<MEMO>	3	0
4	3	0	3	浙江省	<MEMO>	5	0
5	3	0	4	重庆市	<MEMO>	2	0
6	3	0	5	海南省	<MEMO>	8	0
7	3	0	6	台湾省	<MEMO>	5	0

图 31.4 vote 表中的用户投票信息示例

31.1.3.3 论坛分类表 forums

论坛分类表 forums 主要存放论坛分类与分论坛属性信息。forums 表的字段包括分论坛编号 (id)、是否是分类 (isparent)、分类编号 (sortid)、分类名或分论坛名 (forum_name)、版主名单 (admin_author)、

主题总数 (total_topic)、贴子总数 (total_post)、论坛图标 (pic_logo)、论坛描述 (description)。其中论坛编号为主索引、自动增量。表 forums 的结构如表 16.4 所示。

表 31.4 forums 表的结构

字段名	列类型与长度	描述	备注
id	int(3)	分论坛编号	自动增量
isparent	int(1)	是否为论坛分类	不允许为空
sortid	int(3)	分类编号	不允许为空
forum_name	varchar(30)	论坛名称	
admin_author	varchar(60)	版主名单	
total_topic	int(10)	总主题数量	
total_post	int(11)	总贴子数量	
pic_logo	varchar(30)	论坛图标	
description	varchar(160)	论坛描述	

建立表 forums 的 SQL 批处理代码如下。

```
CREATE TABLE `forums` (
  `id` int(3) NOT NULL auto_increment,
  `isparent` int(1) NOT NULL default '0',
  `sortid` int(3) NOT NULL default '0',
  `forum_name` varchar(30) NOT NULL default '',
  `admin_author` varchar(60) default NULL,
  `total_topic` int(10) default NULL,
  `total_post` int(11) default NULL,
  `pic_logo` varchar(30) default 'images/forum_logo1.gif',
  `description` varchar(160) default NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=gbk;
```

图 31.4 是本系统 forums 表中的管理员信息示例。

id	isparent	sortid	forum_name	admin_author	total_topic	total_post	pic_logo	description
1	1	1	程序设计	caxton	0	0	images/forum_logo1	<NULL>
2	0	1	Java	stanley,kaka,royal	8	19	images/forum_logo1	SUN, JSP, Sevlet
3	0	1	VB	kelly	1	2	images/forum_logo1	Microsoft VB
4	0	1	dephi	kelly	0	0	images/forum_logo1	Dephi
5	0	1	VC++	kelly	0	0	images/forum_logo1	Microsoft

图 31.5 forums 表中的论坛信息示例

31.1.3.4 用户信息表 bbsuser

用户信息表 bbsuser 中混合存放了普通用户、管理员、版主信息，它是一个基础数据表。其他表如主题与回复信息表中的 author_id 同 author 等都对应 bbsuser 表相关内容。表示该主题或回复是对应用户所发表。bbsuser 表包括用户编号 (id)、用户名 (username)、密码 (password)、昵称 (nickname)、性别 (sex)、出生年 (birthyear)、出生月份 (birthmonth)、出生日 (birthday)、电子信箱 (email)、博客 (weblog)、QQ 号码 (qq)、SKYPE 账号 (skype)、移动电话 (mobile)、签名 (signature)、注册时间 (reg_time)、最后登录时间 (last_time)、注册时的 IP (reg_ip)、最后登录时的 IP (last_ip)、用户简介 (info)、用户头像 URL (face)、是否在线 (online)、是否被锁定 (islock)、积分 (userscore)、用户权限 (userpower)、总发贴数 (totalpost)、级别 (usergrade)。其中用户编号为主索引、自动增量。username 为惟一索引字段。表 bbsuser 的结构如表 31.2 所示。

表 31.5 bbsuser 表的结构

字段名	列类型与长度	描述	备注
id	int(11)	用户编号	自动增量
username	varchar(20)	用户名	不允许为空，惟一索引
password	varchar(16)	密码	不允许为空
nickname	varchar(20)	昵称	不允许为空
sex	varchar(3)	性别	
birthyear	varchar(100)	出生年	
birthmonth	varchar(4)	出生月	
birthday	varchar(2)	出生日	
email	varchar(2)	电子邮件	
weblog	varchar(25)	博客地址	
qq	varchar(9)	QQ账号	
skype	varchar(16)	Skype账号	
mobile	varchar(13)	移动电话	
signature	tinytext	用户签名	
reg_time	datetime	注册时间	
last_time	datetime	最后一次登录时间	
reg_ip	varchar(16)	注册IP	
last_ip	varchar(16)	最后一次登录IP	
info	tinytext	用户简介	
face	varchar(30)	用户头像URL	
online	int(1)	是否在线	默认值为0
islock	int(1)	是否被锁定	默认值为0
userscore	int(6)	用户积分	默认值为0
userpower	int(2)	用户权限	默认值为0
totalpost	int(6)	总发贴数量	默认值为0
usergrade	varchar(12)	用户级别	默认值为'user'

建立表 bbsuser 的 SQL 批处理代码如下。

```
CREATE TABLE `bbsuser` (
  `id` int(11) NOT NULL auto_increment,
  `username` varchar(20) NOT NULL default "",
  `password` varchar(16) NOT NULL default "",
  `nickname` varchar(20) NOT NULL default "",
  `sex` varchar(3) default NULL,
  `birthyear` varchar(4) default NULL,
  `birthmonth` varchar(2) default NULL,
  `birthday` varchar(2) default NULL,
  `email` varchar(20) default NULL,
  `weblog` varchar(25) default NULL,
  `qq` varchar(11) default NULL,
  `skype` varchar(16) default NULL,
  `mobile` varchar(16) default NULL,
  `signature` tinytext,
  `reg_time` datetime default NULL,
  `last_time` datetime default NULL,
  `reg_ip` varchar(16) default NULL,
  `last_ip` varchar(16) default NULL,
  `info` tinytext,
  `face` varchar(30) default NULL,
  `online` int(1) default '0',
  `islock` int(1) default '0',
```

```

`userscore` int(6) default '0',
`userpower` int(2) default '0',
`totalpost` int(8) default '0',
`usergrade` varchar(12) default 'user',
PRIMARY KEY (`id`),
UNIQUE KEY `name_index` (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=gbk;

```

为使系统能够正常演示运行并可以添加论坛，表 `bbsuser` 中应首先有一条管理员记录，这样管理员可以添加论坛分类和分论坛。图 31.6 是表 `bbsuser` 的示例数据。

select username,password,nickname,sex,birthyear,email,mobile,userpower FROM bbsuser							
username	password	nickname	sex	birthyear	email	mobile	userpower
caxton	a123456	caxton	男	1984	caxton@126.com	13658010799	12
kokoro	a123456	kaka	男	1984	136311890@qq.com	13658010799	0
richard	123456	rich	男	1986	rich@hotmail.com	13658010799	0

图 31.6 表 `bbsuser` 的记录示例

`bbsuser` 表中用于划分用户权限的字段是 `userpower`，普通用户和版主的 `userpower` 值为 0。管理员的 `userpower` 值大于 10。

31.1.3.5 数据库表关系图

本系统中，一些数据库表中所存储的字段内容，与其他表相关联。这种联系只是表中列的相互对应关系。各表字段之间的关系如图 31.7 所示。

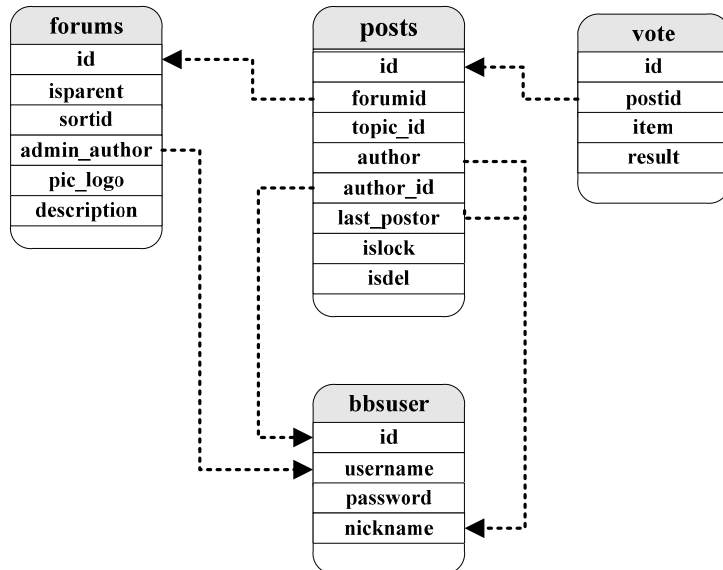


图 31.7 数据库表关系图

表 `forums` 表中的 `admin_author` 对应的是版主的用户名，其中用逗号隔开，图 31.7 只列出了本系统中四个表的部分字段，其中包括与全部其他表有对应关系的字段。

31.2 系统主界面与浏览论坛

BBS 系统主界面是用户进入系统浏览后的默认首页，在用户没有登录的情况下，会显示登录的表单入口。主界面中列出所有的分类与其中的分论坛，系统首页如图 31.8 所示。

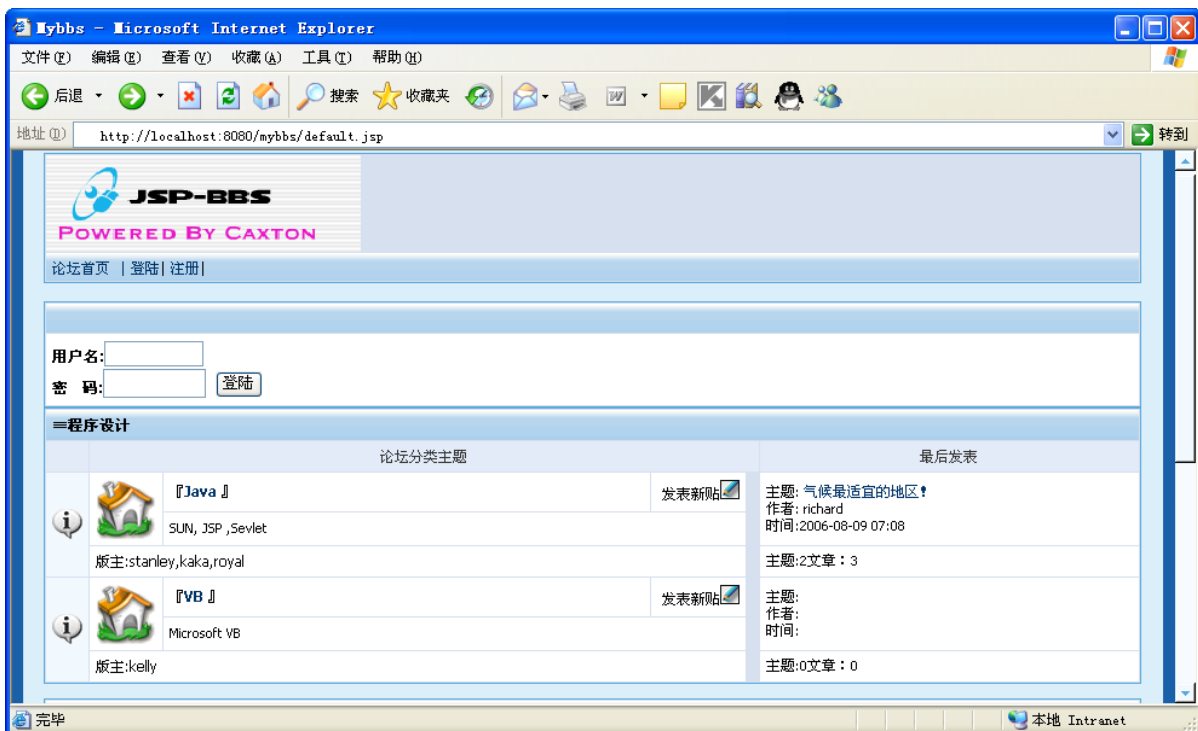


图 31.8 系统首页

上面的系统首页中，顶端的 Logo 区域以及 Logo 下方的导航条，以及图 31.8 看不到的网页下端，都是本系统模板页面有组成部分。中间的登录区域，显示论坛列表的区域都是加载从数据库中读取论坛信息并格式化输出的 Servlet 来实现的。

31.2.1 首页与系统全局模板页面与出错页面

系统全局加载页面主要用于构建系统所有视图页面的显示框架，包括要加载一个顶端 Logo 与导航条，以及底部的论坛版权、论坛标注显示区域。其他 BBS 系统页面通过利用这个模板加载要显示的视图来构建，如在系统首页，显示分论坛列表的是由 listform 这个 Servlet 读取数据库中的论坛信息，从而格式化输出的。

31.2.1.1 框架页面 default.jsp

default.jsp 是由系统全局模板页面加载系统样式表、用户登录区与论坛列表区构建的。

```
<%@ page contentType="text/html; charset=GBK" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
```

```

<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!--*****-->
<%
//如果用户没有登录，显示登录界面
if(session.getAttribute("login")==null)
{ %>
<jsp:include page="include/toplog.jsp"/>
<%} %>
<jsp:include page="listforum" />
<!--*****-->
<BR>
<CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/bottom.jsp"/>
</BODY></HTML>

```

上面的代码中除“*”中间的代码是首页要显示的主要内容外，其他部分都是要在其所页面中显示的框架模板。在首页中判断用户是否已登录，如已登录，不再显示用户登录区，否则加载用户登录区。要加载的顶端 Logo 区域、导航条以及下面的论坛快速转向下拉框的代码如下。

31.2.1.2 Logo 区域 toplogo.jsp

toplogo.jsp 构建顶端 Logo 区域，同样能过加载 Servlet 输出内容构建导航条，toplogo.jsp 位于系统目录下面的 include 文件夹下面。

```

<TABLE width="98%" align=center cellPadding=0
cellSpacing=1 class=i_table style="TABLE-LAYOUT: fixed; WORD-WRAP: break-word">
<TR>
<TD height="80" vAlign=top ></TD>
</TR>
<jsp:include page="/tophead"/>
</TABLE>

```

模板顶端区域加载的导航条是由 TopHead.java 这个 Servlet 动态构建的，管理员的导航条内容与其他用户导航条内容的不同在于管理员多一个“后台管理”的操作入口。图 31.8 所示的首页中是没有登录的情况下的导航条。TopHead.java 的代码如下所示。

```

package mybbs;
import java.io.*;
import java.sql.*;
import java.io.IOException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class TopHead extends HttpServlet {
    PrintWriter out;
    ResultSet rs;
    String action,userid;
    public void doPost(HttpServletRequest request,
                        HttpServletResponse response)
        throws IOException
    {

```

```
response.setContentType("text/html;charset=gbk");
out = response.getWriter();
HttpSession session = request.getSession();
```

- ❑ 判断用户是否已登录，进一步获取用户权限，显示登录后的导航条。

```
if(session.getAttribute("login")!=null){
    int power=0;
    if (session.getAttribute("userpower") != null) {
        power = Integer.parseInt( (String) session.getAttribute("userpower"));
    }
    out.print("<TR><TD height='23' vAlign=top background='images/head.gif' >");
    out.print("<A href='default.jsp'>论坛首页        </A>");
    out.print((String)session.getAttribute("nickname")+"&nbsp;&nbsp;  </A>");
    out.print("<A href='login.do'>重登陆</A>&nbsp;&nbsp; </A>");
    out.print("<A href='user_info.do'>查看资料</A>&nbsp;&nbsp;</A>");
    out.print("<A href='modifyinfo.do'>修改资料</A>&nbsp;&nbsp;</A>");
    out.print("<A href='modifypass.do'>修改密码</A>&nbsp;&nbsp;</A>");
    out.print("<A href='search.do?nickname="+session.getAttribute("nickname")
        +"'>搜索</A>&nbsp;&nbsp;</A>");
    out.print("<A href='Login.do?action=out'>退出</A>&nbsp;&nbsp;</A>");
```

- ❑ 如果权限值大于 10，说明是管理员，显示“后台管理”入口链接。

```
if(power>10){
    out.print("<A href='listbbs.do'>后台管理</A>&nbsp;&nbsp;</A>");
}
out.print("</TD></TR>");
}
else{
```

- ❑ 如果没有登录，则输出以下内容。

```
out.print("<TR><TD height='23' vAlign=top background='images/head.gif' >");
out.print("<A href='default.jsp'>论坛首页        </A>");
out.print("<A href='login.do'>登陆</A>&nbsp;&nbsp;</A>");
out.print("<A href='regist.do'>注册</A>&nbsp;&nbsp;</A>");
out.print("</TD></TR>");
}
}
public void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws IOException
{
    doPost(request, response);
}
}
```

除了上面的导航条、Logo 区域以外，模板页面还要加载论坛风格样式表以及下端区域代码，由于这些内容与 JSP 技术无太大关联，这里不再列出只列出构成下拉论坛转向列表的代码。

```
<SELECT name="forumsid" onchange="Fjump(this.options[this.selectedIndex].value)">
<option value="" selected>快速跳至其他论坛
<jsp:useBean id="bbs" scope="request" class="mybbs.ForumInfo" />
<jsp:getProperty name="bbs" property="allbbs"/>
</SELECT>
<script language="JavaScript">
```

```
function Fjump(value)
{
    if(value!="") {
        window.location=('list-topic.do?forumsid='+value);
    }
}
</script>
```

系统出错或异常将转发到出错页面 `global.jsp`，其中可以获取 Struts 的 `ActionErrors` 封装的错误返回信息以及利用 `request` 保存的出错或异常信息。它也是模板页面加载错误页面错误显示代码构成。其除模板以外的主要代码如下。

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<META HTTP-EQUIV="REFRESH" CONTENT="3;url=javascript:history.back();">
<html:errors/>
<LI><FONT color=red><%=request.getAttribute("throw")%></FONT>
```

31.2.2 显示论坛分类列表

在用户进入的首页中要显示论坛内容列表，主些分论坛列表信息由 `list_forum.java` 这个 Servlet 读取数据表 `forums` 中的内容，格式化输出，在 `default.jsp` 中加载其处理的结果，在首页构建论坛列表的视图。`list_forum.java` 代码如下。

```
package mybbs;
import java.io.*;
import java.sql.*;
import java.io.IOException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class list_forum extends HttpServlet {
    PrintWriter out;
    ResultSet rs, rs1;
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws IOException {
        response.setContentType("text/html; charset=gbk");
        out = response.getWriter();
        try {
```

□ 建立同数据库的连接，检索 `forums` 表中的论坛分类，并显示分类的名称。

```
DBConn conn = new DBConn();
String sql = "SELECT * FROM forums WHERE isparent=1";
rs = conn.query(sql);
while (rs.next()) {
    out.print("<TABLE class=i_table cellSpacing=1 cellPadding=5 width='98%' align=center><TBODY>");
    out.print("<TR><TD class=head align=left colSpan=6 background='images/head.gif'>");
    out.print("<B><A class=cfont href='http://bbs'>");
```

```

out.print("≡" + rs.getString("forum_name") + "</A></B>&nbsp; </TD></TR>");
out.print("<TR class=cbg align=middle height=23>");
out.print("<TD width=4%></TD><TD colspan=4>论坛分类主题</TD>");
out.print("<TD width=35%>最后发表</TD></TR>");

```

- ❑ 检索 forums 表中同一个分类中的分论坛，格式化显示其信息列表。

```

String sql2="SELECT * FROM forums WHERE isparent=0 AND sortid="+rs.getString("sortid");
rs1 = conn.query(sql2);
while (rs1.next()){
    out.print("<TR> <TD rowspan='3' align=middle vAlign=center class=f_two>");
    out.print("<A href='http://bbs' target=_blank><IMG src='images/old.gif'></A></TD>");
    out.print("<TD width=6% rowspan=2 align=left >");
    out.print("<IMG src='" + rs1.getString("pic_logo") + "' /></TD>");
    out.print("<TD width=45% align=left class=f_one onmouseover=this.className='f_two'");
    out.print(" onmouseout=this.className='f_one'>");
    out.print("『<A class=fnamecolor href='list-topic.do?forumsid=" +
        rs1.getString("id") + "'><B>"+ rs1.getString("forum_name") +
        "</B></A>』<BR></TD>");
    out.print("<TD align=right class=f_one onmouseover=this.className='f_two' ");
    out.print("onmouseout=this.className='f_one'>");
    out.print("<p align='right'>发表新贴<A href='post.do?action=1&forumsid=" +
        rs1.getString("id") +
        "'><IMG alt=发表新贴 src='images/icon_12.gif' /></A></P></TD>");
    out.print("<TD width=1' rowspan='3' align=middle></TD>");
}

```

- ❑ 调用 ForumInfo 中的方法，显示分论坛中最后发表的主题信息。

```

ForumInfo fI = new ForumInfo();
out.print(fI.getLastTopic(rs1.getInt("id")));
out.print("</TR><TR><TD colspan=2' align=left >");
out.print(rs1.getString("description")+"</TD></TR>");
out.print("<TR><TD colspan=3' align=left >");
out.print("版主:" + rs1.getString("admin_author") + "</TD>");
out.print("<TD bgColor=#ffffff class=smalltxt>主题:" +
    rs1.getString("total_topic");
out.print("文章: " + rs1.getString("total_post") + "</TD></TR> ");
}
rs1.close();
out.print("</TBODY></TABLE><TABLE><TBODY><TR>");
out.print("<TD height=5 width=98%></TD><TR></TBODY></TABLE>");
}
rs.close();
conn.DBclose();
}

```

- ❑ 异常处理，返回异常信息到出错转向页面。

```

catch (Exception e) {
    request.setAttribute("throw", e.toString());
    response.sendRedirect("global.do");
}
}
public void doGet(HttpServletRequest request,
    HttpServletResponse response) throws IOException {

```

```

        doPost(request, response);
    }
}

```

在 Servlet 中要调用 JavaBean 类 ForumInfo 中的方法 getLastTopic() 获取分论坛最后一个主题的相关信息并输出，另外 ForumInfo.java 中还有一个以后在输出分论坛主题时用到。

ForumInfo.java 代码如下。

```

package mybbs;
import java.sql.*;
public class ForumInfo {
    private int forumid;
    private String forum_name;
    public ForumInfo(){ }
}

```

❑ 设置论坛编号的值，用于获取对应的论坛其他信息。

```

public void setForumid(int fid){
    this.forumid=fid;
}

```

❑ 获取分论坛的名称。

```

public String getForumname()
{
    try{
        DBConn con = new DBConn();
        String sql = "SELECT * FROM forums WHERE id = "+forumid;
        ResultSet rs = con.query(sql);
        if(rs.next()){
            forum_name = rs.getString("forum_name");
            rs.close();
            con.DBclose();
        }
        else{
            forum_name = "没找到";
        }
    }
    catch(Exception e){
        return e.toString();
    }
    return forum_name;
}

```

❑ 获取分论坛中最后一个主题的信息并格式化输出。

```

public StringBuffer getLastTopic(int id) throws Exception
{
    StringBuffer buffer=new StringBuffer();
    DBConn conn =new DBConn();
    String sql1="SELECT topic_id,title,author,post_time FROM posts WHERE istopic>0 AND forumid="+id
        +" ORDER BY post_time ASC";
    try{
        ResultSet rs2 = conn.query(sql1);
        if(rs2.last()){
            buffer.append("<TD rowspan='2' bgColor=#ffffff class=smalltxt>主题: ");
            buffer.append("<A href='disbbs.do?forumsid=" + id + "&topicid=" +

```



```

        rs2.getString("topic_id") + ">");
        buffer.append(rs2.getString("title") + "</A><BR>作者: ");
        buffer.append(rs2.getString("author") + "<BR>时间:" +
            rs2.getString("post_time").substring(0, 16) + "</TD>");
    }

```

❑ 如果没有记录，也就是论坛中没有主题，则输出以下内容以保证页面正常不变形。

```

        else{
            buffer.append("<TD rowspan='2' bgColor=#ffffff class=smalldtxt>主题: ");
            buffer.append("<A href='disbbs.do?forumsid=id'>");
            buffer.append("</A><BR>作者: ");
            buffer.append("<BR>时间:</TD>");
        }
        rs2.close();
    }
    catch(Exception e){
        System.out.print(e.toString());
    }
    conn.DBclose();
    return buffer;
}

```

❑ 获取全部分类中的分论坛，用于以下拉框的方式选择分论坛，进行搜索或直接进入该论坛。

```

public StringBuffer getAllbbs(){
    StringBuffer buffer=new StringBuffer();
    try{
        DBConn conn =new DBConn();
        ResultSet rs=conn.query("SELECT * FROM forums WHERE isparent=0");
        while(rs.next()){
            buffer.append("<option value='"+rs.getString("id")+"'>" +rs.getString("forum_name"));
        }
        rs.close();
        conn.DBclose();
    }
    catch(Exception e){
        buffer.append("数据库操作出错！");
    }
    return buffer;
}
}

```

31.2.3 显示论坛主题列表

浏览论坛主题时，要将该分论坛中的所有未锁定的主题以分页的形式列表显示。通过在模板页面加载列表显示分论坛中话题的方式，构建用户浏览分论坛主题的视图。

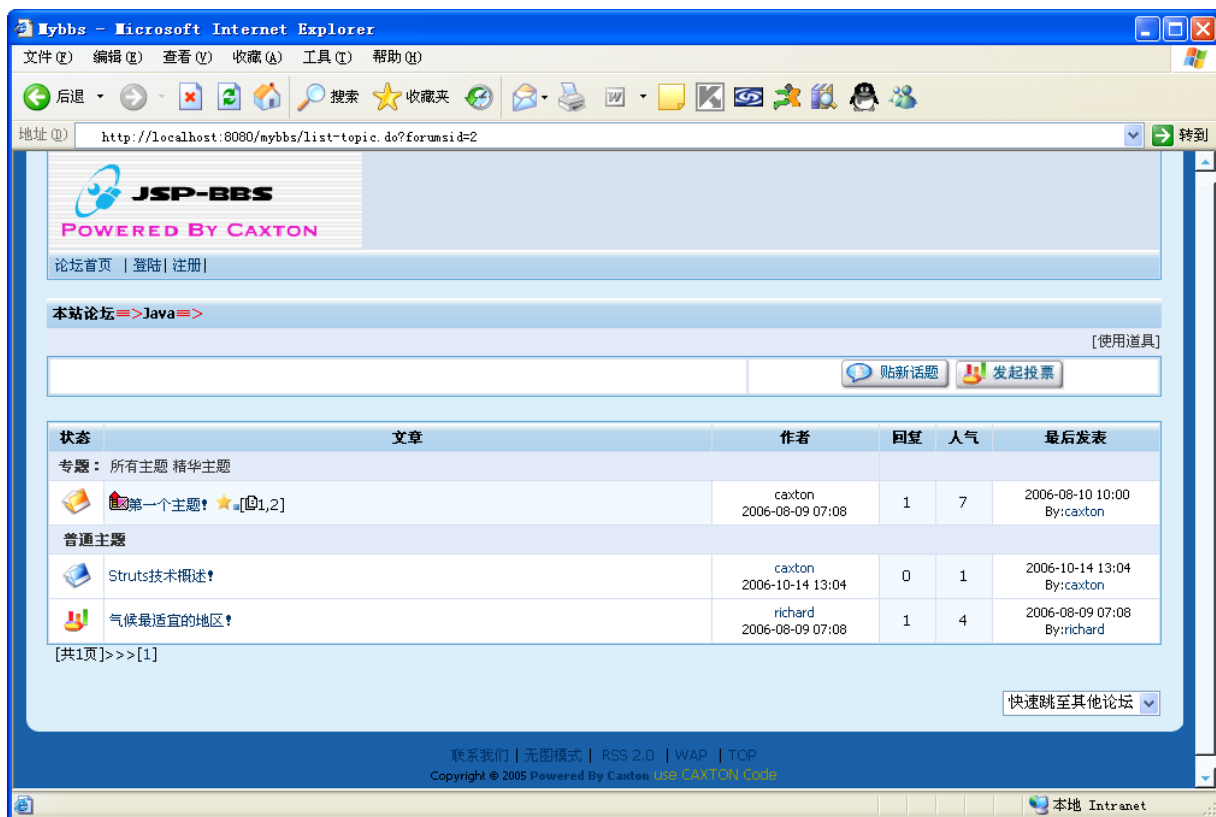


图 31.9 论坛主题列表

list_topic.jsp 利用模板页面加载论坛主题列表的方式构建用户浏览分论坛主题的视图。另外在主题列表的上部要构建一个论坛导航以及发表主题与发起投票的链接。list_topic.jsp 代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
  <TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
    <TBODY>
      <TR><TD>
        <jsp:include page="/include/toplogo.jsp"/>
        <BR>
        <!-- ***** -->
        <%
          String forumid=request.getParameter("forumsid");
        %>
        <jsp:include page="list">
          <jsp:param name="forumsid" value="<%=forumid%>" />
        </jsp:include>
        <!-- ***** -->
        <BR> <CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
        <jsp:include page="include/buttom.jsp"/>
      </BODY>
    </HTML>
```

上面的 list_topic.jsp 要加载的 URL 为“list”的 Servlet 就是 list_topics.java, 它通过调用 ShowTopic.java 是的方法, 获取分论的主题信息并格式化输出。在 list_topics.java 中只是在列表的顶端显示一个导航链接, 同时实现分页的处理逻辑。

list_topics.java 的代码如下。

```
package mybbs;
import java.io.*;
import java.sql.*;
import java.io.IOException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class list_topics extends HttpServlet {
    PrintWriter out;
    ResultSet rs;
    int page=1;
    String forumid,inte;
    public void doPost(HttpServletRequest request,
                        HttpServletResponse response)
        throws IOException
    {
        response.setContentType("text/html;charset=gbk");
        out = response.getWriter();
```

- ❑ 获取当前要访问分页的序号, 进行非空值处理, 直接进入显示第一页。

```
        inte=request.getParameter("page");
        if(inte==null){
            page=1;
        }
        else{
            page=Integer.parseInt(inte);
        }
```

- ❑ 获取访问参数值, 用于确定要浏览的分论坛。获取主题列表分页时每页显示的主题数目。

```
        String fid=request.getParameter("forumsid");
        int forumid=Integer.parseInt(fid);
        int pagesize=params.Topic_Page_Size;
        try{
```

- ❑ 实例化一个 FormInfo 对象, 主要是为了用 forumid 为参数获取对应的分论坛名。以下代码是为了构建主题列表上面的论坛导航, 以及发帖、发起投票链接区域。

```
        ForumInfo fi=new ForumInfo();
        fi.setForumid(forumid);
        String forumname = fi.getForumname();
        out.print("<TABLE width='98%' align=center cellPadding=1 background='images/head.gif'>");
        out.print(" <TBODY> <TR><TD width='69%' align=left background='images/8.gif' class=head>");
        out.print("<B><A href=default.jsp>本站论坛</A><FONT color=red>≡</FONT>"+forumname);
        out.print("<FONT color=red>≡</FONT></B></TD>");
        out.print("<TD width='31%' align=right background='images/header-01.jpg' class=head>");
        out.print(" </TD></TR><TR background='images/head.gif'><TD height=23 colSpan=2 >");
        out.print(" [<A href='#>使用道具</A>]</TD></TR></TBODY></TABLE>");
        out.print("<TABLE width='98%' align=center>");
```

```

out.print("<TBODY><TR>");
out.print(" <TD width='63%' bgColor=#ffffff height='100%'>&nbsp;  </TD>");
out.print("<TD vAlign=top width='37%' bgColor=#ffffff height='100%'><div align='center'>");
out.print("<A href='post.do?action=1&forumsid="+forumid+"'><IMG src='images/post.gif'></A>");
out.print("<A href='post.do?action=3&forumsid="+forumid+"'><IMG src='images/votting.gif'></A> ");
out.print("</div></TD></TR></TBODY></TABLE><TABLE width='99%' align=center><TBODY>");
out.print("<TR><TD height=1></TD></TR></TBODY></TABLE>");

```

- ❑ 实例化一个 ShowTopic 对象，调用其中的方法获取 BBS 的公告主题与普通主题列表。

[illegible]

- ❑ 显示所有的分页链接，用于访问其他分页。

```
for(int j=1;j<=pagecount;j++)
{
    if(j==page){
        out.print("[");
        out.print("<A href=list-topic.do?forumsid="+forumid+"&page="+j+">");
        out.print(""+j);
        out.print("</A>");
        if(j==page){
            out.print("]");
        }
        out.print("&nbsp; ");
    }
}
catch(Exception e)
{
}
```

- ❑ 将异常返回给系统出错转向页面。

```
        request.setAttribute("throw",e.toString());
        response.sendRedirect("global.do");
    }
}

public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws IOException
{
    doPost(request, response);
}
}
```

显示分论坛中话题列表的主要处理程序是 `ShowTopic.java`，其中定义了获取并格式化显示 BBS 系统公告，普通主题的方法。其中在格式化显示论坛主题时，会根据主题的类型、是否被置顶、是否为精华贴等分别显示不同的图标。

ShowTopic.java 代码如下。

```
package mybbs;  
import java.sql.*;  
import java.util.*;
```

```
public class ShowTopic {
    private ResultSet rs=null;
    private int pagecount=0;
    DBConn conn=new DBConn();
    public ShowTopic(){}
```

- ❑ 获取并输出公共主题信息列表。定义一条查询公告的 SQL 语句，检索 groupid 等于 2 的主题，将所有的公告输出。在列表中主题的排序方式是以 istop 的值为优先，其次按照最后回复时间由近到远排序。

```
public StringBuffer ListPublicPost() throws Exception {
    StringBuffer buffer=new StringBuffer();
    String sqln="SELECT * FROM posts WHERE isdel=0 AND groupid=2 AND"+
        " istopic>0 ORDER BY (posts.istop+1)*posts.last_time DESC";
    rs=conn.query(sqln);
    /*-----*/
    buffer.append("<TABLE class=i_table id=ajaxtable ");
    buffer.append("cellSpacing=1 cellPadding=4 width='98%' align=center>");
    buffer.append("<TBODY><TR class=head align=middle><TD width='5%'>状态</TD>");
    buffer.append("<TD align=middle width='55%'>文章</TD>");
    buffer.append("<TD width='15%'>作者</TD><TD width='5%'>回复</TD><TD width='5%'>人气</TD>");
    buffer.append("<TD width='15%'>最后发表</TD></TR><TR ><TD colSpan=3><B>专题: </B> ");
    buffer.append("<A href='http://bbs'>所有主题</A> ");
    buffer.append("<A href='http://bbs'>精华主题</A>");
    buffer.append("</TD> <TD vAlign=center align=right colSpan=3></TD></TR>");
```

- ❑ 遍历所有公告，列表显示其标题、点击次数、回复、发表时间与作者、最后回复的作者与最后回复时间并格式化输出。

```
while(rs.next()){
    buffer.append("<TR class=t_two align=middle><TD><A title=打开新窗口 href='http://bbs'");
    buffer.append(" target=_blank><IMG src='images/topichot.gif' border=0></A></TD>");
    buffer.append("<TD class=t_one id= onmouseover=this.className='t_two' ");
    buffer.append("style=PADDING-LEFT: 8px onmouseout=this.className='t_one' align=left>");
    buffer.append("<IMG alt=置顶帖标志 src='images/headtopic_3.gif'>");
    buffer.append("<A href='disbbs.do?forumsid="+rs.getString("forumid")");
    buffer.append("&topicid="+rs.getString("topic_id")+"'">"+rs.getString("title")+"</A>");
    buffer.append("<IMG alt=精华帖标志 src='images/digest_1.gif'>["");
    buffer.append("<IMG src='images/multipage.gif' border=0>1,2]<DIV></DIV></TD>");
    buffer.append("<TD class=smalltxt>"+rs.getString("author")+"");
    buffer.append("<BR>"+rs.getString("post_time").substring(0,16));
    buffer.append("</TD><TD class=t_one>"+rs.getString("reply_times")+"</TD>");
    buffer.append("<TD class=t_one>"+rs.getString("click_times")+"</TD><TD class=smalltxt>&nbsp;");
    buffer.append(""+rs.getString("last_time").substring(0,16)+"");
    buffer.append("<BR>By: <A href='user_info.do?userid="+rs.getString("author_id")+'>");
    buffer.append(rs.getString("last_postor")+"</A></TD></TR>");
}
buffer.append("<TR class=cbg><TD colSpan=6>&nbsp; <B>普通主题</B></TD></TR>");
rs.close();
return buffer;
}
```

- ❑ 获取并输出非公共主题信息列表。定义一条查询公告的 SQL 语句，检索 groupid 不等 2 的主题，将所有的非公告主题格式化输出，同样以与公告相同的排序方式列表。

```
public StringBuffer DisplayTopic(int forumid,int page)throws Exception{
    StringBuffer buffer2=new StringBuffer();
    String sql="SELECT * FROM posts WHERE isdel=0 AND groupid!=2 AND istopic>=1"+" AND forumid="+
        forumid+" ORDER BY (posts.istop+1)*posts.last_time DESC";
    try{
        rs = conn.query(sql);
```

- ❑ 获取主题的总量，用于计算分页数量。

```
rs.last();
int lastrow = rs.getRow();
int pagesize = params.Post_Page_Size;
//计算分页后的总页数
pagecount = (lastrow % pagesize == 0) ? (lastrow / pagesize) :
    (lastrow / pagesize + 1);
//当前的社始页数
if (page >= pagecount) {
    page = pagecount;
}
int position = (page - 1) * pagesize;
if (position == 0) {
    position = 1;
}
```

- ❑ 将记录指针指向当前分页的第一条记录，然后依次读取主题信息并列表输出。

```
rs.absolute(position);
for (int i = 1; i <= pagesize; i++) {
    buffer2.append("<TR class=t_two align=middle>");
```

- ❑ 如果 istopic 值为 2 显示为投票主题。

```
if (rs.getInt("istopic") == 2) {
    buffer2.append(" <TD><A title=投票 ><IMG src='images/vote.gif' border=0></A></TD>");
}
```

- ❑ 如果 groupid 值为 1 显示为精华主题。

```
if (rs.getInt("groupid") == 1&&rs.getInt("istopic") != 2) {
    buffer2.append(" <TD><A title=精华 ><IMG src=images/isbest.gif border=0></A></TD>");
}
```

- ❑ 其他情况显示为普通主题。

```
if ( (rs.getInt("groupid") != 1) && (rs.getInt("istopic") != 2)) {
    buffer2.append(" <TD><A title=普通帖子 ><IMG src=images/topicnew.gif border=0></A></TD>");
}
buffer2.append("<TD align=left>");
```

- ❑ 如果 islock 值为 1 显示为已锁定的不能回复的主题。

```
if (rs.getInt("islock") == 1) {
    buffer2.append("<A title=不能回复 >");
    buffer2.append("<IMG src=images/locked.gif border=0></A>");
}
```

- ❑ 如果 istop 值大于 0，说明进行过置顶操作，显示为置顶主题。

```
if (rs.getInt("istop") > 0)
    buffer2.append("<IMG src=images/uping.gif >");
buffer2.append("<A href='disbbs.do?forumsid=" + forumid +
    "&topicid=" + rs.getString("topic_id") + ">");
buffer2.append(rs.getString("title") + "</A></TD>");
```

```

buffer2.append("<TD class=smalltxt><A href='user_info.do?userid=" +
    rs.getString("author_id") + "'>" + rs.getString("author") + "</A>");
buffer2.append("<BR>" + rs.getString("post_time").substring(0, 16) + "</TD>");
buffer2.append("<TD class=t_one>" + rs.getString("reply_times") + "</TD>");
buffer2.append("<TD class=t_one>" + rs.getString("click_times") + "</TD>");
buffer2.append("<TD class=smalltxt>" + rs.getString("last_time").substring(0, 16) + "");
buffer2.append("<BR>By:<FONT color=blue><A href='user_info.do?userid=" +
    rs.getString("author_id") + "'>" +
    rs.getString("last_postor") + "</A></FONT></TD></TR>");

if (!rs.next())
    break;
}
rs.close();
}
catch(Exception e){

```

❑ 捕获没有主题或操作异常。

```

    buffer2.append("暂时没有帖子！或操作出错。");
}
return buffer2;
}
public int getPageCount(){
    return pagecount;
}
}

```

在主题列表页面，可以单击顶端导航条的“登录”或“注册”进入登录或注册页面。如果要回到论坛列表，可以单击“本站论坛”回首页或从右下角的“快速跳至其他论坛”直接转向其他分论坛的主题列表。如果要浏览主题的具体内容如相关的回复，可以直接单击列表中的主题标题。

31.2.4 显示主题内容

用户单击论坛主题列表中的标题后，会进入查看主题内容同时查看相关回复的页面。在该页面中程序会根据用户权限在列表上方显示不同的工具条。其中版主和管理员会显示前台管理工具条，图 31.10 所示是未登录用户或普通用户视图。



图 31.10 浏览主题与相关回复

图 31.10 显示主题具体内容的视图也是由模板页面加载 Servlet 输出内容实现的, Servlet 调用 Bean 中的方法读取数据库中主题与相关回复信息, 并格式化输出。

disbbs.jsp 模板加载主题与回复内容列表构建显示主题内容视图, 代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY>
<TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!-- *****-->
<%
String forumid=request.getParameter("forumsid");
String topicid=request.getParameter("topicid");
%>
<jsp:include page="content">
<jsp:param name="forumsid" value="<%=forumid%>" />
<jsp:param name="topicid" value="<%=topicid%>" />
</jsp:include>
<!-- *****-->
```



```
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/bottom.jsp"/>
</BODY>
</HTML>
```

在 disbbs.jsp 中要加载的主题与回复内容列表是由 content.java 这个 Servlet 来组织的，它通过调用 ShowContent 中的方法，输出指定的主题与回复内容

content.java 代码如下。

```
package mybbs;
import java.io.*;
import java.util.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class content extends HttpServlet {
    PrintWriter out;
    String forumid, topic_id, page;
    public void doPost(HttpServletRequest request,
        HttpServletResponse response) throws IOException {
        response.setContentType("text/html;charset=gbk");
        out = response.getWriter();
        HttpSession session = request.getSession();
```

❑ 获取论坛编号、主题编、当前分页序号三个参数的值。

```
forumid = request.getParameter("forumsid");
topic_id = request.getParameter("topicid");
page=request.getParameter("page");
try {
```

❑ 新建一个 ShowContent 类的对象，调用 ListContent() 方法得到主题内容列表的结果集。

```
ShowContent SC=new ShowContent();
Vector VC=SC.ListContent(forumid,topic_id,session);
//将该贴子的点击次数加 1
```

❑ 新建一个 Operate 对象，将主题的点击量加 1。

```
Operate OP=new Operate();
OP.UpdateClick_time(forumid,topic_id);
```

❑ 利用 forumid 做为参数，调用 ForumInfo 的方法获取当前分论坛的名称。

```
int fid=Integer.parseInt(forumid);
ForumInfo fi=new ForumInfo();
fi.setForumid(fid);
String forumname = fi.getForumname();
out.print("<TABLE width='98%' align=center background='images/head.gif' >");
out.print(" <TBODY> <TR><TD width='69%' align=left background='images/8.gif' class=head>");
out.print("<B><A href='default.jsp'>本站论坛</A><FONT color=red>≡</FONT>");
out.print("<A href='list-topic.do?forumsid=" + forumid + "'>"+ forumname + "");
out.print("</A><FONT color=red>≡</FONT>"+SC.getTitle() + "</B></TD>");
out.print(" <TD width='31%' align=right background='images/header-01.jpg' class=head>");
out.print("</TD></TR>");
out.print("<TR><TD align=left class=cbg><FONT color=red>");
```

❑ 新建一个 AdminItem 对象，以 session 等做为参数，调用其中的方法构建前台管理的工具条。

```

AdminItem AI=new AdminItem();
StringBuffer buffer=
    AI.ShowAdminItem(session,SC.getPostid(),forumid,SC.getGroupid(),SC.getIslock());
out.print(buffer);
out.print("</TD></FONT>");
//发贴链接
out.print("<TD height=23 colSpan=2 align=right class=cbg>");
out.print("<A href='post.do?action=1&forumsid=" + forumid +
    "'><IMG src='images/post.gif'></A>");
out.print("&nbsp;<A href='post.do?action=3&forumsid=" +
    forumid + "'><IMG src='images/votting.gif'></A> ");
if (SC.getIslock()== 0) {
    out.print("&nbsp;<A href='post.do?action=2&forumsid=" + forumid +
        "&topicid=" + topic_id + "'><IMG src='images/reply.gif'></A>");
}
out.print("</TD></TR></TBODY></TABLE>");
out.print("<TABLE width='99%' align=center><TBODY>");
out.print("<TR><TD height=1></TD></TR></TBODY></TABLE>");

```

□ 如果是投票主题，显示投票内容结果，构建投票用的表单。

```

/*-----*/
if(SC.getIsTopic()==2){
    out.print("<TABLE width='98%' align=center><TBODY>");
    out.print("<FORM action='vote' method='post'>");
    DisVote disv=new DisVote();
    out.print(disv.ShowVote(String.valueOf(SC.getPostid())));
    //VoteAct vott=new VoteAct();
    out.print("<TR><TD><INPUT type='submit' value='投票' name='submit'></TD><TR>");
    out.print("<INPUT TYPE=hidden name=postid value='"+SC.getPostid()+">");
    out.print("<INPUT TYPE=hidden name=forumsid value='"+forumid+">");
    out.print("<INPUT TYPE=hidden name=topicid value='"+topic_id+">");
    out.print("</FORM>");
    out.print("</TBODY></TABLE>");
}
/*-----*/

```

□ 计算分页，获得当前记录指针应指向的初始位置与结束位置。

```

int pages=0;
if(page==null){
    pages=1;
}
else{
    pages=Integer.parseInt(page);
}
int lastrow=VC.size();
int pagesize=params.Post_Page_Size;
int pagecount=(lastrow%pagesize==0)?(lastrow/pagesize):(lastrow/pagesize+1);
//当前的初始页数
if (pages>=pagecount){
    pages=pagecount;
}
int position=(pages-1)*pagesize;

```



```

HttpSession session=null;
private int postid=0;
private int j=0;
private int istopic=1;
private int groupid=0;
private String title;

```

- 构造一个输出主题与回复内容的方法，以 session 和论坛、主题编者按号为参数。

```

public Vector ListContent(String forumid,String topicid,HttpSession session) throws Exception{
    this.session=session;
    Vector VCon=new Vector();
    DBConn conn=new DBConn();
    ResultSet rs=null;
    int power=0;
    String forum="";
    String userid="";

```

- 获取 session 中的用户权限、版主有权限的论坛编号，以及用户的编号。

```

if(session.getAttribute("userpower")!=null){
    power=Integer.parseInt((String)session.getAttribute("userpower"));
}
if(session.getAttribute("forum")!=null){
    forum=(String)session.getAttribute("forum");
}
if(session.getAttribute("userid")!=null){
    userid=(String)session.getAttribute("userid");
}

```

- 定义一条 SQL 语句用于检索指定的主题与回复信息，最后按照从近到远的交序排列 (? ? ? ?) 。

```

String sql="SELECT posts.*,bbsuser.* FROM posts,bbsuser WHERE posts.isdel=0 AND posts.forumid="+
        +forumid+" AND posts.topic_id="+topicid
        +" AND posts.author_id=bbsuser.id ORDER BY posts.id ASC";
rs=conn.query(sql);
int i=1;
while(rs.next())
{
    StringBuffer buffer=new StringBuffer();

```

- 获取主题的一些属性信息，以便在 content 中调用相对应的 getter 方法获取属性值。

```

if (rs.getInt("istopic") > 0) {
    j = rs.getInt("posts.islock");
    title=rs.getString("title");
    postid = rs.getInt("posts.id");
    groupid = rs.getInt("posts.groupid");
    istopic=rs.getInt("istopic");
}
/*-----*/
buffer.append("<TABLE align=center cellSpacing=0 cellPadding=1 width=\"98%\" border=0>");
buffer.append("<TBODY><TR>");
if(i==1)
    buffer.append("<TD width=\"18%\" background=images/part1_38.gif height=28>");
else

```

□ 如果该作者当前在线，则将作者昵称显示为红色，否则显示为黑色。

[illegible]

```
if(i==1)
    buffer.append(" <TD width=\"58%\" background=images/part1_38.gif>");
else
    buffer.append(" <TD width=\"58%\" bgcolor=\"#f4f8e0\">");
if(i==1){
    buffer.append("<FONT color=red>[楼主]</FONT>");
}
else{
    buffer.append("<FONT color=red>["+i+"楼]</FONT>");
}
buffer.append("&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&主题： "+rs.getString("title")+"</TD>");
if(i==1)
    buffer.append("<TD width=\"20%\" background=images/part1_38.gif>");
else
    buffer.append("<TD width=\"20%\" bgcolor=\"#f4f8e0\">");
buffer.append("<B>Posted:</B>" + rs.getString("post_time").substring(0,16));
buffer.append("</TD></TR></TBODY></TABLE>");
```

```
buffer.append("<TABLE align=center cellSpacing=0 cellPadding=1 width=\"98%\" border=0>");
buffer.append("<TBODY><TR>");
buffer.append(" <TD vAlign=top width=\"18%\" bgColor=#f4d8f2 rowSpan=2>");
buffer.append("&nbsp;&nbsp;&nbsp;<IMG width=130 height=130 src=\""+rs.getString("face")+"\" />");
buffer.append("<BR>&nbsp;&nbsp;&nbsp;等级: "+rs.getString("usergrade"));
buffer.append("<BR>&nbsp;&nbsp;&nbsp;文章: "+rs.getString("totalpost"));
buffer.append("<BR>&nbsp;&nbsp;&nbsp;积分: "+rs.getString("userscore"));
buffer.append("<BR>&nbsp;&nbsp;&nbsp;加入时间: "+rs.getString("reg_time").substring(0,10));
buffer.append("<BR>&nbsp;&nbsp;&nbsp;最后登录: "+rs.getString("bbsuser.last_time").substring(0,10));
buffer.append("<BR></TD>");
```

☐ 显示发贴子内容、作者签名等信息。

```
buffer.append("<TABLE width=90% cellSpacing=0 cellPadding=0 border=0>");
buffer.append("<TBODY> <TR><TD width=14><IMG height=8 src='images/top_l.gif' width=14></TD>");
buffer.append("<TD background='images/top_c.gif'></TD>");
buffer.append("<TD width=16><IMG height=8 src='images/top_r.gif' width=16></TD>");
buffer.append("</TR>");
buffer.append("<TR><TD vAlign=top width=14 background='images/center_l.gif'></TD>");
buffer.append("<TD style='LINE-HEIGHT: 14pt' bgColor=#ffffff>");
```

❑ 如果用户有足够权限，则显示作者的 IP 信息。

- 显示与作者相关的图标与链接。

□ 如果是版主或是管理员，则显示一个删除该贴的图标与链接。

- ❑ 如果用户为管理员，或该贴是当前用户所发，显示“编辑”图标与链接。

```
if(power>10||userid.equals(rs.getString("bbsuser.id")))
{
    buffer.append("<A href='post.do?postid=" + rs.getString("posts.id")+"&action=4">");
    buffer.append("<IMG alt=编辑这个帖子  src='images/edit.gif' ></A>&nbsp;&nbsp; ");
}
buffer.append("<A href='javascript:scroll(0,0)'><IMG alt=顶端  src='images/top.gif'></A>");
```

```
        buffer.append("</TD></TR></TBODY></TABLE>");
        VCon.add(buffer);
        buffer=null;
        i++;
    /*-----*/
    }
    rs.close();
    return VCon;
}
```

□ 获取当前主题在 posts 表中的记录编号，类型等，用于对其进行管理。

```
public int getPostid(){
    return postid;
}
public int getGroupid(){
    return groupid;
}
public int getIstopic(){
    return istopic;
}
public int getIslock(){
    return j;
}
public String getTitle(){
    return title;
}
}
```

查看主题与相关回复页面中，登录用户可以点上面三个图标进行“发新主题”、“发起投票”、“回复话题”。每一个主题或回复都显示了作者的一些信息，在作者签名的下方有 5 个图标。“资料”是查看该用户信息的链接；“主页”是用户博客户主页的链接；“邮件”和“skype”分别对应用户的电子信箱与 skype 账号；“搜索”是搜索该用户所发所有贴子的链接。

31.2.5 搜索论坛内容

在前一节的显示主题和回复内容页面中，有一个“搜索”的链接。它实际上是以用户昵称为关键词，搜索该用户的所有发贴。BBS 系统为用户提供了一个可以用贴子标题、内容、作者、所在分论坛做为限制条件的准确和模糊搜索方式，用户界面如图 31.11 所示。

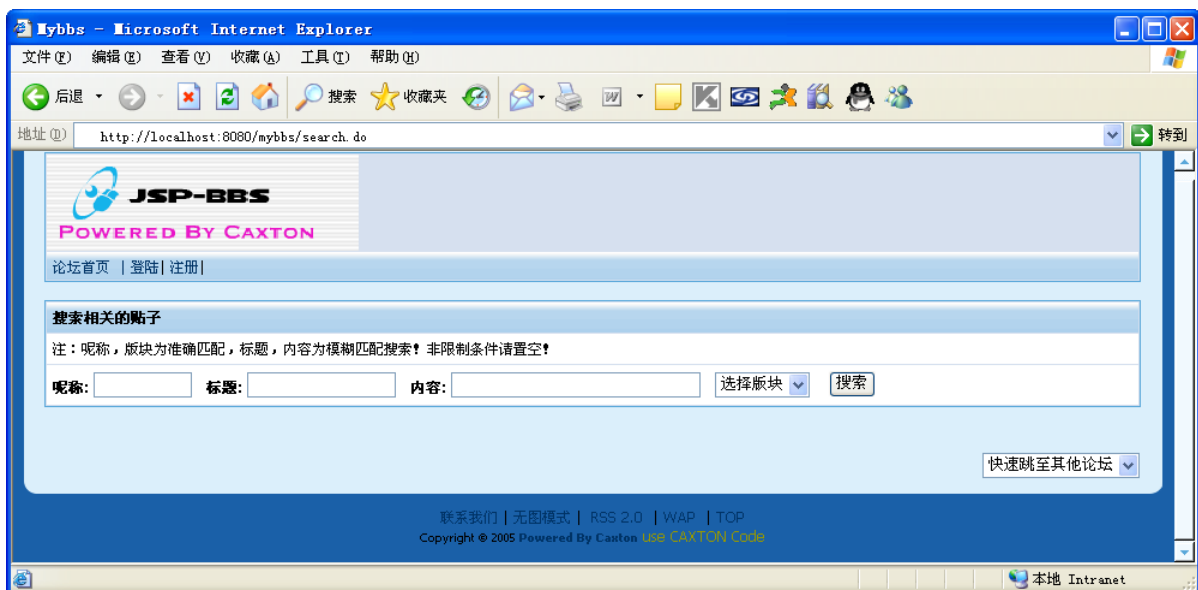


图 31.11 用户搜索界面

图 31.11 中，构建搜索提交表单的是 include 目录下的 searchload.jsp，其中包含了生成搜索表单的代码，还要通过 ForumInfo 的方法获取所有论坛下拉列表，用于选择搜索范围。

searchload.jsp 代码如下。

[illegible]

图 31.11 所示的页面是未提交参数的情况下，直接发出 `search.do` 请求，这时全部参数为空，此时

JavaScript 脚本会运行并弹出警告。运行结果是页面中只包含一个搜索表单，没有搜索结果。如果用户通过表单提交搜索限制条件，同样还是由 search 页面接收这些参数并将这些参数的值传递能处理搜索的 Servlet，最后加载其处理结果。

search.jsp 代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!-- *****-->
<jsp:include page="include/searchload.jsp"/>
<%
```

□ 获取所有参数的值，如果没有参数值，则将该值置空。

```
String forumsid=request.getParameter("forumsid");
String nickname=request.getParameter("nickname");
String keyword=request.getParameter("keyword");
String keyword2=request.getParameter("keyword2");
if(request.getParameter("nickname")==null){
    nickname="";
}
if(request.getParameter("keyword")==null){
    keyword="";
}
if(request.getParameter("keyword2")==null){
    keyword2="";
}
if(request.getParameter("forumsid")==null){
    forumsid="";
}
```

□ 如果三个条件参数都为空，则弹出警告。

```
if(nickname.equals("")&&keyword.equals("")&&keyword2.equals(""))
{
    <script language="JavaScript">
        window.alert('输入全为空，不能进行检索！');
    </script>
}
else{
    %>
```

□ 将参数值传给 searchresult 处理，最终加载其返回的结果。

```
<jsp:include page="searchresult">
    <jsp:param name="forumsid" value='<%=forumsid%>' />
    <jsp:param name="nickname" value='<%=nickname%>' />
    <jsp:param name="keyword" value='<%=keyword%>' />
    <jsp:param name="keyword2" value='<%=keyword2%>' />
</jsp:include>
<%}%>
<!-- *****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/bottom.jsp"/>
```

```
</BODY></HTML>
```

如果用户没有输入任何参数而提交，所有参数处理后的结果都为空，程序会输出 JavaScript 脚本，在客户端运行的结果是弹出警告，如图 31.12 所示。



图 31.12 三个参数为空发出警告

处理搜索的请求 searchresult 由其所对应的 Servlet 类 search_result 来执行，在其中获取从用户提供并处理过的参数值，用这些参数值在数据库表中检索相关的记录，最终将结果以一定的格式输出，由 search.jsp 加载后显示给用户。

search_result.java 代码如下。

```
package mybbs;
import java.io.*;
import java.sql.*;
import java.io.IOException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class search_result extends HttpServlet {
    PrintWriter out;
    ResultSet rs;
    String sql;
    String action,userid;
    public void doPost(HttpServletRequest request,
                       HttpServletResponse response)
        throws IOException
    {
        response.setContentType("text/html;charset=gbk");
        out = response.getWriter();
        try{
            DBConn conn=new DBConn();
            CAEncode cod=new CAEncode();
```

❑ 获取参数值，并进行重编码、字符过滤等处理。

```
String nickname =cod._conv(cod.htmlEncode(request.getParameter("nickname")));
String keyword =cod._conv(cod.htmlEncode(request.getParameter("keyword")));
String keyword2 =cod._conv(cod.htmlEncode(request.getParameter("keyword2")));
String forumid=cod.htmlEncode(request.getParameter("forumsid"));
if(request.getParameter("nickname")==null||nickname.equals("")){
    nickname="%"; }
if(request.getParameter("keyword")==null||keyword.equals("")){
    keyword="_ *"; }
if(request.getParameter("keyword2")==null||keyword2.equals("")){
    keyword2="_ *"; }
if(!forumid.equals("")){
    forumid="="+forumid;
```

```
}
if(request.getParameter("forumsid")==null||forumid.equals("")){
    forumid=">0":    }
```

处理。最终搜索到结果输出并显示的效果如图 31.13 所示。

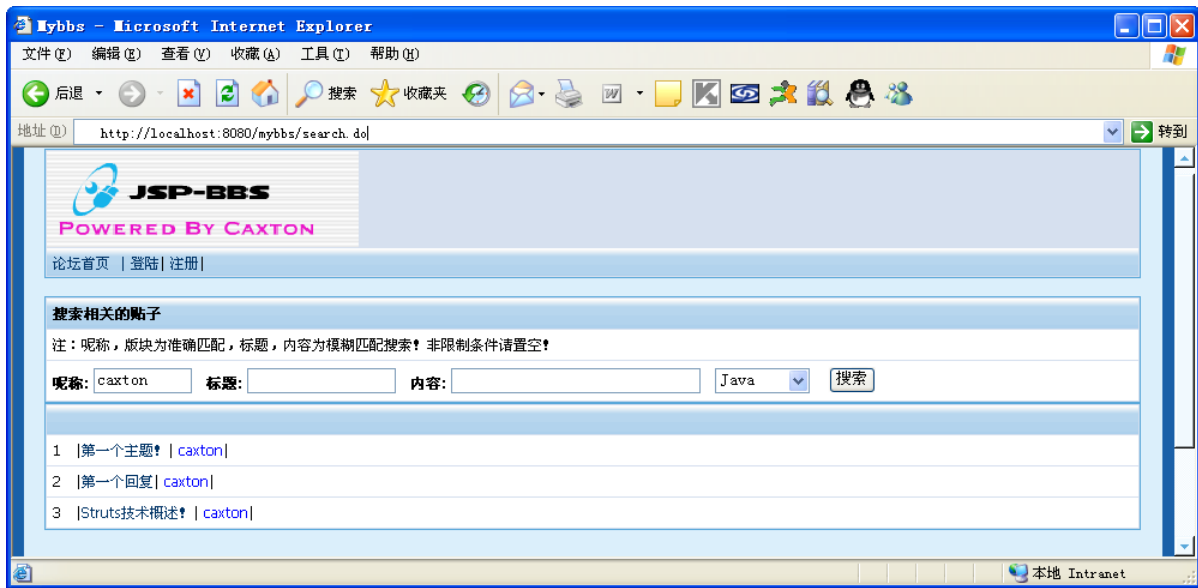


图 31.13 搜索结果

31.3 用户信息处理

BBS 系统的用户信息处理主要是与用户注册信息相关的一些操作。其中包括用户注册、用户登录、用户退出、用户登录后修改注册信息、修改登录密码、查看用户信息时输出数据库中的用户信息。另外要用户登录时将用户状态置为在线，退出或超时置为离线状态。

31.3.1 用户注册

用户注册是 BBS 系统中用户信息处理最重要的一块。进入 BBS 系统后，只能浏览少量的内容。只有注册为 BBS 用户，才能在 BBS 上发贴以及进行其他一些操作。本系统用户注册界面 `regist.jsp` 运行效果如图 31.14 所示。

图 31.14 用户注册界面

上面的用户注册视图是由模板加载中间的表单构建的，表单加载页面的代码全部为 HTML，这里不再列出，有兴趣者可以查看随书光盘中的源程序。

regist.jsp 代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!-- *****-->
<jsp:include page="include/registform.jsp"/>
<!-- *****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>
```

用户输入注册信息并确定必须填写的内容已经填写完毕后，提交注册信息，这些表单参数值首先要经过合法性验证。验证是由 Struts 的 validator 完成的。

用户提交注册信息如果通过合法性验证，则会交由 AddUserAction 处理。AddUserAction 首先要获取这些信息，对用户名和昵称进行检索，如表中不存在相同的记录，则将用户注册信息写入数据库中的

bbsuser 表中。成为一条用户名和昵称惟一的用户记录。

AddUserAction.java 代码如下。

```
package mybbs;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.validator.DynaValidatorForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import java.util.Date;
import java.sql.*;
public final class AddUserAction extends Action {
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {
```

- ❑ 获取用户 session、当前时间和用户的 IP。

```
HttpSession session=request.getSession();
String ip = request.getRemoteAddr();
Date date = new Date();
DynaValidatorForm userform = (DynaValidatorForm) form;
CAEncode encode = new CAEncode();
DBConn conn = new DBConn();
```

- ❑ 获取注册信息并进行处理。

```
String username = encode.htmlEncode( (String) userform.get("username"));
String password = encode.htmlEncode( (String) userform.get("password"));
String password2 = encode.htmlEncode( (String) userform.get("password2"));
String nickname = encode.htmlEncode( (String) userform.get("nickname"));
String sex = encode.htmlEncode( (String) userform.get("sex"));
String birthyear = encode.htmlEncode( (String) userform.get("birthyear"));
String birthmonth = encode.htmlEncode( (String) userform.get("birthmonth"));
String birthday = encode.htmlEncode( (String) userform.get("birthday"));
String email = encode.htmlEncode( (String) userform.get("email"));
String mobile = encode.htmlEncode( (String) userform.get("mobile"));
String qq = encode.htmlEncode( (String) userform.get("qq"));
String skype = encode.htmlEncode( (String) userform.get("skype"));
String weblog = encode.htmlEncode( (String) userform.get("weblog"));
String info = encode.htmlEncode( (String) userform.get("info"));
String signature = encode.htmlEncode( (String) userform.get("signature"));
String face = encode.htmlEncode( (String) userform.get("face"));
String action=(String) userform.get("action");
String PageForward="global";
try{
    CheckUser CU = new CheckUser();
```

- ❑ 如果 action 值为 1，说明操作为新用户注册，调用 CheckUser 中的 isExist()方法，判断用户名和昵称是否在数据中已存在，如果不存在则可以将用户注册信息插入数据库中。

```
if (action.equals("1")) {
    if (!CU.isExist(username, nickname)) {
```

- ❑ 如果用户头像为空，则为其配置默认的头像。

```
        if (face.equals("")) {
            face = "images/a1.jpg";
        }
        String sql = "INSERT INTO bbsuser VALUE(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
```

- ❑ 用预处理的方式将用户注册信息插入数据库中。

```
        PreparedStatement pstmt = conn.getPstmt(sql);
        pstmt.setInt(1, 0);
        pstmt.setString(2, username);
        pstmt.setString(3, password);
        pstmt.setString(4, nickname);
        pstmt.setString(5, sex);
        pstmt.setString(6, birthyear);
        pstmt.setString(7, birthmonth);
        pstmt.setString(8, birthday);
        pstmt.setString(9, email);
        pstmt.setString(10, weblog);
        pstmt.setString(11, qq);
        pstmt.setString(12, skype);
        pstmt.setString(13, mobile);
        pstmt.setString(14, signature);
        pstmt.setString(15, date.toLocaleString());
        pstmt.setString(16, date.toLocaleString());
        pstmt.setString(17, ip);
        pstmt.setString(18, ip);
        pstmt.setString(19, info);
        pstmt.setString(20, face);
        pstmt.setInt(21, 0);
        pstmt.setInt(22, 0);
        pstmt.setInt(23, 100);
        pstmt.setInt(24, 0);
        pstmt.setInt(25, 0);
        pstmt.setString(26, "user");
        int flag = pstmt.executeUpdate();
        pstmt.close();
```

- ❑ 如果插入成功，则向目标页面传递用户的用户名以及密码，显示给用户确认。

```
        if (flag > 0) {
            request.setAttribute("username", username);
            request.setAttribute("password", password);
            PageForward = "regsuccessed"; //新用户注册成功
        }
        else { //新用户注册失败
            PageForward = "regfailed";
        }
    }
    else {
        PageForward = "regfailed";
```

```

        request.setAttribute("throw", new String("用户名或昵称已存在! "));
    }
}

```

- ❑ 如果 action 的值为 2，则说明为用户修改注册信息。

```

if (action.equals("2")) {
    String userid = (String) session.getAttribute("userid");
    String user = (String) session.getAttribute("username");
}

```

- ❑ 判断用户提交的昵称是否已存在，不存在才可以成功更新。

```

if (!CU.nicknameExist(userid, nickname)) {
    String sql2 =
        "UPDATE bbsuser SET nickname=?,sex=?,birthyear=?,birthmonth=?,birthday=?," +
        "email=?,weblog=?,qq=?,skype=?,mobile=?,signature=?,info=?,face=? WHERE username='" +
        user + "' AND id=" + userid;
}

```

- ❑ 用预处理的方式更新数据库中的用户信息。

```

PreparedStatement pstmt = conn.getPstmt(sql2);
pstmt.setString(1, nickname);
pstmt.setString(2, sex);
pstmt.setString(3, birthyear);
pstmt.setString(4, birthmonth);
pstmt.setString(5, birthday);
pstmt.setString(6, email);
pstmt.setString(7, weblog);
pstmt.setString(8, qq);
pstmt.setString(9, skype);
pstmt.setString(10, mobile);
pstmt.setString(11, signature);
pstmt.setString(12, info);
pstmt.setString(13, face);
int u = pstmt.executeUpdate();
pstmt.close();
if (u > 0) {
    PageForward = "updatesuccess";
}
else {
    PageForward = "regfailed";
    request.setAttribute("throw", new String("用户资料更新失败! "));
}
}
}

```

- ❑ 如果 action 的值为 3，则说明为更改密码操作。

```

if (action.equals("3")) {
    String user = (String) session.getAttribute("username");
    String userid = (String) session.getAttribute("userid");
}

```

- ❑ 从 session 中取出用户名，判断用户原始密码是否正确，如果正确才能更新密码。

```

if (CU.checklogin(user, password)) {
    String sql3 = "UPDATE bbsuser SET password='" + password2 +
        "' WHERE username='" +
        user + "' AND id=" + userid;
    int u = conn.update(sql3);
}

```



```

        if (u > 0) {
            PageForward = "updatesuccess";
        }
        else {
            PageForward = "regfailed";
            request.setAttribute("throw", new String("密码更新失败! "));
        }
    }
    else {
        PageForward = "regfailed";
        request.setAttribute("throw", new String("原始密码不正确! "));
    }
}
}
catch(Exception e){
    PageForward = "regfailed";
    request.setAttribute("throw", new String("用户名已存在, 或其他未预知的错误! "));
}
conn.DBclose();
return mapping.findForward(PageForward);
}
}

```

系统用于判断用户名是否存在、用户昵称是否存在、用户登录时检索用户名和密码、用户登录后更新用户最后的登录时间和 IP、判断用户是否为版主、更新用户当前状态等与用户信息相关的方法都封装在 CheckUser.java 中。它主要用于配合其他处理程序进行一些事务处理和逻辑判断，是本系统一个重要的逻辑事务处理 Bean。

CheckUser.java 代码如下。

```

package mybbs;
import java.sql.*;
import mybbs.DBConn;
public class CheckUser {
    String username, password, nickname, userid, userpower, forum;
    public void CheckUser() {}
}

```

❑ 判断用户名和用户昵称是否已存在。

```

public boolean isExist(String user,String nickname) {
    DBConn conn=new DBConn();
    boolean bb = false;
    try {
        String sql2 = "SELECT * FROM bbsuser WHERE username='"+user+"' OR nickname='"+nickname+"'";
        ResultSet rs = conn.query(sql2);
        if (rs.next()) {
            bb = true;
        }
        rs.close();
        conn.DBclose();
    }
    catch (Exception e) {
        System.out.print(e.toString());
    }
}

```

```
return bb;
}
```

- ❑ 判断用户昵称是否已存在。

```
public boolean nicknameExist(String userid,String nickname) {
    DBConn conn = new DBConn();
    boolean bb = false;
    try {
        String sql2 = "SELECT * FROM bbsuser WHERE id!=" + userid +
            " AND nickname=" + nickname + "";
        ResultSet rs = conn.query(sql2);
        if (rs.next()) {
            bb = true;
        }
        rs.close();
        conn.DBclose();
    }
    catch (Exception e) {
        System.out.print(e.toString());
    }
    return bb;
}
```

- ❑ 用户登录时验证密码和用户名。

```
public boolean checklogin(String user, String psw) {
    DBConn conn = new DBConn();
    this.username = user;
    this.password = psw;
    boolean b2 = false;
    try {
        String sql1 = "SELECT * FROM bbsuser WHERE username=" + username +
            " AND password=" + password + " AND islock=0";
        ResultSet rs1 = conn.query(sql1);
        if (rs1.next()) {
            nickname = rs1.getString("nickname");
            userpower = rs1.getString("userpower");
            userid = rs1.getString("id");
            b2 = true;
        }
        rs1.close();
        conn.DBclose();
    }
    catch (Exception e) {
        System.out.print(e.toString());
    }
    return b2;
}
```

- ❑ 如果验证通过，获取用户的昵称、编号、权限值，用于保存它们在 session 中识别用户。

```
public String getNickname() {
    return nickname;
}
```

```
public String getUserid() {  
    return userid;  
}  
public String getUserpower() {  
    return userpower;  
}
```

- ❑ 用户登录成功，则更新用户最后一次登录的时间与 IP 地址。

```
public void UpdateUserInfo(String user,String ip){  
    DBConn conn=new DBConn();  
    String sql="UPDATE bbsuser SET last_time=SYSDATE(),online=1,last_ip='"+ip+"' WHERE  username='"+  
        username + "'";  
    try {  
        conn.update(sql);  
        conn.DBclose();  
    }  
    catch (Exception ex) {  
    }  
}
```

- ❑ 判断用户是否是某个分论坛的版主。如果是则获取其对应的论坛编号。

```
public boolean forumAdmin(String user) {  
    this.username = user;  
    DBConn conn=new DBConn();  
    boolean bb = false;  
    try {  
        String sql2 = "SELECT * FROM forums WHERE admin_author REGEXP '.' +  
            username + '.'";  
        ResultSet rs2 = conn.query(sql2);  
        if (rs2.next()) {  
            forum = rs2.getString("id");  
            bb = true;  
        }  
        rs2.close();  
        conn.DBclose();  
    }  
    catch (Exception e) {  
        System.out.print(e.toString());  
    }  
    return bb;  
}
```

- ❑ 获取版主对应的分论坛编号。

```
public String getForum() {  
    return forum;  
}
```

- ❑ 在用户退出或登录超时用户 session 自动过期时，将用户状态更新。

```
public void UserOut(String userid) {  
    DBConn conn=new DBConn();  
    String sql="UPDATE bbsuser SET online=0 WHERE id="+userid;  
    try{  
        conn.update(sql);  
    }
```

```

        conn.DBclose();
    }
    catch(Exception e){
        System.out.print(e.toString());
    }
}
}

```

本系统并没有对用户密码进行 MD5 加密,前面已经介绍了 MD5 加密的相关方法。只需用前面章节中所构造的 MD5.java 中的方法对密码进行加密转换即可。

用户注册成功后,会转向到 regsu.jsp 页面,其中定义了进入论坛首页和进入论坛登录页面的链接。如果用户没有做出选择,会自动转向到登录页面,regsu.jsp 代码如下。

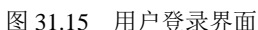
```

<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<META HTTP-EQUIV="REFRESH" CONTENT="6;url=login.do">
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/><BR>
<!-- *****-->
<%
    String forumid=(String)request.getAttribute("username");
    String topicid=(String)request.getAttribute("password");
%>
<TABLE width="98%" align=center cellPadding=1 >
<TBODY><TR>
<TD height="23" align=left background="images/head.gif" class=head>
≡ <B>本页主题: 您已注册成功! </B></TD>
</TR><TR bgcolor="#ADD0EC">
<TD height=151 align=right class=cbg>
<div align="left"><DIR>
<LI>用户名:<FONT color=red><%=forumid%></FONT>
<LI>密码: <FONT color=red><%=topicid%></FONT>
</DIR><DIR>
<LI><A href=default.jsp class="style2"> 返回本站论坛首页</A>
<LI><A href='login.do' class="style2">进入登陆页面 POST</A>
</DIR></div></TD> </TR>
<TR bgcolor="#ADD0EC">
<TD height=30 align=right background="images/head.gif" class=cbg>&nbsp;&nbsp;&nbsp;</TD>
</TR></TBODY></TABLE>

<!-- *****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>

```

用直接进入 BBS 系统首页时,程序在判断用户尚未登录的情况下,会在顶部显示用户登录表单,如果已登录,则不会显示。系统用户登录界面如图 31.15 所示。



用户登录界面是由 `login.jsp` 利用模板页面加载登录表单代码 `toplog.jsp` 构建的，另外用户登录表单还会在用户未登录情况下的首页被加载，`login.jsp` 页面代码略。

```
<% @ page contentType="text/html;charset=GBK" language="java" %>
<TABLE <TABLE class=i_table cellSpacing=1 cellPadding=5 width='98%' align=center>
<TBODY>
<TR>
<TD class=head height=24 align=left colSpan=2 background='images/head.gif'>
</TD></TR>
<TR><TD width='45%' >
<FORM action='Login.do' method='post'><B>
用户名:<INPUT size=10 id=atc type="text" name="username" ><BR>
密 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&码:<INPUT type='password' size=10 id=pass name="password" ></B>
<INPUT type="hidden" name="action" value='in' >
<INPUT type="submit" value="登陆" name="submit">
</FORM></TD></TR>
</TBODY>
</TABLE>
```

用户通过登录表单提交登录验证信息后，系统会转发给 `LoginAction` 处理，其中调用 `CheckUser` 中的 `checklogin()` 方法对用户名和密码进行验证，如果存在于数据库表中，则通过验证登录成功。将用户的一些信息保存在 `session` 中，并将用户当前前状态更新为在线。另外 `LoginAction` 也是用户退出登录的处理程序，其中用参数 `action` 做为操作识别符。

```
package mybbs;
```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.DynaActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import javax.servlet.http.HttpSession;
public final class LoginAction extends Action{
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        DynaActionForm userform = (DynaActionForm) form;
        CAEncode encode=new CAEncode();
        DBConn conn = new DBConn();

```

- ❑ 获取登录操作的参数值并处理。

```

String action=(String)userform.get("action");
String username =encode._conv(encode.htmlEncode((String)userform.get("username")));
String password =encode._conv(encode.htmlEncode((String)userform.get("password")));
HttpSession session = request.getSession();
String ip=request.getRemoteAddr();
String PageForward="global";
CheckUser cu=new CheckUser();

```

- ❑ 参数 action 值为 “in”，说明要进行登录操作，调用 checklogin()方法验证用户名和密码。如果通过验证，则将用户信息保存在 session 中。

```

if(action.equals("in")){
    if(cu.checklogin(username,password)){
        session.setMaxInactiveInterval(params.MaxInactive);
        session.setAttribute("login","OK");
        session.setAttribute("username",username);
        session.setAttribute("nickname",cu.getNickname());
        session.setAttribute("userid",cu.getUserid());
        session.setAttribute("userpower",cu.getUserpower());

```

- ❑ 更新用户最后登录 IP 以及时间，其中时间在 CheckUser 中获取。判断用户是否为版主，如果是，则将其所属的分论坛编号存于 session 中。

```

        cu.UpdateUserInfo(username,ip);
        if(cu.forumAdmin(username)){
            session.setAttribute("forum",cu.getForum());
        }
        PageForward="LoginOkay";
    }
    else{
        PageForward="LoginFailed";
    }
}

```

- ❑ action 值为 “out”，则为退出登录操作，销毁用户 session 并更新其当前状态，最后定向页面到系统首页。

```
if(action.equals("out")){
    String userid=(String)session.getAttribute("userid");
    cu.UserOut(userid);
    session.invalidate();
    PageForward="LoginOut";
}
return mapping.findForward(PageForward);
}
```

用户登录成功后,系统的导航条会发生变化,出现一些用于更新注册信息、修改密码等操作的链接。首页中的登录区也消失了。如图 31.16 所示是普通用户登录成功后的首页。

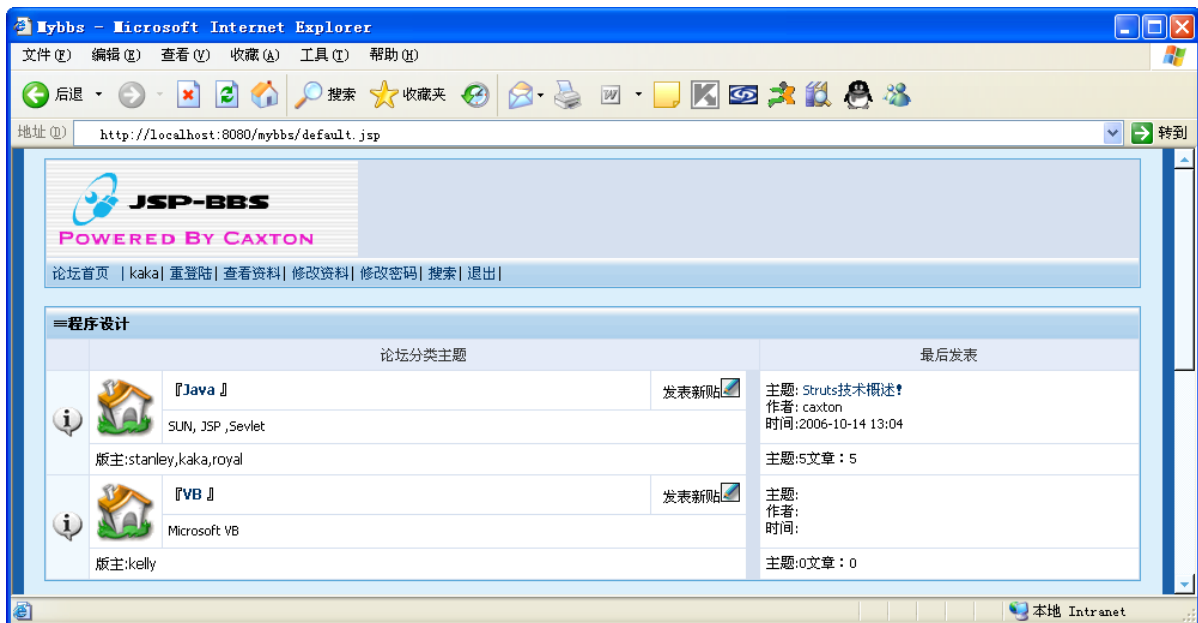


图 31.16 用户成功登录

31.3.3 用户退出

用在登录的状态下,如果要退出登录,可以单击上方工具条的“退出”链接。经 LoginAction 处理,销毁用户 session,同时更新用户状态。最后会将页面转向到系统的首页,

由于在用户登录时,系统将用户 session 不活动的最大时间设为 5 分钟。因此如果用户离开浏览器而没有注销,或是在 5 分钟没有进行任何操作,则 session 自动过期而销毁。这是一种用户被迫退出的情况,由 session 监听器 UserListener 监听到 session 销毁后,自动更新该用户的状态。

31.3.4 修改注册信息

用户在登录的情况下,可以对自己的信息进行修改,除了用户名以外,其他信息都可以修改。用户修改注册信息的页面是由模板页面加载修改数据表单构建的,在加载页面中获取用户对应的信息,以可修改状态显示在表单文本中,如图 31.17 所示。

填写信息	
*号注明的都需要填写，密码请区分大小写。头像请输入URL，如"upload/a1.jpg"	
昵称:	<input type="text" value="kaka"/> *
头像:	<input type="text" value="images/a1.gif"/>
性别:	<input checked="" type="radio"/> 男 <input type="radio"/> 女
生日:	1980 年 01 月 01 日 *
E-mail:	<input type="text" value="136311890@qq.com"/> *
手机号码	<input type="text" value="13658010799"/>
腾讯QQ:	<input type="text" value="136311890"/> *
Skype:	<input type="text" value="kokoro"/>
Weblog:	<input type="text" value="http://"/>
个人简介:	<input type="text"/>

图 31.17 修改用户注册信息

修改用户信息的视图几乎同用户注册信息的界面相同，只是利用 JavaBean 程序 DBUser 将用户信息输出到文本框中。下面是构建修改用户信息的界面代码。

modifyinfo.jsp 代码。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!-- ***** -->
<jsp:include page="include/modify.jsp"/>
<!-- ***** -->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>
```

modify.jsp 用于构建修改用户信息的表单，在其中要对获取用户 session 中的用户编号，用于获取对应的用户信息。如果用户未登录而进入该页面，会自动转向到登录页面。

```
<%@ page contentType="text/html; charset=GBK" %>
<%@ page language="java" %>
<%
String userid="0";
if(session.getAttribute("userid")!=null)
    userid=(String)session.getAttribute("userid");
else
    response.sendRedirect("login.do");
%>
<jsp:useBean id="user" scope="request" class="mybbs.DBUser" />
<jsp:setProperty name="user" property="userid" value="<%=userid%>"/>
<TABLE cellSpacing=0 cellPadding=4 width=657 align=center border=0>
<form action="ModifyInfo.do" method="post">
<INPUT type="hidden" name="action" value='2' >
```


上面只列出了部分 modify.jsp 的代码，其他代码同查看用户信息的代码几乎相同，只是多了一个表单，其中用到的 DBUser.java 将在查看用户信息一节中列出。用户修改注册信息后，提交修改信息处理请求 ModifyInfo.do，并最终转发给 AddUserAction 处理更新数据库。

31.3.5 修改用户密码

用户要修改自己的登录密码，同修改其他信息不同的是，修改密码要输入用户原始密码进行验证。这样可以提高用户密码的安全性。用户可在登录的情况下，单击工具条中的“修改密码”链接，进入修改密码表单界面，如图 31.18 所示。

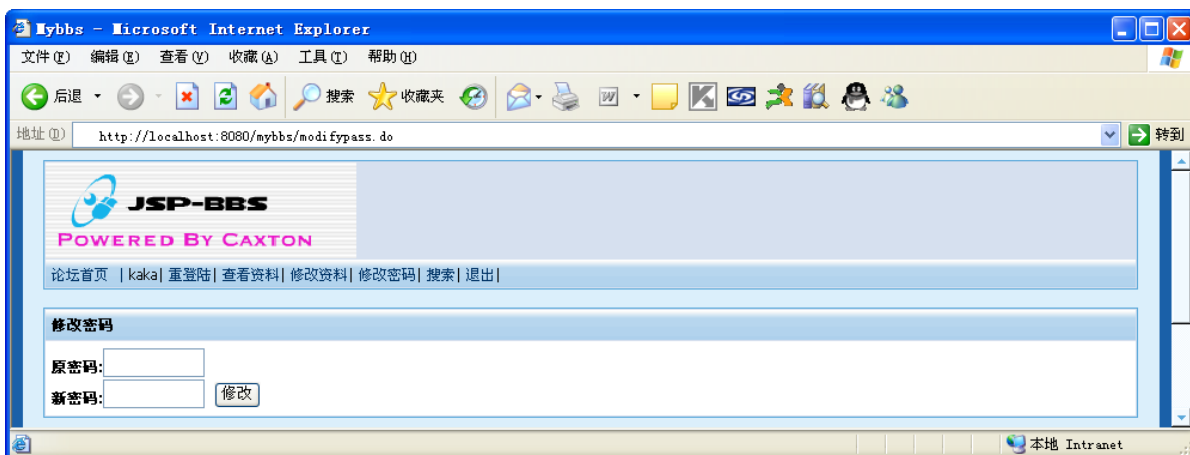


图 31.18 修改用户密码

图 31.18 所示的修改密码视图同样是由模板加载修改密码表单构成。修改用户密码的 JSP 程序 modifypass.jsp 的代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#d9e1f2 border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!-- *****-->
<TABLE class=i_table cellSpacing=1 cellPadding=5 width='98%' align=center>
<TBODY><TR>
<TD class=head height=24 align=left colSpan=2 background='images/head.gif'>
修改密码</TD></TR>
<TR><TD width='45%' align=left onmouseout=this.className='f_one'>
<FORM action='ModifyPass.do' method='post'><B>
原密码:<INPUT size=10 id=atc type="password" name="password" ><BR>
新密码:<INPUT type='password' size=10 id=pass name="password2" ></B>
<INPUT type="hidden" name="action" value='3' >
<INPUT type="submit" value="修改" name="submit">
</FORM></TD></TR>
```

```

</TBODY>
</TABLE>
<!--*****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>

```

修改密码表单中有一个隐藏的参数提交，也就是 `action` 其值为“3”，目标处理程序最终仍是 `AddUserAction`，根据 `action` 的值为“3”识别操作为修改密码。

31.3.6 查看用户资料

用户的资料可能会被其他用户在查看该用户所发的贴子时查看，因为在显示贴子具体内容列表时，同时显示了一个“资料”按钮。所有用户都可以单击该按钮，查看该用户的资料。另外用户也可以在登录的情况下，单击工具条的“查看资料”链接，查看自己的资料，如图 31.19 所示。

用户信息:	
昵称:	kaka
发贴数量:	1
最后登录:	2006-10-14
注册时间:	2006-09-09
性别:	男
生日:	1984年 01月 01日
E-mail:	136311890@qq.com
WebLog:	http://
QQ:	136311890
Skype:	kokoro
Mobile:	13658010799
签名:	每一步都有一个新的高度，我能!
简介:	中国人民! 年轻人!
用户信息:	



图 31.19 查看用户资料

上面的查看用户资料的视图由模板页面加载用户信息列表构成，利用 `JavaBean DBUser` 获取用户对应的信息，并列表显示。

`profile.jsp` 代码如下。

```

<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/>
<BR>
<!--*****-->

```

☐ 判断用户是否登录，如果没有登录，则不能查看任何用户信息，自动转向到登录页面。

```

<%
    if(session.getAttribute("login")==null)
        response.sendRedirect("login.do");
    String userid="0";

```

☐ 获取用户以两种方式进入查看用户资料页面时的参数取值。

```

    if(request.getParameter("userid")==null&&session.getAttribute("userid")!=null)

```

```

        userid=(String)session.getAttribute("userid");
    else
        userid=request.getParameter("userid");
    %>

```

❑ 利用 DBUser 获取对应用户的注册信息，列表显示。

```

<jsp:useBean id="user" scope="request" class="mybbs.DBUser" />
<jsp:setProperty name="user" property="userid" value="<%=userid%>"/>
<CENTER>
<table border="white" cellSpacing="0" width="76%" border="darkblue" border="1">
<TR><TD background='images/head.gif' colSpan=3>
    用户信息:</TD></TR>
<TR><TD colSpan=3></TD></TR>
<TR bgcolor=#f2f8fd><TD width=180 align=right>昵称:</TD>
<TD><jsp:getProperty name="user" property="nickname"/></TD>
<TD rowspan=7 width=160>
<IMG height=160 width=160 src='<jsp:getProperty name="user" property="face"/>' />
</TD></TR>
<TR><TD align=right>发贴数量:</TD>
<TD><jsp:getProperty name="user" property="totalpost"/></TD></TR>
<TR><TD align=right>最后登录:</TD>
<TD><jsp:getProperty name="user" property="lasttime"/></TD></TR>
<TR><TD align=right>注册时间:</TD>
<TD><jsp:getProperty name="user" property="regtime"/></TD></TR>
<TR><TD align=right>性别:</TD>
<TD><jsp:getProperty name="user" property="sex"/></TD></TR>
<TR><TD align=right>生日:</TD>
<TD><jsp:getProperty name="user" property="birthyear"/>年
<jsp:getProperty name="user" property="birthmonth"/>月
<jsp:getProperty name="user" property="birthday"/>日</TD></TR>
<TR><TD align=right>E-mail:</TD>
<TD><jsp:getProperty name="user" property="email"/></TD></TR>
<TR><TD align=right>WebLog:</TD>
<TD colSpan=2><jsp:getProperty name="user" property="weblog"/></TD></TR>
<TR><TD align=right>QQ:</TD>
<TD colSpan=2><jsp:getProperty name="user" property="qq"/></TD></TR>
<TR><TD align=right>Skype:</TD>
<TD colSpan=2><jsp:getProperty name="user" property="skype"/></TD></TR>
<TR><TD align=right>Mobile:</TD>
<TD colSpan=2><jsp:getProperty name="user" property="mobile"/></TD></TR>
<TR><TD align=right>签名:</TD>
<TD colSpan=2><jsp:getProperty name="user" property="signature"/></TD></TR>
<TR><TD align=right>简介:</TD>
<TD colSpan=2><jsp:getProperty name="user" property="info"/></TD></TR>
<TR><TD background='images/head.gif' colSpan=3>
    用户信息:</TD></TR>
<TR><TD colSpan=3></TD></TR>
</table>
</CENTER>
<!-- *****-->
<BR> <CENTER></CENTER><BR></TD></TR></TBODY></TABLE>

```

```
<jsp:include page="include/buttom.jsp"/>
</BODY>
</HTML>
```

DBUser.java 主要是通过用户编号获取对应的用户信息，构造这些用户信息的 `getter` 方法，从而可以在 JSP 中便用 JavaBean 标签获取这些参数的值。DBUser.java 代码如下。

```
package mybbs;
import java.sql.*;
public class DBUser
{
    private String userid;
    String username,password,nickname,sex,birthday,birthmonth,birtheyear,e_mail,weblog,qq,skype,mobile,
        signature,usergrade,reg_time,last_ip,reg_ip,last_time,info,face;
    int userscore,userpower,totalpost;
    public void DBUser(){}
}
```

❑ 设置用户编号。

```
public void setUserid(String userid)
{
    this.userid=userid;
}
public String getNickname(){
    try{
        DBConn conn=new DBConn();
```

❑ 检索 bbsuser 表，获取指定编号所对应的用户信息。

```
String sql="SELECT * FROM bbsuser WHERE id="+userid;
ResultSet rs=conn.query(sql);
rs.next();
username=rs.getString("username");
password=rs.getString("password");
nickname=rs.getString("nickname");
sex=rs.getString("sex");
birthday=rs.getString("birthday");
birthmonth=rs.getString("birthmonth");
birtheyear=rs.getString("birtheyear");
e_mail=rs.getString("email");
qq=rs.getString("qq");
skype=rs.getString("skype");
mobile=rs.getString("mobile");
signature=rs.getString("signature");
usergrade=rs.getString("usergrade");
reg_time=rs.getString("reg_time");
last_ip=rs.getString("last_ip");
reg_ip=rs.getString("reg_ip");
last_time=rs.getString("last_time");
info=rs.getString("info");
weblog=rs.getString("weblog");
userscore=rs.getInt("userscore");
userpower=rs.getInt("userpower");
totalpost=rs.getInt("totalpost");
face=rs.getString("face");
```

```
rs.close();
}
catch(Exception e){}
return nickname;
}
```

❑ 构造用户信息相对应的 `getter` 方法。

```
public String getEmail()
{
    return e_mail;
}
public String getQq()
{
    return qq;
}
public String getSkype()
{
    return skype;
}
public String getRegtime()
{
    return reg_time.substring(0,10);
}
public String getFace()
{
    return face;
}
//.....省去其他属性的 getter 方法
}
```

为节省篇幅，一些其他的 `getter` 方法在这里省去。`DBUser` 在获取用户信息时首先获取 `nickname` 的值，同时在其中与数据库建立连接时一次性获取其他属性的值。应该注意的是，在利用 `DBUser` 获取用户信息时，应首先获取 `nickname` 的值，这样其他值才能得到，否则先获取其他属性值时会出现错误得到“`null`”，而导致获取失败。

31.4 发贴与投票

发贴和投票包括发主题、发起投票、回复主题。这三种操作都是用户提交信息向数据库中写入数据。这些用户操作都是论坛最常用的操作，也是论坛主要的也最基本或不可缺少的功能。

31.4.1 发表新主题

主题就是用户要讨论的方向，也就是话题，一个话题通常会引来其他用户对此话题的看法或见解，其中用户针对此话题发表的评论称为回复，也叫跟贴。主题与回复不同的是它将在分论坛的主题列表中显示，回复隶属于对应主题。发表主题时一定要要有标题，主题在写入数据库中时比回复要多一些属性字段。图 31.20 是发表主题的界面。

图 31.20 用户发帖界面

发表主题的界面也是由模板加载中间的表单页面形成。由于使用图 31.20 所示的一些自定义标签要用到 JavaScript，它可以实现自动插入。其中包含了大量的代码，由于本书的写作方向，不对这些做介绍，其代码也不再列出。本章所构建的发帖界面只实现基本的发文本、各类文件等内容。不能利用转换代码向主题内容中加入文字格式、图形等。用户上传图片将由程序自动定位到主题内容中。

postform.jsp 是发帖页面加载的构建中间表单的程序，它可以动态的构建发主题、发投票主题、回复主题、编辑帖子内容四类操作的用户界面，代码如下。

```
<%@ taglib uri="/WEB-INF/struts-bean.tld" prefix="bean" %>
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
<%@ taglib uri="/WEB-INF/struts-logic.tld" prefix="logic" %>
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<%@ page import="mybbs.modifyPost" %>
```

- ❑ 获取用户要发帖的位置参数，以及发帖的种类，由 action 控制，根据 action 的值构建不同视图。forumid 定位了发帖的论坛，topicid 定位了回复所对应的主题。

```
<%
String forumid=request.getParameter("forumsid");
String topicid=request.getParameter("topicid");
String action=request.getParameter("action");
```

- ❑ 如果是非登录用户，自动转向登录页面。

```
if(session.getAttribute("login")==null){
    response.sendRedirect("login.do");
}
%>
<TABLE cellSpacing=0 cellPadding=0 width="99%" align=center>
<TBODY><TR><TD>
```

```
<FORM action=postup.do method=post encType="multipart/form-data">
<!--*****-->
```

❑ 用隐藏表单向处理程序提交操作的类型，以及编辑贴子时的记录号。

```
<INPUT type=hidden name=action value=<%=action%> >
<INPUT type=hidden name=postid value=<%=request.getParameter("postid")%> >
<!--*****-->
<TABLE class=i_table cellSpacing=1 cellPadding=3 width="100%" align=center>
<TBODY><TR><TD class=head colSpan=3>&nbsp;</TD></TR>
<TR><TD class=t_one width="20%" align=right><B>标题</B></TD>
<TD class=t_one>
```

❑ 利用逻辑标签判决 action 值为 4 时，构建编辑贴子的视图，获取该贴的内容，显示在可以修改的文本区中。

```
<logic:equal parameter="action" value="4">
<jsp:useBean id="beans" scope="request" class="mybbs.modifyPost" />
<jsp:setProperty name="beans" property="postid" value="<%=request.getParameter("postid")%>"/>
<INPUT tabIndex=1 size=60 id=atc_title name="c_title" value='<%=beans.getTitle()%>'>
<INPUT type=hidden name=topicid value=<%=beans.getTopicid()%> >
<INPUT type=hidden name=forumsid value=<%=beans.getForumid()%> >
</logic:equal>
```

❑ Action 值不等于 4 时，标题文本区显示为空，重新发新贴。

```
<logic:notEqual parameter="action" value="4">
<INPUT tabIndex=1 size=60 id=atc_title name="c_title" >
<INPUT type=hidden name=topicid value=<%=topicid%> >
<INPUT type=hidden name=forumsid value=<%=forumid%> >
</logic:notEqual>
</TD></TR>
```

❑ 如果 action 值为 3，构建用户发起投票主题的界面，除了构建同发表主题相同的表单区外，还要加一个选择投票方式的单选框和一个输入投票条目的文本区。

```
<logic:equal parameter="action" value="3">
<TR>
<TD align=right><B>投票模式</B></TD>
<TD>
<INPUT type=radio checked value=1 name=model>不记名单选
<INPUT type=radio value=2 name=model>记名单选
<INPUT type=radio value=3 name=model>不记名复选
<INPUT type=radio value=4 name=model>记名复选
</TD></TR>
<TR>
<TD align=right><B>投票选项</B></TD>
<TD>
<TEXTAREA name="items" ROWS="8" COLS="60" WRAP="physical">
</TEXTAREA><BR>
每一个选项为一行，按 Enter 换行，系统将以换行符区分各选项！
</TD>
</TR>
</logic:equal>
<!--*****-->
<TR><TD class=t_one align=right><B>发贴内容</B></TD>
<TD class=t_one align=left colSpan=2>
```

❑ 如果 action 值为 4 则显示要编辑贴子的内容在文本区。

```
<logic:equal parameter="action" value="4">
<TEXTAREA name=c_content rows=20 cols=80 >
<jsp:getProperty name="beans" property="content"/>
</TEXTAREA>
</logic:equal>
```

❑ 如果 action 值不为 4, 显示空的文本区域。

```
<logic:notEqual parameter="action" value="4">
<TEXTAREA name=c_content rows=20 cols=80 >
</TEXTAREA>
</logic:notEqual>
</TD></TR>
```

❑ 构建上传文件的输入区域。

```
<TR><TD class=t_one vAlign=top align=left>附件最大:1000 (K)</TD>
<TD class=t_one vAlign=top align=left>
<B>有效文件类型:gif jpg rar txt zip swf jpf fla</B><BR>
描述: <INPUT size=25 name=theText>
&nbsp; 附件: <INPUT class=subject type=file name=theFile>
</TD></TR></TBODY></TABLE><BR>
<CENTER>
<INPUT type=submit value="提 交" >
<INPUT type=reset value="重置" ></CENTER>
</TD></TR></FORM>
</TBODY></TABLE><BR>
```

上面的代码只是图 31.20 用界面的简化, 能实现基本的发贴功能。其他与本书内容关系不大的样式表, JavaScript 内容已被省去。有兴趣的读者可以在网上查找相关的资料, 对于有编程基础的读者, 这些都不是难事。

31.4.2 处理发贴操作的Action和ActionForm

用户提交发贴的内容后, 会首先将这些参数值填充进 PostForm, 再处理器转发给 TopicAction 处理时一并传递给 TopicAction, 这些内容已在 Struts 技术相关章中介绍。PostForm.java 构造与发贴表单参数相对应 ActionForm Bean, 代码如下。

```
package mybbs;
import org.apache.struts.action.*;
import javax.servlet.http.*;
import org.apache.struts.upload.FormFile;
public class PostForm
    extends ActionForm {
    private String theText, username, userid, c_title, c_content, action,
        forumsid, topicid, postid, model, items;
    private org.apache.struts.upload.FormFile theFile;           //文件框对应的是 formFile 类型
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
}
```



```

    }
    public String getItems() {
        return items;
    }

    //省去其他的一些属性的 setter 和 getter 方法

    public FormFile getTheFile() {
        return theFile;
    }
    public void setTheFile(org.apache.struts.upload.FormFile theFile) {
        this.theFile = theFile;
    }
    public void reset(ActionMapping actionMapping,
                      HttpServletRequest httpRequest) {
        username = null;
        userid = null;
        c_title = null;
        c_content = null;
    }
}

```

上面的 PostForm 中都是一些简单的 `getter` 和 `setter` 方法，只有一个特殊的 `FormFile` 类型的属性应值得读者注意，它是为了获取和设置要上传的文件内容。这里省去一些没必要列出的方法。

提交发贴内容写入数据库的操作全部是由 `TopicAction` 这个类来完成，其中可以根据 `action` 值的不同处理不同类型的内容写入数据库中。并调用其他类的方法实现数据库一些值的更新。

`TopicAction.java` 是系统最大的一个程序，可以处理多种不同的请求，代码如下。

```

package mybbs;
import org.apache.struts.action.*;
import org.apache.struts.upload.FormFile;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;
import java.util.*;
import javax.servlet.http.HttpSession;
public class TopicAction extends Action {
    public ActionForward execute(ActionMapping actionMapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response) {

        PostForm postform = (PostForm) form;
        DBConn conn = new DBConn();
        CAEncode ncode = new CAEncode();
        String PageForward = "global";
    }
}

```

❑ 判断用户是否登录，如果没有，停止执行以下操作，直接转向登录页面，否则继续。

```

HttpSession session = request.getSession();
if(session.getAttribute("nickname")==null)
{
    PageForward = "NotLogin";
}

```

```
else{
```

- 获取登录发帖用户 session 中的昵称、用户编号。获取从 ActionForm 中传递的其他参数的值，进行转换类型、字符重编码、过滤等处理。

```
String nickname=(String) session.getAttribute("nickname");
String uid = (String) session.getAttribute("userid");
int userid = Integer.parseInt(uid);
String fid = ncode.htmlEncode(postform.getForumsid());
int forumid = Integer.parseInt(fid);
String tid = ncode.htmlEncode(postform.getTopicid());
int topicid = 0;
try {
    topicid = Integer.parseInt(tid);
}
catch (Exception ex) {
    topicid = 0;
}
FormFile file = postform.getTheFile();
String c_title = ncode._conv(ncode.htmlEncode(postform.getC_title()));
String c_content = ncode._conv(ncode.htmlEncode(postform.getC_content()));
String action = postform.getAction();
String theText = ncode._conv(ncode.htmlEncode(postform.getTheText()));
String IP = request.getRemoteAddr();
String items = postform.getItems();
String model = postform.getModel();
String postid = ncode.htmlEncode(postform.getPostid());
```

- action 值为 3 时，操作为发表投票主题，可以获取到投票条目与类型的值。

```
if (action.equals("3")) {
    items = ncode._conv(ncode.repltem(items));
    model = ncode.htmlEncode(model);
}
```

- 当文件不为空时，说明发帖时有附件上传，调用 uploadFile 的方法将文件写入指定文件夹。最后在贴子内容中定位显示。

```
if (! (file.toString()).equals("")) {
    uploadFile uf = new uploadFile();
    uf.UPLOADFILE(file, theText);
    String upload_text = uf.getFiletext();
    c_content = c_content + upload_text;    //在贴子内容中加入文件描述内容。
}
try {
```

- 获取当前的时间，也就是发帖时间。

```
java.util.Date dd = new java.util.Date();
String dtime = dd.toLocaleString();
java.sql.Date sd = new java.sql.Date(dd.getTime());
PreparedStatement pstmt;
Operate OP = new Operate();    //实例化一个 Operate 类对象。
int uu = 0;
ResultSet rs = null;
```

- action 值为 3 时，进行写入投票主题的操作，获取 posts 表中的最大记录号，加 1 也就是将要写

入的贴子的记录号，用于在投票信息表中识别投票项目属于对应的主题。

```
if (action.equals("3")) {
    ResultSet rs = conn.query("SELECT * FROM posts");
    rs.last();
    int postt = rs.getInt("id") + 1;
    rs.close();
```

- ❑ 将投票内容写入数据库中。

```
VoteAct votat = new VoteAct();
String postdd = String.valueOf(postt);
votat.upvote(items, postdd, c_title, model);
```

- ❑ 获取要发贴的分论坛中主题的数量，获取其最大主题编号，加 1 后得到要写入主题的编号。

```
ResultSet rsm = conn.query(
    "SELECT * FROM posts WHERE istopic>0 AND forumid=" + forumid);
rsm.last();
int lastN = rsm.getRow() + 1;
rsm.close();
```

- ❑ 定义预处理语句，将贴子内容写入数据库中。

```
String psql =
    "INSERT INTO posts (forumid,groupid,istopic,topic_id,title,content,author," +
    "author_id,post_ip,post_time,last_postor,last_time)VALUES(?,?,?,?,?,?,?,?,?,?,?)";
pstmt = conn.getPstmt(psql);
pstmt.setInt(1, forumid);
pstmt.setInt(2, 0);
pstmt.setInt(3, 2);
pstmt.setInt(4, lastN);
pstmt.setString(5, c_title);
pstmt.setString(6, c_content);
pstmt.setString(7, nickname);
pstmt.setInt(8, userid);
pstmt.setString(9, IP);
pstmt.setString(10, dtime);
pstmt.setString(11, nickname);
pstmt.setString(12, dtime);
uu = pstmt.executeUpdate();
```

- ❑ 插入成功，更新分论坛的主题总数以及用户的发贴总数。

```
if (uu > 0) {
    OP.UpdateTotalTopic(forumid);
    OP.UserTotalPost(userid);
    PageForward = "postsucceeded";
    request.setAttribute("topicid", String.valueOf(lastN));
}
else {
```

- ❑ 如果插入失败，返回一条出错信息。

```
PageForward = "postfailed";
request.setAttribute("throw", new String("发贴失败! "));
}
}
```

- ❑ 如果 action 值为 1，说明要执行插入普通主题的操作。同样要获取当前要插入主题的编号。

```
if (action.equals("1")) {
```

```
rs = conn.query("SELECT * FROM posts WHERE istopic>0 AND forumid=" +
                forumid);

rs.last();
int lastNo = rs.getRow() + 1;
rs.close();
```

- ❑ 定义预处理语句，将主题内容写入数据库中。

```
String psql2 =
    "INSERT INTO posts (forumid,groupid,istopic,topic_id,title,content,author," +
    "author_id,post_ip,post_time,last_postor,last_time)VALUES(?,?,?,?,?,?,?,?,?,?,?)";
pstmt = conn.getPstmt(psql2);
pstmt.setInt(1, forumid);
pstmt.setInt(2, 0);
pstmt.setInt(3, 1);
pstmt.setInt(4, lastNo);
pstmt.setString(5, c_title);
pstmt.setString(6, c_content);
pstmt.setString(7, nickname);
pstmt.setInt(8, userid);
pstmt.setString(9, IP);
pstmt.setString(10, dtime);
pstmt.setString(11, nickname);
pstmt.setString(12, dtime);
uu = pstmt.executeUpdate();
```

- ❑ 插入成功，更新分论坛的主题总数以及用户的发帖总数。

```
if (uu > 0) {
    OP.UpdateTotalTopic(forumid);
    OP.UserTotalPost(userid);
    PageForward = "postsucceeded";
    request.setAttribute("topicid", String.valueOf(lastNo));
}
else {
    PageForward = "postfailed";
    request.setAttribute("throw", new String("发帖失败！"));
}
}
```

- ❑ action 值为 2 时，说明要执行插入回复内容的操作，直接将内容写入数据库即可。

```
if (action.equals("2")) {
    String psql3 =
        "INSERT INTO posts (forumid,groupid,istopic,topic_id,title,content,author," +
        "author_id,post_ip,post_time) VALUES(?,?,?,?,?,?,?,?,?)";
    pstmt = conn.getPstmt(psql3);
    pstmt.setInt(1, forumid);
    pstmt.setInt(2, 0);
    pstmt.setInt(3, 0);
    pstmt.setInt(4, topicid);
    pstmt.setString(5, c_title);
    pstmt.setString(6, c_content);
    pstmt.setString(7, nickname);
    pstmt.setInt(8, userid);
```

```
pstmt.setString(9, IP);
pstmt.setString(10, dtime);
uu = pstmt.executeUpdate();
```

- ❑ 插入成功，更新分论坛的贴子总数以及用户的发贴总数。将回复主题的最后回复者与最后回复时间更新。

```
if (uu > 0) {
    OP.UpdateTotalPost(forumid);
    OP.UserTotalPost(userid);
    String sql3= "UPDATE posts SET click_times=click_times+1,
        reply_times=reply_times+1,last_postor=" +
        nickname + ",last_time=SYSDATE() WHERE topic_id=" + topicid
        + " AND forumid=" + forumid + " AND istopic>0";
    if (conn.update(sql3) > 0) {
        request.setAttribute("topicid", String.valueOf(topicid));
        PageForward = "postsucceeded";
    }
}
else {
    PageForward = "postfailed";
    request.setAttribute("throw",new String("发帖失败！"));
}
}
```

- ❑ 如果 action 值为 4，说明要执行重新编辑贴子后的更新操作。

```
if (action.equals("4")) {
    c_content = c_content + "<FONT color=red>Editted</FONT>";
    String sql4 = "UPDATE posts SET title=" + c_title + ",content=" +
        c_content
        + " WHERE id=" + postid + " AND author=" + nickname + """;
    if (conn.update(sql4) > 0) {
        request.setAttribute("topicid", String.valueOf(topicid));
        PageForward = "postsucceeded";
    }
    else {
        PageForward = "postfailed";
        request.setAttribute("throw",new String("发帖失败！"));
    }
}
catch (Exception e) {
    PageForward = "global";
    request.setAttribute("throw", e.toString());
}
}
```

- ❑ 向操作成功页面传递论坛编号，以便于在该页面停留后自动转向查看发帖页面。

```
request.setAttribute("forumsid", String.valueOf(forumid));
return actionMapping.findForward(PageForward);
}
}
```

如果发帖成功，则页面转向到发帖成功页面 right.jsp，其中构建三个分别转向到论坛首页、主题所

在的论坛、显示所发贴子页面的链接。如果用户在指定时间没有做出选择，则自动转向到用户所发贴子内容的查看页面，其代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<%
```

□ 获取所发贴子所属的论坛、主题的编号，用于定义转向到查看其内容页面的 URL。

```
String forumid=(String)request.getAttribute("forumsid");
String topicid=(String)request.getAttribute("topicid");
String url="#";
if(topicid!=null)
{
    topicid="&topicid="+topicid;
    url="disbbs.do?forumsid="+forumid+topicid;
}
else{
    topicid="";
    url="list-topic.do?forumsid="+forumid;
}
%>
<HTML>
<jsp:include page="include/head.txt"/>
```

□ 如果用户没有做出转向选择，三秒钟后自动转向到查看到查看发贴内容的页面。

```
<META HTTP-EQUIV="REFRESH" CONTENT="3;url=<%=url%>">
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY><TR><TD>
<jsp:include page="/include/toplogo.jsp"/><BR>
<!-- *****-->
<TABLE width="98%" align=center cellPadding=1 >
<TBODY><TR>
<TD height="23" align=left background="images/head.gif" class=head>
<B>本页主题: 发贴成功! </B></TD> </TR>
<TR bgcolor="#ADD0EC"> <TD height=151 align=right class=cbg>
<div align="left"> <DIR>
```

□ 构建三个转向链接。

```
<LI><A href=default.jsp class="style2"> 返回本站论坛首页</A>
<LI><A href='list-topic.do?forumsid=<%=forumid%>' class="style2">返回 POST 所在的分论坛</A>
<LI><A href='<%=url%>' class="style2">进入您刚才所发 POST</A>
</DIR> </div></TD>
</TR><TR bgcolor="#ADD0EC">
<TD height=25 align=right background="images/head.gif" class=cbg>&nbsp;  </TD>
</TR></TBODY></TABLE>
<!-- *****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>
```

发贴成功后，转向到发贴成功页面，其视图效果如图 31.21 所示。

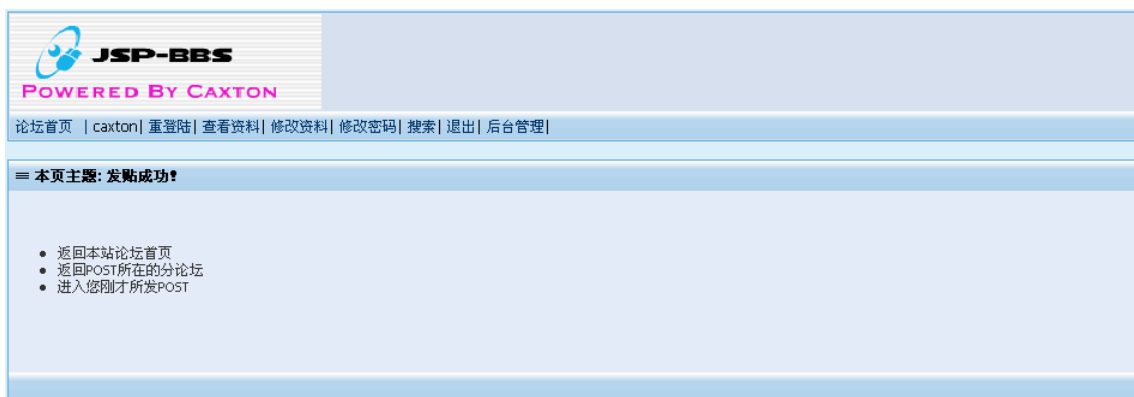


图 31.21 发帖成功

31.4.3 处理文件上传

如果用户在发帖时提交了附件，程序就会调用 `uploadFile` 的方法将文件存入指定目录。并将文件的描述信息添加到贴子的内容中去。最后根据文件的类型将其定位到贴子中。

`uploadFile.java` 实现文件的上传，并将文件定位到贴子中，代码如下。

```
package mybbs;
import org.apache.struts.upload.FormFile;
import java.io.*;
public class uploadFile {
    private String filetext="";
    public uploadFile(){}
    public void UPLOADFILE(FormFile file,String theText)
    {
```

❑ 获取文件的大小，判断是否超出了允许的范围。

```
int len=file.getFileSize();
if(!((file.toString()).equals(""))&&(len<params.Upload_Size)){
    try {
```

❑ 获取文件保存的路径，将文件内容写入到指定文件夹中。

```
String path1=params.UPLOAD_PATH;
InputStream input = file.getInputStream();//能从 FormFile 中获得输入流
String path=path1+file.getFileName();
OutputStream output = new FileOutputStream(path);
int bytesRead = 0;
byte[] readbuffer = new byte[len];
while ( (bytesReader = input.read(readbuffer, 0, len)) != -1) {
    output.write(readbuffer, 0, bytesRead);
}
output.close();
}
catch (Exception e) {
    e.printStackTrace();
}
```

❑ 如果没有文件描述信息，则以“相关附件”代之。

```
if(theText.equals("")){
```

```

        theText="相关附件:<BR>";
    }

```

- ❑ 获取文件的类型，以准确将其定位到内容中去。

```

        String FT=file.getContentType();
        if(FT.equals("image/jpeg")||FT.equals("image/gif")||FT.equals("image/bmp")){
            filetext=filetext+"<BR>"+theText+"<BR><IMG src=upload/"+file.getFileName()+ " ></IMG>";
        }
        else{
            filetext=filetext+"<BR><A href=upload/"+file.getFileName()+ " >"+theText+"</A>";
        }
    }
}

```

- ❑ 附件超出大小范围时，显示附件上传不成功。

```

else{
    filetext=filetext+"<BR><FONT color=blue>没有附件或文件超出限制大小！ </FONT>"+len;
}
}
public String getFiletext(){
    return this.filetext;
}
}

```

在上传附件时，程序会判断附件的类型，利用不同的 HTML 代码定位这些文件。

31.4.4 处理发贴操作的Bean

用户在发贴时，如果贴子内容插入数据库成功，则需要更新论坛的贴子、主题总数以及用户的发贴总数。这些更新操作都是在 Operate 这个类中构造的，Operate.java 代码如下。

```

package mybbs;
import java.sql.*;
public class Operate {
    public Operate(){}

    ❑ 在用户查看主题的内容与回复时，将主题的点击量增加 1。

    public void UpdateClick_time(String fid,String tid){
        DBConn conn=new DBConn();
        try {
            String sql2 = "UPDATE posts SET click_times=click_times+1 WHERE forumid=" +
                fid + " AND topic_id=" + tid + " AND istopic>0";
            conn.update(sql2);
        }
        catch (Exception e) {
            System.out.print("点击量增加不成功: "+e.toString());
        }
    }
}

```

- ❑ 将对应论坛的贴子总数加 1。

```

public void UpdateTotalPost(int fid){
    DBConn conn=new DBConn();
    try {
        String sql3 =
            "UPDATE forums SET total_post=total_post+1 WHERE id=" + fid;
    }
}

```



```
        conn.update(sql3);
        conn.DBclose();
    }
    catch (Exception e) {
        System.out.print("话题总数增加不成功: "+e.toString());
    }
}
```

❑ 将对应论坛的主题总数加 1。

```
public void UpdateTotalTopic(int fid){
    DBConn conn=new DBConn();
    try {
        String sql4 =
            "UPDATE forums SET total_topic=total_topic+1 WHERE id=" + fid;
        conn.update(sql4);
        conn.DBclose();
    }
    catch (Exception e) {
        System.out.print("话题总数增加不成功: "+e.toString());
    }
}
```

❑ 将用户的发贴总数加 1。

```
public void UserTotalPost(int userid){
    DBConn conn=new DBConn();
    try {
        String sql5 =
            "UPDATE bbsuser SET totalpost=totalpost+1 WHERE id=" + userid;
        conn.update(sql5);
        conn.DBclose();
    }
    catch (Exception e) {
        System.out.print("用户发表话题总数增加不成功: "+e.toString());
    }
}
```

31.4.5 回复主题

回复主题是对某一主题发表个人看法，所发内容隶属于该主题。在主题列表显示时，会依次显示在主题的后面。回复的用户界面同发表主题是一样的，最后处理写入数据库也是由 **TopicAction** 来完成，只是在写入数据库时由程序自动判断并写入可以区分二者的字段值。

本系统主题必须有要有标题，回复可以有标题，也可以没有标题。用户对某一主题回复成功后，会自动对用户发贴总数、分论坛总帖子数、所回复主题的点击数量、回复数量、最后回复作者、最后回复时间进行更新。

31.4.6 重新编辑发表的内容

用户可以对自己所发表的贴子进行编辑，管理员可以对任何贴子进行编辑。对发贴进行编辑也就是

要改动原来的内容。同修改用户资料相同，系统会调出要编辑贴子的内容，以可修改的形式显示在与发贴界面相同的表单文本区内，用户可对其进行修改后提交。最终会由 TopicAction 处理并更新数据库。

modifyPost.java 实现获取指定贴子的内容，代码如下。

```
package mybbs;
import java.sql.*;
public class modifyPost
{
    String userid;
    String title,content,topicid,forumid,postid;
    public void modifyPost(){}
}
```

❑ 设置要获取贴子的记录编号，以在数据库中惟一确定这条记录。

```
public void setPostid(String postid)
{
    this.postid=postid;
}
```

❑ 获取贴子的标题，同时获取其他一些内容。

```
public String getTitle(){
    try{
        DBConn conn=new DBConn();
        String sql="SELECT * FROM posts WHERE id="+postid;
        ResultSet rs=conn.query(sql);
        rs.next();
        content=rs.getString("content");
        title=rs.getString("title");
        topicid=rs.getString("topic_id");
        forumid=rs.getString("forumid");
        rs.close();
        conn.DBclose();
    }
    catch(Exception e){}
    return title;
}
```

❑ 获取贴子的文本内容。

```
public String getContent()
{
    return content;
}
```

❑ 获取贴子所属的分论坛与所属的主题编号。

```
public String getForumid()
{
    return forumid;
}

public String getTopicid()
{
    return topicid;
}
}
```

在重新编辑贴子时，向 `post.do` 传递一个贴子记录号参数，通过 `modifyPost` 获取该贴子的内容显示于文本区中。必须是首先获取标题，这样其他值才能得到。

31.4.7 发起投票主题

发表投票主题，是在发表主题的同时，发表与该主题对应的投票项目。它比发表普通主题多了两个参数，一个是投票的方式，另一个是以换行相隔的投票条目。系统提供了四种投票方式，投票的条目可以全部输入在“投票选项”中，每一个选项为一行。如果用户要发投票主题，系统会根据 `action` 的值由 `postbbs.jsp` 加载 `postform.jsp` 自动构建发表投票主题的视图，如图 31.22 所示。

The image shows a web form for creating a voting topic. It has several sections: '标题' (Title) with a text input; '投票模式' (Voting Mode) with radio buttons for '不记名单选' (Anonymous Single), '记名单选' (Named Single), '不记名复选' (Anonymous Multiple), and '记名复选' (Named Multiple); '投票选项' (Voting Options) with a large text area and a note '每一个选项为一行，按Enter换行，系统将以换行符区分各选项！'; 'Icon' with a row of 16 different icons; '论坛代码' (Forum Code) with radio buttons for '提示插入' (Prompt Insert), '直接插入' (Direct Insert), and '帮助信息' (Help Information); '内容' (Content) with a note 'HTML 代码不可用' and a link '点击这里看使用方法'; '代码选项' (Code Options) with checkboxes for '自动分析url', '使用签名', 'Code自动转换', and '新回复邮件通知'; '附件' (Attachment) with a note '附件最大:1000 (K)', a file type list '有效文件类型:gif jpg rar txt zip swf jqr fla', a description field, and a '浏览...' button; and finally '提交' (Submit) and '预览' (Preview) buttons at the bottom.

图 31.22 发投票主题界面

用户提交后，处理程序会自动以一个字符串的形式获取投票的选项。最后将这些选项从字符串中分离出来，分别做为一个记录写入 `vote` 表中。

31.4.8 用户投票处理

系统有四种投票方式，但处理投票写入数据库只有单选投票和多项投票两种不同的处理方式。因为记名和不记名只在输出时进行控制。单选投票只是将用户所选的项的结果加 1，多项投票是将所有用户所选的项结果加 1。无论是多选还是单选，用户都只能投票一次。

`VoteSev.java` 实现用户投票后更新数据库投票结果，可以处理单项投票和多项投票。

```
package mybbs;
import java.io.*;
import java.io.IOException;
import javax.servlet.http.HttpServlet;
```

```

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class VoteSev extends HttpServlet {
    PrintWriter out;
    int flag=0;
    String pp,keys,postid,forumid,topicid;
    public void doPost(HttpServletRequest request,
                        HttpServletResponse response)
        throws IOException
    {
        response.setContentType("text/html;charset=gbk");
        out = response.getWriter();
        HttpSession session = request.getSession();

```

- ❑ 如果用户尚未登录，不能投票。否则获取用户的昵称和投票内容。

```

    if(session.getAttribute("login")!=null){
        String nickname=(String)session.getAttribute("nickname");
        keys=request.getParameter("key");
        forumid=request.getParameter("forumsid");
        topicid=request.getParameter("topicid");
        //CAEncode enco=new CAEncode();
        pp=request.getParameter("p");
        postid=request.getParameter("postid");
        int kkey=Integer.parseInt(keys);
        VoteAct vot=new VoteAct();

```

- ❑ 如果投票模式识别符值小于 3，即单项投票，更新数据库中该选项投票结果，返回更新是否成功的状态值 flag。单项投票是在 voting()方法内部判断用户是否已投票。

```

        if(kkey<3){
            if(vot.voting(nickname,pp,postid)==1)
                flag=1;
        }

```

- ❑ 如果投票模式识别符值大于 2 也就是多项投票，这里首先判断用户是否已投过票。

```

        if(kkey>2){
            if(!vot.Hadvoted(nickname,postid)){

```

- ❑ 如果用户未投票，将投票选项结果逐个更新并返回是否更新成功状态值。

```

                String[] res= request.getParameterValues("checkb");
                for(int i=0;res!=null&&i<res.length;i++){
                    flag=vot.vott(nickname,res[i],postid);
                }
            }
        }
    }

```

- ❑ 如果更新成功，自动返回查看投票结果页面，否则输出错误信息。

```

        if(flag==1){
            String RLL="disbbs.do?forumsid="+forumid+"&topicid="+topicid;
            response.sendRedirect(RLL);
        }
        else{
            out.print("您已经投过票了，也有可能你非法操作！");

```

```

    }
}
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws IOException
{
    doPost(request, response);
}
}

```

31.4.9 处理投票的Bean

处理投票的 Bean 中封装了从一个整字符串中分析出所有投票选项的方法 Setvote(), 将投票内容写入数据库的方法 upvote(), 判断用户是否已投票的方法 HadVoted()等。

VoteAct.java 实现了投票操作所有处理方法, 代码如下。

```

package mybbs;
import java.sql.*;
import mybbs.DBConn;
import java.util.*;
public class VoteAct
{
    public void VoteAct(){}
}

```

❑ 分析投票选项, 将结果存入数组中。

```

public String[] Setvote(String items){
    StringTokenizer fenxi=new StringTokenizer(items,"#");
    String b[]=new String[fenxi.countTokens()];
    int n=0;
    while(fenxi.hasMoreTokens()){
        b[n]=fenxi.nextToken();
        n++;
    }
    return b;
}

```

❑ 将投票内容写入数据库。

```

public void upvote(String voteitem,String postid,String title,String mode){
    try{
        DBConn conn=new DBConn();
        String CBB[]=this.Setvote(voteitem);
        conn.update("INSERT INTO vote(postid,iskey,keyid,item) VALUES("+postid+", "+mode+",0,"+title+"");
        for(int u=0;u<CBB.length;u++){
            if(!CBB[u].equals("null")){
                int M=u+1;
                conn.update("INSERT INTO vote(postid,iskey,keyid,item,voter)
                            VALUES("+postid+",0,"+M+", "+CBB[u]+"', '#')");
            }
        }
    }
    catch(Exception e){

```

```
        System.out.print(e.toString());    }  
    }
```

❑ 利用用户昵称判断用户是否已投票。

```
public boolean Hadvoted(String user,String postid){  
    String username=user;  
    String postidn=postid;  
    boolean voted;  
    try{  
        DBConn cc=new DBConn();  
        ResultSet rs=cc.query("SELECT * FROM vote WHERE postid="+  
                                postidn+" AND voter REGEXP '"+username+"#");  
  
        if(rs.next())  
            voted=true;  
        else  
            voted=false;  
        rs.close();  
    }catch(Exception e){  
        System.out.print(e.toString());  
        voted=false;  
    }  
    return voted;  
}
```

❑ 将用户投票结果写入数据库。

```
Public int vott(String user,String keyid,String postid){  
    String keyid3=keyid;  
    String user3=user;  
    String postid3=postid;  
    int ii=0;  
    try{  
        DBConn con=new DBConn();  
        con.update("UPDATE vote SET result=result+1 WHERE postid="+postid3+" AND iskey>0 AND islock=0");  
        ii=con.update("UPDATE vote SET result=result+1,voter=CONCAT(voter,'" +user3+"', '#') WHERE  
                        postid="+postid3+" AND iskey=0 AND keyid='"+keyid3);  
    }  
    catch(Exception e){  
        System.out.print(e.toString());  
    }  
    return ii;  
}
```

用户投票成功后,将用户的昵称写入到所投票选项的 voter 列中。多选则写入每一个项目对应的 voter 列中。在判断用户是否已投票时,只需要检索用户的昵称是否在这些记录的 voter 列中存在,如果已存在,说明用户已经投票。

31.4.10 显示投票

显示投票是在用户查看投票贴子内容时,在顶端显示投票的结果,用彩色条形表示各选项的得票结果。在显示投票结果的同时构建用于投票的表单,已登录且没有进行过投票的用户可以直接选择要投票

的选项后。单击“投票”提交投票结果，程序在处理后会自动返回这个页面，用户可以即时看到投票结果。图 31.23 是查看投票主题时显示的投票界面。



图 31.23 用户投票与显示投票结果

显示投票界面就是如果主题的 `istopic` 值为 2 时，说明是投票主题，从数据库中读取与其相关的投票项目并显示在主题的顶端。

`DisVote.java` 是一个专门用于显示投票结果的类，将投票结果以百分比的形式格式化输出，它同时构建了一个用于投票的表单。

```
package mybbs;
import java.sql.*;
import java.util.*;
public class DisVote{
    String postid;
    public void DisVote(){
    public StringBuffer ShowVote(String postid){
        StringBuffer buffer=new StringBuffer();
        this.postid=postid;
        int total=0,iskey=0;
        try{
            DBConn conn=new DBConn();
```

- ❑ 从 `vote` 表中检索与主题相对应的投票标题项目，也就是 `iskey` 不等于 0 的指定记录。找到该记录后获取记录中 `iskey` 的值，用于在后面判断投票的模式。获取 `result` 的值，它是项目总的投票数，用于在后面计算各选项的得票率。

```
String sql1="SELECT * FROM vote WHERE iskey>0 AND postid="+postid;
ResultSet rs1=conn.query(sql1);
if(rs1.next()){
    iskey=rs1.getInt("iskey");
```

```
total=rs1.getInt("result");
}
rs1.close();
```

- ❑ 检索投票项目对应的选项条目。

```
String sql2="SELECT * FROM vote WHERE iskey=0 AND postid="+postid;
ResultSet rs2=conn.query(sql2);
int j=1;
while(rs2.next()){
```

- ❑ 如果是单项投票，构建单选框。

```
if(iskey<3){
    buffer.append("<TR><TD width='6%' align=right><INPUT type='radio' name='p' value='"+
        rs2.getString("keyid")+" "></TD>");
}
else{
```

- ❑ 如果是复选投票，构建复选框。

```
buffer.append("<TR><TD class=cbg><input type=checkbox name='checkb' value='"+
    rs2.getString("keyid")+" "></TD>");
}
```

- ❑ 显示投票选项描述内容。

```
buffer.append("<TD class=cbg width='34%'>"+rs2.getString("item")+"</TD>");
int n1=0;
float n2=0;
```

- ❑ 计算各选项的得票率，显示结果时体现在图片的长度上。

```
if(rs2.getInt("result")!=0){
    n1=rs2.getInt("result")*100/total;
    n2=(float)rs2.getInt("result")*100/total;
}
buffer.append("<TD class=cbg width='60%'><IMG src='images/bar"+j
    +".gif' height=10 width='"+n1+"%'>"+n2+"%</TD>");
```

- ❑ 如果是记名投票，从 voter 列中获取投票用户名单，分析后以下拉列表的形式显示在用户所投选项条目的后面。

```
if(iskey==2||iskey==4){
    StringTokenizer fenxi=new StringTokenizer(rs2.getString("voter"),"#");
    buffer.append("<TD class=cbg><SELECT>");
    while(fenxi.hasMoreTokens()){
        buffer.append("<option value='"+fenxi.nextToken()+">");
    }
    buffer.append("</SELECT></TD>");
}
buffer.append("</TR>");
buffer.append("<INPUT TYPE=hidden name=key value='"+iskey+" ">");
j++;
if(j==5)
    j=1;
}
rs2.close();
conn.DBclose();
}
catch(Exception e){ }
```



```
return buffer;  
}  
}
```

本系统所构建的投票可以是一个独立的模块在网站中使用，为多种方式投票，随意定义投票项目，分条目提供了解决方案。它可以实现不限制投票的选项数目，为使用者提供较大的灵活性。另处在论坛中使用，也使得用户在发表投票主题是易于操作。

31.5 论坛管理

本系统提供了简单的论坛管理处理方法，只涉及了一些常用的管理操作。论坛管理主要分为前台管理和后台管理。前台管理主要是针对用户发贴的管理，如置顶、加为精华等。后台管理主要是针对分论坛属性以及用户的管理。如添加分论坛，锁定用户等。

31.5.1 前台管理

管理员和分论坛的版主有对用户发贴进行管理操作的权限。如果版主登录后进入自己所管的分论坛，在查看主题内容时，会在顶端构建显示一个管理工具条。版主和管理员由于权限不同，管理员会多出一个“设为公告”的操作链接。如果要对当前主题做相应的操作，可以点这些链接进行操作。图 31.24 是管理员的管理工具条。

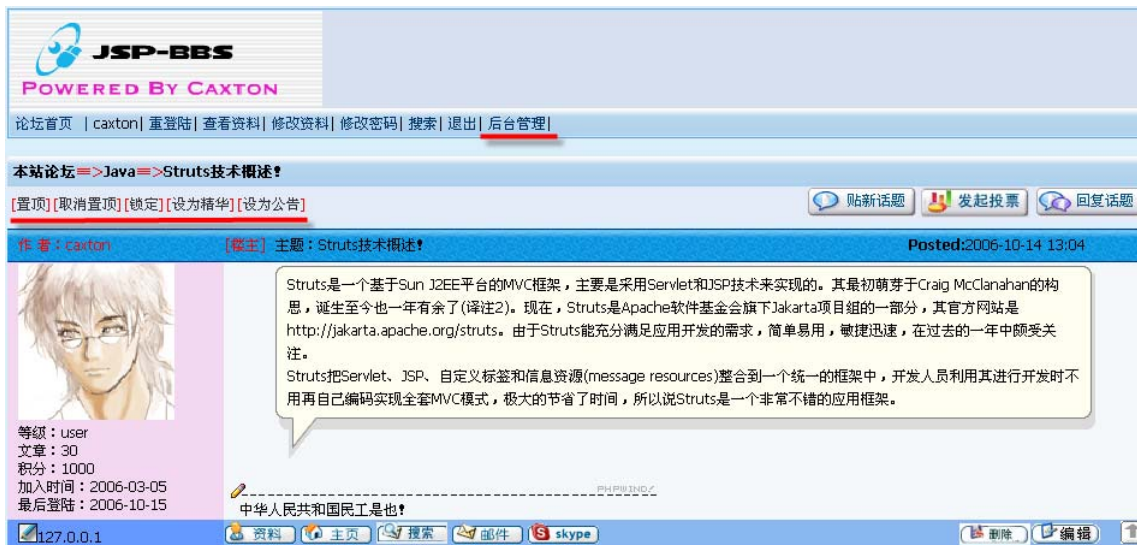


图 31.24 前台管理链接

管理工具条是由 AdminItems 这个 Servlet 来显示的，其中要对用户的权限进行判断。根据权限的不同显示不同的管理工具条，普通用户显示“普通用户”，不能进行任何操作。

AdminItems.java 构建管理工具条，代码如下。

```
package mybbs;  
import javax.servlet.http.HttpSession;  
public class AdminItem {  
    HttpSession session;  
    public AdminItem() {}  
}
```

```

public StringBuffer ShowAdminItem
    ( HttpSession session,int id,String forumid,int groupid,int islock)
{
    StringBuffer buffer=new StringBuffer();
    this.session=session;
    int power=0;
    String forum="";

```

- 如果是登录用户获取其权限与管理论坛的编号。

```

    if(session.getAttribute("userpower")!=null){
        power=Integer.parseInt((String)session.getAttribute("userpower"));
    }
    if(session.getAttribute("forum")!=null){
        forum=(String)session.getAttribute("forum");
    }

```

- 判断如果是管理员或当前论坛的版主，显示以下链接。

```

    if(power>10||forum.equals(forumid))
    {

```

- 本系统采用的是可连续置顶以改变主题排序的方式来进行置顶操作。因此“置顶”和“取消置顶”同时显示。

```

        buffer.append("&nbsp;[<A href=adminAction.do?motion=1&postid="+id+"&forumsid="+forumid
            +"]>置顶</A>]");
        buffer.append("&nbsp;[<A href=adminAction.do?motion=11&postid="+id+"&forumsid="+forumid
            +"]>取消置顶</A>]");

```

- 如果该主题已被锁定，显示“取消锁定”操作链接，否则显示“锁定”链接。

```

        if(islock==0){
            buffer.append("&nbsp;[<A href=adminAction.do?motion=3&postid="+id+"&forumsid="+forumid
                +"]>锁定</A>]");
        }
        if(islock!=0){
            buffer.append("&nbsp;[<A href=adminAction.do?motion=31&postid="+id+"&forumsid="+forumid
                +"]>取消锁定</A>]");

```

- 如果该主题已被设为精华，显示“取消精华”操作链接，否则显示“设为精华”链接。

```

        }
        if(groupid==0){
            buffer.append("&nbsp;[<A href=adminAction.do?motion=4&postid="+id+"&forumsid="+forumid
                +"]>设为精华</A>]");
        }
        if(groupid==1){
            buffer.append("&nbsp;[<A href=adminAction.do?motion=41&postid="+id+"&forumsid="+forumid
                +"]>取消精华</A>]");
        }

```

- 如果当前用户是管理员，该主题已被设为公告的情况下，显示“取消公告”操作链接，否则显示“设为公告”链接。

```

        if(power>10&&groupid==0){
            buffer.append("&nbsp;[<A href=adminAction.do?motion=5&postid="+id+"&forumsid="+forumid
                +"]>设为公告</A>]"); }
        if(power>10&&groupid==2){
            buffer.append("&nbsp;[<A href=adminAction.do?motion=51&postid="+id+"&forumsid="+forumid

```

```
+ ">取消公告</A>]"); }
```

```
}
```

- 如果当前用户是普通用户或没有登录，显示“普通用户”。

```
else{
    buffer.append("[普通会员]");
}
return buffer;
}
}
```

对主题进行的管理操作除了上面所列的对主题的操作以外，还有在贴子下方显示的“删除”图标，用于对任何一个贴子进行删除操作。在数据库中将 islock 设为 1，这样在查看主题内容时就不会显示。主题如果 islock 值为 1，在主题列表中也不会列出。

管理员和版主进行主题或发帖管理操作最终是由 AdminAction 来完成的，根据操作符的不同，分别处理以上所有的前台管理操作，AdminAction.java 代码如下。

```
package mybbs;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.Action;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.DynaActionForm;
import org.apache.struts.action.ActionForward;
import org.apache.struts.action.ActionMapping;
import javax.servlet.http.HttpSession;
import java.sql.*;
public final class AdminAction extends Action {
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        CAEncode encode = new CAEncode();
        String PageForward;
        HttpSession session = request.getSession(true);
        int power = 0;
        String forum = "";
```

- 获取用户权限与版主对应的分论坛编号。

```
if (session.getAttribute("forum") != null) {
    forum = (String) session.getAttribute("forum");
}
if (session.getAttribute("userpower") != null) {
    power = Integer.parseInt( (String) session.getAttribute("userpower"));
}
try {
    DBConn conn = new DBConn();
```

- 获取要管理主题或贴子的一些属性与管理的类型。

```
DynaActionForm userform = (DynaActionForm) form;
String motion = (String) userform.get("motion");
String postid = encode.htmlEncode( (String) userform.get("postid"));
```

```
String topicid = (String) userform.get("topicid");
String forumid = (String) userform.get("forumsid");
/*-----
*1:置顶 2:删除 3: 锁定 4:加精 11:取消置顶      *
-----*/
```

❑ 判断用户权限否可以进行如下操作，如果可以则更新数据库。

```
if (power > 10 || forum.equals(forumid)) {
    if (motion.equals("1")) {
        conn.update("UPDATE posts SET istop=istop+1 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("11")) {
        conn.update("UPDATE posts SET istop=0 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("2")) {
        conn.update("UPDATE posts SET isdel=1 WHERE id="+postid);
    }
    if (motion.equals("21")) {
        conn.update("UPDATE posts SET isdel=0 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("3")) {
        conn.update("UPDATE posts SET islock=1 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("31")) {
        conn.update("UPDATE posts SET islock=0 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("4")) {
        conn.update("UPDATE posts SET groupid=1 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("41")) {
        conn.update("UPDATE posts SET groupid=0 WHERE istopic>0 AND id="+postid);
    }
}
```

❑ 只有管理员才能进行设为公告与取消公告的操作。

```
if (power > 10) {
    if (motion.equals("5")) {
        conn.update("UPDATE posts SET groupid=2 WHERE istopic>0 AND id="+postid);
    }
    if (motion.equals("51")) {
        conn.update("UPDATE posts SET groupid=0 WHERE istopic>0 AND id="+postid);
    }
}
conn = null;
request.setAttribute("forumsid", forumid);
PageForward = "adminOkay";
}
catch (Exception e) {
```

❑ 将页面重新定向到论坛主题列表。

```
PageForward = "adminFailed";
request.setAttribute("throw", e.toString());
```

```

    }
    return mapping.findForward(PageForward);
}
}

```

上面的程序在处理管理操作时再一次对用户权限进行审核，是为了保证管些操作的安全性。确保无权限用户进行这些操作的无效。

31.5.2 添加论坛分类

添加论坛分类是最常用的后台管理，也可以进行删除分类、添加分类中的分论坛、删除分类中的分论坛等操作。系统在开始运行时往往是没有任何分类和分论坛的，需要管理员去添加。添加论坛分类与分论坛的管理界面如图 31.25 所示。



图 31.25 管理论坛分类

图 31.24 中的管理界面是由模板页面加载 JavaBean 和表单页面构成的，用户可以直接在添加分论坛的表单中输入分论坛名，提交添加分类中的分论坛。用户也可以在底部表单中输入论坛分类名，提交添加论坛分类。构成上面视图的 listbbs.jsp 代码如下。

```

<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<%

```

❑ 判断用户是否有权限进行后台操作，如果没有而进入该页面，则直接转向到登录页面。

```

if(session.getAttribute("userpower")==null)
    response.sendRedirect("login.do");
else if(Integer.parseInt((String)session.getAttribute("userpower"))<10)
    response.sendRedirect("login.do");
%>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY> <TR><TD>
<jsp:include page="/include/toplogo.jsp"/>

```

```

<BR>
<!--*****-->
<TABLE width="98%" align=center cellPadding=1 >
<TBODY>
<TR ><TD background="images/head.gif"></TD>
<TD height="23" background="images/head.gif">≡ <B>本页主题: 论坛分类
</B></TD></TR>
<TR bgcolor="#ADD0EC"><TD width=10%></TD>
<TD align=right class=cbg></TD></TR>

```

□ 调用 JavaBean 的方法，显示所有分类和分论坛列表。

```

jsp:useBean id="forum" scope="request" class="mybbs.BBSAdmin" />
<jsp:getProperty name="forum" property="bbs"/>
<TR bgcolor="#ADD0EC"><TD width=10%></TD>
<TD align=left >

```

□ 构建添加论坛分类表单。

```

<FORM action='AddForum.do' method='post'>
添加论坛分类:<BR>
分类名:<INPUT type=text name='forumname' value=''>
<INPUT type=hidden name='action' value='1'>
<INPUT type='submit' value='提交'>
</FORM></TD></TR>
<TR><TD width=10% background="images/head.gif"></TD>
<TD height=23 background="images/head.gif">&nbsp;&nbsp;&nbsp;</TD>
</TR></TBODY></TABLE>
<!--*****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>

```

上面的视图要通过 JavaBean BBSAdmin 来显示论坛分类和分论坛列表。另外在编辑论坛属性时出通过 BBSAdmin 的方法来输出指定分论坛的信息。

BBSAdmin.java 代码如下。

```

package mybbs;
import java.sql.*;
public class BBSAdmin {
    public BBSAdmin(){
    }
}

```

□ 输出所有论坛分类以及分类中的论坛列表。同时构建向分类中添加分论坛的表单。

```

public StringBuffer getBbs(){
    StringBuffer buffer=new StringBuffer();
    DBConn conn =new DBConn();
    try{
        String sql="SELECT * FROM forums WHERE isparent=1";
        ResultSet rs=conn.query(sql);
        while(rs.next()){
            buffer.append("<TR bgcolor='#ADD0EC'><TD width=10%></TD><TD align=left>");
            buffer.append("<FONT color=blue>" +rs.getString("forum_name")+"</FONT>&nbsp;&nbsp;&nbsp;");
            buffer.append("<a href='AddForum.do?action=5&forumid="+rs.getInt("sortid")+"'>");
            buffer.append("<img src='images/del.gif'></a></TD></TR>");
            buffer.append("<TR><TD width=10%></TD><TD align=left>");

```

□ 以论坛分类的编号为参数获取其中的所有分论坛。

- 获取指定分论坛的属性信息，并构建编辑分论坛的表单。

```
public StringBuffer getForumdetail(String id){
StringBuffer buffer=new StringBuffer();
DBConn conn =new DBConn();
try{
    String sql="SELECT * FROM forums WHERE isparent=0 AND id="+id;
    ResultSet rs=conn.query(sql);
    if(rs.next()){
        buffer.append("论坛名称:");
        buffer.append("<INPUT type=text name='forumname' value='"+rs.getString("forum_name")+"'>");
        buffer.append("<BR>论坛描述:");
        buffer.append("<INPUT type=text name='description' value='"+rs.getString("description")+"'>");
        buffer.append("<BR>论坛图标:");
        buffer.append("<INPUT type=text name='logo' value='"+rs.getString("pic_logo")+"'>");
        buffer.append("<BR>管理名单:");
        buffer.append("<INPUT type=text name='admin' value='"+rs.getString("admin_author")+"'>");
    }
}
```

```

    }
}
catch(Exception e){
    buffer.append("目前无分论坛");
}
return buffer;
}
}

```

处理添加论坛分类、添加分论坛以及编辑分论坛内容的操作是由 `AddForumAction` 来完成的。同样根据操作符的不同进行不同操作的处理。

`AddFormAction.java` 实现处理后台论坛管理操作，其代码如下。

```

package mybbs;
import org.apache.struts.action.*;
import javax.servlet.http.*;
import org.apache.struts.action.DynaActionForm;
import java.sql.*;
import javax.servlet.http.HttpSession;
public class AddForumAction extends Action {
    public ActionForward execute(ActionMapping actionMapping,
                                ActionForm form,
                                HttpServletRequest request,
                                HttpServletResponse response) {
        DynaActionForm adminform = (DynaActionForm) form;
        DBConn conn = new DBConn();
        CAEncode ncode = new CAEncode();
        HttpSession session = request.getSession();

```

❑ 获取用户权限，确保只有管理员才能进行操作。

```

String pp = (String) session.getAttribute("userpower");
int power=Integer.parseInt(pp);

```

❑ 获取管理操作中提交的参数值，并对这些参数值进行处理。

```

String forumname = ncode.htmlEncode((String)adminform.get("forumname"));
String fid = ncode.htmlEncode((String)adminform.get("forumid"));
String admin = ncode.htmlEncode((String)adminform.get("admin"));
String logo = ncode.htmlEncode((String)adminform.get("logo"));
String description = ncode.htmlEncode((String)adminform.get("description"));
String action=(String)adminform.get("action");
String PageForward="";
try{
    if(power>10){

```

❑ 如果 `action` 值为 1，则要进行添加论坛分类的操作。将提交的论坛分类名写入数据库中。

```

        if (action.equals("1")) {
            String sql = "SELECT * FROM forums WHERE isparent=1";
            ResultSet rs = conn.query(sql);
            rs.last();
            int maxid = rs.getRow()+1;
            rs.close();
            String sql2 =
                "INSERT INTO forums(isparent,sortid,forum_name)VALUE(?,?,?)";
            PreparedStatement pstmt = conn.getPstmt(sql2);

```



```

pstmt.setInt(1, 1);
pstmt.setInt(2, maxid);
pstmt.setString(3, forumname);
int u = pstmt.executeUpdate();
if (u > 0)
    PageForward = "addsuccess";
else
    PageForward = "addfailed";
pstmt.close();
}

```

- ❑ 如果 action 值为 2，则说明操作是添加分论坛操作，将分论坛名写入数据库。

```

if (action.equals("2")) {
    int forumid = Integer.parseInt(fid);
    String sql2 =
        "INSERT INTO forums(isparent,sortid,forum_name)VALUE(?,?,?)";
    PreparedStatement pstmt = conn.getPstmt(sql2);
    pstmt.setInt(1, 0);
    pstmt.setInt(2, forumid);
    pstmt.setString(3, forumname);
    int u = pstmt.executeUpdate();
    if (u > 0)
        PageForward = "addsuccess";
    else
        PageForward = "addfailed";
    pstmt.close();
}

```

- ❑ 如果 action 值为 3，则进行更新分论坛属性的操作。

```

if (action.equals("3")) {
    int forumid = Integer.parseInt(fid);
    String sql2 = "UPDATE forums SET forum_name=?,admin_author=?,pic_logo=?,description=? WHERE
id="+fid;
    PreparedStatement pstmt = conn.getPstmt(sql2);
    pstmt.setString(1, forumname);
    pstmt.setString(2, admin);
    pstmt.setString(3, logo);
    pstmt.setString(4, description);
    int u = pstmt.executeUpdate();
    if (u > 0)
        PageForward = "addsuccess";
    else
        PageForward = "addfailed";
}

```

- ❑ 如果 action 值为 5，处理删除分论中的分论坛操作。

```

if(action.equals("4")){
    String sql4="DELETE FROM forums WHERE id="+fid;
    conn.update(sql4);
    PageForward = "addsuccess";
}

```

- ❑ 如果 action 值为 6，则处理删除分类的操作，同时删除所有分类中的分论坛。

```

if(action.equals("5")){
    String sql5="DELETE FROM forums WHERE sortid="+fid;
    conn.update(sql5);
    PageForward = "addsuccess";
}
}
else{
    PageForward = "global";
    request.setAttribute("throw", new String("无权操作! "));
}
conn.DBclose();
}
catch (Exception e) {
    PageForward = "global";
    request.setAttribute("throw", e.toString());
}
}

```

❑ 如果操作处理成功，返回分论和分论坛管理页面。

```

return actionMapping.findForward(PageForward);
}
}

```

在管理分类和分论坛后台管理页面（图 31.24 所示）中，如果单击分论中的分论坛，可以进入编辑分论坛属性。如果要删除该分论坛，可以点其后面的图标，实现删除的操作。

31.5.3 编辑论坛属性

论坛后台管理的编辑分论坛页面中，提供了一个可以输入并提交分论坛信息的表单，可以进行编辑分论坛属性操作，如添加版主，编辑论坛描述信息等，如图 31.26 所示。

The screenshot shows a web browser window with the title 'JSP-BBS POWERED BY CAXTON'. The navigation bar includes links for '论坛首页', 'caxton', '重登陆', '查看资料', '修改资料', '修改密码', '搜索', '退出', and '后台管理'. The main content area is titled '主题: 分论坛内容'. Below this, there is a form with the following fields:

- 论坛名称: Java
- 论坛描述: SUN, JSP, Servlet
- 论坛图标: images/forum_logol.gif
- 管理名单: stanley, kaka, royal

At the bottom of the form are two buttons: '提交' (Submit) and '重置' (Reset).

图 31.26 编辑论坛属性

modifyinfo.jsp 构建管理员编辑分论坛属性的视图，主要是一个表单，代码如下。

```

<%@ page contentType="text/html; charset=gbk" %>
<%@ page language="java" %>
<HTML>
<jsp:include page="include/head.txt"/>
<BODY vLink=#333333 link=#333333>
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
<TBODY> <TR><TD>
<jsp:include page="/include/toplogo.jsp"/>

```

```
<BR>
<!--*****-->
```

❑ 判断用户是否有权进行后台操作，如果没有而进入该页面，则直接转向到登录页面。

```
<%
if(session.getAttribute("userpower")==null)
    response.sendRedirect("login.do");
else if(Integer.parseInt((String)session.getAttribute("userpower"))<10)
    response.sendRedirect("login.do");
%>
<TABLE width="98%" align=center cellPadding=1 >
<TBODY> <TR><TD background="images/head.gif" width=10%></TD>
    <TD height="23" align=left background="images/head.gif" class=head>
        ≡ <B>主题: 分论坛内容</B></TD></TR>
    <TR bgcolor="#ADD0EC"><TD class=cbg></TD>
        <TD align=left class=cbg>
            <FORM action='AddForum.do' method='post'>
```

❑ 调用 BBSadmin 的方法获取分论坛的信息，以可修改表单的视图显示。

```
<jsp:useBean id="forum" scope="request" class="mybbs.BBSadmin" />
<%
    String fid=request.getParameter("forumid");
    out.print(forum.getForumdetail(fid));
%>
<INPUT type=hidden name='forumid' value='<%=fid%>'>
<INPUT type=hidden name='action' value='3'><BR>
<INPUT type='submit' value='提交'>
<INPUT type='reset' value='重置'>
</FORM> </TD> </TR>
<TR bgcolor="#ADD0EC"><TD background="images/head.gif" width=10%></TD>
    <TD height=23 align=right background="images/head.gif" class=cbg>&nbsp;&nbsp;&nbsp;</TD>
</TR> </TBODY></TABLE>
<!--*****-->
<BR><CENTER></CENTER><BR></TD></TR></TBODY></TABLE>
<jsp:include page="include/buttom.jsp"/>
</BODY></HTML>
```

管理员输入分论坛相关信息后，提交 AddforumAction 处理，更新 forums 表中的分论坛信息，最终会将页面重新返回管理论坛分类和分论坛页面。

31.5.4 锁定用户

管理员锁定用户，就是将用户表中的 islock 的值设为 1。这样用户就不能正常登录。不能进行发帖等操作。锁定用户的管理界面如图 31.27 所示。



图 31.27 锁定用户界面

图 12.26 所示的管理视图是由模板加载锁定用户的表单构成。在页面底端显示被锁定的用户昵称列表，锁定用户管理界面 lockuser.jsp 代码如下。

```
<%@ page contentType="text/html; charset=gbk" %>
```

```
<%@ page language="java" %>
```

❑ 判断用户是否为管理员，如果不是转向其他页面。

```
<%
```

```
if(session.getAttribute("userpower")==null)
```

```
response.sendRedirect("login.do");
```

```
else if(Integer.parseInt((String)session.getAttribute("userpower"))<10)
```

```
response.sendRedirect("login.do");
```

```
%>
```

```
<HTML>
```

```
<jsp:include page="include/head.txt"/>
```

```
<BODY vLink=#333333 link=#333333>
```

```
<TABLE cellSpacing=0 cellPadding=8 width="98%" align=center bgColor=#dbeffb border=0>
```

```
<TBODY><TR><TD>
```

```
<jsp:include page="/include/toplogo.jsp"/>
```

```
<BR>
```

```
<!--*****-->
```

```
<TABLE width="98%" align=center cellPadding=1 >
```

```
<TBODY><TR><TD background="images/head.gif">≡ <B>本页主题: 锁定用户</B> </TD>
```

```
<TD height="23" background="images/head.gif">
```

```
[<a href='listbbs.do'><FONT color=red>添加分类</FONT></a>]
```

```
</TD></TR><TR bgcolor="#ADD0DC">
```

```
<TD align=right width=30%>选择输入内容</TD>
```

❑ 构建提交要锁定用户识别信息的表单。

```
<TD> <FORM action='LockUser.do' method='post'>
```

```
<INPUT type=radio checked value='userid' name=ref>Userid
```

```
<INPUT type=radio value='nickname' name=ref>Nickname
```

```
</TD></TR>
```

```
<TR bgcolor="#ADD0DC"><TD align=right width=30%>
```

```
锁定用户信息: </TD>
```

```
<TD><INPUT type=text name='var' value='>
```

```
</TD></TR> <INPUT type=hidden name=action value='1'>
```

```
<TR bgcolor="#ADD0DC"><TD align=right width=30%></TD>
```

```
<TD><INPUT type='submit' value='提交锁定'>
```

```
</FORM></TD></TR>
```

```
<TR bgcolor="#ADCCDC"><TD align=right width=10%></TD>
```

- 调用 `JavaBean AdminUser` 的方法获取被锁定用户列表。

在显示被锁定用户列表中，如果想要对某个被锁定用户解除锁定，可以直接单击被锁定用户列表中的用户昵称。直接对该用户进行解除锁定操作，显示被锁定用户列表的 `JavaBean AdminUser` 中主要构造一个获取锁定用户列表的方法，代码如下。

❑ 显示被锁定用户列表，同时构建解除该用户锁定的链接。

处理锁定用户操作与解除用户锁定操作都是 `LockUserAction`。根据操作符 `action` 值的不同，可以进行相应的操作。`LockUserAction.java` 的代码如下。

```
package mybbs;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.*;
import javax.servlet.http.HttpSession;
public final class LockUserAction extends Action {
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
```

```

    HttpServletRequest request,
    HttpServletResponse response) throws Exception {
    CAEncode encode = new CAEncode();
    String PageForward;
    HttpSession session = request.getSession(true);
    int power = 0;

```

- ❑ 获取操作者权限，用于再一次确认用户身份为管理员。

```

    if (session.getAttribute("userpower") != null) {
        power = Integer.parseInt( (String) session.getAttribute("userpower"));
    }
    try {
        DBConn conn = new DBConn();
        DynaActionForm userform = (DynaActionForm) form;

```

- ❑ 获取管理员提交的参数值。

```

        String action = (String) userform.get("action");
        String ref = (String) userform.get("ref");
        String var = encode._conv( (String) userform.get("var"));
        String sql="";

```

- ❑ 如果 ref 的值为 userid，则 var 的值为要锁定用户的编号。如果值为 nickname，则 var 的值为要锁定用户的昵称。

```

        if(ref.equals("userid")){
            sql="UPDATE bbsuser SET islock=1 WHERE id="+var;
        }
        if(ref.equals("nickname")){
            sql="UPDATE bbsuser SET islock=1 WHERE nickname='"+var+"'";
        }
        if (power > 10 ) {

```

- ❑ 如果 action 的值为 1，进行锁定用户操作。

```

            if (action.equals("1"))
                conn.update(sql);

```

- ❑ 如果 action 的值为 2，进行解除锁定用户操作。

```

            if (action.equals("2"))
                conn.update("UPDATE bbsuser SET islock=0 WHERE id="+var);
        }
        conn.DBclose();
        PageForward = "userlocked";
    }
    catch (Exception e) {
        PageForward = "global";
        request.setAttribute("throw", e.toString());
    }
}

```

- ❑ 如果操作成功，将页面定向到锁定用户页面。

```

    return mapping.findForward(PageForward);
}
}

```

31.6 系统全局模块与配置

本章 BBS 系统除了前些节所介绍的专用户处理功能模块的程序外，还有一些在整个系统中用于处理数据库连接、字符重编码、字符过滤的程序，以及一些系统全局统一的常量。本节对这些要在整个系统中用到的程序以及系统的总体配制进行介绍。

31.6.1 系统全局常量

本系统的全局常量主要是一些用于系统运行的数值和字符。如论坛显示时的分页、上传文件保存的文件夹等。将这些类型的常量放在一个类中是为了在系统中更好的管理这些数值，便于修改和维护。本章系统中惟一的一个全局常量类 `params.java` 类代码如下。

```
package mybbs;
public final class params {
    //The global parameters
    public static final int Upload_Size =1024*10000;    //上传文件大小上限
    public static final int Topic_Page_Size =10;        //主题列表时每页显示的主题数量
    public static final int Post_Page_Size =10;         //主题内容与回复列表时每页显示的帖子数
    public static final int MaxInactive =5*60;         //最大不活跃时间
    public static final String UPLOAD_PATH = "D:/mybbs/upload/";    //上传文件路径
}
```

31.6.2 字符的处理

在系统中要对用户提交的字符型参数值进行处理，包括重新编码、字符替换。重新编码是为了防止一些中文字符在定入数据库时发生乱码。字符替换是替换到一些可能用于操作数据库的字符，以及一些 HTML 代码。本系统用于字符处理的是 `CAEncode.java`，其代码如下。

```
package mybbs;
import java.util.*;
public class CAEncode {
    String re, re1;
    public void CAEncode() {}
```

❑ 对字符进行重新编码。

```
public String _conv(String n) {
    byte[] temp;
    try {
        temp = n.getBytes("ISO-8859-1");
        n = new String(temp, "GBK");
    }
    catch (Exception e) {}
    return n;
}
```

❑ 构造用于字符替换的方法，将源字符串中的某个指定字符替换成另一个字符。

```
public String replace(String con, String tag, String rep) {
    int j = 0, i = 0, k = 0;
    String RETU = "";
```

```

String temp = con;
int tagc = tag.length();
while (i < con.length()) {
    if (con.substring(i).startsWith(tag)) {
        temp = con.substring(j, i) + rep;
        RETU += temp;
        i += tagc;
        j = i;
    }
    else {
        i += 1;
    }
}
RETU += con.substring(j);
return RETU;
}

```

- 主要用于将发起投票时将用户提交的换行替换为“#”，以利于对其中的条目进行分离取出。

```

public String repltem(String mm) {
    re1 = replace(mm, "<", "&lt;");
    re1 = replace(re1, ">", "&gt;");
    re1 = replace(re1, "\n", "#");
    re1 = replace(re1, " ", "&nbsp;");
    re1 = replace(re1, "'", "&#39");
    return re1;
}

```

- 将一些 HTML 代码过滤掉，同时将换行和空格替换为 HTML 的换行和空格。

```

public String htmlEncode(String s) {
    re = replace(s, "<", "&lt;");
    re = replace(re, ">", "&gt;");
    re = replace(re, "\n", "<br>");
    re = replace(re, " ", "&nbsp;");
    re = replace(re, "'", "&#39");
    return re;
}
}

```

除了进行以上这些处理外，利用 CAEncode 中的字符替换方法，还可以用于解析论坛使用的自定义标签，在输出时显示为 HTML 代码，以产生用户预期的视图。

31.6.3 Session监听

系统对用户 Session 的动听的监听主要是为了随时更新主户的在线状态。如果用户不正常退出，如 session 由于长时间不活动而过期，这样要由 session 监听器来获取用户 session 被销毁，将数据库中的用户状态更新。另外它也可以将用户登录的时间，与退出的时间记录在日志文件中。

UserListener.java 实现对用户 session 创建与销毁的监听，代码如下。

```

package mybbs;
import javax.servlet.http.*;
import java.util.Date;
public class UserListener implements HttpSessionListener {

```



```

public void sessionCreated(HttpSessionEvent e1){
    □ Session 创建时，设置其最大不活动时间为 5 分钟。
    e1.getSession().setMaxInactiveInterval(params.MaxInactive);
    System.out.print(new Date().toLocaleString()+"用户登陆! ");
}
public void sessionDestroyed(HttpSessionEvent e2){
    String userid=(String)e2.getSession().getAttribute("userid");
    □ session 被销毁时，更新数据库中的状态。
    (new CheckUser()).UserOut(userid);
    System.out.print((String)e2.getSession().getAttribute("nickname"));
    System.out.print(new Date().toLocaleString()+"退出");
}
}

```

如果要对用户当前的位置进行跟踪，可以由 session 属性监听器来完成。利用这些监听程序配合 JSP 中的 application 对象，可以监控所有用户的在线状态。

31.6.4 数据源的配置

本系统采用容器自动管理的数据源，其配置方法是在系统根目录下新建一个 META-INF 目录，在其中新建一个 context.xml 文件，写入以下配制内容。

```

<Context>
  <Resource
    name="jdbc/mybbs"
    type="javax.sql.DataSource"
    password="123456"
    driverClassName="com.mysql.jdbc.Driver"
    maxIdle="10"
    maxWait="5000"
    username="root"
    url="jdbc:mysql://localhost:3306/mybbs?characterEncoding=GBK"
    maxActive="20"/>
    removeAbandoned="true"
    removeAbandonedTimeout="60"
    <!-- Default set of monitored resources
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
  </Context>

```

这样的配置使数据库只对 BBS 系统有效，而不需要对服务器的 server.xml 进行配置。有效的确保了系统的完整性。

31.6.5 系统配置文件

本系统中所有用户的请求都将由 Struts 控制器来统一转发，也就是都是以“.do”结尾。这样就需要对系统的 struts-config.xml 进行配置，其中也包含对 ActionForm Bean 的配置。

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE struts-config PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 1.2//EN"
  "http://jakarta.apache.org/struts/dtds/struts-config_1_2.dtd">

```

```
<struts-config>
  <form-beans>
```

❑ 配置系统要用到的 ActionForm。

```
    <form-bean name="postform" type="mybbs.PostForm"/>
    <form-bean name="registform"
      type="org.apache.struts.validator.DynaValidatorForm">
      <form-property name="username" type="java.lang.String"/>
      <form-property name="password" type="java.lang.String"/>
      <form-property name="password2" type="java.lang.String"/>
      <form-property name="nickname" type="java.lang.String"/>
      <form-property name="sex" type="java.lang.String"/>
      <form-property name="birthyear" type="java.lang.String"/>
      <form-property name="birthmonth" type="java.lang.String"/>
      <form-property name="birthday" type="java.lang.String"/>
      <form-property name="email" type="java.lang.String"/>
      <form-property name="mobile" type="java.lang.String"/>
      <form-property name="skype" type="java.lang.String"/>
      <form-property name="qq" type="java.lang.String"/>
      <form-property name="weblog" type="java.lang.String"/>
      <form-property name="info" type="java.lang.String"/>
      <form-property name="face" type="java.lang.String"/>
      <form-property name="signature" type="java.lang.String"/>
      <form-property name="action" type="java.lang.String"/>
    </form-bean>
    <form-bean name="LogForm"
      type="org.apache.struts.action.DynaActionForm">
      <form-property name="username" type="java.lang.String"/>
      <form-property name="password" type="java.lang.String"/>
      <form-property name="action" type="java.lang.String"/>
    </form-bean>
    <form-bean name="lockForm"
      type="org.apache.struts.action.DynaActionForm">
      <form-property name="var" type="java.lang.String"/>
      <form-property name="ref" type="java.lang.String"/>
      <form-property name="action" type="java.lang.String"/>
    </form-bean>
    <form-bean name="adminform"
      type="org.apache.struts.action.DynaActionForm">
      <form-property name="motion" type="java.lang.String"/>
      <form-property name="postid" type="java.lang.String"/>
      <form-property name="topid" type="java.lang.String"/>
      <form-property name="userid" type="java.lang.String"/>
      <form-property name="forumsid" type="java.lang.String"/>
    </form-bean>
    <form-bean name="addform"
      type="org.apache.struts.action.DynaActionForm">
      <form-property name="action" type="java.lang.String"/>
      <form-property name="forumname" type="java.lang.String"/>
      <form-property name="admin" type="java.lang.String"/>
      <form-property name="logo" type="java.lang.String"/>
```

```

        <form-property name="description" type="java.lang.String"/>
        <form-property name="forumid" type="java.lang.String"/>
    </form-bean>
</form-beans>
<global-forwards>

```

❑ 配置系统 Action 处理结束后的转向映射。

```

        <forward name="addsuccess" path="/listbbs.jsp"/>
        <forward name="updatesuccess" path="/profile.jsp"/>
        <forward name="LoginOkay" path="/default.jsp"/>
        <forward name="NotLogin" path="/login.jsp"/>
        <forward name="LoginOut" path="/default.jsp"/>
        <forward name="postsuccesed" path="/right.jsp"/>
        <forward name="regsuccesed" path="/regsu.jsp"/>
        <forward name="global" path="/global.jsp"/>
        <forward name="adminOkay" path="/right.jsp"/>
        <forward name="userlocked" path="/lockuser.jsp"/>
    </global-forwards>

```

❑ 配置系统的请求处理转发映射关系。

```

<action-mappings>
    <action path="/Login" type="mybbs.LoginAction" name="LogForm" scope="request" validate="true"
    input="/global.jsp"/>
    <action path="/adduser" type="mybbs.AddUserAction" name="registform" scope="request"
    validate="true" input="/global.jsp"/>
    <action path="/ModifyInfo" type="mybbs.AddUserAction" name="registform" scope="request"
    validate="false" input="/global.jsp"/>
    <action path="/ModifyPass" type="mybbs.AddUserAction" name="registform" scope="request"
    validate="false" input="/global.jsp"/>
    <action path="/postup" type="mybbs.TopicAction" name="postform" scope="request"
    validate="true" input="/global.jsp"/>
    <action path="/adminAction" type="mybbs.AdminAction" name="adminform" scope="request"
    validate="true" input="/global.jsp"/>
    <action path="/AddForum" type="mybbs.AddForumAction" name="addform" scope="request"
    validate="true" input="/global.jsp"/>
    <action path="/LockUser" type="mybbs.LockUserAction" name="lockForm" scope="request"
    validate="true" input="/global.jsp"/>
    <action path="/listbbs" forward="/listbbs.jsp"/>
    <action path="/forumdetail" forward="/forumdetail.jsp"/>
    <action path="/login" forward="/login.jsp"/>
    <action path="/regist" forward="/regist.jsp"/>
    <action path="/list-topic" forward="/list_topic.jsp"/>
    <action path="/disbbs" forward="/disbbs.jsp"/>
    <action path="/post" forward="/postbbs.jsp"/>
    <action path="/user_info" forward="/profile.jsp"/>
    <action path="/global" forward="/global.jsp"/>
    <action path="/search" forward="/search.jsp"/>
    <action path="/default" forward="/default.jsp"/>
    <action path="/modifyinfo" forward="/modifyinfo.jsp"/>
    <action path="/modifypass" forward="/modifypass.jsp"/>
    <action path="/lockuser" forward="/lockuser.jsp"/>
</action-mappings>

```

❑ 配置用户提交信息合法性验证的验证规则配置。

```
<plug-in className="org.apache.struts.validator.ValidatorPlugIn">
  <set-property
    property="pathnames"
    value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>
  <set-property property="stopOnFirstError" value="false"/>
</plug-in>
</struts-config>
```

除了以上这些配置文件外，系统还有其他一些配制文件。主要是 web.xml 和 validation.xml 文件。web.xml 中配置了系统用到的 Servlet。validation.xml 中配置了用户提交注册信息时的合法性验证规则。

31.7 小结

本章所实现的 BBS 系统，在编程实现中有一些较大较长的程程序，如处理发帖写入数据库的程序 TopicAction.java，其中定义了处理本系统所有发帖以及编辑贴子后写入数据库的处理方法。另外还有显示论坛主题列表，显示主题和相关回复内容的程序。它们都是本系统的最核心处理程序。另外还有其他一些相同处理逻辑的集成处理程序，读者都应尽可能理解。

BBS 系统中只实现了一些 BBS 的基本功能，如后台管理和用户管理都只提供了一些最常用，可以基本实现论坛维护的管理操作。在学习完本书所有内容后，读者可以根据 Internet 上一些主流论坛的功能和处理方式对论坛进行功能和安全性方面的完善。