****

**企业级框架课程**

**设计报告**

**作业题目：** **陇院论坛系统的设计与实现**

**学 院： 信息工程学院**

**专 业： 计算机科学与技术**

**班 级： 17级（2）班**

**小组成员： 杜晨敏 杨乐 张莉娟**

**授课教师： 李娜**

**完成日期： 2020 年 6 月 20 日**

**1、功能要求:**

根据用户关联的对象是自己、版面、公共信息、帖子。分别根据用户与这些对象联系给出用户功能表，如表1.1所示。

**表1.1** 用户功能表

|  |  |
| --- | --- |
| 功能名称 | 说明 |
| 用户注册 | 用户在论坛填写必要信息后就可以注册为论坛用户，并能在登陆后对注册信息进行更改 |
| 用户登陆 | 注册后的用户可利用用户名和密码在论坛登陆 |
| 浏览其他用户信息 | 用户注册后可以查看其他用户的公开信息 |
| 与其他用户联系 | 可以在论坛内给其他用户私下留言，或直接发送邮件等 |
| 浏览版面 | 用户可以看到以列表形式展现的版面信息和版面说明 |
| 浏览发布公共信息 | 未登陆的用户可以浏览公共信息，登陆后可以发布允许范围内的公共信息 |
| 浏览帖子 | 用户可以以列表形式浏览帖子主题及其全部内容 |
| 发布修改删除帖子 | 用户在登陆后可以发布帖子，修改自己的帖子，被给与版主权限后还可以编辑、修改、删除权限范围内的其他用户帖子 |
| 对帖子特殊操作 | 主要针对版主权限用户，对帖子进行置顶，加精华等特殊操作，是帖子主题不同于其他主题 |

版面、公共信息、帖子都没有和其他对象的关联，都属于被动的对象，是被其他对象所操作的，因此针对它们的功能主要由有其他对象产生。版面和帖子是有一定关系的，可以看做是对帖子内容的归类，类似文件夹与文件的关系，有时候可以认为版面是帖子的一个属性。

**1.1 管理员功能设计**

对于管理员的功能，应该认为管理员是论坛的特殊群体，工作是设置论坛参数规则和维护论坛的正常运转。管理员可以和所以对象产生关联，所以其功能设定也很丰富。管理员功能表，如表1.2所示。

**表1.2** 管理员功能表

|  |  |
| --- | --- |
| 功能名称 | 说明 |
| 用户删除 | 对于一些蓄意破坏的用户，删除其用户，不允许其存在 |
| 用户权限 | 对于一个具有管理能力的用户给予一些特殊的权限，让他们在权限范围内帮助管理和维护论坛 |
| 版面管理 | 版面是论坛的枝干，要设定论坛需要的合适版面，让论坛的功能更明细，分类更规范，方便大家的浏览 |
| 公共信息管理 | 公共信息是向用户展现论坛状态的部分，由于信息的普遍性，其删除和修改权限需要由管理人员来完成 |
| 帖子管理 | 帖子相当于论坛的叶子，是基本数据库，反映了论坛使用者和管理者的水平，所以需要管理人员的全权管理来规范其内容 |
| 设置系统参数 | 系统参数包含的内容很多，直接关系到论坛的正常运行，包括网址在内的各种参数全部要由管理人员统一设置和变更 |

**1.2扩展功能设计**

具备了发帖、浏览帖子、回帖的Web系统就构成论坛了，但是这样的论坛很简单，没有吸引力，用户使用起来也不会很舒服，无论是作为商业还是个人论坛都是没有竞争力的。所以需要为其添加一些更人性化的功能使内容更丰富。扩展功能设计，如表1.3所示。

**表1.3** 扩展功能表

|  |  |
| --- | --- |
| 功能名称 | 说明 |
| 用户控制的自己的信息 | 用户可控制自己的信息是否被公开 |
| 用户登陆添加验证码 | 登陆时需要输入验证码，如登陆信息有误则重新生成验证码。这样可以防止一些软件偷取用户名和密码 |
| 站内短信 | 让两个用户能在论坛内更方便的交流 |
| 帖子评价 | 给予用户一个帖子评价好坏的功能 |
| 帖子发布添加多媒体 | 如何在帖子内使用表情图片，插入flash，音乐等功能 |
| 帖子类别多样化 | 让用户可以发起投票，买卖等功能的帖子 |
| 帖子收藏 | 在论坛中加入搜索但效率却不一定很高，对于好的帖子为了方便用户以后更方便找到，可以收藏帖子 |
| 用户等级 | 建立一个荣誉体系，对于对论坛贡献大的用户给予一定的荣誉称号或者高的论坛等级 |
| 论坛情况显示 | 让用户直接看到一些论坛统计数据，如主题总数，在线人员等 |
| 数据备份 | 论坛最重要的就是数据库，但数据系统的安全并不是100%的，所以为了论坛能持续发展，需要定期对论坛数据进行备份 |
| 非法言论过滤 | 对于一些非法言论或者不良词汇可以在论坛进行过滤设置，一旦有人使用也会被过滤而不至于造成不良影响 |
| 非法IP限制 | 对于一个蓄意捣乱的用户，限制他的IP不能访问论坛，但这个功能有很多的局限性，一般情况下不会使用 |

**2、考核的知识点：**

（1）MyBatis的配置文件；MyBatis的映射文件

（2）MyBatis的关联映射

（3）MyBatis的动态SQL和注解

（4）使用Spring管理Bean

（5）Spring管理数据库事务

（6）SpringMVC的参数绑定

（7）SpringMVC的异常处理和拦截器

（8）SpringMVC的文件上传和国际化

**3、考核标准：（满分100分）**

（1） 能实现所有的功能（需要学生进行演示答辩），并且系统结构设计合理（70）。

（2）能应用到所有的知识点（需要学生进行演示答辩）（+10）。

（3）能应用到课程延伸到的知识内容（需要学生进行演示答辩）（+10）。

（4）提交作业的设计文档（其中要有一定量的系统分析内容，并给出数据库设计、核心配置文件、文件列表清单（列出文件名以及其功能））及核心源代码（+10）。

备注：设计文档格式要求，作业题目（宋体，小二）一级标题（宋体 小四）

正文（宋体，小四，行间距 1.5倍，段前段后均0.5行） 图表（宋体，小五）

**小组组员成绩评定**

班级： 17级计算机科学与技术2班

小组成员： 杜晨敏 杨乐 张莉娟

任务完成情况表

|  |  |  |
| --- | --- | --- |
| 姓名 | 具体任务 | 完成情况（老师填） |
| 杜晨敏 |  |  |
| 杨乐 |  |  |
| 张莉娟 |  |  |

成绩评定表

|  |  |  |
| --- | --- | --- |
| 姓名 | 完成成绩（老师填） | 答辩成绩（老师填） |
| 杜晨敏 |  |  |
| 杨乐 |  |  |
| 张莉娟 |  |  |

**项目完成情况表**

班级： 17级计算机与科学2班 小组成员： 杜晨敏 杨乐 张莉娟

任务分配表

|  |  |  |
| --- | --- | --- |
| 姓名 | 具体任务 | 完成情况 |
| 杜晨敏 | 数据库连接 | 已完成 |
| 杨乐 | 页面设置 | 已完成 |
| 张莉娟 | 文档编辑 | 已完成 |

项目完成情况表

|  |  |  |
| --- | --- | --- |
| 项目功能 | 完成情况 | 备注 |
| 注册登录功能 | 已完成 |  |
| 分页功能 | 已完成 |  |
| 浏览话题 | 已完成 |  |
| 新建话题 | 已完成 |  |
| 删除话题 | 已完成 |  |
| 编辑话题 | 已完成 |  |
| 回复功能 | 已完成 |  |
| 用户管理 | 已完成 |  |

**目录**

[一．绪论 1](#_Toc301)

[二．需求分析 2](#_Toc13158)

[2.1 功能需求 2](#_Toc9389)

[2.2 非功能需求 3](#_Toc4274)

[三．系统设计 3](#_Toc17926)

[3.1系统功能模块划分 3](#_Toc20239)

[3.2 系统模块设计 6](#_Toc14678)

[四．数据库设计 12](#_Toc10512)

[4.1 数据库分析 12](#_Toc21026)

[4.2 数据库表设计 14](#_Toc8797)

[五． 程序结构 16](#_Toc29383)

[六． 核心配置 17](#_Toc5098)

[七．系统实现与测试 29](#_Toc2924)

[7.1 前台页面设计 29](#_Toc15178)

[7.2 后台页面设计 33](#_Toc27167)

[7.3 数据库连接 34](#_Toc13821)

[八．用户管理的详细设计与实现 35](#_Toc20032)

[8.1 用户管理详细设计 35](#_Toc17175)

[九．系统测试 39](#_Toc22559)

[9.1后台子系统主要功能测试 39](#_Toc30988)

[9.2 测试结论 41](#_Toc1578)

[十．总结 41](#_Toc8012)

**陇院论坛系统的设计与实现**

**一．绪论**

校园论坛也就是我们常说的校园BBS，BBS是英文Bulletin Board System的简称。BBS开发技术发展至今，从CGI,ASP，到PHP已经日趋成熟，功能也更加丰富，但携着Sun公司的Java技术所实现的“一次编写，到处运行”的优势，继承这一衣钵的JSP技术越来越受到人们的关注。BBS作为一种对外的展示窗口，进行内外信息交流，已成为大众的广泛需要。为了进行更好的交流，用户想就自己的专业和爱好能和其他的用户进行及时专业的交流，这就有了论坛，这样一来我们就可以在网络这个虚拟的空间中方便地实现交流。Internet上发布信息主要是通过网站来实现的，获取信息是要在网站论坛中和Internet“海洋”中按照一定的检索方式将所需要的信息查找出的。因此论坛建设在Internet应用上的地位显而易见，它已成为现代人沟通和获取信息的重要组成部分，从而倍受人们的重视。现在各个大学网站都有BBS论坛，在BBS上，同学与同学之间，老师之间可以非常轻松的进行交流，有网络的地方，不同的地区和国家的各种不同的人都可加入到BBS上进行交流。这也节省了大量的教育资源和充分的利用教育资源。

校园论坛是通过网络社区，提供给人们一个虚拟的空间释放压力。现代生活中，无论是成年人还是未成年人都面临巨大的压力，在网络这个没有压力的世界，人们能够更好地放松自己，以便更好地投入现实生活。虚拟的社区能够创造一个全新的展现自我的舞台，能够在其中做许多平时不能完成的事情，发泄一下内心的郁闷。校园论坛同时也是一个信息的港湾，它集结了许许多多的内容，在这里青少年可以接触到许多平时难以接触的信息。当代大学生喜爱结交新朋友，寻找有共同兴趣的人交流讨论，校园论坛为这些有共同爱好的年轻人创造了另一片交流的空间，深受大学生青睐，因此网络上涌现了以各所高校为背景的大量优秀大型校园论坛。

随着计算机技术的飞速发展，信息时代的到来，信息改变了我们这个社会。各类行业在日常经营管理各个方面也在悄悄地走向规范化和网络化。为了陇院中老师、学生们更方便地交流，更快捷地获取所需地信息，所以开发此论坛，方便大家的应用。

# 二．需求分析

软件需求分析是软件开发期的第一个阶段，是软件生存周期最重要的一步，是关系到软件开发成败的关键步骤。他在问题定义和可行性研究阶段之后进行。它的基本任务是准确地回答“系统必须做什么？”这个问题。虽然在可行性研究阶段粗略了解了用户的需求，甚至还提出了一些可行的方案，但是可行性研究的基本目的是最小的代价在尽可能短的时间内确定问题是否存在可行的解法，因此许多细节都被忽略了，一个微笑的错漏都可能导致误解或铸成系统的大错，在纠正时付出巨大的代价。因而可行性研究不能代替需求分析，他实际上并没有准确地回答“系统必须做什么？”这个关键问题。

软件需求分析是整个系统开发的基础。在此阶段结束前，系统分析员应该写出软件需求规格说明书，以书面的形式准确地描述软件需求。在此过程中，分析员和用户都是起着关键的、必不可少的作用。

双方确认对软件系统有下述几方面的综合要求：

功能、性能、可靠、可用、出错、接口、约束、逆向以及将来可能发生的。

所谓需求分析，就是要弄清楚问题的要求，我们要清楚地知道对于用户来讲，他要实现什么功能，他希望我们做到什么。在这个阶段与客户的交流十分重要，只有明确了客户的需求，才能制定出正确的计划，才能做出完整的、另其满意的项目，若一开始没有明确客户需求，那我们所追求的目标就是不正确的，制定的计划和努力的方向也随之发生了偏转，最后可能与客户需要的想去甚远，需要重头再来，这样就造成了大量资源、人力、财力的浪费。需求分析完成的好，对今后项目的开发百益而无一害，可以加快设计开发的效率，避免返工，节省时间，避免造成不必要的浪费。

# 2.1 功能需求

指所开发软件系统必须提供的服务，划分出系统必须完成的所有功能。在本论坛系统中，存在下列功能需求：

1. 面向普通用户的功能需求

注册用户：如果用户为非会员用户，通过注册，经审核通过之后成为会员，获得一个登陆身份。

登录：如果用户已经是注册会员，可以进行登录，登陆后实现别的功能。

浏览帖子：用户查看系统中的帖子，了解最新的资讯和各种动态等信息。

发布新帖：用户可以发表一些帖子，来与其他论坛用户共享信息。

回复帖子：对于自己或者别的会员发表的帖子，可以进行回复。

（2）面向管理员的功能需求

管理员登录：对于已经登录的用户，若权限为管理员，可以进行管理员的登录，登陆之后才能有权限进行下一步操作。

帖子管理：管理员可以对发表的帖子进行查询、修改、删除等操作，对好的帖子进行标识，删除或转移不适合的帖子。

模块管理：论坛中的模块，管理员可对其进行添加、修改、删除等操作。

用户管理：用户注册成功后，管理员可以对用户进行添加、删除操作。

（3）面向游客的功能需求

用户可以对论坛进行访问，浏览帖子的功能，但不能参与回复操作，也没有发布帖子的权限。

**2.2 非功能需求**

陇院论坛系统比较重要的一点莫过于界面的设计，一个动漫论坛是否能吸引用户的第一印象取决于该界面是否友好，因此友好的界面也是该系统必不可少的。与其他论坛系统风格不同的是，动漫论坛要有自己独特的二次元风格，让爱好者有亲切、熟悉的感觉。

# 三．系统设计

## **3.1系统****功能模块划分**

本系统有前台会员登录系统，也有后台管理员登录系统。一般的会员和用户只在前台系统登录，只有管理员才能登录后台系统，对整个系统进行操作。

系统结构框图如图3-1所示。

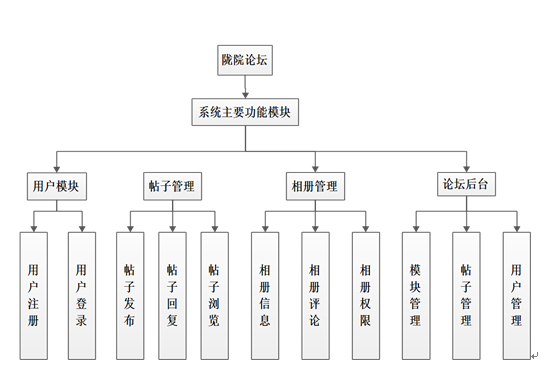


图3-1系统结构图

### 3.1.1 系统前台功能结构设计

在前台会员登录界面，会员通过注册，称为本系统的会员，如果没有注册，只能算是游客身份，在本系统，游客只有浏览帖子和查看帖子回复的权限，所以如果想要发表帖子等权限，就必须通过注册称为本系统的会员。一旦注册成功，成为了会员，=就自动登录到本系统。在登录后，会员可以浏览整个论坛的索引帖子。也可以再子论坛发表自己的帖子，也可以恢复别人发表的帖子。如果会员用户想要修改个人信息的功能，修改完成后可以查看自己个人信息。如果会员用户想要退出本系统，则可以点击退出，会员用户就退出了登录系统，则系统自动跳转到前台用户登录界面。

前台过年好是会员用户登录后能实现的功能，可见的功能模块主要分为：

1. 会员用户登录模块
2. 注册模块
3. 修改用户信息模块
4. 发表帖子模块
5. 浏览，恢复帖子模块
6. 短信息模块

前台系统功能结构图如下图3-2所示：

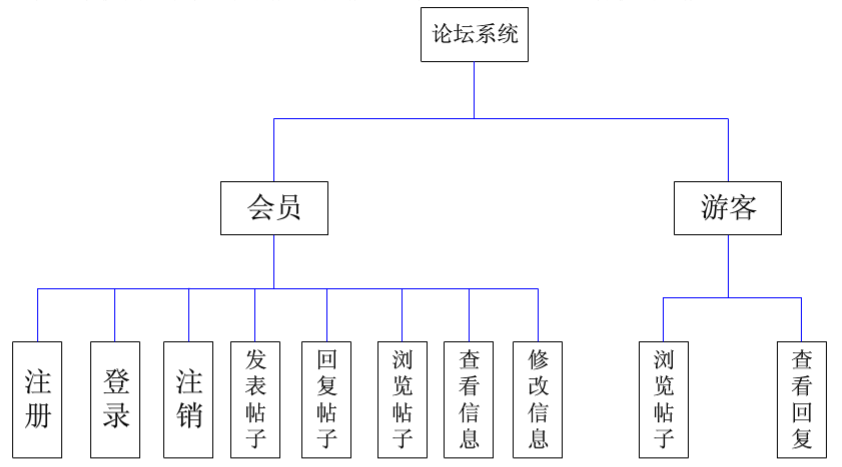


图3-2 前台系统功能结构图

### 3.1.2 系统后台功能结构设计

这里是后台管理员登录系统，在本系统中，管理员登录界面进入管理员管理界面，在管理系统界面，管理员可以对整个论坛进行管理操作其中包括对整个论坛模块的管理，对模块进行增删改查。会员的管理，如有用户违反了论坛规定，管理员有权对该用户进行删除操作。权限管理，如有某个子论坛需要更换版主，则在管理员系统下，可以对子论坛的版主更换进行操作，只要选定要作为版主的用户，点击设为版主，就能改变此会员用户的权限了。公告管理，在整个论坛里，有很多公告，管理员都可以对这些公告进行增加，删除和编辑操作。后台管理的好处就是只要对后台系统的参数进行改变，前台页面就能得到相应的改变，方便快捷。后台功能是管理员登录系统后实现的功能，可见的功能模块主要有：

1. 管理员登录模块
2. 管理首页模块
3. 论坛管理模块
4. 版主管理模块
5. 论坛公告管理模块
6. 用户管理模块
7. 论坛变量管理模块

后台系统功能结构图，如下图3-3所示：

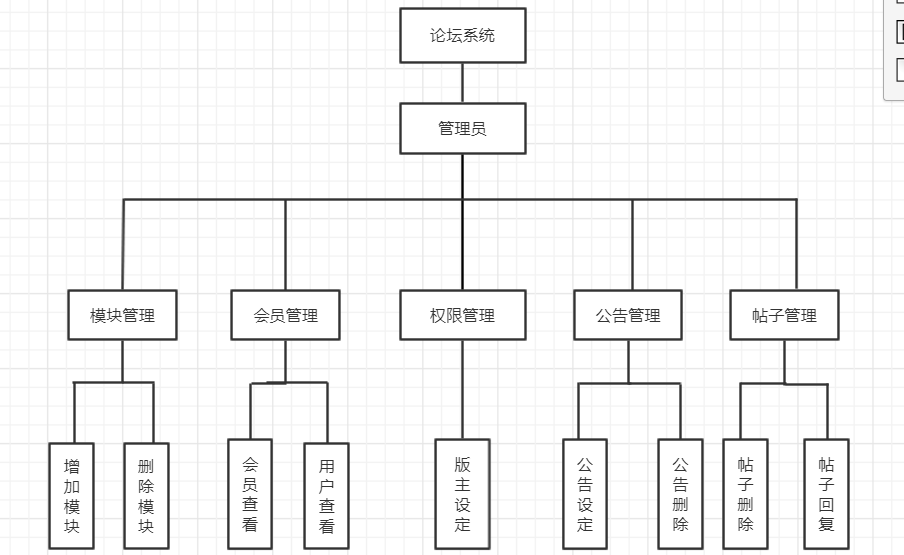


图3-3 后台系统结构图

## **3.2 系统模块设计**

本系统有很多功能，每个功能实现都有自己的流程图，每个流程图都表达具体的实现步骤，下面就具体的功能设计流程图：

### 3.2.1 会员注册

在登录的时候，如果你没有本论坛的用户名和密码，则必须注册，会员注册是在注册界面进行操作的，首先用户在登录界面点击会员注册，进入注册界面，在注册界面用户填写注册信息，如果注册信息合法，则就注册成功，如果不合法就必须重新填写用户注册信息，直到注册成功。成功后便自动跳转到论坛首页。

会员注册流程图如下图3-4所示：

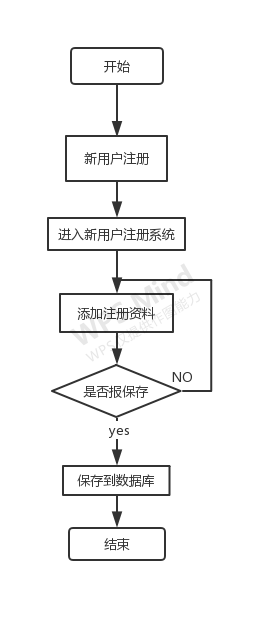
****

图3-4 会员注册流程图

### 3.2.2 会员登录

会员在拥有了论坛登录用户名和密码后，再下一次进入论坛，就可以不用注册了，会员用户在登录界面，输入用户名和密码，点击登录，就能登录了，如果会员用户输入的用户名和密码错误或者输入的信息不合法，则登录失败，页面跳转登录失败页面。会员必须重新登录。

会员登录流程图如图3-5所示：

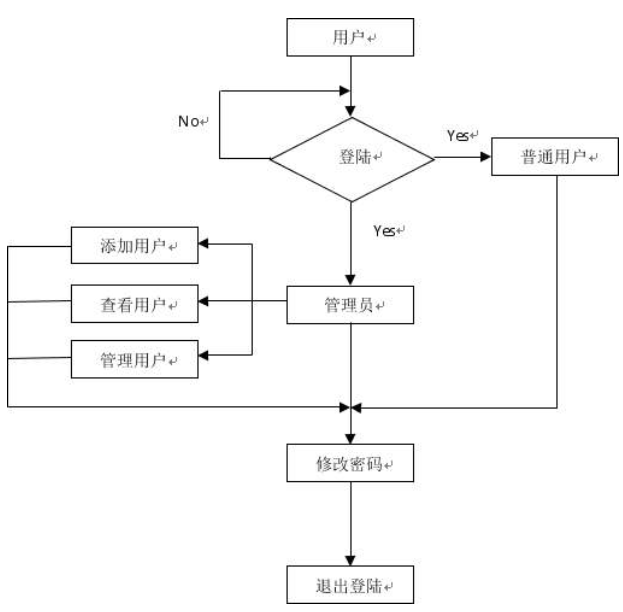


图3-5 登录流程图

### 3.2.3 会员帖子浏览

在会员用户登录成功后，会员就可以浏览论坛的所有帖子了，会员只要点击子论坛名称，就会进入子论坛中，找到自己感兴趣的帖子点击进入就可以浏览帖子的所有内容和回复了。进入帖子，只要点击帖子名称就能进入浏览和回复状态了。

帖子浏览如下图3-6所示：

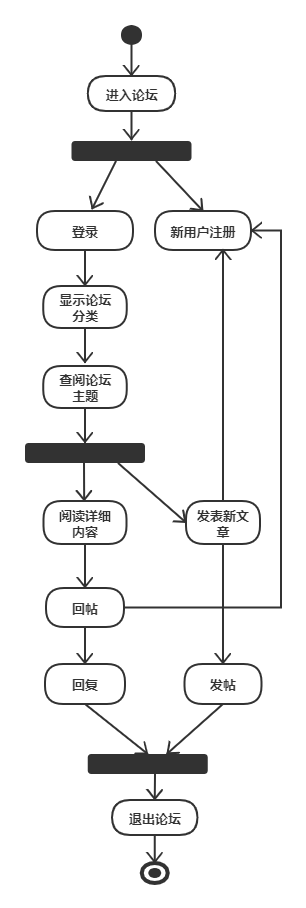


图3-6 帖子浏览图

### 3.2.4 会员帖子发表

在子论坛界面，点击发表帖子按钮，如果你是会员用户，则会进入发表帖子界面，在发表帖子界面，会员用户输入帖子主题，帖子类型和帖子内容等要发表的帖子信息。然后点击发表，帖子就会上传到子论坛上去显示。如果只是游客身份，则不会进入发表帖子界面，直接显示需要注册才能发表帖子。

帖子发表流程图如下图3-7所示：

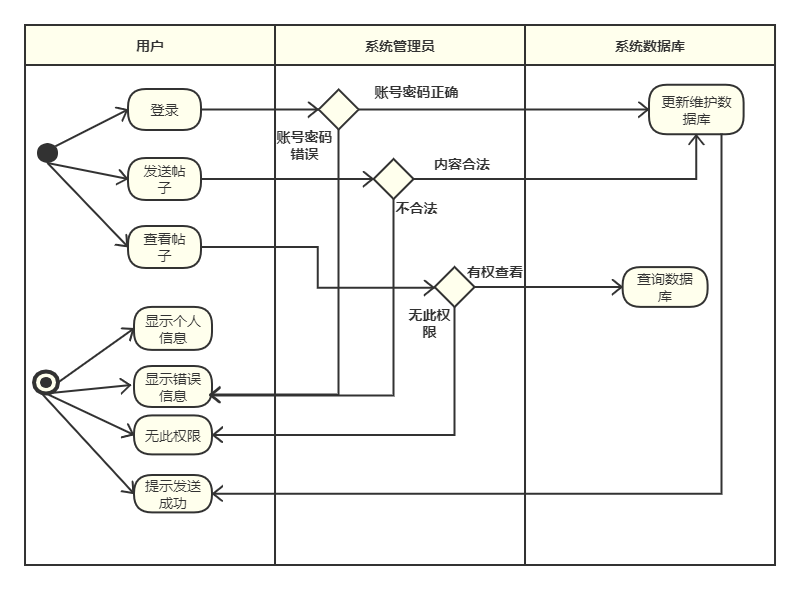


图3-7帖子发表流程图

### 3.2.5 管理员模块管理

首先要进入管理员系统，则必须要管理要登录，如果登录的不是管理员，则如法进入管理员系统，如果是管理员，就能在管理页面对整个系统的模块进行管理，对模块的管理包括增加模块，删除模块和编辑模块等操作。

模块管理如下图3-8所示：

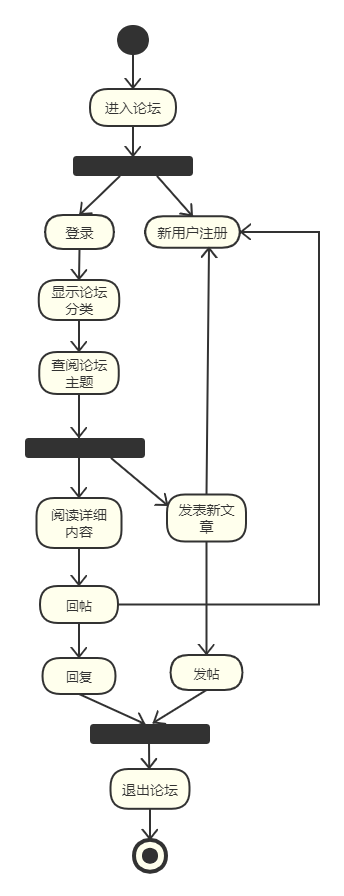


图3-8模块管理流程图

### 3.2.6 管理员帖子管理

首先要进入管理员系统，则必须要管理员登录，如果登录的不是管理员，则无法进入管理员系统，如果是管理员，就能在管理页面对整个论坛的帖子管理。管理员可以对整个论坛的帖子进行删除，置顶等操作。

帖子管理流程图如下图3-9所示：

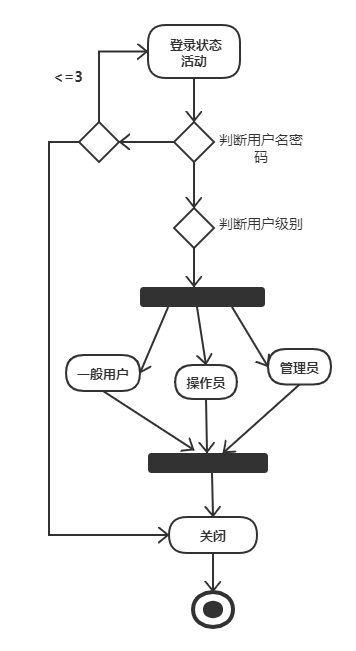


图3-9 帖子管理流程图

# 四．数据库设计

## **4.1 数据库分析**

数据库的设计，在程序开发中起者至关重要的作用，它往往决定了在后面开发进行怎么样的程序编码，一个合理，有限的数据库设计可降低程序的负责性，使得程序开发过程更为容易。

本系统是一个小型论坛系统，考虑到用户量，开发经济性，决定用MySQL作为数据库系统。

MySQL数据库系统虽然说规模小，功能有限，但是丝毫也没有减少它受欢迎的程度，对于一般的个人使用者和中小企业来说，MySQL提供的功能已经绰绰有余了，又因为MySQL是开发源码软件，因此大大降低了总体成本，它可以处理千万条记录的大型数据库，支持多种存储引擎。

1. R模型是对现实世界的一种抽象，它主要成分是实体，联系和属性。使用这三种成分，我们可以建立许多应用环境E-R模型。
2. 系统E-R图如下图4-1所示：

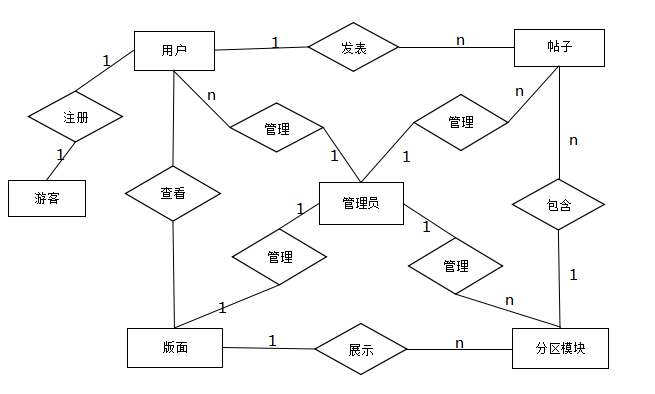


图4-1 系统E-R

1. 用户属性图

用户实体有ID,用户名，昵称，性别，密码，邮箱和QQ等属性。下图4-2为用户实体与属性图：

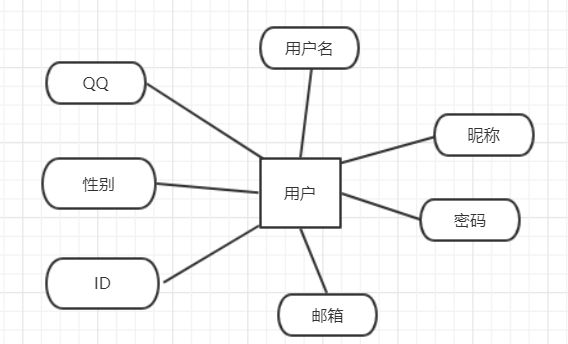


图4-2 用户实体图

帖子实体有帖子主题，发表人，发表时间，帖子ID，帖子内容，所属模块，帖子类型等属性。下图4-3为帖子实体图：

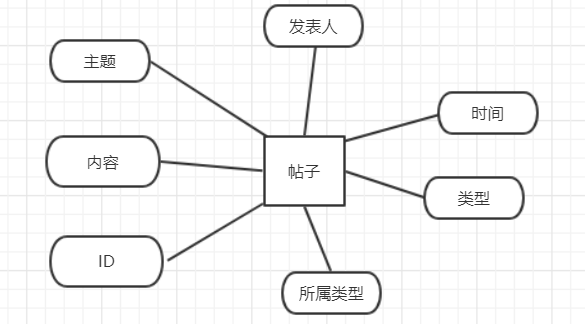


图4-3 帖子实体图

1. 模块属性图

模块实体图包括模块ID,模块名称，下图4-4为模块的实体属性图：

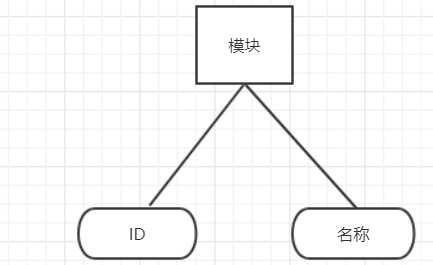


图4-4 模块实体图

## **4.2 数据库表设计**

本系统的数据库主要有四个数据库表，这四张数据库表分别是：t\_user用户信息表，t\_forums论坛类别介绍表，t\_topic版面信息表，t\_threads贴子信息表，这主要四张表的内容如下所示：

（1）user用户表主要字段信息表如下表4-1所示：

表4-1 t\_user（用户信息表）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度 |
| 1 | id | 编号 | int | 11 |
| 2 | userName | 用户名 | varchar | 50 |
| 3 | userPW | 用户密码 | varchar | 50 |
| 4 | realName | 真实姓名 | varchar | 50 |
| 5 | email | 邮箱 | varchar | 100 |
| 6 | userType | 用户权限 | int | 11 |
| 7 | isDel | 删除标记 | int | 11 |

（2）t\_forums（论坛类别介绍表），如下表4-2所示。

表4-2 t\_forums表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度 |
| 1 | fid | 模块编号 | int | 11 |
| 2 | type | 类型 | varchar | 10 |
| 3 | name | 名字 | varchar | 100 |
| 4 | fup | 上级编号 | int | 11 |
| 5 | status | 状态 | smallint | 6 |

1. t\_topic（版面信息表）如下表4-2-3所示。

表4-2-3 t\_topic表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度 |
| 1 | pid | 新帖编号 | int | 11 |
| 2 | fid | 模块编号 | int | 11 |
| 3 | author | 作者 | varchar | 100 |
| 4 | authorid | 作者id | int | 11 |
| 5 | subject | 标题 | varchar | 100 |
| 6 | view | 观看 | int | 11 |
| 7 | replies | 回复 | int | 11 |
| 8 | addtime | 发表时间 | datetime | 0 |

1. t\_threads（贴子信息表），如下表4-2-4所示。

表4-2-4 t\_threads表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 名称 | 别名 | 类型 | 长度 |
| 1 | tid | 帖子编号 | int | 11 |
| 2 | pid | 新帖编号 | int | 11 |
| 3 | fid | 题目编号 | int | 11 |
| 4 | ftype | 类型 | int | 11 |
| 5 | author | 作者 | varchar | 50 |
| 6 | authorid | 作者id | int | 11 |
| 7 | subject | 标题 | varchar | 100 |
| 8 | Content | 内容 | varchar | 10000 |
| 9 | addtime | 发表时间 | datetime | 0 |
| 10 | tnum | 帖子数量 | int | 11 |

# 程序结构

程序结构如下图5-1所示：

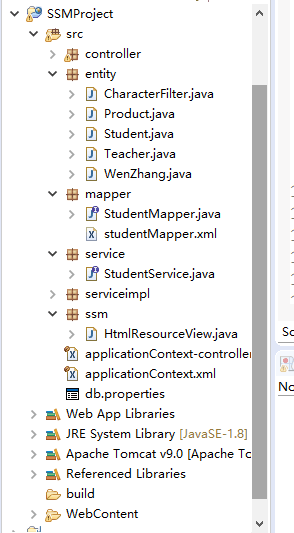


图5-1 程序结构图

# 核心配置

1.SpringMVC核心配置代码如下：

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:c="http://www.springframework.org/schema/c"

xmlns:jdbc="http://www.springframework.org/schema/jdbc"

xmlns:jee="http://www.springframework.org/schema/jee"

xmlns:lang="http://www.springframework.org/schema/lang"

xmlns:task="http://www.springframework.org/schema/task"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:util="http://www.springframework.org/schema/util"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd

http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-4.3.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-4.3.xsd

http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd

http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang-4.3.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.3.xsd">

<!-- 扫描器 扫描有注解的类 -->

<context:component-scan base-package="controller"></context:component-scan>

<!-- 配置视图解析器 -->

<!-- 配置视图解析器 InternalResourceViewResolver -->

<bean

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="prefix" value="/context/"></property>

<property name="suffix" value=".jsp"></property>

</bean>

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<property name="defaultEncoding" value="UTF-8"></property><!-- 文件编码 -->

<property name="maxUploadSize" value="409600"></property> <!-- 单个文件上传最大值单位byte 如果是-1，表示无限制-->

</bean>

<mvc:annotation-driven></mvc:annotation-driven>

<mvc:default-servlet-handler />

<mvc:resources mapping="/context/\*\*" location="context/" />

<mvc:resources mapping="/picture/\*\*" location="picture/" />

</beans>

<!-- jsp文件和HTML文件共存 -->

<!-- <bean id="htmlviewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass" value="ssm.HtmlResourceView"></property> <property

name="order" value="0"></property> <property name="prefix" value="/context/"></property>

<property name="suffix" value=".html"></property> </bean> <bean id="jspviewResolver"

class="org.springframework.web.servlet.view.InternalResourceViewResolver">

<property name="viewClass" value="org.springframework.web.servlet.view.JstView"></property>

<property name="order" value="1"></property> <property name="prefix" value="/context/"></property>

<property name="suffix" value=".jsp"></property> </bean> -->

<!-- 配置（CommonsMultipartResolver） 将其加载到SpringIOC容器中 -->

<!-- <bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<property name="defaultEncoding" value="UTF-8"></property>文件编码 <property

name="maxUploadSize" value="102400"></property> 单个文件上传最大值单位byte 如果是-1，表示无限制

</bean> -->

<!-- SpringMvc基础配置 标配 -->

<!-- 此配置是springmvc的基础配置 多功能需要此注解来调解 -->

1. Spring核心配置如下：

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:c="http://www.springframework.org/schema/c"

xmlns:jdbc="http://www.springframework.org/schema/jdbc"

xmlns:jee="http://www.springframework.org/schema/jee"

xmlns:lang="http://www.springframework.org/schema/lang"

xmlns:mvc="http://www.springframework.org/schema/mvc"

xmlns:task="http://www.springframework.org/schema/task"

xmlns:tx="http://www.springframework.org/schema/tx"

xmlns:util="http://www.springframework.org/schema/util"

xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd

http://www.springframework.org/schema/jee http://www.springframework.org/schema/jee/spring-jee-4.3.xsd

http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-4.3.xsd

http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd

http://www.springframework.org/schema/lang http://www.springframework.org/schema/lang/spring-lang-4.3.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd

http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-4.3.xsd">

<!-- 依赖注入 给service 注入dao -->

<bean id="studentService" class="serviceimpl.StudentServiceimpl">

<property name="studentmapper" ref="studentMapper"></property>

</bean>

<!-- controller控制器依赖service

<bean id="studentController" class="controller.StudentController">

<property name="studentservice" ref="studentService"></property>

</bean>

-->

<!-- 配置数据源和mapper文件 -->

<!-- 加载db.properties文件 -->

<bean id="congig" class="org.springframework.beans.factory.config.PreferencesPlaceholderConfigurer">

<property name="locations">

<value>classpath:db.properties</value>

</property>

</bean>

<!-- 数据源 -->

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">

<property name="driverClassName" value="${driver}" />

<property name="url" value="${url}" />

<property name="username" value="${username}" />

<property name="password" value="${password}" />

</bean>

<!-- mapper.xml -->

<!-- SqlSessionFactory 然后交给spring -->

<!-- 配置mybatias 需要的核心类 SqlSessionFactory -->

<bean id="SqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="dataSource" ref="dataSource"></property>

<!-- 加载映射文件路径 -->

<property name="mapperLocations" value="classpath:mapper/\*.xml"></property>

</bean>

<!-- 扫描器 Spring 整合 mybatias 将mybatias的 SqlSessionFactory 交给 Spring-->

<!-- 将SqlSessionFactory交给spring -->

<bean id="mappers" class="org.mybatis.spring.mapper.MapperScannerConfigurer">

<property name="sqlSessionFactoryBeanName" value="SqlSessionFactory"></property>

<!-- 批量产生那个包的 -->

<property name="basePackage" value="mapper"></property>

</bean>

<!-- 上面basePackage所在的property的作用：

将 mapper包 所有接口与之产生动态代理对象 （对象名就是首字母小写的接口名 ） 比如 studentMapper.queryStudentByStuno();

-->

</beans>

1. MyBatis的核心代码如下：

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<!-- 接口的全类名 -->

<mapper namespace="mapper.StudentMapper">

<!-- 单个查询 -->

<select id="queryStudentByStuno" parameterType="int"

resultType="entity.Student">

select \* from student where stuno = #{stuNo}

</select>

<insert id="addstudent" parameterType="entity.Student">

insert into student(stuno,stuname,password,stusex,stuage) values

(#{stuNo},#{stuName},#{password},#{stuSex},#{stuAge})

</insert>

<select id="selectstudent" parameterType="entity.Student"

resultType="entity.Student">

select \* from student where stuno = #{stuNo}

</select>

<insert id="insertwen" parameterType="entity.WenZhang">

insert into wen(bname,author,authors,press,wen) values

(#{bname},#{author},#{authors},#{press},#{wen})

</insert>

<select id="selectwen" resultType="entity.WenZhang">

select \* from wen

</select>

<insert id="Addtu" parameterType="entity.Product">

insert into product values (#{bname},#{author},#{authors},#{press},#{wen})

</insert>

<select id="selectpic" resultType="entity.Product">

select \* from product

</select>

<!-- 老师 -->

<insert id="addteacher" parameterType="entity.Teacher">

insert into teacher(tno,tname,password,tsex,tage) values

(#{tNo},#{tName},#{password},#{tSex},#{tAge})

</insert>

<select id="selecteacher" parameterType="entity.Teacher"

resultType="entity.Teacher">

select \* from teacher where tno = #{tNo}

</select>

<insert id="insertwen2" parameterType="entity.WenZhang">

insert into wen2(bname,author,authors,press,wen) values

(#{bname},#{author},#{authors},#{press},#{wen})

</insert>

<select id="selectwen2" resultType="entity.WenZhang">

select \* from wen2

</select>

<insert id="Addtu1" parameterType="entity.Product">

insert into product1 values (#{bname},#{author},#{authors},#{press},#{wen})

</insert>

<select id="selectstudent1" resultType="entity.Student">

select \* from student

</select>

</mapper>

1. web.xml的核心配置代码如下：

<!-- web项目中引入spring -->

<!-- needed for ContextLoaderListener -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:applicationContext.xml</param-value>

</context-param>

<!-- Bootstraps the root web application context before servlet initialization -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- 配置springmvc -->

<!-- The front controller of this Spring Web application, responsible for handling all application requests -->

<servlet>

<servlet-name>springDispatcherServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:applicationContext-controller.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<!-- Map all requests to the DispatcherServlet for handling -->

<servlet-mapping>

<servlet-name>springDispatcherServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

<filter>

<filter-name>characterEncoding</filter-name>

<filter-class>entity.CharacterFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>utf-8</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>characterEncoding</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

</web-app>

# 七．系统实现与测试

## **7.1 前台页面设计**

### 7.1.1 用户注册

用户通过注册成为论坛会员，就可以浏览帖子，发表帖子以及回复帖子等功能。用户进入login.jsp页面后，如果是没有注册的用户点击注册按钮，页面就跳转到register.jsp页面，通过serson的判断，如果同意条约，就真正进入注册页面，在注册页面，用户填写重要的注册信息，填写完成后用户点击注册，用户信息通过UserSerlet.java提交给数据库，就完成了注册功能，完成注册后，就可以通过登录论坛进入论坛首页。论坛注册界面如图5-1所示：



图7-1坛注册界面

用户注册核心代码如下：

if ("add".equals(path)) {// 用户注册

user .setUserName (request .getParameter ("userName"));

user .setUserEmail (request.getParameter ("userEmail"));

user.setArticle(0);//已发表的文章数

user. setuserPassword (request . getParameter ("psw"));

user.setsign (request .getParameter("sign"));//个性签名

user.setsex (request . getParameter("sex"));//性别

user . setHomepage ("www. fly . com") ;

user . setAddDate (cutils.getTime () ) ;

user. setLogins(0) ;//登录次数

user . setFace (request. getParameter ("face")) ;//头像

user. setHeight (32) ;

user. setwidth (32) ;

user . setQq (request. getParameter ("qq")) ;

user. setBbstype(1);//普通会员身份

user. setLockuser (0) ;

user . setUserWealth (config. getWealthReg() ) ;

user . setUserEP (config. getCpReg()) ; //用户经验

user . setUserCP (config.getEpReg()) ;//用户魅力

userbiz.add (user) ;

tourl = "index.jsp";

### 7.1.2 用户登录

已经注册的会员用户，通过登录，进入论坛首页，用户如果没有登录就不能浏览帖子，回复帖子。用户登录是在login.jsp这个页面下登录的。用户输入了用户名和密码后点击登录，然后系统根据用户名和密码在后台数据库找到用户名和密码是否匹配，如果匹配成功，则进入论坛首页，如果匹配不成功，将进入失败页面。用户名和密码的匹配过程是通过UserServlet.java来匹配的，这里可以根据用户输入的用户名和密码来进行数据库连接。从而达到前台和后台数据库的连接。用户登录界面时如图7-2所示：



图7-2 用户登录界面

用户登录代码如下：

if ("login" .equals (path)) {// 登录论坛

Message message = new Message () ;

user . setUserName ( request. getParameter ( "userName") ) ;

user . setUserPassword (request. getParameter ("userPassword") ) ;

List list = userbiz. searchById(2，user) ;

MessageBiz messagebiz = new MessageBiz () ;

message. setIncept (user . getUserName () ) ;

List listMess = messagebiz. searchById(2，message) ;

if (list.size () > 0) {

user .

(User) list.get (0) ;

session.setAttribute ("user", user);// 记录登录后的用户信息.

session. setAttribute ("userName", user . getUserName ()) ; //

记录用户名

// session.setAttribute ("news"， user .getLockuser () ) ;

session. setAttribute ("news", listMess.size() ) ;

session. setAttribute ("bbstype", user .getBbstype() ) ;

Online onl ine = new Online () ;

online = (Online) session. getAttribute ("online") ;

online. setUsername (user . getUserName () ) ;

online. setBbstype (user .getBbstype () ) ;

String t\_ onlineid =session.getAttribute("onlineid").toString() ;

if (t\_ onlineid != nu1l) {

int onlineid = cutils.strToInt (t\_ onlineid) ;

online. setId(onlineid) ;

}

onl inebiz. update (online) ;

session. setAttribute ("onl ine"

online) ;

String type = request.getParameter ("type") ;

if (!"admin"'. equals (type))

tourl = " "index. jsp";

else

tourl ="manager/admin\_ main.jsp"; }

### 7.1.3 用户发表帖子

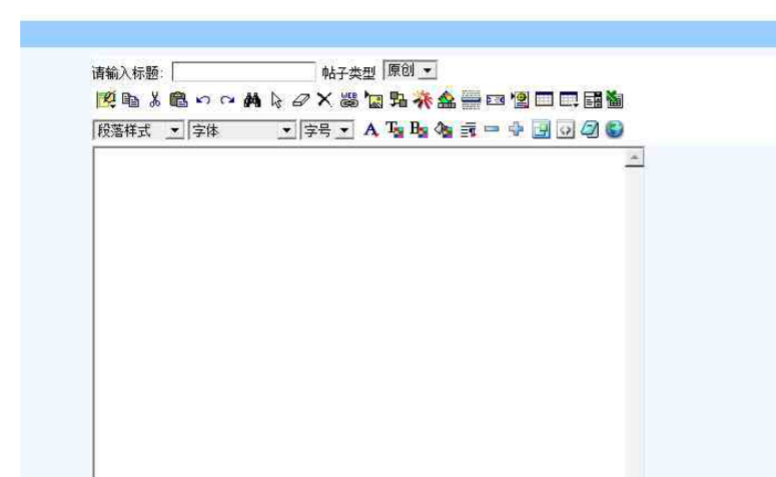
用户在进入一个论坛专区之后可以在那个论坛专区里发表帖子按钮，页面通过一个serson进入发表帖子的页面，在这个页面，用户输入要发表帖子的主题和内容，完成后点击提交，系统根据BbsServlet.java来提交用户发表的帖子进入数据库。主要还是通过get和set方法来接受和设置用具提交的信息。发表帖子界面如下图7-3所示：

图7-3发表帖子界面

## **7.2 后台页面设计**

后台页面主要是论坛管理员登录系统，对论坛进行维护，对模板的管理，用户的管理，论坛管理以及帖子的管理，在后台主要是对整个论坛的数据的操作。

### 7.2.1 管理员登录

管理员登录后台，对整个论坛进行管理，管理员登录界面如下图7-4所示：

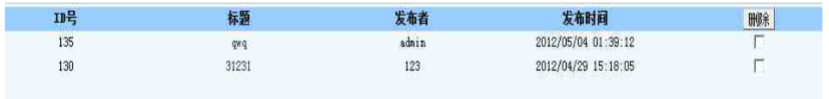


### 图7-4管理员登录界面

后台管理是进入admin\_login.jsp文件，在这个页面，管理员登录管理员用户名和密码，根据UserServlet.java匹配，如果匹配成功，则进入管理员首页。

### 7.2.2 论坛帖子管理

### 论坛帖子管理如下图7-5所示：



### 图7-5 论坛帖子管理

### 7.2.3 论坛用户管理

用户的管理主要是删除用户和查看用户信息，如果某个用户违反了论坛规定，管理员有权对该用户进行删除，进入用户管理页面，然后选择要删除的用户，在后面勾选。然后点击删除，就可以把违发规定的用户删除了。

论坛用户及权限管理如下图7-6所示：



### 图7-6 用户及权限管理

## **7.3 数据库连接**

论坛系统使用的是MySQL数据库，在整个系统中国，数据库的作用是非常重要的，论坛要实现，就必须连接数据库。

数据库连接的代码实现如下：

private ConnectionFactory () throws Exception {

driver = "com. mysql .jdbc. Driver" ;

dbURL="jdbc:mysql://localhost:3306/bbs?useUnicode=true&characterEncoding=gbk" ;

user = "root" ;

password = "123456";

}

public static Connection getConnection() {

Connection conn = nu11;

if ( factory == null) {

try {

factory = new ConnectionFactory() ;

} catch (Exception e) {

e.printStackTrace () ;

return nu1l;

}

}

try {

Class. forName ( factory.driver) ;

conn = Drive rManager . ge tConnection ( factory . dbURL，factory. user,

factory .password) ;

} catch (Exception e) {

e.printStackTrace () ;

return conn;

}

}

# 八．用户管理的详细设计与实现

## **8.1 用户管理详细设计**

登录功能序列图如图8-1所示

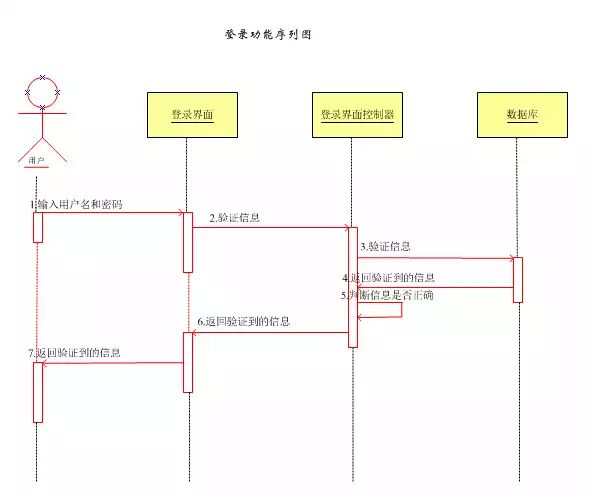


图8-1 登录功能序列图

用户输入用户名和密码之后，系统开始在数据库中检索信息是否一致，如果一致，将返回验证信息，并将验证信息反馈给用户，用户将登陆成功，进入主页面。

注册功能序列图如图8-2所示：

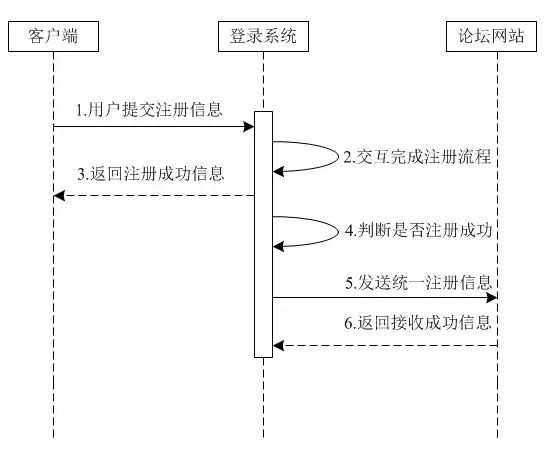


图8-2 注册功能序列图

用户提交注册信息，系统判断提交的信息是否符合设定规范，如果信息符合规范，将发送注册成功信息给用户，用户即可成功注册账号。

相册管理序列图如图8-3所示：

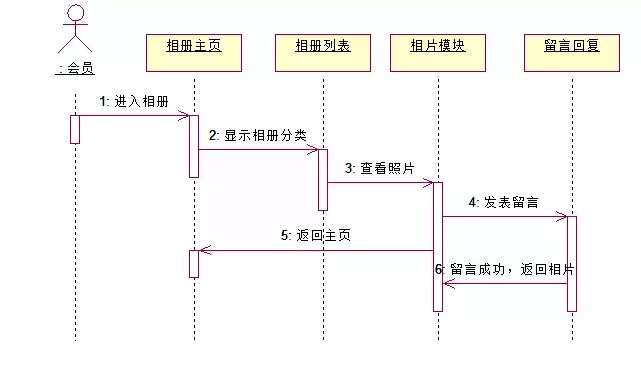


图8-3 相册管理序列图

以上为相册管理序列图，用户在论坛系统上可以管理自己的相册，并对相册模块进行操作，主要有上传相册，删除相册，查看相册以及设定可查看相册的权限，并可以对其他用户的评论进行回复等操作。

帖子管理序列图如图8-4所示

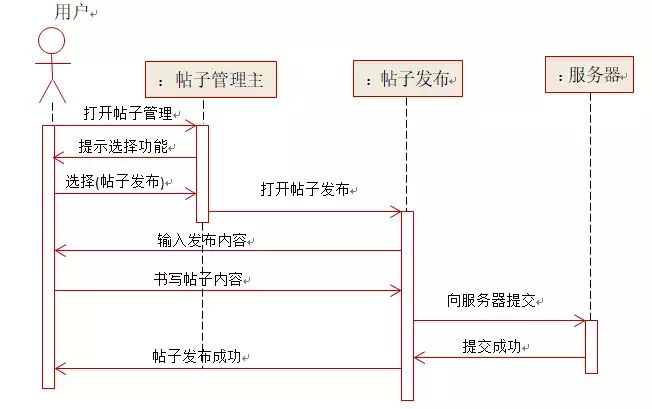


图8-4 帖子管理序列图

以上的帖子管理序列图，是指用户可以对自己的帖子进行的一系列操作，当用户想要发布帖子时，选择需要发布的模块，输入发布内容，之后点击发布，即可向服务器提交帖子内容，后台管理人员经过审核符合规定的，便可提示发布成功。

后台管理序列图如图8-5所示：

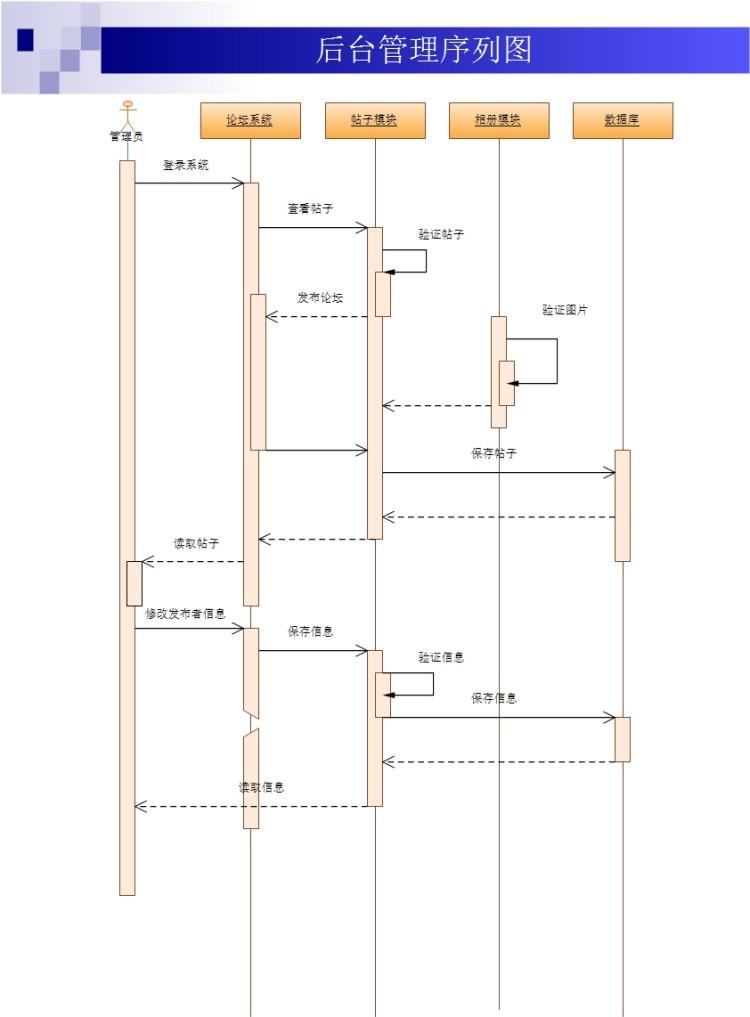


图8-5 后台管理时序图

后台管理系统主要是用于对网站前台的信息管理，如文字、图片、影音、和其他日常使用文件的发布、更新、删除等操作，同时也包括会员信息、订单信息、访客信息的统计和管理。简单来说就是对网站数据库和文件的快速操作和管理系统，以使得前台内容能够得到及时更新和调整。

# 九．系统测试

## **9.1后台子系统主要功能测试**

（1）系统管理模块

添加管理员用例测试表部分测试数据见表9-1所示。

表9-1 添加管理员功能用例测试表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能特性 | 添加管理员验证 | | | | |
| 测试目的 | 验证输入的信息是否合法 | | | | |
| 测试数据 | 管理员账号、管理员密码、管理员姓名、管理员电话、管理员邮箱、管理员  角色 | | | | |
| 测试内容 | 操作描述 | 数据 | 期望结果 | 实际结果 | 测试状态 |
| 1 | 仅输入管理员账号，点击提交 | admin | 提示“密码不能为空” | 提示“密码不能为空” | 正确 |
| 2 | 仅输入管理员密码，点击提交 | 111111 | 提示“账号不能为空” | 提示“账号不能为空” | 正确 |
| 3 | 仅输入管理员姓名，点击提交 | 张三 | 提示“账号不能为空” | 提示“账号不能为空” | 正确 |
| 4 | 输入不合法的电话号码，其它信息填写合法数据，点击提交 | 12345678 | 提示“电话号码格式不正确” | 提示“电话号码格式不正确” | 正确 |
| 5 | 输入不合法的邮箱号码，其它填写合法数据，点击提交 | zzf#.com | 提示“邮箱号码格式不正确” | 提示“邮箱号码格式不正确” | 正确 |
| 6 | 不选择管理员角色，点击提交 |  | 提示“请选择角色” | 提示“请选择角色” | 正确 |
| 7 | 在添加管理员时不输入年龄 |  | 提示“年龄不能为空” | 提示“年龄不能为空” | 正确 |

## **9.2 测试结论**

这次测试总体比较成功，表示本系统的功能基本都完成了，因为时间有限，对于论坛的其他功能不能做到很完美的实现，在后面的时间里，继续对该系统做最后的完善。

# 十．总结

在经过我们小组的不懈努力以及李老师的辛勤指导，本次设计已具备基本雏形，虽然离我们的期望还有一段距离，但由于自身技术等方面的原因，不能做到所期望的那样。所以，后期继续加强技术锻炼。

虽然系统不尽人意，但在此次设计过程中，依然学到了SSM框架的好多知识。在小组成员以及老师的相互交流探讨中，也学到了一些解决问题的方法与途径。整个设计的过程不仅是对专业知识的考验，更是处理问题的各方面能力的检验，从一开始的无从下手，资料的查询，整理，再到老师的帮助，是对我们处理问题的能力的全面提高，帮助的不仅是对本次课程设计，更是对我们以后的工作生活影响深远。

本次课程设计，让我们小组成员深切体会到，脚踏实地，认真严谨，实事求是才是处理事情的最根本的要求。