成都學院
CHENGDU UNIVERSITY

JAVAEE 项目设计

题　　目 文章发布系统

学　　院 信息科学与工程学院

专　　业 计算机科学与技术

学生姓名 陈香宇 蔡春燕

指导教师 袁飞

姓名	学号	主要任务
陈香宇	201510411404	文档撰写，后台，前端
蔡春燕	201510411401	文档撰写，数据库设计，前台

2018 年 6 月 21 日

目录

1 开发背景

随着网络的发达，越来越多的人更加的依赖网络，它已经成为我们生活中必不可少的一部分。人们获取知识的方式不仅是通过书本和通过别人的讲解，还可以通过网络来学习自己想获取的知识。这也给人们带来了极大地方便，不用因为要学习而随时随地都带着自己的书本，自己心中的疑惑不能够解决而着急。正因为网络带给人们的方便，越来越多的人愿意在网上发布电子文档、在网上进行交流讨论和分享自己在某一模块的学习心得。同时，以往人们对于知识的记录仅仅是通过手写来保存，但是这样的保存记录方式不是永久性的，不方便携带并且容易丢失。因此通过电子文档或者利用一些网站或者软件来记录保存自己的笔记和心得。这样不仅是自己的内容可以不容易丢失永久保存，还可以随时随地查看和编辑自己的文章。这样更加有利于人们相互学习、共同进步。

因此为了给此类用户提供一个学习交流的平台，同时为了让该文章发布系统更加规范合理，添加了管理员来处理文章的审核。

2 需求分析

2.1 需求概要描述

2.1.1 开发概述

本产品前台是为喜欢分享或者发表文章的用户而开发的一套文章发布管理系统，旨在向用户提供一个方便快捷的记录事物的平台，让用户更加方便快捷地记录自己的学习心得、分享自己的学习心得和学习他人的学习笔记。用户还可以设置自己的文章权限。

后端是为了管理整个文章发布系统，因为前台为用户所提供的功能和信息过于复杂，而且有些文章内容是否合理，网站维护人员难以合理地管理并及时地处理各种数据信息，所以需要后台管理界面来方便快捷地管理网站内容，并能够审核用户新发布的文章是否合理，有权删除不合理的新文章。系统开发人员所提供的平台，用于管理整个系统的内容和信息的便捷化提取。

2.1.2 功能概要需求

该文章发布管理系统涉及到了普通用户、管理员和文章。所有用户需要输入用户名和密码才能够进入相应的系统中。管理员进入系统后可以查看所有用户的文章详情、可以审核用户新发布的文章决定用户的文章是否满足审核条件，如果不满足，则管理员有权删除用户新发布的文章，同时管理员也拥有和普通用户一样的权利就是发布自己的文章；普通用户进入系统后，可以查看文章，在这一模块查看用户的文章可以分为自己的所有文章和可见的文章，也可以修改、删除和回收自己的文章以及设置自己的文章授权。一个用户和管理员可以是有多篇文章。

2.1.3 用例图

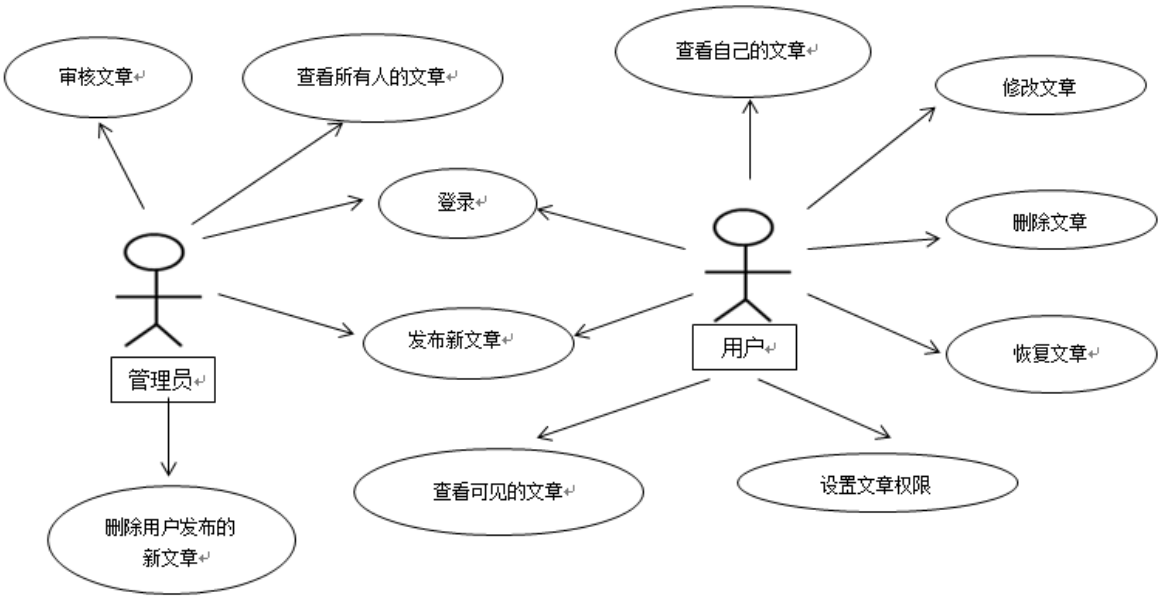

图 2-1　用例图

2.2 功能需求

2.2.1 用户模块

本模块实现了对用户的分类以及每一类用户所拥有权限的管理，下面可分为三种用户，分别为：文章作者、管理员。

文章作者：拥有此权限的用户在实现登录的基础上可以发布文章、修改文章、查看自己的文章、查看授权的文章和设置自己文章的权限以及从回收站恢复自己的文章。

管理员：此用户拥有系统最高操作权限，拥有此权限的用户也是在登录的基础上可以发布文章、查看所有作者的文章。负责规范用户管理即是用于审核作者新发布的文章是否合法，如果不合法则删除响应的内容。

2.2.2 前台页面

登录页面：提供普通用户权限及以上进行登录，需要输入用户名和密码。

主界面：本模块属于显示操作模块，文章作者和管理员登录以后可以根据界面上方的提示按钮进行相关操作。

查看文章界面：可以显示文章列表，用户可以根据提示按钮，可以对文章进行修改、删除和查看文章详情。

文章编辑页面：用于新闻编辑者进行新闻编辑，文章要通过管理员的审核，审核通过才能在新闻页面浏览。

文章授权界面：文章作者可以对自己的文章进行授权操作。

回收站界面：文章作者可以查看自己删除的文章、同时也可以恢复删除的文章。

2.2.3 后台管理

后台管理模块的前端拥有一个主页面，而其跳转则是实现模块之间的切换，在上方有文档审核、查看所有文章和发布文章三大分类。管理员可以通过一系列的操作进行浏览和操作整个文章发布系统，比如，删除作者发布的新文章，可以查看所有用户的文章的特殊权限。

2.3 开发环境

软件环境
操作系统 Windows8.1
开发平台 Intellij IDEA 2016.3
数据库 MySQL5.5、Navicat for MySQL
WEB 服务器：Tomcat 8　　JDK 1.8
硬件环境

CPU 多核、500G 内存大小

3 开发平台和技术简介

3.1 开发平台

IDEA 全称 IntelliJ IDEA，是 java 语言开发的集成环境。最突出的功能是调试(Debug)，可 IntelliJ IDEA 与其他 IDE 对比图以对 Java 代码，JavaScript，JQuery,Ajax 等技术进行调试。 IDEA 所提倡的是智能编码，减少程序员的工作。但是有着插件开发困乏、资源消耗较大的缺点。

3.2 数据库简介

MySQL 是一个小型关系型数据库管理系统，是存放数据的仓库，主要存储各种数据，包括（文章，图片，用户信息得），一个动态的网站，基本上是在页面上看到的所有的数据都是存放到数据库里的，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的 RDBMS (Relational Database Management System，关系数据库管理系统)应用软件。

MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。

3.3 Tomcat 服务器

Tomcat 是为 web 开发提供的 Servlet 容器，属于轻量级服务器，是当前较为流行的服务器，多用于中小型系统的开发，我们首相将 tomcat 安装在系统上，并配置好其工作环境，然后将 web 项目部署到 tomcat 上，而 tomcat 来帮助 web 项目响应并处理网络传输过程中的一些 http 页面请求，并返回数据。

Apache 是普通服务器，本身只支持 html 即普通网页。不过可以通过插件支持 php,还可以与 tomcat 连通。Apache 只支持静态网页，但像 php,cgi,jsp 等动态网页就需要 tomcat 来处理，tomcat 是有 Apache 软件基金会下属的 Jakarta 项目开发的一个 Servlet 容器，按照 Sun Microsystems 提供的技术规范，实现了对 Servlet 和 JavaServerPage(JSP)的支持，并提供了作为 Web 服务器的一些特有功能，如 Tomcat

管理和控制平台，安全域管理和 Tomcat 阀等，由于 Tomcat 本身也内含了一个 HTTP 服务器，它也可以被视作为一个单独的 Web 服务器。但是，不能将 Tomcat 和 Apache Web 服务器混淆，Apache Web Server 是一个用 c 语言实现的 HTTP web server;这两个 HTTP web server 不是捆绑在一起的。Apache Tomcat 包含了一个配置管理工具，也可以通过编辑 XML 格式的配置文件来进行配置。Apache,nginx,tomcat 并称为网页服务三剑客，可见其应用度之广泛。

3.4 主要技术简介

（1） AJAX

AJAX 就是 异步 JavaScript 和 XML。AJAX 是一种在无需重新加载整个网页的情况下，能够更新部分网页的技术。AJAX 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。传统的网页（不使用 AJAX）如果需要更新内容，必需重载整个网页面。

（2） Servlet

Servlet（Server Applet）是 Java Servlet 的简称，称为小服务程序或服务连接器，用 Java 编写的服务器端程序，主要功能在于交互式地浏览和修改数据，生成动态 Web 内容。

（3） JSP

Java Servlet 是 JSP 的技术基础，而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成。JSP 具备了 Java 技术的简单易用，完全的面向对象，具有平台无关性且安全可靠，主要面向因特网的所有特点。

（4） JAVABEAN

JavaBean 是一种 JAVA 语言写成的可重用组件。为写成 JavaBean，类必须是具体的和公共的，并且具有无参数的构造器。

JavaBean 是一个遵循特定写法的 Java 类，它通常具有如下特点：

- 这个 Java 类必须具有一个无参数的构造函数。
- 属性必须私有化。

- 私有化的属性必须通过 public 类型的方法暴露给其它程序，并且方法的命名也必须遵守一定的命名规范。

JavaBean 的属性可以是任意类型，并且一个 JavaBean 可以有多个属性。每个属性通常都需要具有相应的 setter、 getter 方法，setter 方法称为属性修改器，getter 方法称为属性访问器。

属性修改器必须以小写的 set 前缀开始，后跟属性名，且属性名的第一个字母要改为大写，例如，name 属性的修改器名称为 setName，password 属性的修改器名称为 setPassword。

属性访问器通常以小写的 get 前缀开始，后跟属性名，且属性名的第一个字母也要改为大写，例如，name 属性的访问器名称为 getName，password 属性的访问器名称为 getPassword。

一个 JavaBean 的某个属性也可以只有 set 方法或 get 方法，这样的属性通常也称之为只写、只读属性。

（5） CSS

层叠样式表(英文全称：Cascading Style Sheets)是一种用来表现 HTML（标准通用标记语言的一个应用）或 XML（标准通用标记语言的一个子集）等文件样式的计算机语言。CSS 不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。

（6） JavaScript

JavaScript 一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为 JavaScript 引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在 HTML（标准通用标记语言下的一个应用）网页上使用，用来给 HTML 网页增加动态功能。

（7） Html

超文本标记语言，标准通用标记语言下的一个应用。使用 HTML 来建立自己的 WEB 站点，HTML 运行在浏览器上，由浏览器来解析。

3.5 设计模式

设计模式是面向对象的程序设计人员用来解决编程问题的一种形式化表示。本系统开发采用的是 MVC 设计模式。全名是 Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特地发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。MVC 分层有助于管理复杂的应用程序，因为可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。MVC 模式同时提供了对 HTML、CSS 和 JavaScript 的完全控制。

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。

通常模型对象负责在数据库中存取数据。

View（视图）是应用程序中处理数据显示的部分。

通常视图是依据模型数据创建的。

Controller（控制器）是应用程序中处理用户交互的部分。

通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

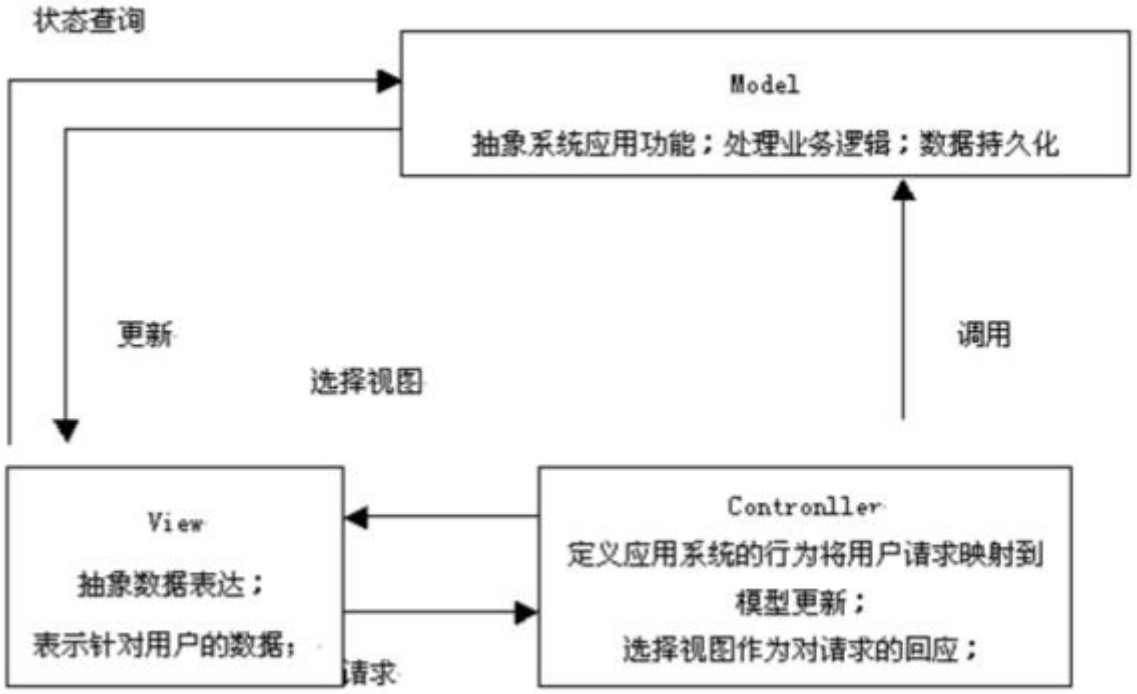
状态查询
Model
抽象系统应用功能；处理业务逻辑；数据持久化
更新
选择视图
调用
View
抽象数据表达；
表示针对用户的数据；
Contronller
定义应用系统的行为将用户请求映射到
模型更新；
选择视图作为对请求的回应；

图 3-1　MVC 模式

4 概要设计

4.1 业务流程

该文章发布管理系统要实现普通用户和管理进行相关的操作都是需要进入登录界面输入相应的用户名和密码。整体的可以分为普通用户模块和管理员模块。具体的业务流程如下图。

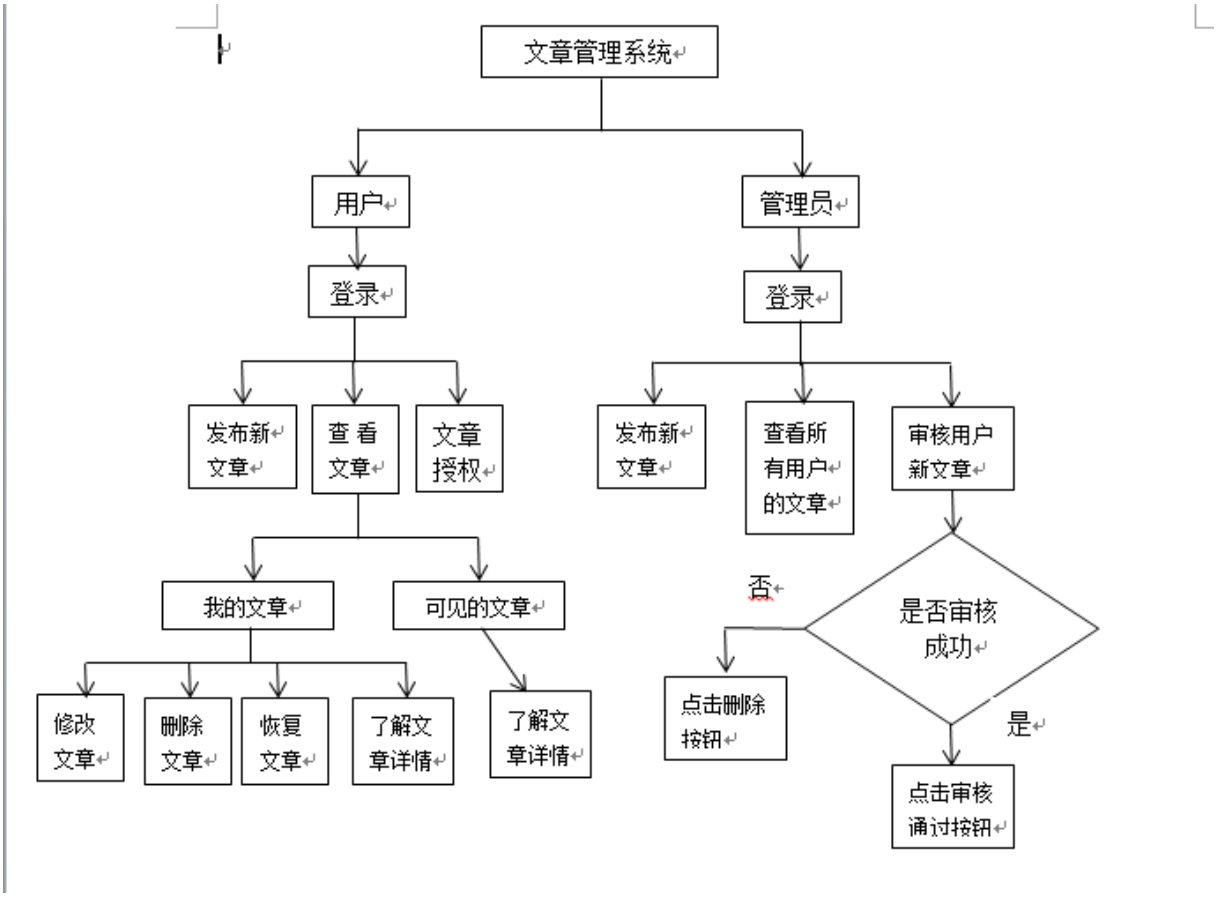

图 4-1 业务流程图

在普通用户模块，用户需要输入正确的用户名和密码才可以进入用户界面，实现自己的相关操作。用户可以根据界面的提示进行操作。点击查看文章按钮，用户便可以看到自己的文章列表和自己可见的文章列表，同时用户也可以 对自己的文章进行修改、删除和查看详情的操作，对于可见的文章只能够查看他人的文章详情。也可以在该界面实现添加文章、批量删除文章和查看回收站；点击文章授权按钮，用户可以将自己的文章设置权限，能够只让部分人可以看；点击发布文章，用户便可以编辑自己的文章，写

下自己的心得，根据界面文字提示在相应模块写入标题、内容和类别等信息；点击回收站，用户可以看到自己删除的文章列表，同时可以选择恢复自己删除的文章。

在管理员模块。管理员与用户一样需要输入正确的用户名和密码才可以进入管理员界面，实现管理员的操作。管理员可以根据界面提示进行相关处理。点击文档审核，管理员可以查看审核文章，可以选择是否通过此篇文章，如果不通过，管理员可以选择删除；点击查看所有文章，管理员可以看到所有的文章列表，选择要查看的文章的详细情况；点击发布文章，管理员和普通用户一样可以发布文章，根据文字提示输入标题、内容和类别等信息。

4.2 数据库设计

4.2.1 E-R 图

该文章发布系统有三个实体分别是用户、文章和管理员，文章发布系统的 E-R 图如下：

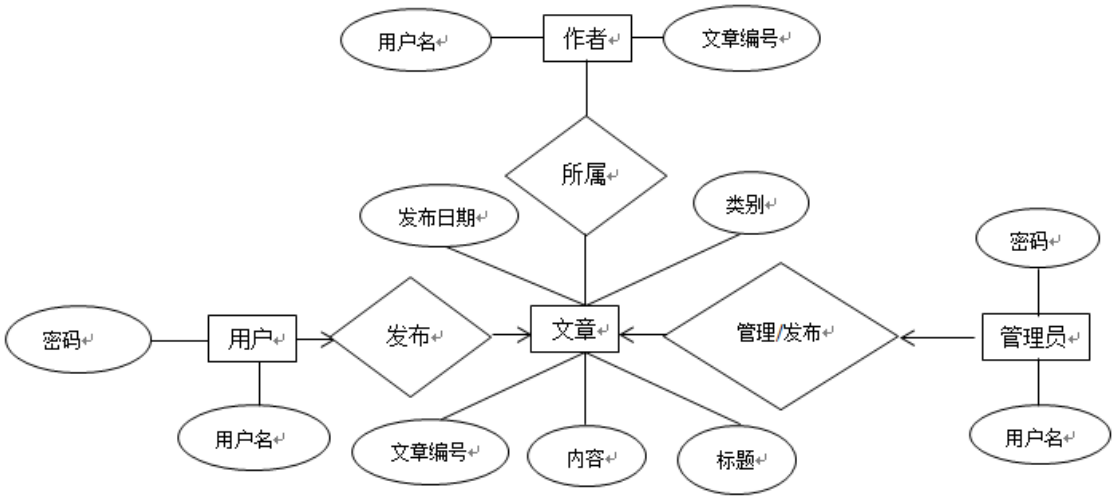

图 4-2　E-R 图

4.2.2 数据结构

（1）　admin 表，该表是管理员的信息，从上到下依次是管理员的用户名、密码。

属性名称	数据类型	
Username	Varchar（20）	主键
Password	Varchar(30)	

图 4-1　admin 表

（2） authority 表，该表是设置文章的权限，从上到下依次是文章 ID、用户名。

属性名称	数据类型	
Newid	Varchar(30)	主键
username	Varchar(30)	主键

图 4-2 authority 表

（3） check_pending 表，该表是用于管理员审核用户新发布的文章，从上到下依次是文章 ID、标题、内容、发布时间、作者、类型。

属性名称	数据类型	
Newsid	Varchar(30)	主键
Head	Varchar(100)	
content	Varchar(8000)	
Publish_time	Varchar(30)	
issueuser	Varchar(30)	
newstype	Varchar(30)	

图 4-3 check_pending 表

（4） Dutbin 表，该表是用于用户查看或回收回收站文章，从上到下依次是文章 ID、标题、内容、发布时间、作者、类型。

属性名称	数据类型	
Newsid	Varchar(30)	主键
Head	Varchar(100)	
content	Varchar(8000)	
Publish_time	Varchar(30)	
issueuser	Varchar(30)	
newstype	Varchar(30)	

图 4-4 dutbin 表

（5） News 表，该表是用于发布或者编辑文章，从上到下依次是文章 ID、标题、内容、发布时间、作者、类型。

属性名称	数据类型	
Newsid	Varchar(30)	主键
Head	Varchar(100)	
content	Varchar(8000)	
Publish_time	Varchar(30)	
issueuser	Varchar(30)	
newstype	Varchar(30)	

图 4-5 news 表

（6） user 表，该表是用于普通用户实现登录，从上到下依次是用户名、密码。

属性名称	数据类型	
Username	Varchar(32)	主键
Password	Varchar(32)	

图 4-6 user 表

（7） userAuthor 表，该表是关于作者的属性，从上到下依次是用户名、文章 ID。

属性名称	数据类型	
username	Varchar(32)	主键
newsId	Varchar(30)	

图 4-2 userAuthor 表

5 功能模块详细设计及技术处理

5.1 用户功能模块

（1） 登录界面

图 5-1 登录界面

（2） 发布文章界面

图 5-2 发布文章界面

（3） 用户界面

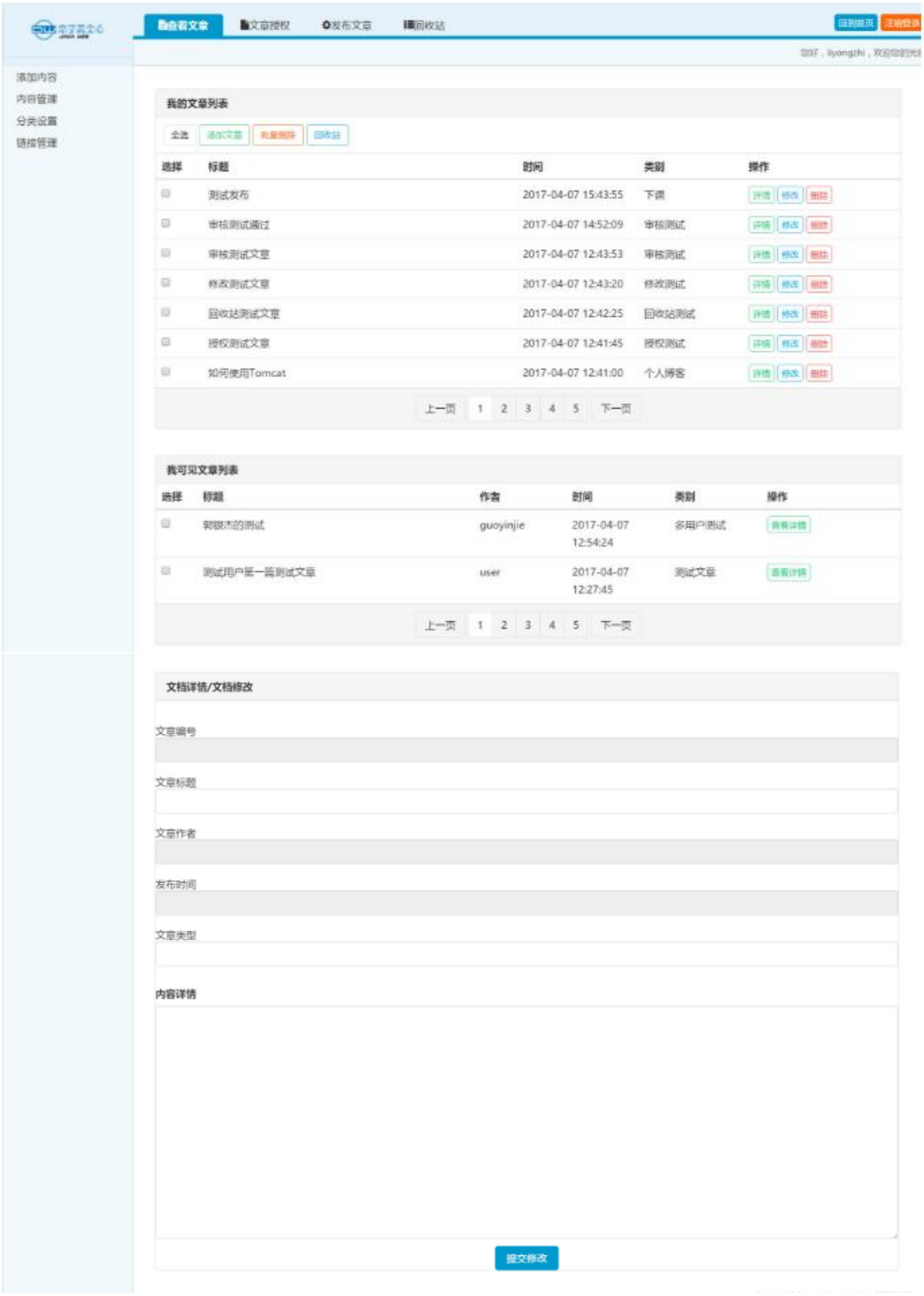

图 5-3 用户界面

（4） 用户登录功能

主要的就是 form 标签里的 action 属性，表示将表单里的内容提交给后台的 checkLogin_user 这个 Servlet 进行处理，其中 input 标签里的 name 属性标记其中的值，可以在 Servlet 中使用 request.getparameter（）方法得到标签中填入的值，如下图所示：

```
<form method="post" action="checkLogin_user">
    <div class="panel">
        <div class="panel-head"><strong>用户登录</strong></div>
        <div class="panel-body" style="padding:30px;">
            <div class="form-group">
                <div class="field field-icon-right">
                    <input type="text" class="input" name="user" placeholder="Username"/>
                    <span class="icon icon-user"></span>
                </div>
            </div>
```

图 5-4 用户登录

处理登录的 Servlet：处理 JSP 发送过来的数据，调用后台程序进行处理，并返回结果。如果用户输入的信息匹配则登录到用户界面，失败则返回到登录界面重新进行登录验证。

```
String user=request.getParameter("user");
String pass=request.getParameter("pass");

NewsRealeseDao newsRealeseDao=new NewsRealeseDao();
try {
    boolean checked=newsRealeseDao.ischecked(user,pass,"user");//调用后台的Dao方法利用user表进行身份验证
    if(checked)
    {
        HttpSession session=request.getSession();
        session.setAttribute("username",user);//设置用户的姓名
        //  response.sendRedirect("adminManager.jsp");
        response.sendRedirect("content_user.jsp");
    }
    else
    {
        response.sendRedirect("login_user.jsp");

    }
}
```

图 5-5 处理登录

（5） 用户发布文章功能

这部分功能主要是通过把用户输入的文章信息按照不同的属性例如文章名、标题、文章编号、发布时间、作者、文章分类用 put 的方法加入到 HashMap 的类里的 addnews_list 中，然后同步更新数据库，如果添加成功则提示成功否则提示失败，如下图所示：

```
System.out.println(table+"==================================================");
NewsRealeseDao newsRealeseDao = new NewsRealeseDao();
HashMap<String,String> addnews_list=new HashMap<>();
addnews_list.put("newsid",request.getParameter("newsid"));
addnews_list.put("head",request.getParameter("head"));
addnews_list.put("content",request.getParameter("content"));
addnews_list.put("publish_time",request.getParameter("publish_time"));
addnews_list.put("issueuser",request.getParameter("author"));
addnews_list.put("newstype",request.getParameter("newstype"));
```

图 5-6 用户发布文章

（6） 用户查看文章详情功能

实际上就是通过 JDBC 连接数据库然后直接调用 NewsRealeseDao.java 中的方法来直接查询数据库中的记录，然后将结果放到一个 ArrayList 中直接返回，然后在 JSP 页面上显示出来，如下图所示：

```
ArrayList newsRealese = new ArrayList();
try {
    //
    Class.forName(driver);

    conn = DriverManager.getConnection(url, databseUser, password);
    //
    String sql = "select * from " + table + " where issueuser='" + username + "' order by publish_time desc";
    Statement stat = conn.createStatement();
    newsRealese = getNews(stat, sql);
    if (newsRealese.size() == 0) {
        System.out.println("没有查询到信息==========");
        return null;
    }
```

图 5-7 用户查看文章

（7） 用户删除文章信息

这部分功能主要也是通过获取用户传入的想要删除的文章 id 号，然后通过在数据库中搜索到这个 id 然后删除掉这个文章，如下图所示：

```
String sql_move = "insert into dustbin select * from news where newsId='" + newsid + "'";

String sql_delete = "DELETE FROM " + table + " WHERE newsId='" + newsid + "'";//要传递newsid
Statement ps = conn.createStatement();
```

图 5-8 用户删除文章

（8） 用户修改文章信息

这部分主要是通过把用户重新修改后的文章信息重新传入到数据库中，并用

UPDATE 命令实现数据的时事更新，如下图所示：

```
Class.forName(driver);
conn = DriverManager.getConnection(url, databseUser, password);//
//newsid
String sql = "UPDATE news set head=?,content=?,publish_time=?,issueuser=?,newstype=? where newsId=?";
```

图 5-9 用户修改文章

（9） 用户恢复文章功能

这部分功能的实现主要是通过实现传入用户想要恢复文章的 id 号，然后通过 sql 语句进行查找这个 id 号，把它从 dustbin 的数据表中删除掉从而把文章从垃圾箱中恢复出来，如下图所示：

```
public boolean ResumeNews(String newsid)//
{
    String sql_insert = "insert into news select * from dustbin where newsId=?"
    String sql_delete = "delete from dustbin where newsId=?";
    Connection con = null;
```

图 5-10 用户恢复文章

（10） 用户授权自己文章功能

这部分功能主要是通过实现获取用户想要授权给的文章的 id 号，然后通过 sql 语句在数据库中把这个 id 对应的文章添加到 authority 数据表中，如下图所示：

```
for (int i = 0; i < userlist.size(); i++) {
    String sql = "insert into authority  VALUES(?,?)";
    PreparedStatement ps = conn.prepareStatement(sql);
    ps.setString(1, newsid);
    ps.setString(2, userlist.get(i));
    System.out.println(ps.toString());
    count += ps.executeUpdate();
```

图 5-11 用户授权文章

5.2 管理员功能模块

（1） 管理员界面

图 5-12 管理员界面

（2） 管理员查看所有文章详情功能

实际上就是通过 JDBC 连接数据库然后直接调用 NewsRealeseDao.java 中的方法来直接查询数据库中的记录，然后将结果放到一个 ArrayList 中直接返回，然后在 JSP 页面上显示出来，如下图所示：

```
conn = DriverManager.getConnection(url, databseUser, password);
//构造  SQL数据集
String sql = "select * from news order by publish_time desc";//查询语句
Statement stat = conn.createStatement();
newsRealese = getNews(stat, sql);
```

图 5-13 管理员查看文章

（3） 管理员发布新文章功能

这部分功能也就是通过 sql 语句 insert 命令在数据库中的表中新增加数据，如下图所示：

```
Class.forName(driver);
conn = DriverManager.getConnection(url, databseUser, password);
//构造  SQL数据集
Statement stat = conn.createStatement();
String sql = "insert into " + table + "  VALUES(?,?,?,?,?,?)";
PreparedStatement ps = conn.prepareStatement(sql);
```

图 5-14 管理员发布文章

（4） 管理员删除文章功能

这部分功能主要通过 sql 语句的 delete 命令来对数据库表中的数据进行删除，如下图所示：

```
String sql_delete = "DELETE FROM  check_pending WHERE newsId='" + newsid + "'";
Statement ps = conn.createStatement();


rs = ps.executeUpdate(sql_delete);
```

图 5-15 管理员删除文章

（5） 管理员审核文章功能

这部分功能主要是通过管理员选择审核通过按钮还是删除按钮而传入不同的操作值，如果是审核通过则通过 sql 语句中的 insert 命令来把相应的文章信息增添到审核数据表中，如果是删除则表示审核不通过，然后通过 sql 语句 delete 把相应的文章信息从审核数据表中删除掉，如下图所示：

```
String sql_insert = "insert into news select * from check_pending where newsId=?";
String sql_delete = "delete from check_pending where newsId=?";
```

图 5-16 管理员审核文章

5.3 技术应用

（1）AJAX 技术

查看所有文章列表 JSP 页面最后有几种 button，每种 button 里面有一个 onclick 方法，该方法就是使用 JavaScript 实现的方法，然后使用 Ajax 技术将数据传送给后台的 Servlet，如下图所示：

```
<td><a href="#"><img src="images/edit.png" alt="" title="" border="0"
                     onclick="queryInfo('<%=tem.getNewsId()%>','news')"/></a>
</td>

<td><a href="#"><img src="images/trash.gif" alt="" title="" border="0"
                     onclick="javascrtpt:window.location.href = 'DeleteOneNews?destination
                     =AllNews_admin.jsp&table=news&newsid=<%=tem.getNewsId()%>'"/></a>
</td>
```

图 5-17 Ajax

（2）XMLHttpRequest

所有的增删查改功能的实现都是通过 js 创建一个 XMLHttpRequest 对象，然后通过 post 向服务器发送相应的操作请求，如果返回 successful 则成功，否则失败，如下图所示：

```
xmlHttp.open("post",url,true);      //向服务器端发送请求
xmlHttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=utf8");
xmlHttp.send(null);
function callback()
{
    if(xmlHttp.readyState==4)
    {
        if(xmlHttp.status==200)
        {
            var data= xmlHttp.responseText;
            data=data.toString();
            if(data=="successful")
            {
                alert("恢复成功");
                window.location.href="content_user.jsp";
            }
            else
            {
                alert("恢复失败，请重试");
                location.reload();
```

图 5-17 XMLHttpRequest

（3）JavaBean

由于文章实体在普通用户和管理员都需要用到的内容，需要被多次调用重复使用，为了减少代码量，用 JavaBean 来建立文章的类，具体代码如下：

```
public class NewsRealese {
```

```
private String newsId;
private String head;
private String content;
private String publish_time;
private String issueuser;
private String newstype;
public String getNewsId() { return newsId; }
public void setNewsId(String newsId) { this.newsId = newsId;}
public String getHead() {    return head;}
public void setHead(String head) { this.head = head; }
public String getContent() {return content; }
public void setContent(String content) { this.content = content;}
public String getPublish_time() { return publish_time;}
public void setPublish_time(String publish_time)
{    this.publish_time = publish_time; }
public String getIssueuser() { return issueuser; }
public void setIssueuser(String issueuser)
{    this.issueuser = issueuser; }
public String getNewstype(){ return newstype;}
public void setNewstype(String newstype) {    this.newstype = newstype;    }}
```

（4）Servlet

Serlet 是用 Java 编写的服务器端程序，主要功能在于交互式地浏览和修改数据，生成动态 Web 内容。在 MVC 模式中相当于控制器接收用户的请求，委托给模型进行处理（状态改变），处理完毕后把返回的模型数据返回给视图，由视图负责展示，也就是说控制器做了个调度员的工作。查询文章的部分代码如下：

```
public class QueryOneNews extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String newsid= request.getParameter("newsid");
            String table=request.getParameter("table");
            System.out.println(newsid+table);
            NewsRealeseDao newsRealeseDao = new NewsRealeseDao();
            try {
               HashMap<String,String> result = newsRealeseDao.queryOneNews(newsid,table);
                if (result.size()==0) {
                System.out.println("查询失败 ========================QueryOneNews");
                 } else {
```

```
                String data=result.get("newsid")+"||"+result.get("head")+"||"+result.get("author")+"||"+
                        result.get("time") +"||"+result.get("newstype")+"||"+result.get("content");
                System.out.println(data);
                out.write(data.toString());
                    }
                } catch (Exception ex) {
                    Logger.getLogger(checkLogin.class.getName()).log(Level.SEVERE, null, ex);
                } } }
        @Override
        protected void doGet(HttpServletRequest request, HttpServletResponse
response)
                throws ServletException, IOException
                 {  processRequest(request, response); }
        @Override
        protected void doPost(HttpServletRequest request, HttpServletResponse
response)
                throws ServletException, IOException {
            processRequest(request, response);
        }
        @Override
        public String getServletInfo() {  return "Short description";  } }
```

6 结 论

该文章发布系统已经达到团队的预期效果，实现了基本的功能。由于其中选择的某些技术以前未曾接触过，所以开发过程中是边学边做，中间遇到了很多技术的问题和难点，力所能及的就解决，超出能力范围之外的就回退到上一个版本，在做的过程中其中不断摸索，查阅相关资料，查看一些博客，慢慢地在技术方面也有一些提升，慢慢的对整个系统开发的流程也有了新的认识。

在实际项目开发中，不能有一丝含糊，在整个流程中，首先得明确确立需求，因为需求的确定才能使得开发过程目标明确，假如需求频繁变更，则需要付出的功能修改和重构的代价也是相当的大，在设计方面尽量做到细心，严谨。在开发过程中一定要遵守最初的逻辑结构，不能为了一时方便而省略掉一些结构层次，这样会导致整个系统的结构乱掉，后续的扩展将变得杂乱 ，假如在开发过程中发现有设计方面的缺陷需要及时的回滚和修复，不然越累计到后面则需要付出的代价相当大。在开发阶段需要腾出的单元测试和自测时间必须足够，才能保证开发的稳定前进。在过程中自己也接触了很多的框架，这些框架不一定要选出高低，因为不同的框架都有各自的优劣势，根据自己的需求选择不同的框架才会有更好的辅助作用。