

《 博客系统》

系统设计与开发说明

课程名称 Java 语言与 WEB 应用程序开发

开课学期 2022 至 2023 学年 第 一 学期

年 级 2021 级 专业班级 网络工程

学 生 李依琳

学 号 222021321072051

第一章 绪论

1.1 系统的目的和意义

该系统仿照博客和微博，用于网友日常分享文字和图片。系统功能与老师所讲解的网上书店相似，实现了用户注册与登录、博客文字和图片内容的发布及收藏取消收藏、管理员在后台对博客内容的查看、删除及修改等简单功能。由于能力实在有限，本系统的功能还不太完善，但可以满足网友分享日常生活中简单的分享欲，传递幸福、快乐或者一时的难过、失意，相信通过倾诉，快乐可以加倍，痛苦可以折半。

第二章 需求分析

2.1 业务功能需求

2.1.1 用户登录与注册功能

首个页面实现了用户登录和注册的功能：已注册过账号和密码的用户可以直接登录，未注册过帐号和密码的用户可以通过注册按钮跳转到注册页面进行注册后再进行登录。同时，页面下方提供了免密登录，用户无需注册也能进入到博客页面。

2.1.2 博客页面展示功能

博客首页面展示网友所发的博客内容，进入该页面可以看到网友所发的图片和文字，同时，用户可以把收藏感兴趣的博客内容。并且，该页面实现了四个其他功能，分别是：注册会员、进入收藏、进入后台和发布博客。

2.1.3 过滤器功能

管理员登陆后执行管理员的相关功能，并且已登陆的用户可以发布博客，未登录的游客无法发布博客。

2.1.4 管理员的相关功能

管理员可以在博客首页进入后台，登陆后执行管理员的相关功能。例如新增博客，修改博客、删除博客、返回首页、退出系统。

2.1.5 收藏取消收藏功能

用户可以选择感兴趣的博客内容进行收藏和取消收藏的操作。

2.2 数据存储需求

2.2.1 博客内容的存储

已发的博客的文字内容、图片内容、是否发布等信息都存在数据库中。

2.2.2 用户信息的存储

用户的用户名、密码等都存在数据库中，并且该系统实现了对密码的加密处理。

2.2.3 收藏内容的存储

该系统对网友感兴趣的内容可进行收藏。

2.3 权限管理需求

2.3.1 游客权限

游客可以不登录就进入到博客首页，但执行发布功能时会被拦截。

2.3.2 已登录用户权限

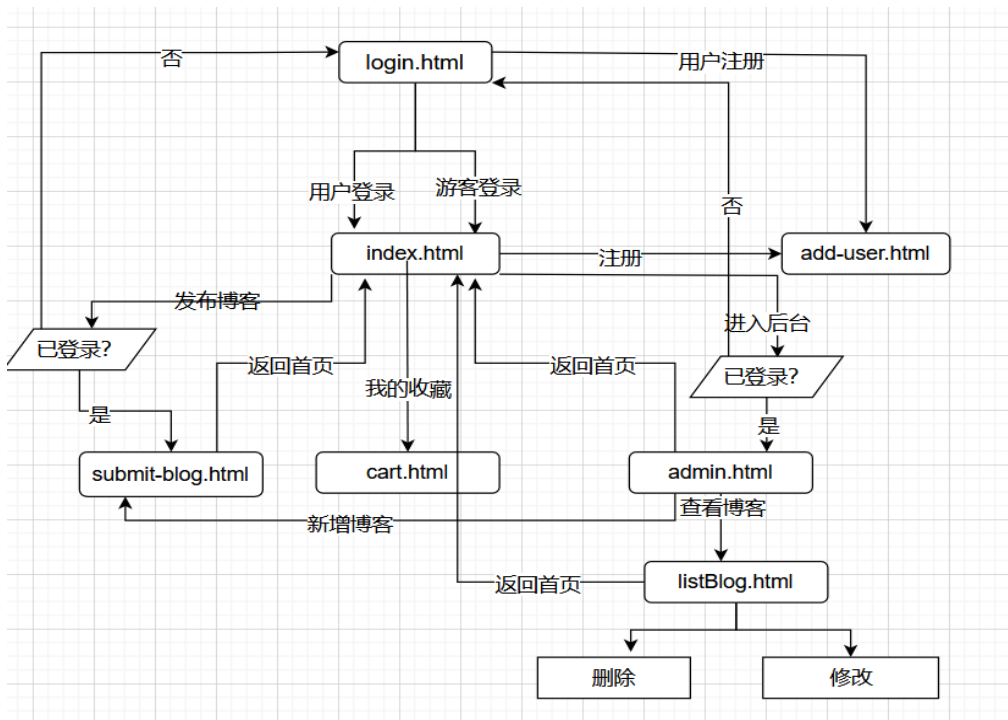
已登录的用户可以实现发布功能。

2.3.3 管理员权限

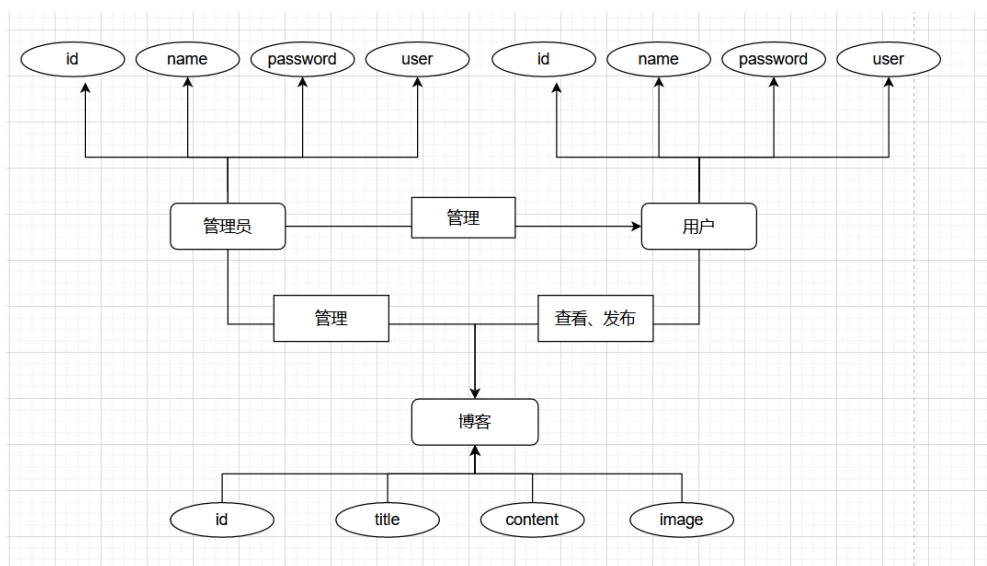
管理员可以在博客首页进入后台，登陆后可执行管理员的相关功能，例如新增博客，修改博客、删除博客、返回首页、退出系统。

第三章 系统设计与技术

3.1 系统模块结构设计



3.2 系统数据库结构设计



3.3 关键技术

在代码的编写过程中，使用 HttpServlet 传值有误的问题我一直未能得到很好的解决。由于对代码的理解还不是很深，所以我选择了跟老师所讲的网上书店系统相似的博客系统来完成此次期末设计。

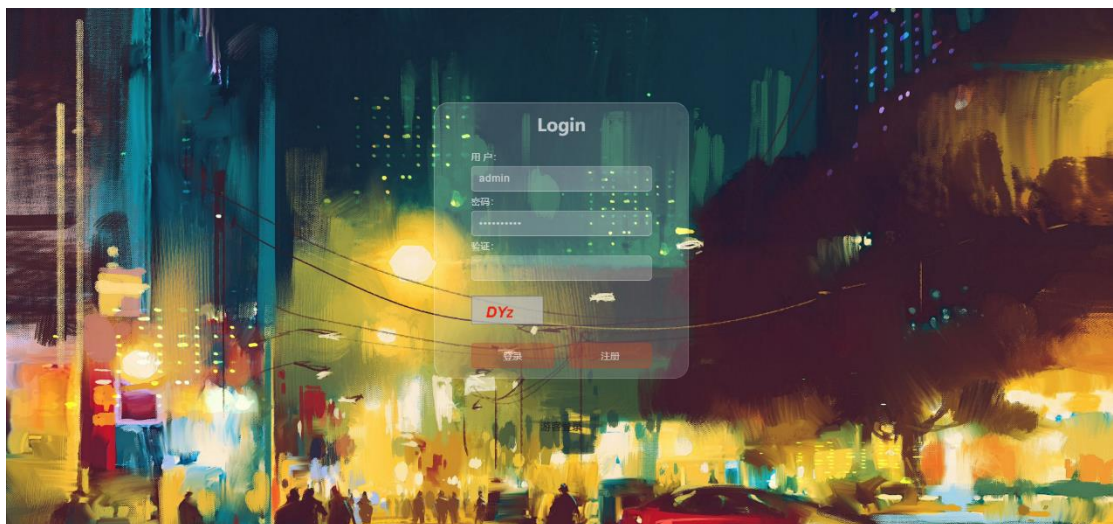
关键技术：

1. 查找获取表内所有信息时，由于获取的对象不确定，所以使用泛型。
2. 客户端通过 Ajax 的方式进行异步访问，服务器端通过 Servlet 提供功能服务的 API 接口。
3. 后端数据对象以 json 输出，在前端获取并使用。
4. 通过 HttpSession 对象实现对话管理，实现了收藏、取消收藏的功能。

第四章 系统实现

4.1 用户注册与登录功能实现

此为用户登录和游客登陆页面：



此为新用户注册页面



此为登陆页面的 html 代码

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="style.css" rel="stylesheet" type="text/css">
  <title>Login</title>
</head>
<body>
<div class="box">
  <h2>Login</h2>
  <form action="./login" method="post">
    <div class="input-box">
      <label for="">用户: </label>
      <input type="text" name="user" value="admin">
    </div>
    <div class="input-box">
      <label for="">密码: </label>
      <input type="password" name="password" value="swu.edu.cn">
    </div>
    <div class="input-box">
      <label for="">验证: </label>
      <input type="text" name="code" autocomplete="off"><br>
      
    </div>
    <div class="btn-box">
      <div>
        <button onclick="window.location.href('index.html')"> <a href="index.html">登录</a></button>
        <button onclick="window.location.href('add-user.html')"> <a href="add-user.html">注册</a></button>
      </div>
    </div>
  </form>

```

对登录页面的设计采用 html + css ,其中 css 采用外联式。（<link href=" style.css" rel="stylesheet" type=" text/css" ），外联式 css 设计可以简化 style 部分代码。

此为新增用户代码

```

@WebServlet("/api/addUser")
public class AddUserServlet extends HttpServlet {

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        String name = request.getParameter( s: "name");
        String userName = request.getParameter( s: "user");
        String password = request.getParameter( s: "password");
        User user = new User();
        user.setUser(userName);
        user.setName(name);
        user.setPassword(password);
        try {
            UserRepo.getInstance().save(user);
            System.out.println("提交信息保存成功");
        } catch (SQLException e) {
            e.printStackTrace();
            System.out.println("提交信息保存失败! ");
        }
        response.setContentType("text/html; charset=UTF-8");

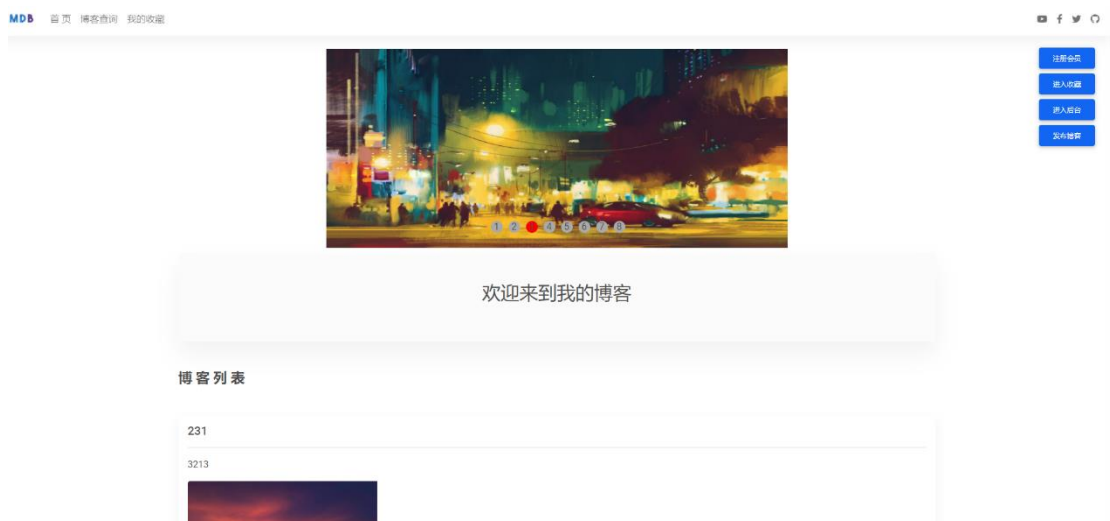
        try (Writer writer = response.getWriter()) {
            writer.write( str: "注册成功");
            writer.write( str: "<br><br>");
            writer.write( str: "<a href='"+request.getServletContext().getContextPath()+"/login.html'>返回登录页面</a>");
        }
    }

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

```

若注册成功则会响应内容“注册成功”，设置为 html 样式，在此网页我们可以跳转回登陆页面。

4.2 博客页面展示功能实现



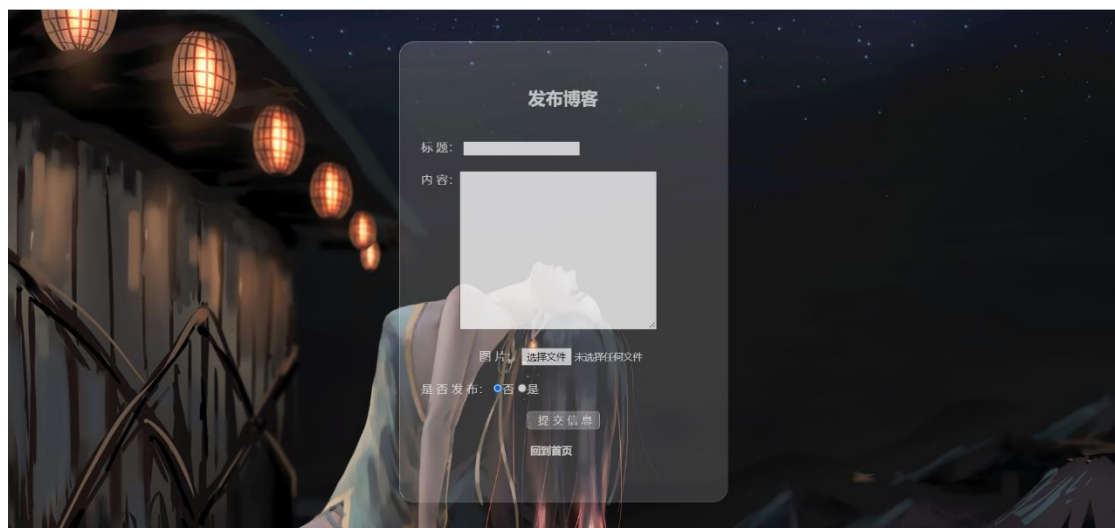
此为博客首页

代码较多，在这里不做展示。博客首页的设计采用 html + css +js，可以较好的完成一个简单但相对美观的页面。在此页面，用户可以浏览博客内容、收藏感兴趣的内容，同时右上角有四个功能，分别是：注册会员（同 4.1 新增用户）、进入收藏、进入后台（4.4 管理员相关功能）和发布博客（4.3）。

4.3 发布博客模块实现

Servlet Api 中有 Servlet 接口，当用户登录成功后将用户信息保存在 session 当中在过滤器中对 session 中的 user 信息进行判断，如果 session 不为空，则放行，否则跳回登录页面。

由于 Filter 的过滤功能，博客内容的发布功能只能由登录后的用户及管理员后台才能实现，检测到用户为游客时，系统会跳转到登录页面，游客只有登录之后才能体验发布功能。网页和代码如下：




```

String uri = request.getRequestURI();
if (uri.endsWith("login.html")
    || uri.endsWith("/api/blogs")
    || uri.endsWith("/api/addToCart")
    || uri.endsWith("/api/getBlogsInCart")
    || uri.endsWith("/api/removeFromCart")
    || uri.endsWith("cart.html")
    || uri.endsWith("login")
    || uri.endsWith("verifyCode")
    || uri.endsWith("index.html")
    || uri.endsWith("png")
    || uri.endsWith("jpg")
    || uri.endsWith("css")
    || uri.endsWith("/add-user.html")
    || uri.endsWith("/api/addUser")
    || uri.endsWith("/saveBlog")) {
    // 对于不需要进行登录认证的资源，直接放行，继续往后执行
    chain.doFilter(request, response);
    return;
}

HttpSession session = request.getSession();
if (session == null) {
    response.sendRedirect(s: "../login.html");
} else {
    Boolean token = (Boolean) session.getAttribute(LoginServlet.LOGIN_TOKEN);

    if (token != null && token.equals(Boolean.TRUE)) {
        System.out.println("已经是登录用户");
        chain.doFilter(request, response);
    } else {
        System.out.println("未获得登录标记");
        response.sendRedirect(s: "../login.html");
    }
}

```

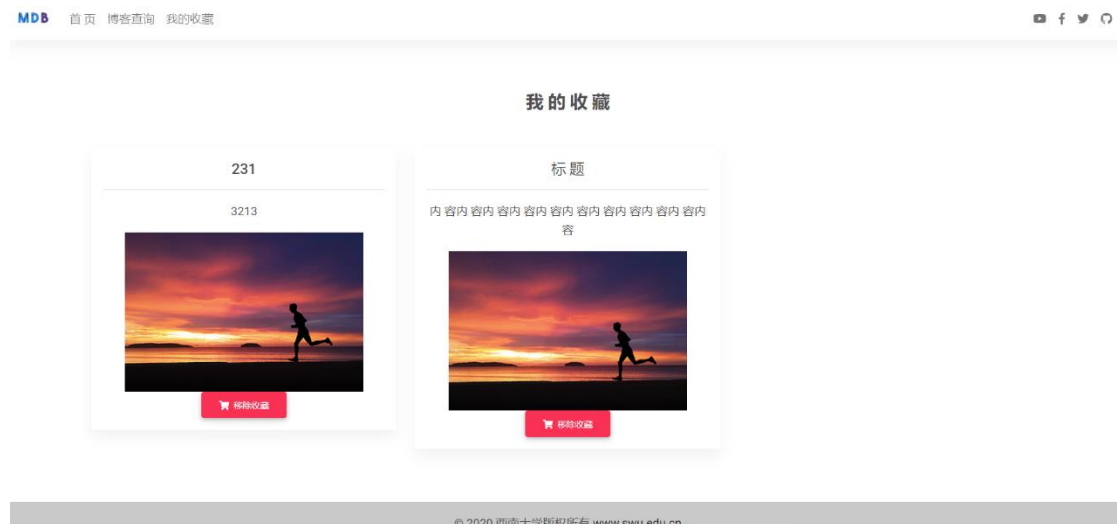
4.4 管理员功能模块实现

管理员在博客首页通过进入后台的功能进行登陆页面，管理员用户名为：admin，密码为:123456，登录成功后进入到 admin.html，在该页面可以实现新增博客、查看博客、删除博客、修改博客等功能。

下为管理员登录页面：

4.5 收藏功能模块实现

博客首页用户浏览时提供收藏功能，右上角功能可以进入我的收藏页面。



同时，在我的收藏中提供了移除收藏的功能。

```
@WebServlet("/api/addToCart")
public class AddToCartServlet extends HttpServlet {

    public final static String CART = "cart";

    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        String blogId = request.getParameter("blogId");
        List<Long> cart = this.getCart(request);
        cart.add(Long.valueOf(blogId));

        for (Long id : cart) {
            System.out.println("博客ID: " + id);
        }

        response.setContentType("application/json; charset=UTF-8");
        try (Writer writer = response.getWriter()) {
            writer.write("{\"code\": 200, \"message\": \"success\"}");
        }
    }

    private List<Long> getCart(HttpServletRequest request) {
        HttpSession session = request.getSession(true);
        List<Long> cart = (List<Long>) session.getAttribute(CART);
        if (cart == null) {
            cart = new ArrayList<Long>();
            session.setAttribute(CART, cart);
        }
        return cart;
    }
}
```

由于时间问题此处代码还不够完善，只能实现简单的收藏和取消收藏功能。

4.6 博客系统实现的不足

由于时间紧张，本系统不足之处较多。第一，网页设计比较简陋，页面不够美观；第二，本系统的权限管理做得不好，还没有实现管理员的增删改查，只在数据库中添加了一个用户为 admin、密码为 123456 的管理员，在这方面代码还有待改善；第三，交互性差，没能实现用户之间的评论等交互功能；第四，本系统安全性不高，只是使用了验证码登录，没能实现短信验证功能。总之，本系统的需要改进的地方还有很多。