

中山大学饭堂菜品评价系统-测试文档

目录

中山大学饭堂菜品评价系统-测试文档	1
一、背景	1
二、测试人员	1
三、测试时间	1
四、测试平台	1
五、测试内容	1
1. 基本功能（黑盒）测试	1
2. 白盒测试	5
六、测试总结	8

一、背景

本文档为许嘉璋（学号 21307275）于中山大学 2024 年春季软件工程期末大作业中为其设计的“中山大学饭堂菜品评价系统”编写的测试文档。

二、测试人员

许嘉璋，负责所有测试，包括基本功能（黑盒）测试、白盒测试，验证了软件设计方案和流程的可行性。

三、测试时间

2024.5 ~ 2024.7

四、测试平台

在 Ubuntu20.04 下，使用 IDE 为 eclipse-2024-03（已配置好 dynamic web 开发组件）、数据库为 MySQL-8.0.37、服务器为 tomcat-9.0.89。

五、测试内容

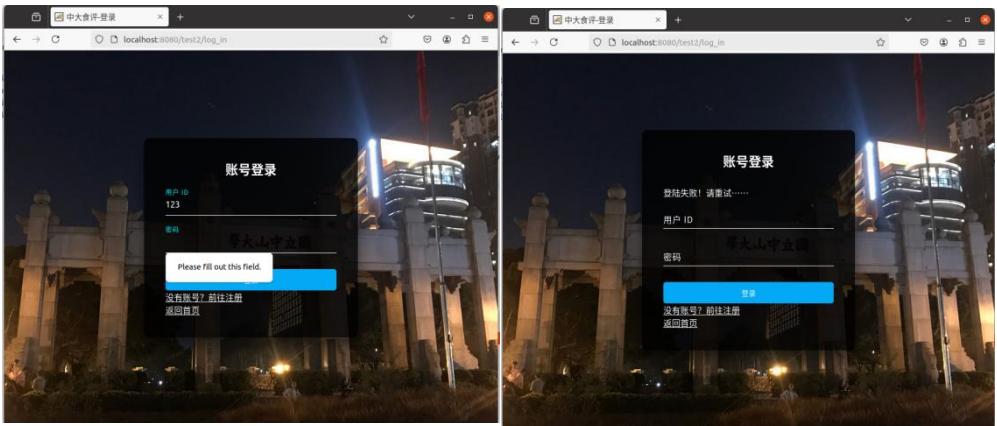
1. 基本功能（黑盒）测试

（1）访问首页。初始时在浏览器中输入正确的网站地址，网站正常显示；否则访问不

到该网站。功能正常。



(2) 登录功能。从主页右上方按钮可以跳转到登录界面, 仅当输入正确的用户 ID 和匹配的密码时才能登录成功; 否则警告登录失败。功能正常。



(用户 ID 缺失或密码缺失)

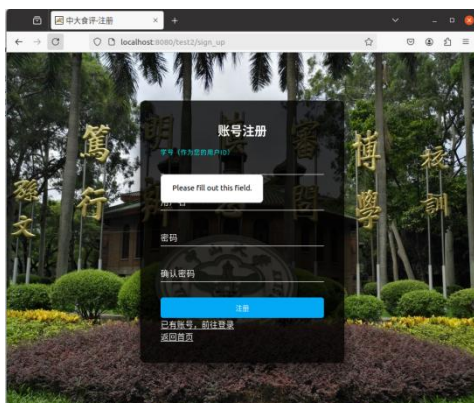
(用户 ID 错误或密码错误)



(登录成功后自动跳转到首页, 注意此时右上方按钮发生变化。原登录按钮被“查

询账户信息”替换。)

(3) 注册功能。通过登录界面可以跳转到注册界面, 此时必须合理设置各个值才能注册成功并跳转到登录界面、并在登录界面提示注册成功, 否则会给出警告。功能正常。



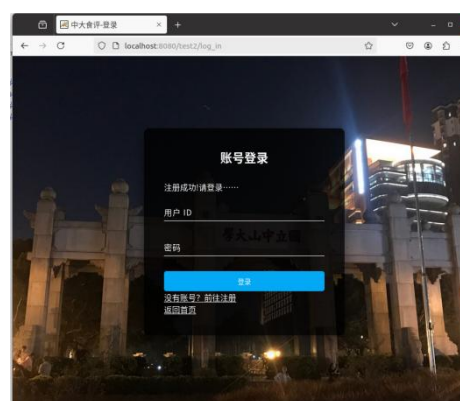
(缺失某个值时)



(填入一个已有用户 ID 时)

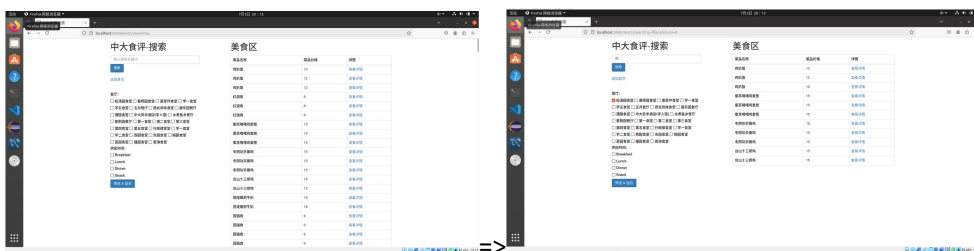


(两次密码不匹配时)



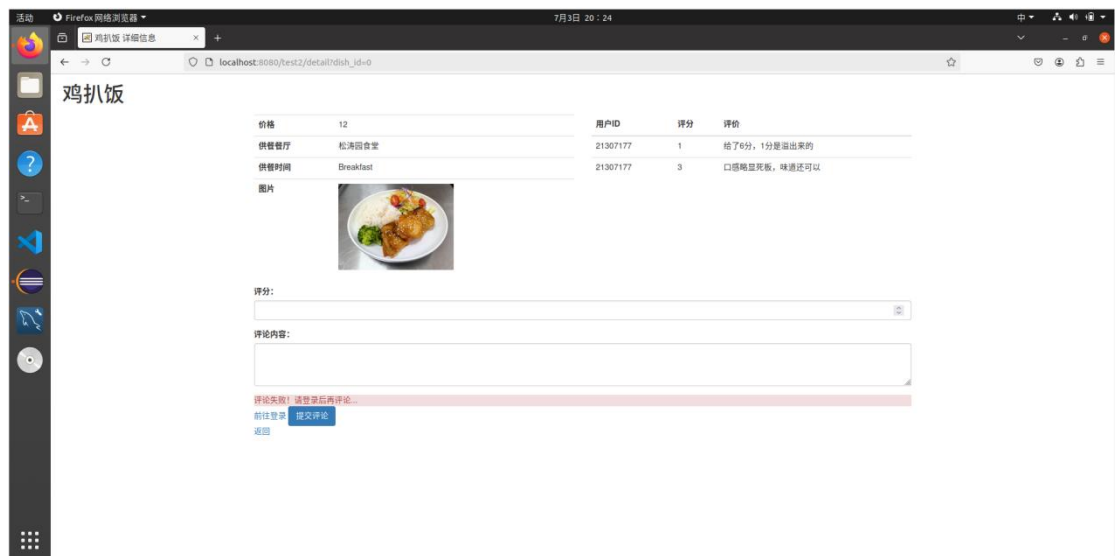
(注册成功时)

(4) 评论查询筛选功能。通过主页可以正常跳转到查询界面, 在查询页面下进行的任何查询都可以得到正确结果。功能正常。

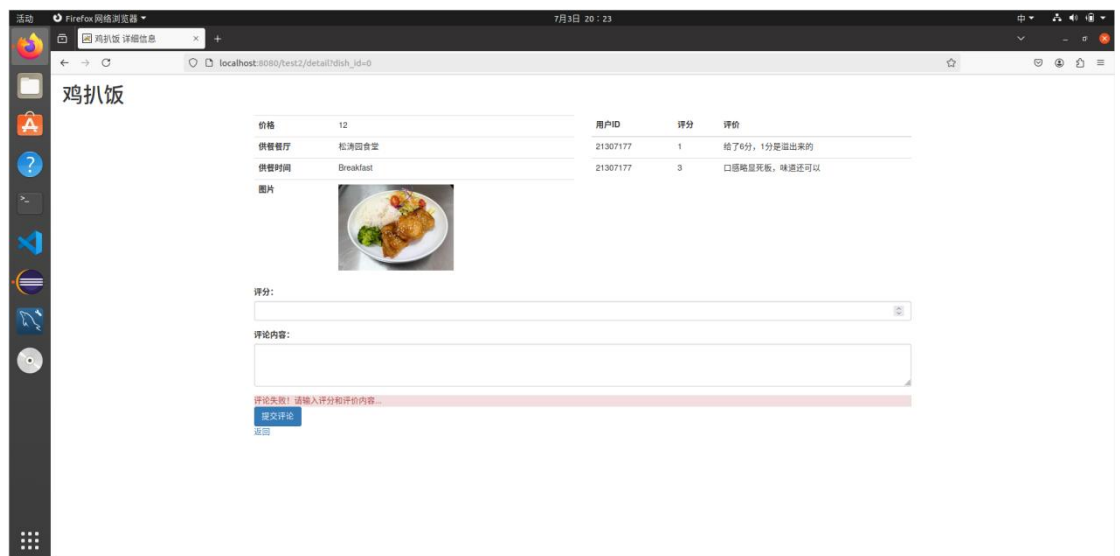


(查询页面及查询过程)

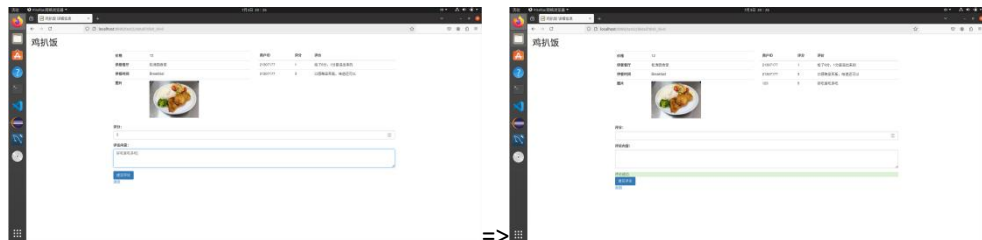
(5) 菜品详细信息查看和评论功能。通过查询界面菜品后的按钮可以正常跳转到菜品详情页, 详情页中必须在登录状态下填入 1~5 之间的评分和评价内容才可以提交评论。功能正常。



(非登录状态下尝试评价会警告并跳出登录按钮)

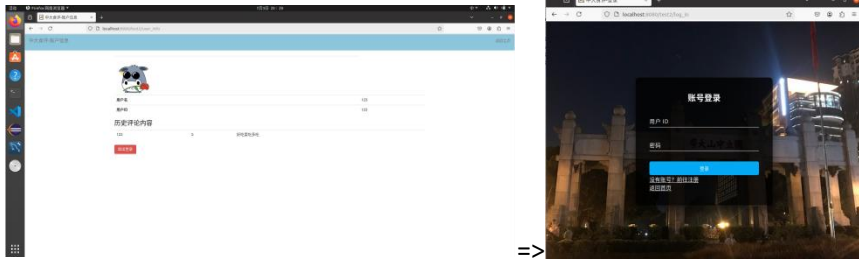


(登录状态下尝试提交非法评价会警告)



(登录状态下提交合法评价, 可以看到评价被网页所接受。)

(6) 退出登录功能。登录状态下进入账户信息页面, 点击“退出登录”按钮即可退出并跳转到登录界面。功能正常。



(退出登录)

(7) 网页制作者信息查看功能。从主页右上方“关于我们”按钮可以跳转到相关页面。功能正常。



(about_us 页面访问)

2. 白盒测试

(1) 时间显示功能。Index.jsp 和 about_us.jsp 下都有下列关于实时时间显示的 JavaScript 代码, 由此测试并观察到: 对应网页下方可见实时时间显示正确。功能正常。

```
<script>
function updateCurrentTime() {
    var now = new Date();
    var year = now.getFullYear();
    var month = now.getMonth() + 1;
    var date = now.getDate();
    var hours = now.getHours();
    var minutes = now.getMinutes();
    var seconds = now.getSeconds();

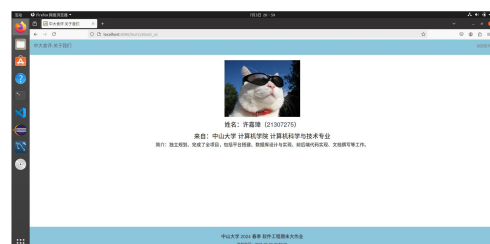
    var monthString = month < 10 ? "0" + month : month;
    var dateString = date < 10 ? "0" + date : date;
    var hoursString = hours < 10 ? "0" + hours : hours;
    var minutesString = minutes < 10 ? "0" + minutes : minutes;
    var secondsString = seconds < 10 ? "0" + seconds : seconds;

    var currentTimeString = year + "-" + monthString + "-" + dateString + " " +
        hoursString + ":" + minutesString + ":" + secondsString;
    document.getElementById("current-time").innerText = "当前时间: " + currentTimeString;
}

// 每秒更新一次时间
setInterval(updateCurrentTime, 1000);
</script>
```

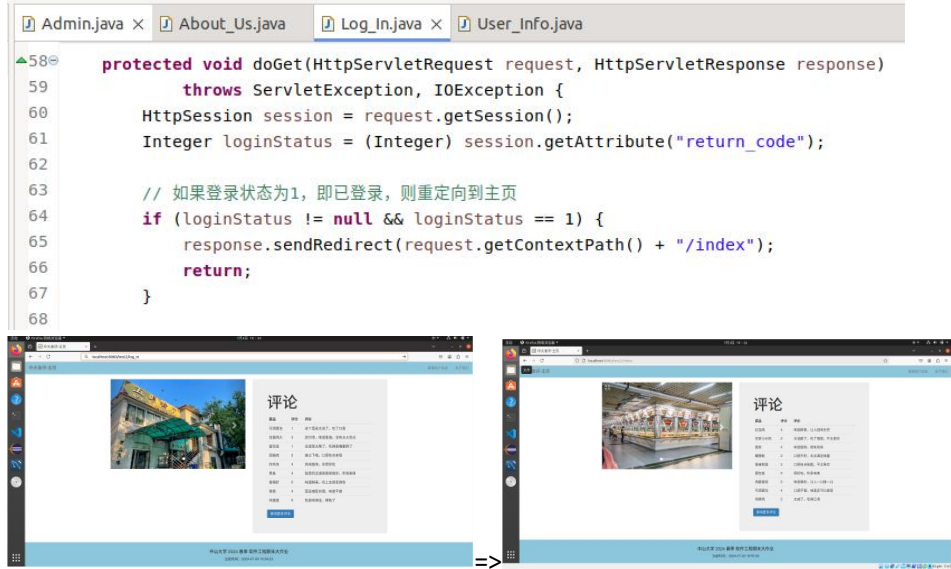


(/index)



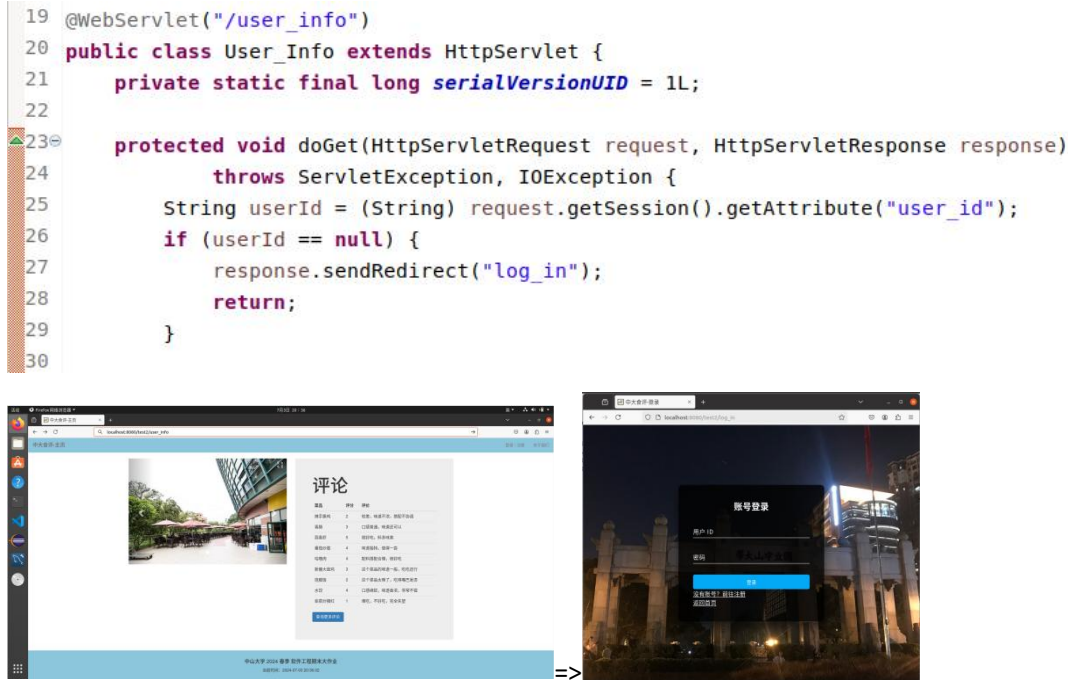
(about_us)

(2) 异常登录排除功能。log_in.java 和 sign_up 下有类似如下重定向代码，由此测试并观察到：在登录状态下试图直接通过网址、或通过注册界面跳转访问登录界面以造成系统异常是不可行的，此时会将网址重定向到首页。功能正常。



(登录状态下访问/log_in 或/sign_up 会重定向回主页)

(3) 异常访问排除功能。user_info.java 下有类似重定向代码，由此测试并观察到：非登录状态下尝试通过网址访问账户信息界面会被重定向到登录界面。功能正常。



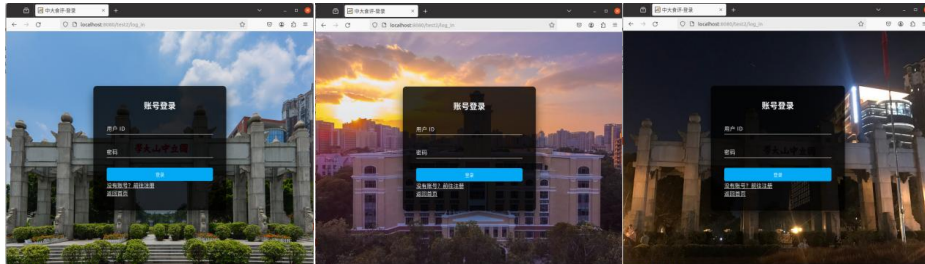
(非登录状态下访问/user_info 会重定向到登录界面)

(4) 界面自适应时间功能。log_in.java 下有如下代码，由此测试并观察到：在一天的 3 个时段内登录界面会有不同的背景，实现美化效果。功能正常。

```

69 // 根据时间确定登录界面背景
70 LocalTime currentTime = LocalTime.now();
71 String backgroundImage;
72 String imagePath = request.getContextPath() + "/static/images/Background/";
73 if (currentTime.isAfter(LocalTime.of(7, 0)) && currentTime.isBefore(LocalTime.of(14, 0))) {
74     backgroundImage = imagePath + "day1.jpg";
75 } else if (currentTime.isAfter(LocalTime.of(14, 0)) && currentTime.isBefore(LocalTime.of(18, 0))) {
76     backgroundImage = imagePath + "day2.jpg";
77 } else {
78     backgroundImage = imagePath + "day3.jpg";
79 }
80
81 request.setAttribute("backgroundImage", backgroundImage);

```



(7a.m.-2p.m.)

(2p.m.-6p.m.)

(else)

(5) 查询状态保持功能。search.jsp 下有如下 JavaScript 脚本，由此测试并观察到：每次查询条件都会保持。功能正常。

```

<script>
// 当页面加载时，检查并设置复选框的状态以及输入框的内容
window.onload = function () {
    var form = document.getElementById('filterForm');
    var checkboxes = form.querySelectorAll('input[type="checkbox"]');
    var queryInput = document.getElementById('query');

    var urlParams = new URLSearchParams(window.location.search); // 获取URL参数并解析为对象
    var savedQuery = urlParams.get('q'); // 设置查询输入框的值
    if (savedQuery) {
        queryInput.value = savedQuery;
    }

    checkboxes.forEach(function (checkbox) { // 设置复选框的状态
        if (urlParams.getAll(checkbox.name).includes(checkbox.value)) {
            checkbox.checked = true;
        } else {
            checkbox.checked = false;
        }
    });

    checkboxes.forEach(function (checkbox) { // 监听复选框状态变化，并更新URL参数
        checkbox.addEventListener('change', function () {
            updateURLParams();
        });
    });

    queryInput.addEventListener('input', function () { // 监听输入框内容变化，并更新URL参数
        updateURLParams();
    });

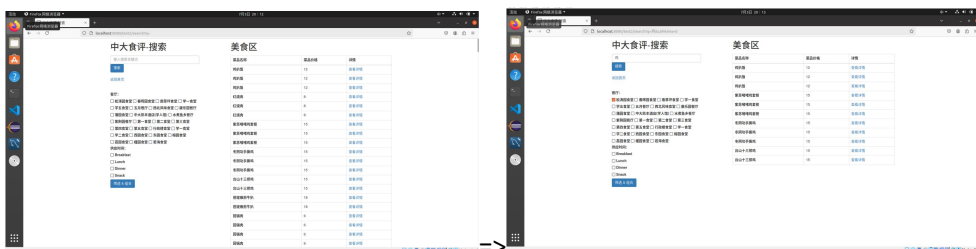
    function updateURLParams() {
        var url = new URL(window.location);
        var params = new URLSearchParams(url.search);

        params.set('q', queryInput.value); // 更新查询参数

        params.delete('cafeteria'); // 清空并更新复选框参数
        params.delete('time');
        checkboxes.forEach(function (checkbox) {
            if (checkbox.checked) {
                params.append(checkbox.name, checkbox.value);
            }
        });

        window.history.replaceState({}, '', `${url.pathname}?${params}`); // 更新浏览器URL但不刷新页面
    }
};
</script>

```



(查询页面及查询过程)

(6) 首页随机展示功能。index.java 下有如下代码，由此测试并观察到：每次访问首页都会有不同的饭堂图片和不同的评论内容展示。功能正常。

```
// 生成幻灯片内容
StringBuilder indicators = new StringBuilder();
StringBuilder slides = new StringBuilder();
List<Integer> numbers = new ArrayList<>();
for (int i = 0; i < 11; i++) {
    numbers.add(i);
}
Collections.shuffle(numbers);
List<Integer> randomNumbers = numbers.subList(0, 5);
for (int i = 0; i < 5; i++) {
    int num = randomNumbers.get(i);
    String imagePathCafeteria = request.getContextPath() + "/static/images/Cafeteria/" + num + ".png";
    indicators.append("<li data-target='#myCarousel' data-slide-to='").append(i).append("'")
        .append(i == 0 ? " class='active'" : "").append("></li>");
    slides.append("<div class='item'").append(i == 0 ? " active" : "").append(">").append("<img src='")
        .append(imagePathCafeteria).append("' alt='Image '").append(num + 1).append(">").append("</div>");
}
request.setAttribute("indicators", indicators.toString());
request.setAttribute("slides", slides.toString());

// 从数据库获取随机 20 条评论
List<String[]> comments = new ArrayList<>();
try {
    Class.forName("com.mysql.cj.jdbc.Driver");
    Connection connect = DriverManager.getConnection("jdbc:mysql://localhost:3306/my_homework", "root",
        "xu2580");
    String query = "SELECT d.dish_name, c.score, c.content " + "FROM Comments c "
        + "JOIN Dish d ON c.dish_id = d.dish_id " + "ORDER BY RAND() LIMIT 9";
    PreparedStatement stmt = connect.prepareStatement(query);
    ResultSet rs = stmt.executeQuery();
    while (rs.next()) {
        String[] comment = new String[3];
        comment[0] = rs.getString("dish_name");
        comment[1] = rs.getString("score");
        comment[2] = rs.getString("content");
        comments.add(comment);
    }
    rs.close();
    stmt.close();
    connect.close();
} catch (Exception e) {
    e.printStackTrace();
}
request.setAttribute("comments", comments);
```

(7) *其它测试内容与黑盒测试部分相近，只是可以通过代码解释网页行为，故不做赘述。

六、测试总结

网站现有功能都可以正常完成，可见软件设计方案和流程的可行性是没有问题的。