

**软件工程**

智能垃圾分类系统测试报告

学 院： 计算机工程学院 班 级：网络2113

小 组： 老鼠爱大米 指导老师：张 敏

时 间：2024/5/8

目录

[1. 导言 3](#_Toc3178)

[1.1 编写目的 3](#_Toc9656)

[1.2 项目范围 3](#_Toc12690)

[1.3 引用标准 3](#_Toc7844)

[1.4 参考资料 3](#_Toc7614)

[1.5 版本更新信息 3](#_Toc9458)

[2. 测试项目介绍 4](#_Toc15229)

[2. 1 项目介绍 4](#_Toc1622)

[2.2 测试需求 4](#_Toc402)

[3. 应用环境 4](#_Toc9974)

[3.1 软件环境 4](#_Toc12849)

[4. 测试过程 5](#_Toc7227)

[4.1 静态测试 5](#_Toc282)

[4.2 动态测试 9](#_Toc22264)

[5. 测试结论 17](#_Toc16287)

[5.1 测试用例执行结论 17](#_Toc8150)

[5.1.1 功能测试 17](#_Toc26077)

[5.1.2 性能测试 17](#_Toc50)

[5.2 测试结论 18](#_Toc19115)

[6. 测试体会 18](#_Toc17096)

**1.** 导言

**1.1** 编写目的

该文档是关于智能垃圾分类系统项目的测试和可信性评估的报告。本文档的预期读者包括：

 软件评估人员

 编程人员

 测试人员

**1.2** 项目范围

该文档主要包括了智能垃圾分类系统项目的功能和性能测试结果分析，以及相应的可信研究报告。

**1.3** 引用标准

[1] 《软件工程案例教程第 2 版》韩万江等 机械工业出版社

**1.4** 参考资料

[1] 《软件项目管理案例教程第 2 版》韩万江等机械工业出版社

**1.5** 版本更新信息

本文档的更新记录如表 1- 1 所示。

表 1- 1 版本更新信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 001 | 2024.5.8 | 1.0 | 全部 | 初始发布版本 |
| 002 | 2024.5.9 | 1.1 | 模块设计 | 增加 |
| 003 | 2024.5.10 | 2.0 | 模块设计 | 修改 |
| 004 | 2024.5.12 | 3.0 | 性能测试 | 增加 |
| 005 | 2024.5.13 | 3.1 | 性能测试 | 修改 |
| 006 | 2024.5.14 | 4.0 | 全部 | 修改 |

**2.** 测试项目介绍

**2. 1 项目介绍**

自19年垃圾分类政策试行，全国从46个垃圾分类试点城市实行开始，垃圾分类就变成了城市生活中的我们的一个重要需求。垃圾分类有助于我们日常生活的清洁方便，同样也有助于垃圾回收的高效利用。但对于许多人来说，如何甄别垃圾类型并正确投放是一件比较困难的事情。在大城市生活的人们日常的时间精力已经被生活所压榨殆尽，没有更多的时间和意愿去系统的学习垃圾分类的专业知识。 传统的人工垃圾分拣技术存在着效率低、耗时长、需要大量人力、分挑环境差等问题，如何有效地智能化地进行垃圾分类以及成为亟待解决的问题。

**2.2 测试需求**

本次测试的主要需求如下：

1. 文档、代码的检测

通过需求规格、概要设计、详细设计文档评审，以及源代码的走查，作为评估标准之一。

2. 功能需求测试

参见需求规格中的功能需求。功能测试覆盖到每个功能能点，检查并记录异常的功能点。

3. 性能需求测试

参见需求规格中的性能需求。其中，用户机型要求 48 款以上。性能测试覆盖到常规反应时间测试、兼容性测试。

**3.** 应用环境

应用环境可以分为硬件环境、软件环境。

**3.1** 软件环境

操作系统：Windows11

开发工具：IDEA

版本管理工具：Github

测试工具：Postman

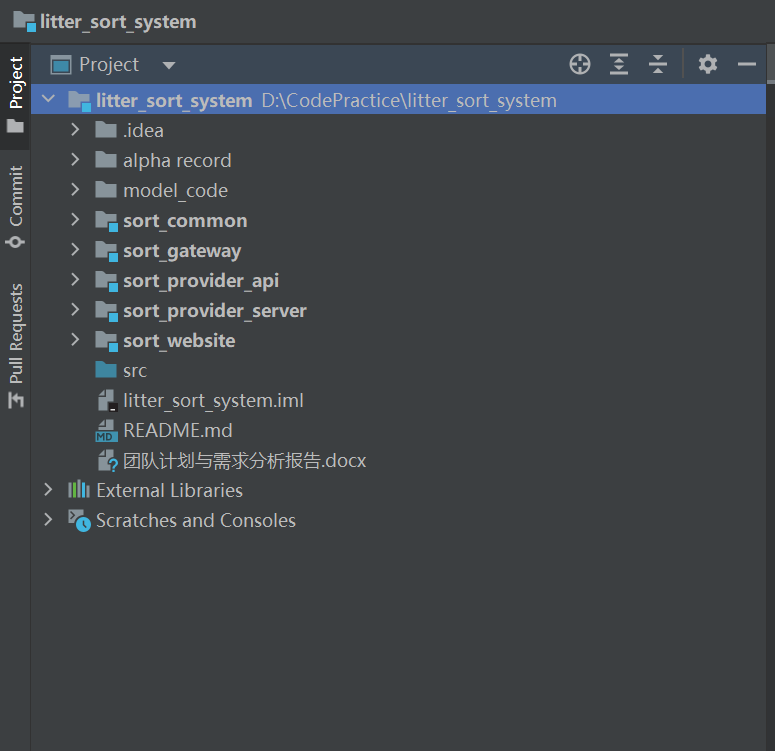
**4.** 测试过程

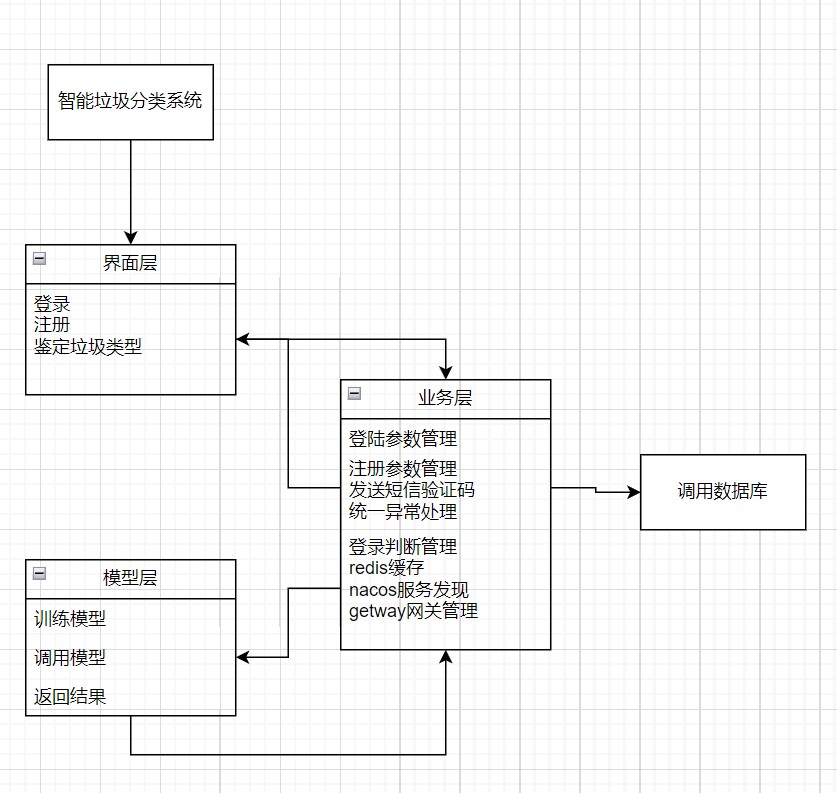
**4.1** 静态测试

静态测试中包括对项目文档、系统设计和代码的评审检测过程，文档属性包括：完整性、规范性、正确性、可维护性、创新性、复用。设计主要分为两个属性检测：架构设计、模块设计。代码中我们将其分为六个属性，包括：完整性、规范性、正确性、可维护性、创新性、复用。

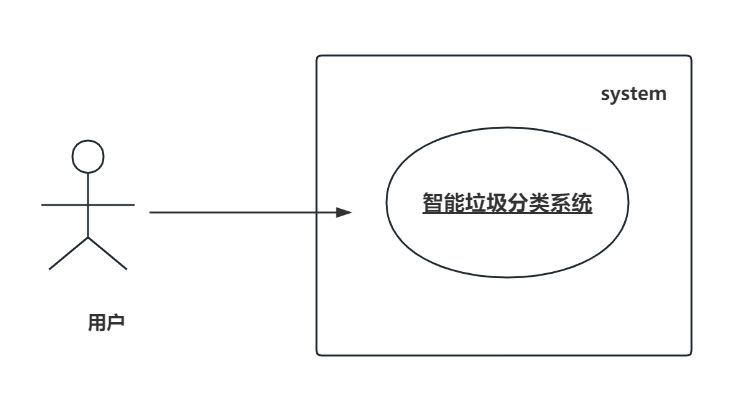
（一）设计属性

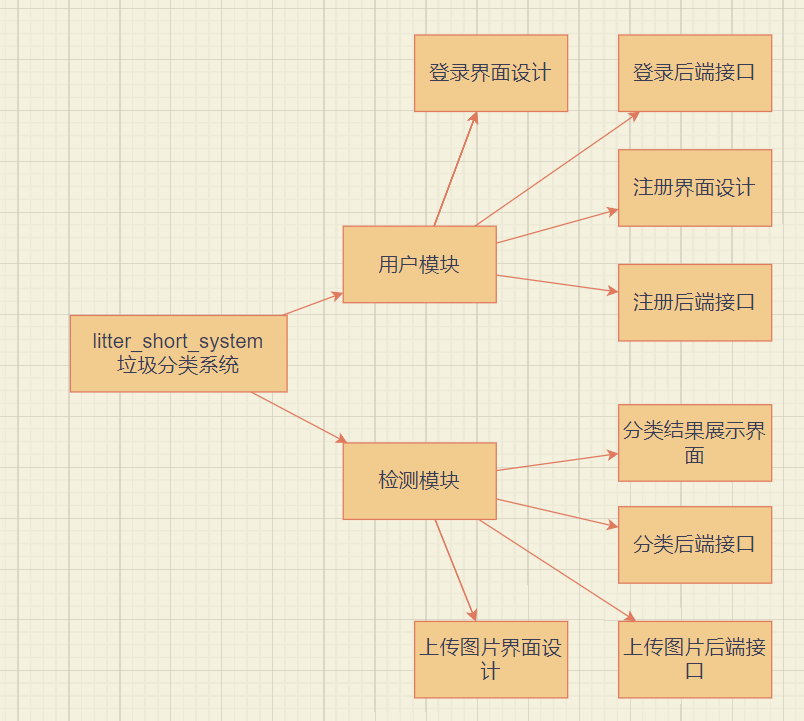
架构设计：



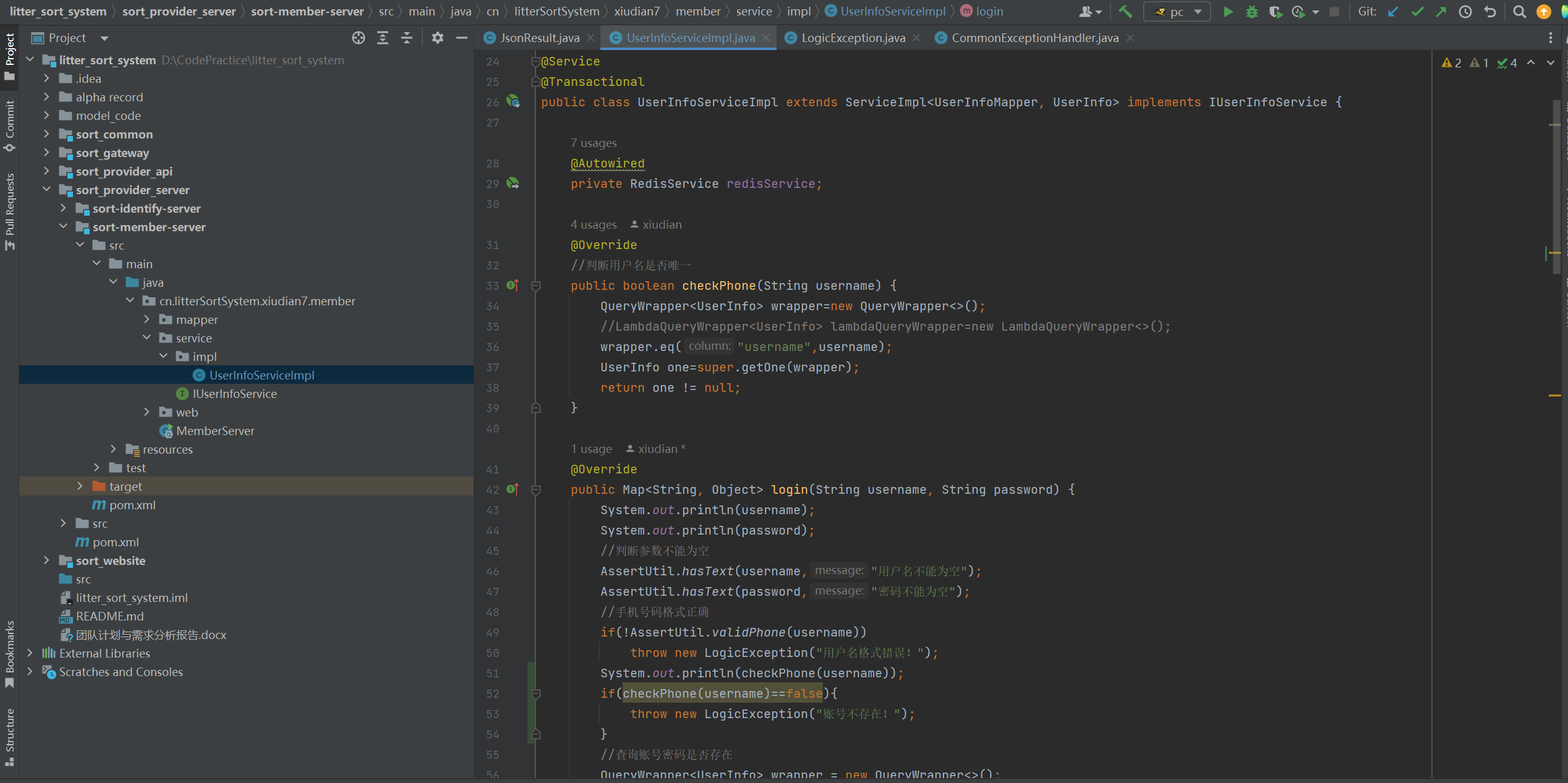


模块设计：





（二）代码属性



**4.2** 动态测试

动态测试主要针对产品的功能性、性能性，其中包括安全性，基本上是基于

测试用例的执行。

|  |  |
| --- | --- |
| 测试工作安排 | |
| 测试人员 | 测试任务 |
| 陈曦 | 登录功能检查、垃圾分类类型鉴别 |
| 李睿 | 深度学习准确性 |
| 郑博文 | 页面检测 |
| 杨雨佳 | 注册功能 |
| 范兴宇 | 前端人性化安排检查 |

功能属性证据：

1 、功能测试用例执行统计

|  |  |  |  |
| --- | --- | --- | --- |
| 步骤编号 | 测试步骤 | 预测结果 | 测试结果 |
| LitterSort-1 | 登录账号为空 | 提示账号为空 | Pass |
| LitterSort-2 | 登录账号格式错误 | 提示格式错误 | Pass |
| LitterSort-3 | 登录密码为空 | 提示密码为空 | Pass |
| LitterSort-4 | 登录密码错误 | 提示密码错误 | Pass |
| LitterSort-5 | 登入账号密码正确 | 提示账号密码正确并跳转 | Pass |
| LitterSort-6 | 未登录上传图片鉴别 | 提示请先登录 | Pass |
| LitterSort-7 | 上传图片鉴别结果 | 输出垃圾类型 | Pass |
| LitterSort-8 | 照片中有些杂物 | 准确识别 | Fail |
| LitterSort-9 | 鉴定代码 | 后端控制台返回正确结果 | Pass |
| LitterSort-10 | 页面正确显示 | 点击每个按键有回应 | Pass |
| LitterSort-11 | 页面参数正确传送到后台 | 后台正确接受参数 | Pass |
| LitterSort-12 | 账号已经被注册 | 提示账号已经被注册 | Pass |
| LitterSort-13 | 发送验证码 | 正确发送验证码并给出提示 | Pass |
| LitterSort-14 | 注册账号格式错误 | 提示账号格式错误 | Pass |
| LitterSort-15 | 注册时两次密码不一致 | 提示两次密码不一致 | Pass |
| LitterSort-16 | 正确填写信息 | 注册成功数据库添加代码并且跳转登录界面 | Pass |
| LitterSort-17 | 使用拍照功能进行图片识别输出 | 输出垃圾类型并且进行语言输出 | Fail |

2 、功能测试缺陷统计

本团队针对智能垃圾分类系统项目，共设计17类功能测试用例。并通过预期结果与实际结果的比较，发现缺陷共计2处。缺陷对应的编号以及操作步骤如下图所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 测试用例名称 | 测试环境 | 操作步骤 |
| 1 | 拍照识别垃圾 | IDEA | 点击拍摄 |
| 2 | 图片里面有杂物 | IDEA | 识别精度不够 |

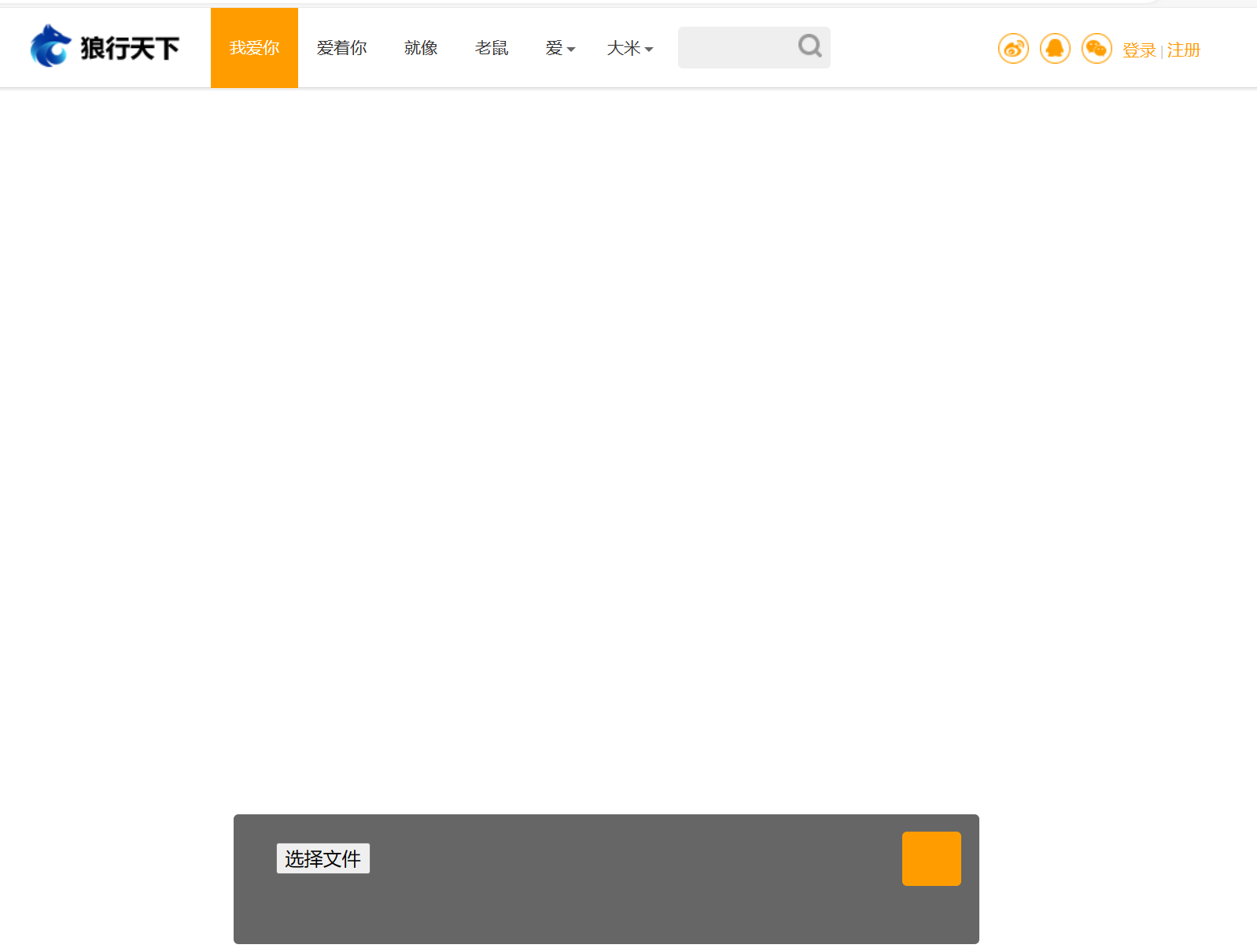
缺陷统计结果如下所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 预测结果 | 实测结果：问题描述 | 能否重现 | 测试发生概率 | Bug状态 |
| 实现此功能 | 未开发拍照识别功能 | 能 | 100% | 严重 |
| 输出正确垃圾类型结果 | 识别垃圾类型错误 | 能 | 100% | 严重 |

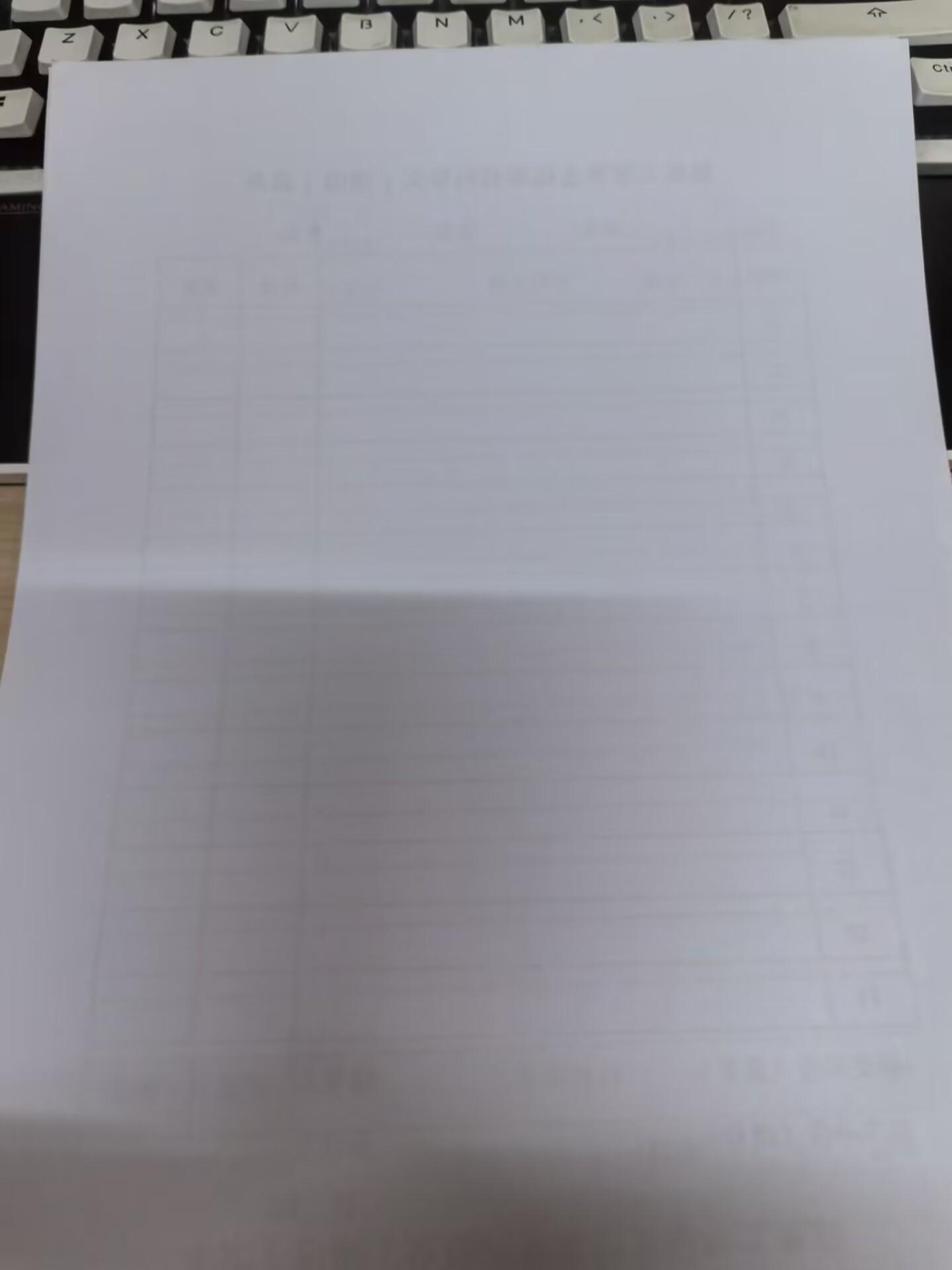
相应的截图为：

1. 用例一：拍照识别

未实现拍照识别功能



1. 用例二：有杂物的垃圾识别

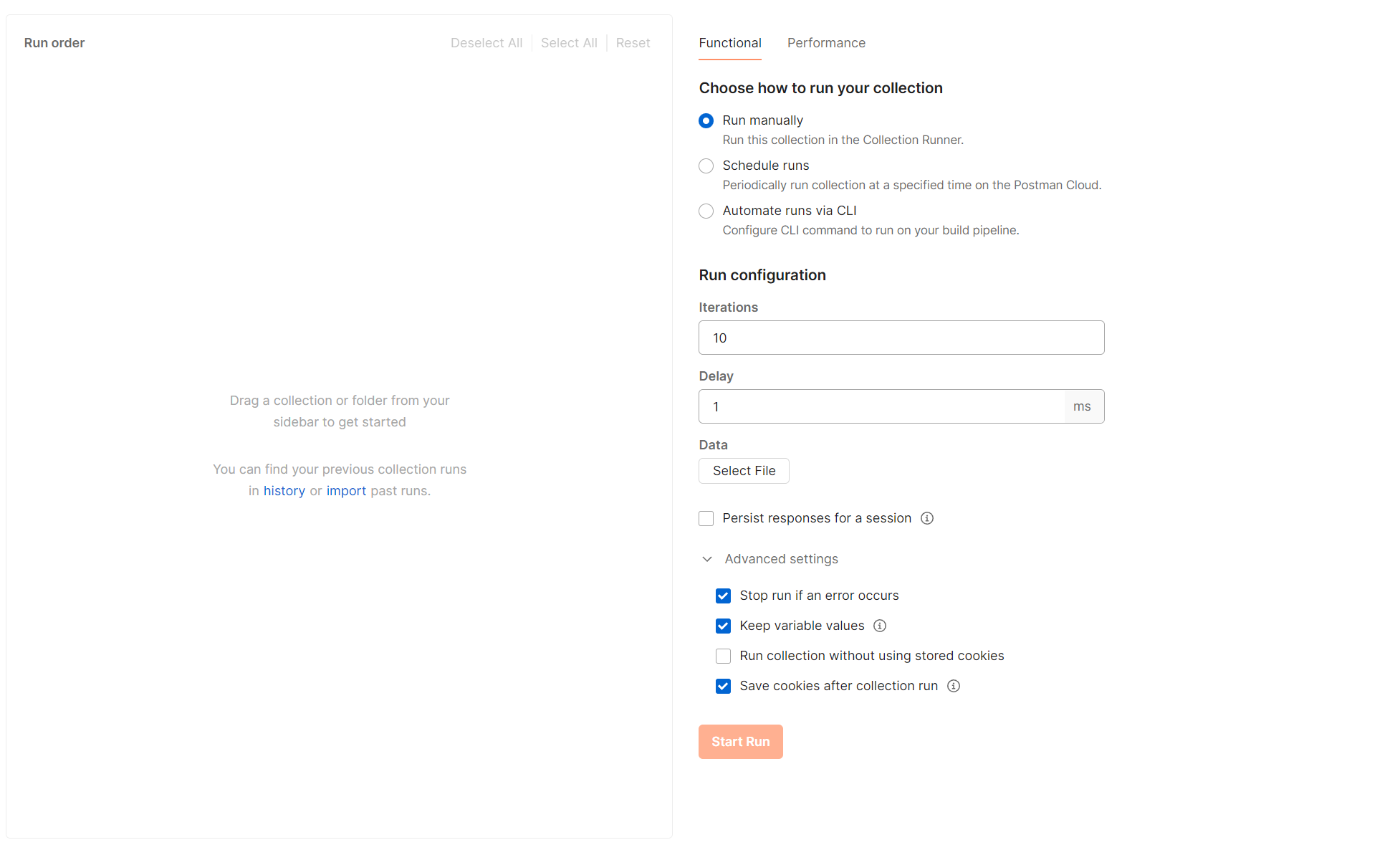


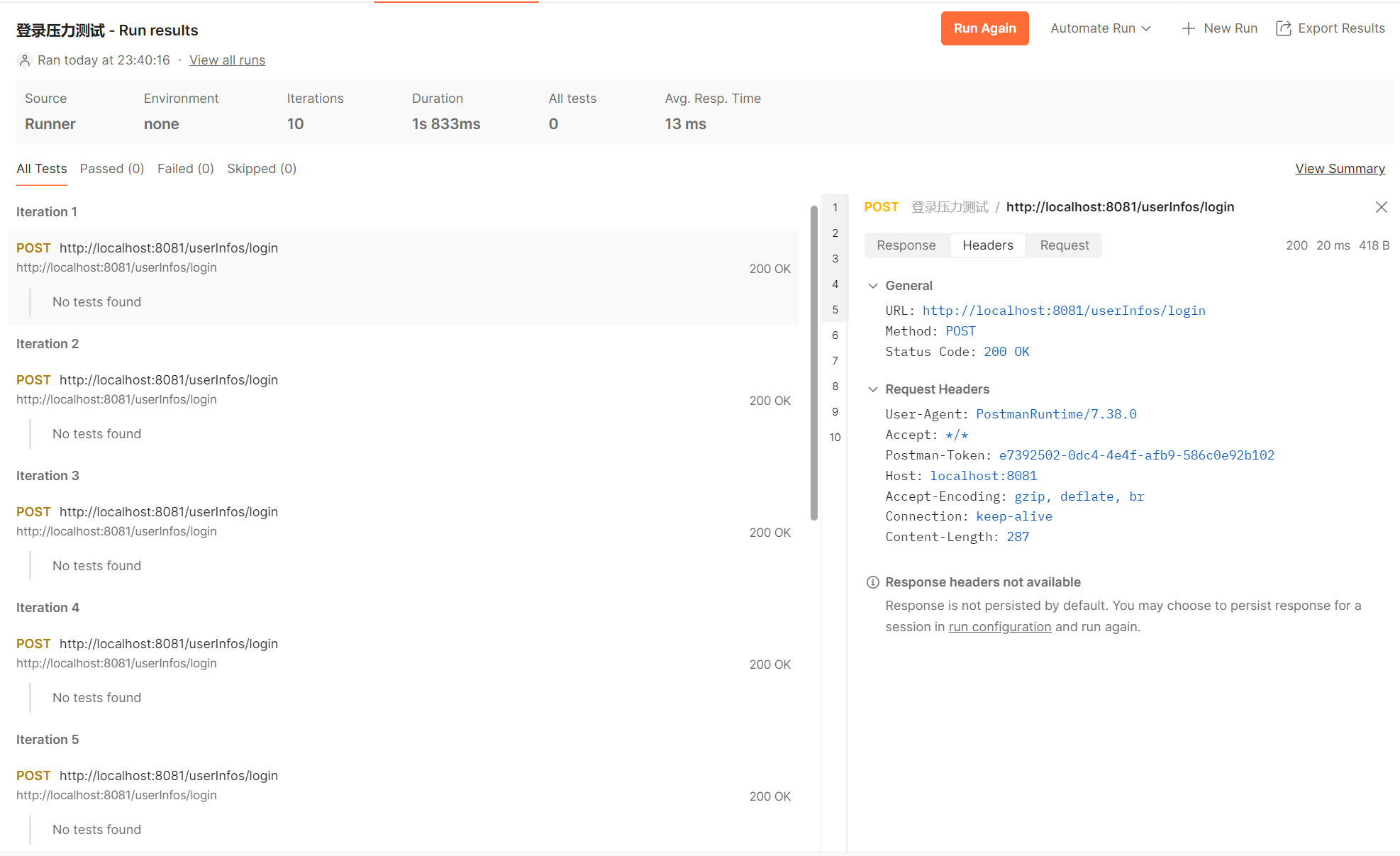


识别成为纸皮

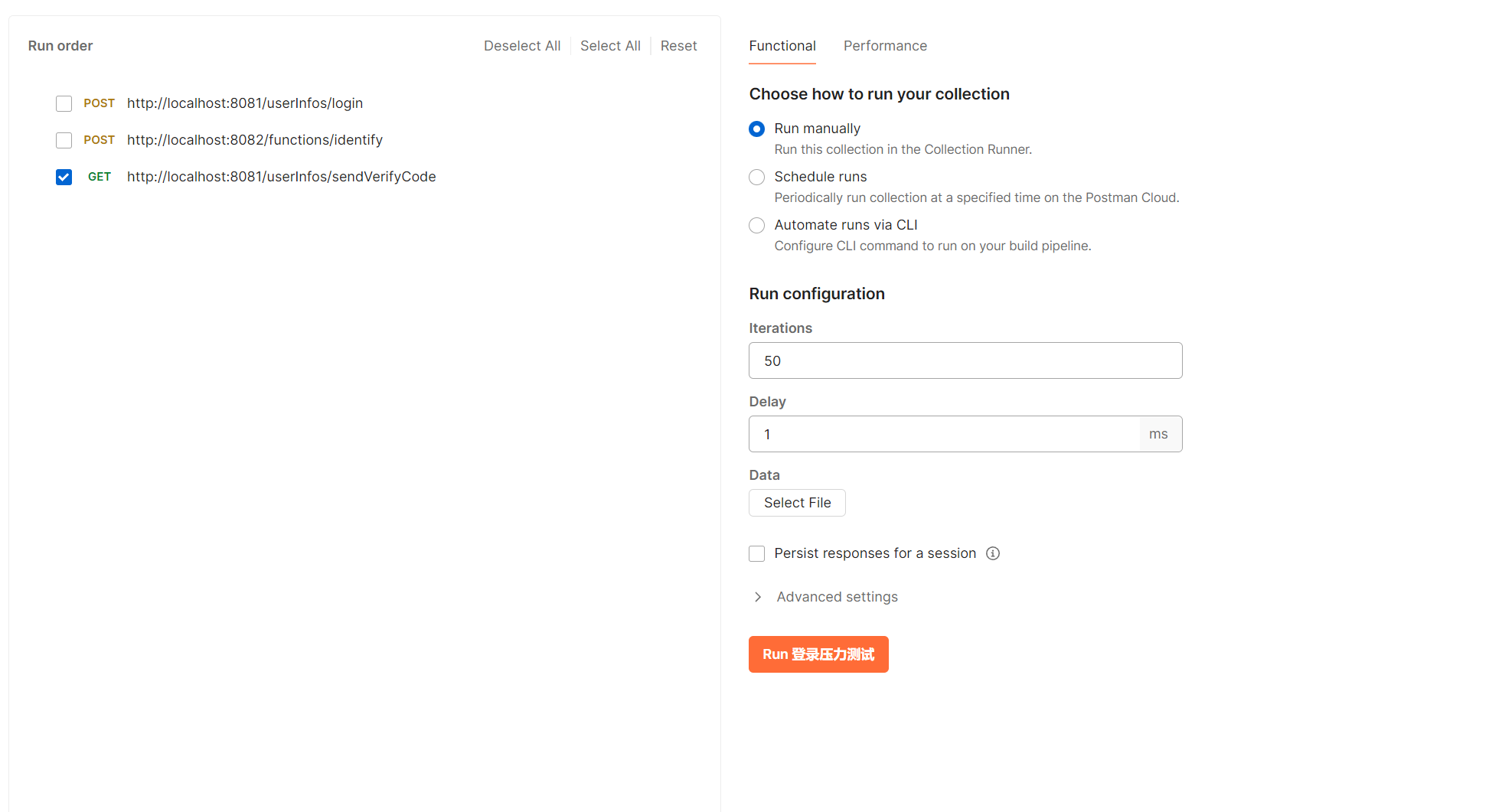
登录压力测试：

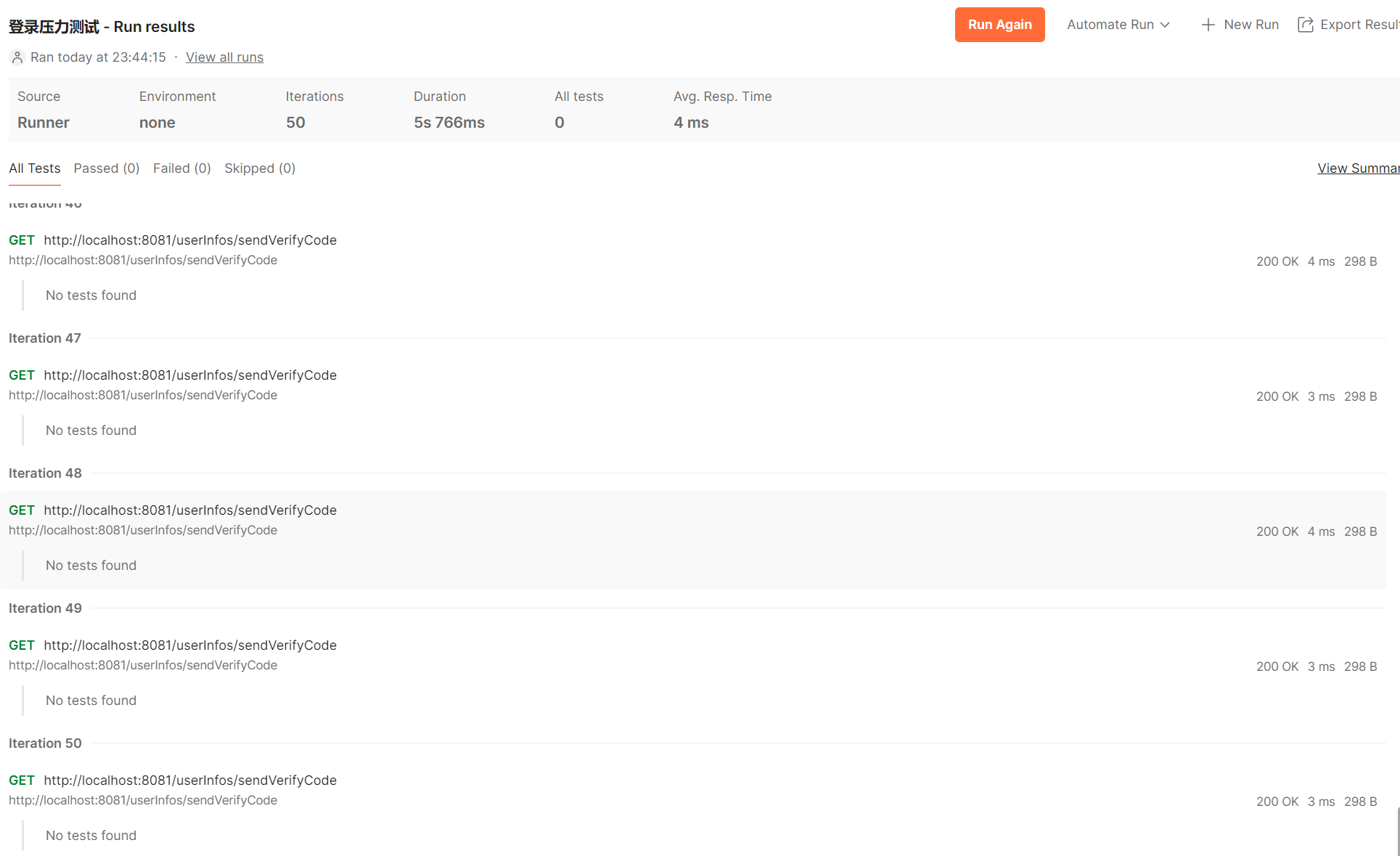
连续发送10次请求，间隔1ms





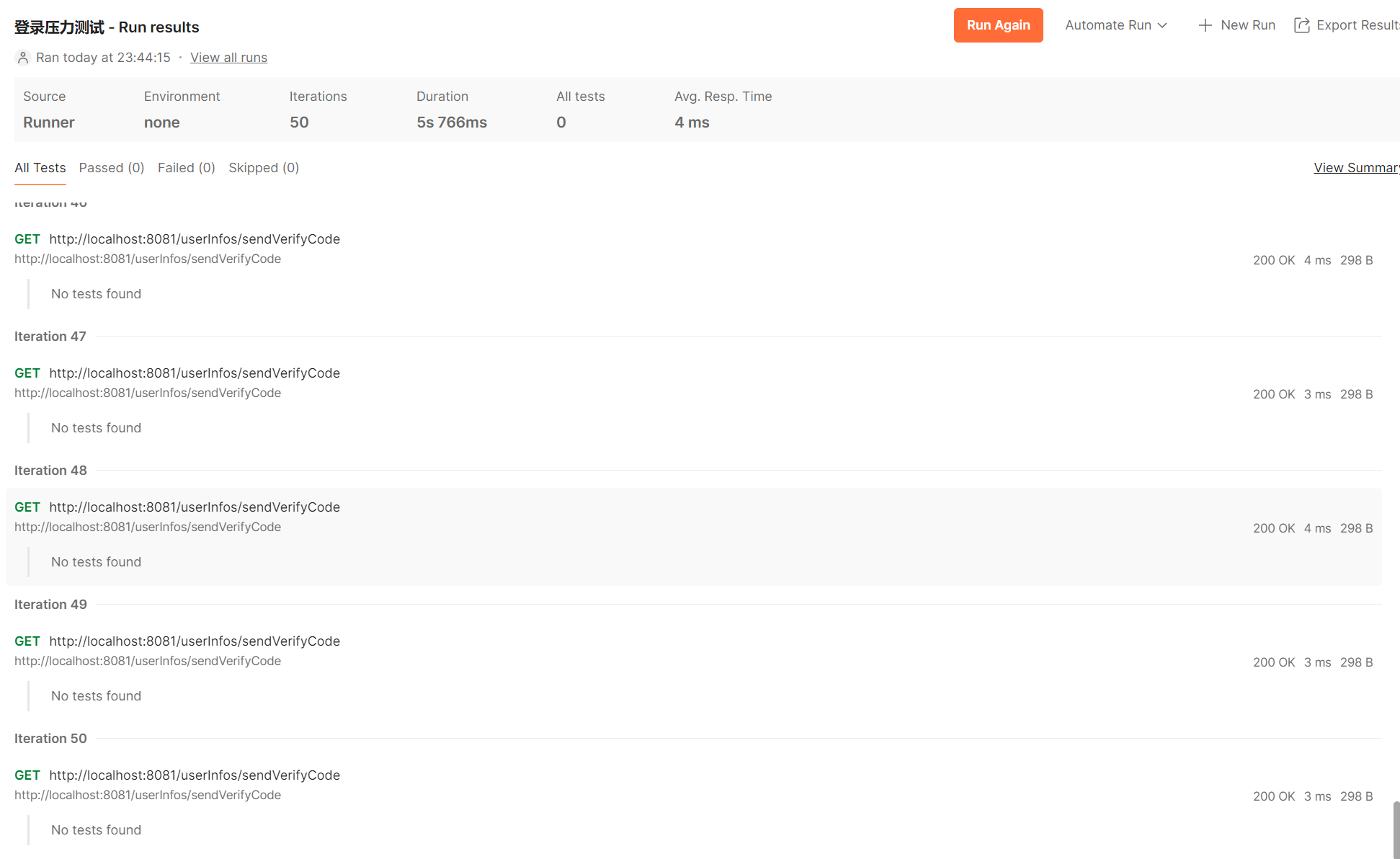
对发送验证码进行50次压力测试，间隔时间1ms



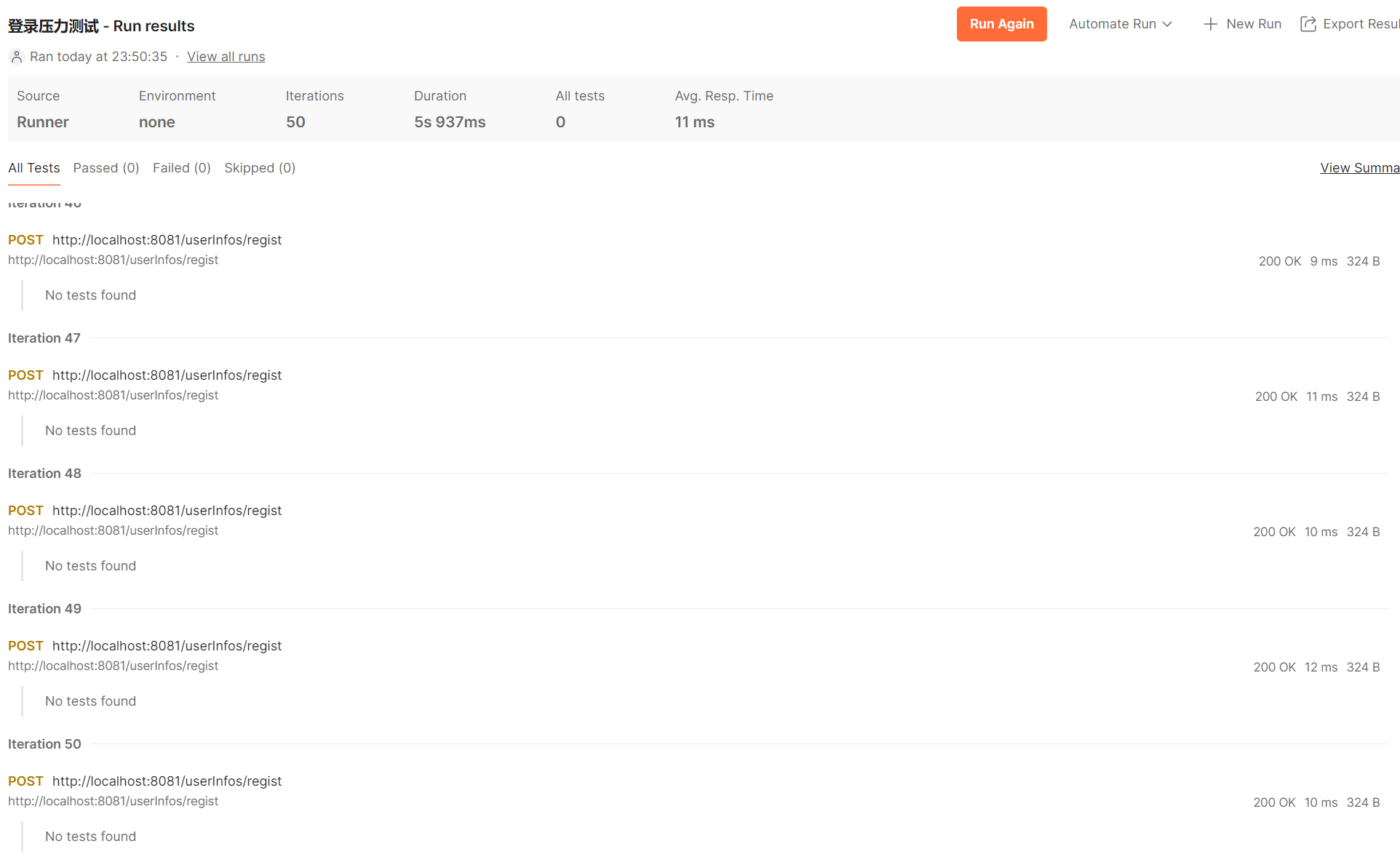


运行正常

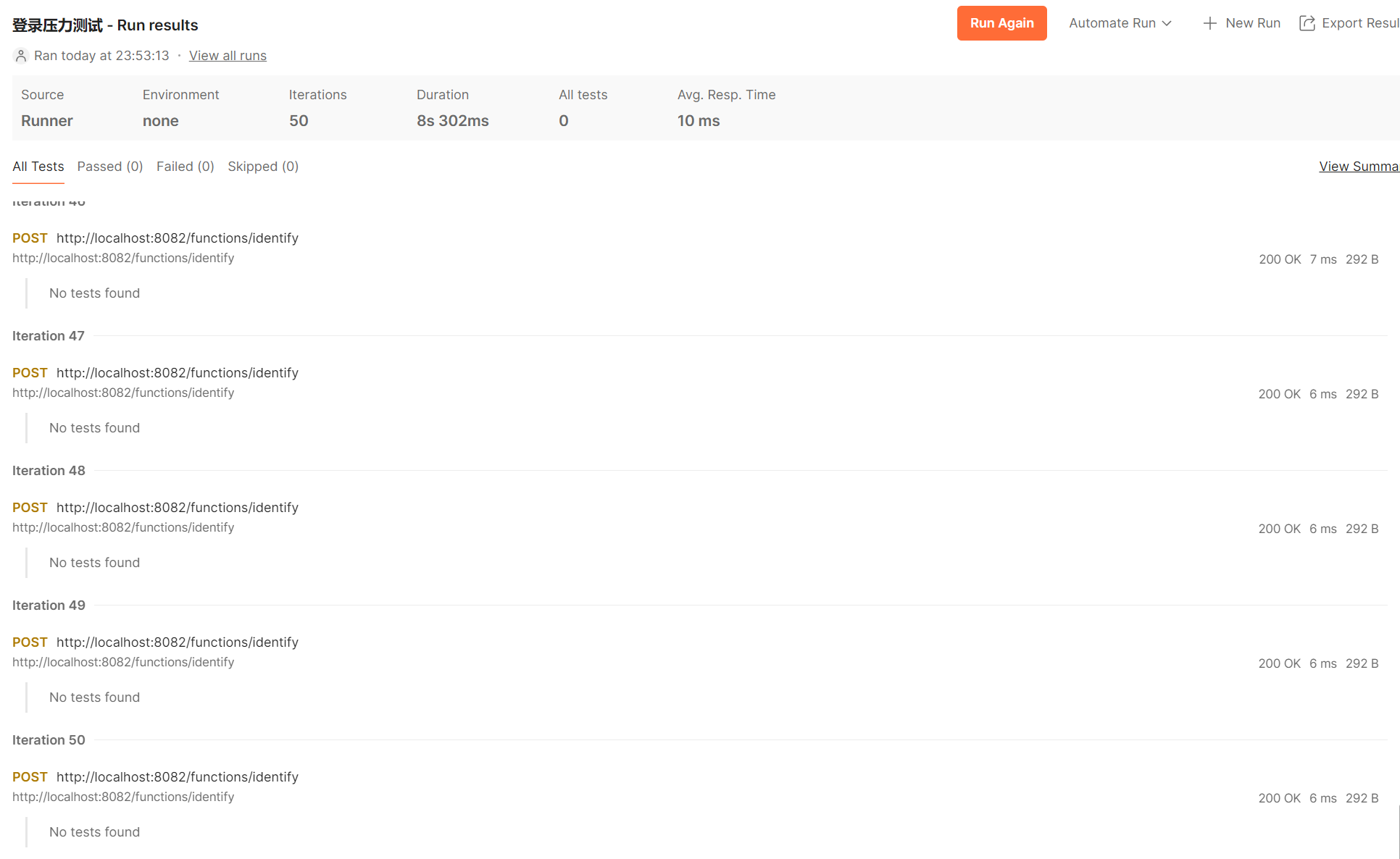
对检查账号是否存在模块进行压力测试 50次，间隔时间1ms



对注册模块进行压力测试 并发数50 1ms



鉴定模块



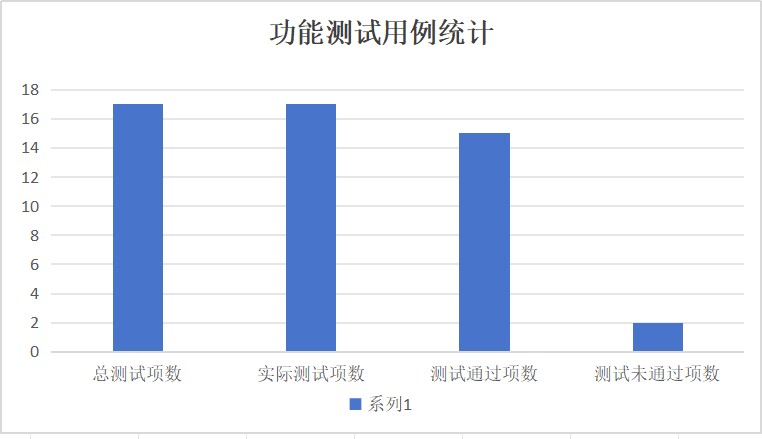
**5.** 测试结论

**5.1 测试用例执行结论**

本次测试分功能测试和性能测试两个方面。

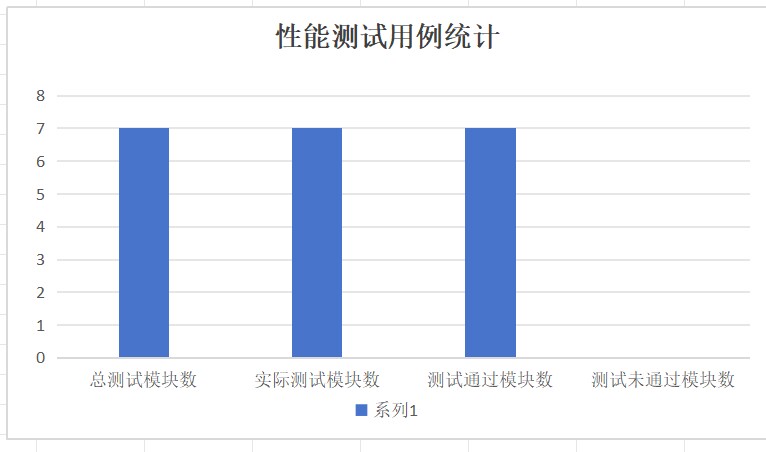
**5.1.1 功能测试**

功能测试中测试项目数共27个，实际测试项目数27个，未通过测试项目数4个。测试项通过率85.19%。测试用例结果统计图如下所示：



**5.1.2 性能测试**

性能测试中测试机型共50款，实际测试机型有48款，其中有9款未能通过安装耗时的性能测试，测试通过率为81.25%。测试用例结果统计图如下所示：



**5.2 测试结论**

测试过程中，我们使用大量的功能测试、并发测试和性能测试，发现我们所做的系统功能还有一些不完善，比如未实现拍照识别功能还有识别准确度的问题，后来还需要进一步改善，对于性能测试，我们系统表现的还是比较完美的，面对高并发数据，能够很快的进行响应，这一点还是比较让我欣慰的。

1. **测试体会**

在完成一个Web项目的测试工作后，我有了一些心得和体会。首先，在接口测试方面，Postman提供了强大的功能和用户友好的界面，使得创建和发送HTTP请求变得非常简单。通过Postman，可以对API进行CRUD操作的测试，并通过编写JavaScript测试脚本实现自动化测试，提高了测试效率和覆盖率。在压力测试方面，虽然Postman本身不专注于此，但我们可以结合Newman在命令行中执行批量请求来模拟高负载场景，或者将Postman的测试用例导入JMeter或Locust等专业工具，实现更复杂的压力测试。功能测试方面，通过手动测试直观地发现界面和交互上的问题，而自动化测试工具（如Selenium、Cypress）则大大提高了重复性测试的效率和一致性。通过这次Web项目的测试，我深刻体会到测试在软件开发中的重要性，接口测试确保系统功能的正确性和可靠性，压力测试检验系统在高负载下的表现，功能测试全面验证应用的各个功能模块。未来，我将继续深入研究和使用各种测试工具和方法，不断提高测试质量和效率，为开发高质量的软件产品贡献力量。