

网上垃圾分类系统测试计划

V1.0

北京长江软件公司

评审日期：2025 年 10 月 20 日

项目编号：202510001

文档编号：TP-01

密级：内部

第 1 页共 11 页

目录

1 导言	3
1.1 目的	3
1.2 范围	3
1.3 缩写说明	4
1.4 术语定义	4
1.5 引用标准	4
1.6 参考资料	4
1.7 版本更新信息	5
2 测试项目	5
2.1 测试项目的背景	5
2.2 测试要点	5
2.3 测试内容	6
2.3.1 功能测试内容	6
2.3.2 性能与其他非功能测试内容	7
3 测试方法	7
3.1 测试环境	7
3.2 测试工具	8
3.3 测试方法	8
4 测试标准	9
4.1 测试通过/失败标准	9
4.2 测试挂起/恢复条件	9
4.3 系统测试交付结果	10

5 测试计划	10
5.1 角色和职责	10
5.2 测试设计工作任务分解和工作安排	10
6 审批	11

1 导言

1.1 目的

该文档的目的是描述《智能垃圾分类系统》项目的系统测试计划，主要内容包括：

- 测试系统简介；
- 测试方法；
- 测试标准；
- 测试计划。

本文档的预期读者为：

- 开发人员；
- 项目管理人员；
- 测试人员；
- 运维人员。

1.2 范围

该文档定义了智能垃圾分类系统的系统测试方法、测试标准和时间计划，不包括具体的测试用例设计，测试用例将体现在《智能垃圾分类系统测试设计》中。

测试范围主要覆盖：

- 文本垃圾分类接口： /api/classify；
- 批量分类接口： /api/batch-classify；
- 规则管理接口： /api/rules（增删改查与 CSV 导入导出）；
- 统计分析接口： /api/statistics；
- 相似物品推荐接口；
- 图片识别接口： /api/image-classify；
- 主站首页及静态资源。

1.3 缩写说明

API Application Programming Interface, 应用程序编程接口;

CSV Comma-Separated Values, 逗号分隔值文件;

HTTP Hyper Text Transfer Protocol, 超文本传输协议;

REST Representational State Transfer, 一种 Web 服务架构风格。

1.4 术语定义

智能垃圾分类系统 基于规则与模型, 对文本描述或图片中的生活垃圾进行分类, 并提供统计分析与规则管理能力的 Web 系统;

系统级测试 基于系统需求, 对整个系统进行端到端验证的测试;

功能性测试 验证系统是否实现需求规格中规定的功能;

非功能性测试 验证性能、安全性、可靠性、可用性等非功能属性;

测试用例 为验证某一具体需求而设计的输入、操作步骤及预期结果集合。

1.5 引用标准

1. 《企业文档格式标准》, 北京长江软件有限公司;
2. 《软件测试计划报告格式标准》, 北京长江软件有限公司软件工程过程化组织。

1.6 参考资料

1. 《智能垃圾分类系统需求规格》;
2. 《智能垃圾分类系统概要设计》;
3. 《智能垃圾分类系统详细设计》;
4. 《智能垃圾分类系统测试报告》;
5. 《软件测试技术概论》, 古乐、史九林编著;
6. 《Software Testing (Second Edition)》, Paul C. Jorgensen。

1.7 版本更新信息

本文档的更新记录如表 1 所示。

表 1: 版本更新记录

修改编号	修改日期	版本	修改位置	修改内容概述
000	2025-10-20	1.0	全部	初始发布版本，形成完整系统测试计划

2 测试项目

2.1 测试项目的背景

智能垃圾分类系统面向居民用户和社区管理者，支持对生活垃圾的文本描述或图片进行自动分类，并提供规则配置与统计分析能力。本次测试的对象为系统服务端及其开放的 REST API，以及基础 Web 页面。

本次测试的目标是验证：

- 是否满足需求规格中定义的各项功能需求；
- 在预期并发量与数据规模下是否能保持可接受的性能；
- 当规则数据异常、依赖缺失或输入异常时，系统能否稳定运行并给出合理提示。

2.2 测试要点

被测特性：

- 文本分类、批量分类、图片分类的正确性与健壮性；
- 垃圾分类规则的新增、修改、删除、导入导出及并发编辑控制；
- 统计报表的正确性，包括分类结果统计、时间维度统计等；
- 相似物品推荐准确性及边界情况；
- 系统在典型并发场景下的响应时间、吞吐量和资源占用；
- 异常输入、依赖缺失、配置错误等场景下的容错与日志记录。

不被测特性：

- 第三方框架本身的内部实现逻辑（如 Flask、PyTorch 等内部实现）；

- 操作系统、数据库等基础设施的性能调优；
- UI 视觉设计细节（颜色、图标等非功能体验部分，仅做基本可用性检查）。

2.3 测试内容

测试内容即测试需求，包括功能测试和非功能测试两大类。

2.3.1 功能测试内容

功能测试按照需求规格划分模块，主要包括：

- 分类识别功能：
 - 单条文本分类：支持中文描述、常见误拼、混合描述等；
 - 批量文本分类：CSV/JSON 批量提交，正确过滤空行与异常记录；
 - 分类结果校验：返回垃圾类型、置信度、可能的推荐规则等。
- 规则管理功能：
 - 规则新增、修改、删除；
 - 规则查询与分页过滤；
 - 规则 CSV 导入导出，非法数据校验；
 - 并发编辑冲突检测与处理。
- 统计分析功能：
 - 分类结果分布统计；
 - 时间区间统计（按日/周/月）；
 - 当无数据时返回空数据结构而非错误。
- 相似物品推荐功能：
 - 基于名称、关键字的相似物品推荐；
 - 名称大小写、简繁体等场景下的匹配行为。
- 图片识别功能：
 - 上传合法格式（JPG/PNG 等）图片进行分类；
 - 超出大小限制、分辨率过大时的处理；
 - 模型依赖缺失、模型文件损坏时的错误提示与降级策略。

- 通用与安全性功能:

- 认证授权控制（若有）：未登录用户禁止访问相关统计接口；
- 错误处理与返回码规范：常见异常请求返回统一格式的错误信息；
- 日志记录：关键操作与异常场景记录必要上下文信息。

2.3.2 性能与其他非功能测试内容

非功能测试重点包括：

- 性能测试：

- 在并发用户数为 50、100、120 等不同负载下，对主要接口进行压测；
- 统计各业务事务的平均响应时间、95% 响应时间、吞吐量及错误率；
- 特别关注图片识别接口在 GPU/CPU 不同配置下的性能表现。

- 可靠性测试：

- 长时间运行（如 2 小时以上）稳定性；
- 多次重启、规则数据频繁更新情况下的可用性。

- 安全性与健壮性：

- 对非法输入、超长输入、恶意构造数据的抵御能力；
- 认证授权机制的正确性（如有）。

3 测试方法

3.1 测试环境

系统测试环境拟定如下（与测试报告保持一致或略有冗余）：

- 硬件环境：

- CPU：Intel i7 或同等级别处理器；
- 内存：16GB 及以上；
- 磁盘：SSD 512GB 及以上。

- 软件环境：

- 操作系统：Windows 11 / WSL2 Ubuntu 22.04；

- Python 运行环境: Python 3.11;
 - Web 框架: Flask 2.x, Flask-RESTful;
 - 依赖库: Torch、Transformers、Pillow 等;
 - 数据存储: 本地 CSV 文件 (`garbage_rules.csv`);
 - Web 服务器: 内置开发服务器或 Nginx+uWSGI (准生产环境)。
- 网络环境:
 - 局域网环境, 带宽不低于 100Mbps;
 - 模拟互联网访问延迟可通过测试工具配置延时。

3.2 测试工具

计划采用以下测试工具:

- Postman/HTTPPie: 接口功能测试;
- PyTest: 单元与接口自动化测试;
- Locust 或 JMeter: 性能与压力测试;
- 浏览器开发者工具: 前端调试与网络请求分析;
- 日志与监控工具: 如内置日志、Prometheus+Grafana (如有), 用于资源监控和日志分析。

3.3 测试方法

测试方法主要采用黑盒测试, 必要时结合白盒测试手段:

- 功能测试: 基于需求规格, 采用等价类划分、边界值分析、错误推测等策略设计用例, 手动与自动化结合执行;
- 接口测试: 重点验证各 REST 接口的请求方法、参数校验、返回码、返回结构及异常处理;
- 性能测试: 通过 Locust/JMeter 构造典型业务场景 (单条分类、批量分类、规则读写、图片识别等), 执行不同负载的压测;
- 回归测试: 在缺陷修复后, 针对改动范围及相关模块进行回归测试;
- 安全与健壮性测试: 对接口进行非法输入、边界值及异常场景验证。

4 测试标准

4.1 测试通过/失败标准

测试中发现的缺陷按照严重程度划分为 4 个级别，如表 2 所示。

表 2: 缺陷严重级别

严重级别	严重程度说明
1-提示 (Low)	微小的错误，不影响系统功能，只影响易用性或文案描述等
2-一般 (Medium)	问题增加测试或用户操作的复杂度，或轻微降低系统性能，系统仍可正常工作
3-严重 (High)	严重降低系统性能，或影响主要功能的正确使用，不符合需求说明
4-致命 (Very High)	系统无法正常启动，或核心功能无法使用，导致业务中断

系统测试完成的标准：

- 计划中的系统测试用例全部执行完毕；
- 无未关闭的 3 级（严重）及以上缺陷；
- 2 级缺陷已给出明确处理计划并评估为可接受风险；
- 关键性能指标满足需求规格（主要接口 95% 响应时间在约定阈值以内）。

4.2 测试挂起/恢复条件

测试挂起的条件：

- 出现致命缺陷（4 级），导致核心业务流程无法继续测试；
- 测试环境出现重大故障，短期内无法恢复；
- 被测版本不稳定，频繁崩溃影响测试结果可靠性。

测试恢复的条件：

- 引起挂起的致命缺陷已修复并经回归测试验证通过；
- 测试环境恢复正常可用；
- 相关版本变更已完成必要的冒烟测试。

4.3 系统测试交付结果

系统测试阶段的主要交付物包括：

- 《智能垃圾分类系统测试计划》；
- 《智能垃圾分类系统测试设计》（测试用例与脚本）；
- 《智能垃圾分类系统测试跟踪日志》；
- 《智能垃圾分类系统测试报告》；
- 测试脚本、测试数据及相关配置文件。

5 测试计划

5.1 角色和职责

测试相关角色和职责分配如表 3 所示。

表 3: 角色和职责表

角色	小组成员	职责
测试经理	张晓红	制定测试计划，组织测试工作；系统测试用例评审、测试报告评审；提交测试输出文档
测试工程师	吴亮	编写系统测试用例与脚本；执行系统测试用例；填写测试跟踪日志；编写测试总结报告
测试系统管理员	姜中	测试环境搭建与维护；测试工具安装与配置；测试数据准备与备份
开发代表	李明	协助缺陷定位与修复；提供技术支持与设计说明
运维代表	王强	协助模拟生产部署环境；参与性能测试与问题分析

5.2 测试设计工作任务分解和工作安排

系统测试的任务分解及时间安排如表 4 所示。

表 4: 测试时间计划

工作任务	工作量	开始日期	结束日期	备注
复习需求与设计文档，确定测试范围	1 工作日	2025-10-18	2025-10-18	测试组全员参与
讨论并制定测试计划	1 工作日	2025-10-19	2025-10-19	测试经理主导
测试用例设计与评审	4 工作日	2025-10-20	2025-10-23	含功能与性能用例
搭建与验证测试环境	1 工作日	2025-10-24	2025-10-24	含依赖安装与冒烟测试
系统功能测试并记录跟踪日志	8 工作日	2025-10-27	2025-11-05	覆盖所有功能用例
性能与可靠性测试	4 工作日	2025-11-06	2025-11-11	Locust/JMeter 压测
回归测试	3 工作日	2025-11-12	2025-11-14	重点验证缺陷修复
编写测试报告并评审	2 工作日	2025-11-15	2025-11-16	形成最终测试结论

6 审批

本测试计划经项目组成员和相关专家评审通过后生效。

测试经理签名: _____

日期: _____

项目经理签名: _____

日期: _____