垃圾分类系统

-- 老鼠爱大米

[垃圾分类系统 1](#_Toc16481)

一、团队计划........................................................................................................2

[二、 用Isseues管理项目 3](#_Toc11788)

[三、 需求分析报告 5](#_Toc1672)

[1.背景 5](#_Toc29189)

[2.功能需求 5](#_Toc5519)

[3.非功能需求 6](#_Toc11611)

[4.限制条件 6](#_Toc365)

[5.假设条件 6](#_Toc2758)

[6.结论 6](#_Toc5692)

[四、目标用户 6](#_Toc5324)

[五、功能分析和任务分解WBS 11](#_Toc12277)

[六、编码规范 13](#_Toc27873)

[七、需求报告上传github 1](#_Toc21014)4

八、系统设计.......................................................................................................14

九、审核标准.......................................................................................................15

# 一、团队计划

|  |  |
| --- | --- |
| 第 7 周 | 1. 制定每周的团队计划，具体到任务分工 |
|  | 2. 需求规格说明书，对项目的框架有整体的把握 |
|  | 1. 开展会议讨论确定项目扩展功能并评估可行性 |
|  | 4. 原型设计，队员估计任务难度并开始学习必要的技术 |
| 第8周 | 1. 编码规范完成、平台环境搭建完成、初步架构搭建 |
|  | 2. 架构设计，WBS，团队成员估计各自任务所需时间 |
|  | 3. 测试计划制定 |
| 第9周 | 1. 团队项目Alpha任务分配计划 |
|  | 2.每日会议确定队员负责项目或学习技术的进展 |
| 第10周 | 1. 连续7天的Alpha敏捷冲刺，7 篇 每日Scrum Meeting报告+代码提交 |
| 第11周 | 1. 用户反馈+测试计划改进 |
|  | 2. 团队Alpha阶段个人总结 |
| 第12周 | 1. 团队项目Alpha报告：发布说明、测试报告、展示报告、项目管理 |
| 第13周 | 1. 项目复审，测试完善功能 |
|  | 2. 团队项目Alpha报告：事后分析 |
| 第14周 | 1. 团队项目Beta任务分配计划 |
|  | 2. 连续4天的Beta敏捷冲刺，4 篇 每日Scrum Meeting报告+代码提交 |
| 第15周 | 1. 团队项目Beta报告：发布说明、测试报告、展示 |
|  | 2. 团队Beta阶段个人总结 |
| 第16周 | 1. 项目复审，确定扩展功能的可行性 |
|  | 2. 团队项目Beta阶段事后分析， 宣布每人的贡献分 |
|  | 3. 团队整个阶段总结，分析用户数据，整理文档，保证以后的团队能接手。 |

# 用Isseues管理项目

# 

# 需求分析报告

## 1.背景

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. 功能需求

**（1）数据收集：**收集大量垃圾图像数据，并进行标记分类，建立训练所需的数据集。

**（2）特征提取：**使用图像处理技术提取垃圾图像的特征，如颜色、形状、纹理等，以便于机器学习模型识别。

**（3）模型选择与训练：**选择合适的机器学习模型，如卷积神经网络（CNN），进行训练以实现图像分类任务。

**（4）模型优化与评估：**通过交叉验证、超参数调优等方法优化模型性能，并使用测试集评估模型准确率和泛化能力。

**（5）系统集成：**将训练好的模型集成到垃圾分类系统中，实现实时监测和自动分类功能。

**（6）用户界面设计：**设计用户友好的界面，使用户可以轻松地与系统交互和操作。

## 3. 非功能需求

（1）**性能：**系统需要能够处理大量数据和复杂查询，并且在短时间内呈现结果。

**（2）可靠性：**系统应该是可靠的，并具备一定的容错机制，以避免识别错误和系统崩溃等问题。

**（3）易用性：**系统应该具有简单易用、识别精准、不分人群、全天候多环境工作等特点。

## 4. 限制条件

垃圾分类目前还停留在人工识别分类上，智能分类的普及度不高。

系统应该能够适应不同的屏幕大小和分辨率，以便用户在不同的设备上使用。

## 5. 假设条件

（1）收集的大量垃圾图像数据已经标记分类，并利用图像处理技术提取垃圾图像的特征，选择适当的机器学习模型进行训练，优化模型性能并评估其准确率，最终将训练好的模型集成到垃圾分类系统中，实现实时监测和自动分类功能。

（2）设计用户友好的界面，方便用户（不具备专业知识的用户也可进行操作）交互和操作

## 6. 结论

基于以上需求分析，我们的项目旨在开发一套基于机器学习的智能垃圾分类系统，以提高垃圾分类的准确率并减少误分类现象。同时，该系统将满足性能、可靠性和易用性的需求。

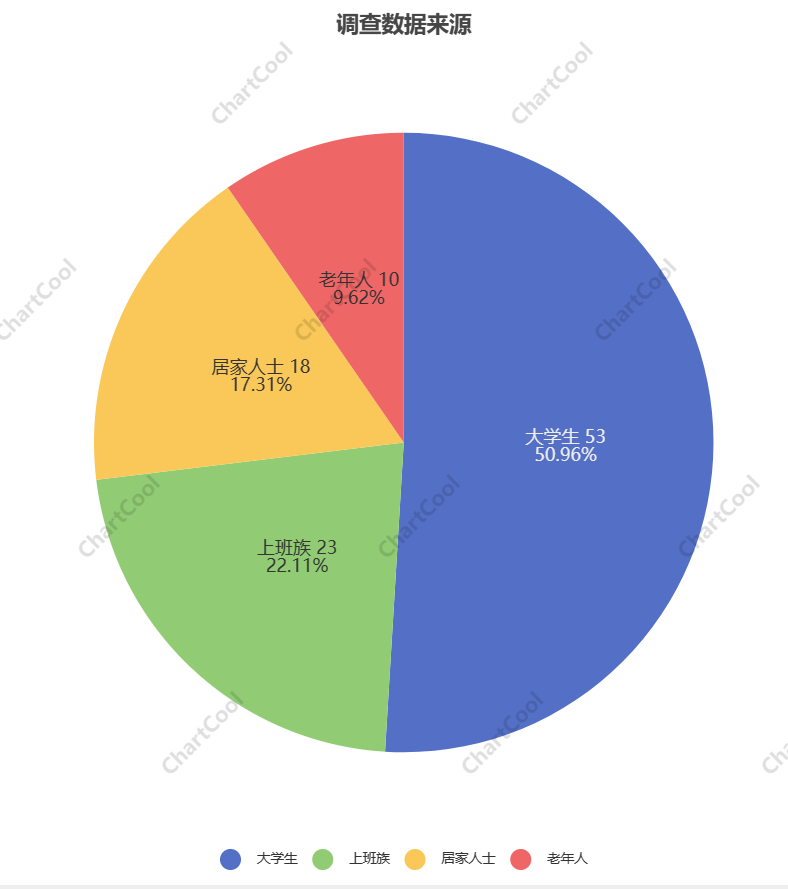
# 四、目标用户

垃圾分类智能监测的目标市场主要包括城市管理部门、垃圾处理企业和居民社区等。 垃圾分类智能监测系统的应用将帮助城市管理部门更加高效地监测和管理垃圾分类工作，提高垃圾分类的准确率和效率，从而解决垃圾问题，促进城市的可持续发展。 垃圾处理企业也是垃圾分类智能监测的重要用户。

**问卷调查内容以及结果分析（统一以饼状图来呈现）；**

在开始项目之前，我们需要先进行一些基础调查。

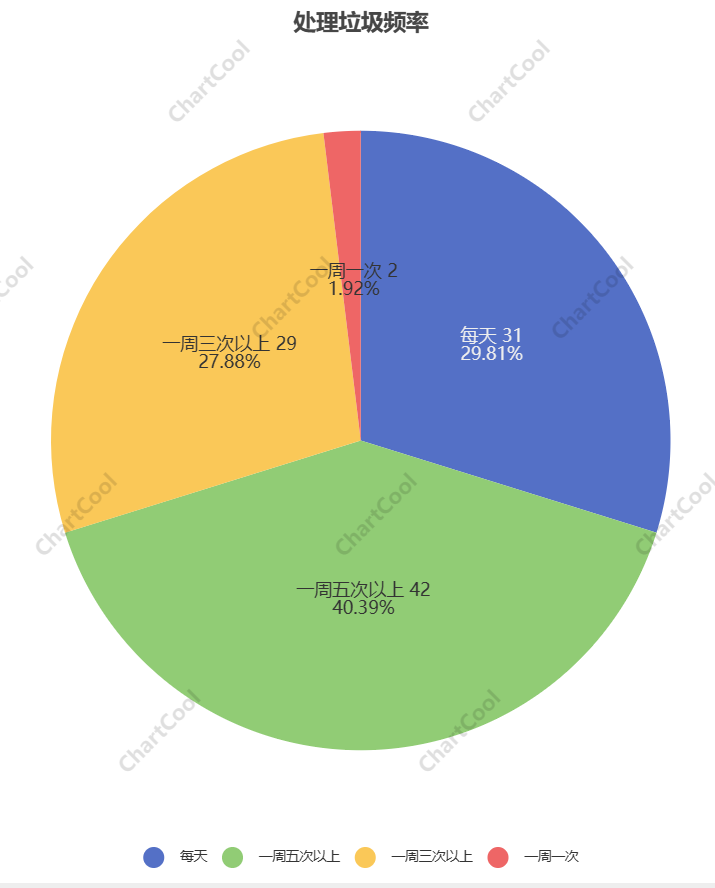
首先我们需要先确定我们的调查人群。结果如下



可以看到，大学生占了大多数群体，上班族和居家人士差不多，老年人较少。这和我们的目标群体略有偏差，我们会在之后加大对于社会人士的调查，社会人士也是使用我们这套系统的主力存在。

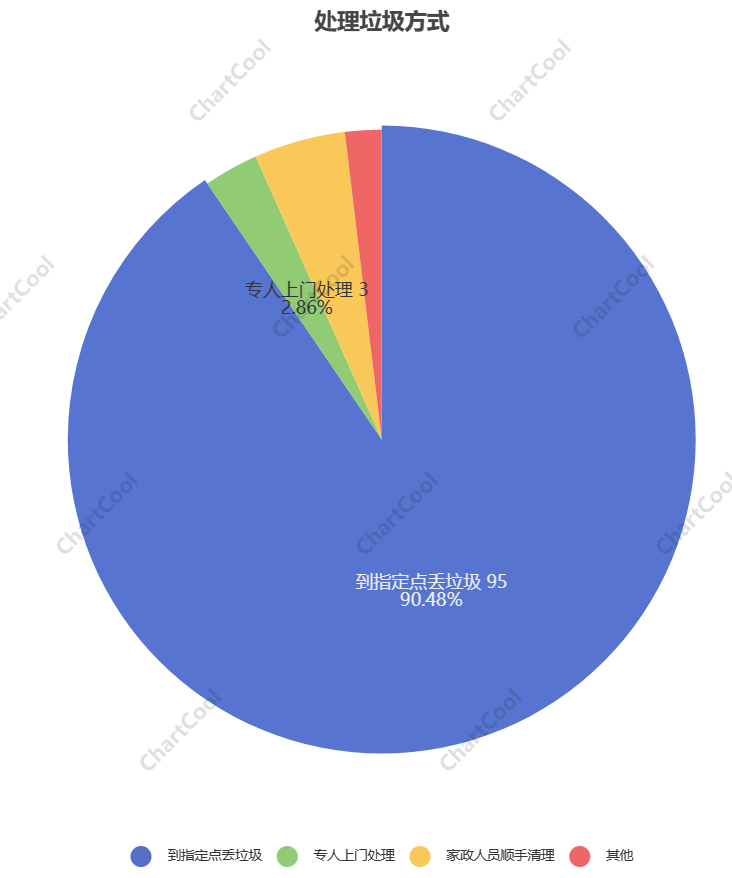
在了解了调查人群后，我们主要的调查项目有五个：处理垃圾的频率、处理垃圾的方式、是否了解垃圾处理和分类的知识、是否觉得垃圾处理麻烦、生活中是否有接触过智能垃圾分类产品。

首先是处理垃圾的频率，如下图



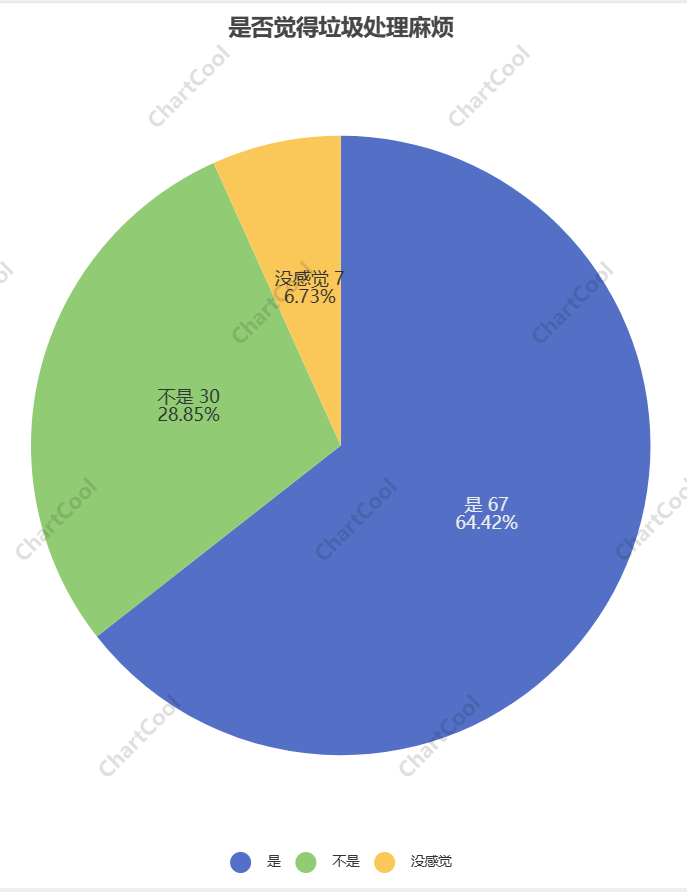
可以看到，一周处理五次以上的人最多，其次是每天处理和一周处理三次以上，这三者的数据区别不大，最少的是一周一次的，占绝对少数，说明大部分人对处理垃圾有比较积极的需求。

其次是处理垃圾的方式，如下图



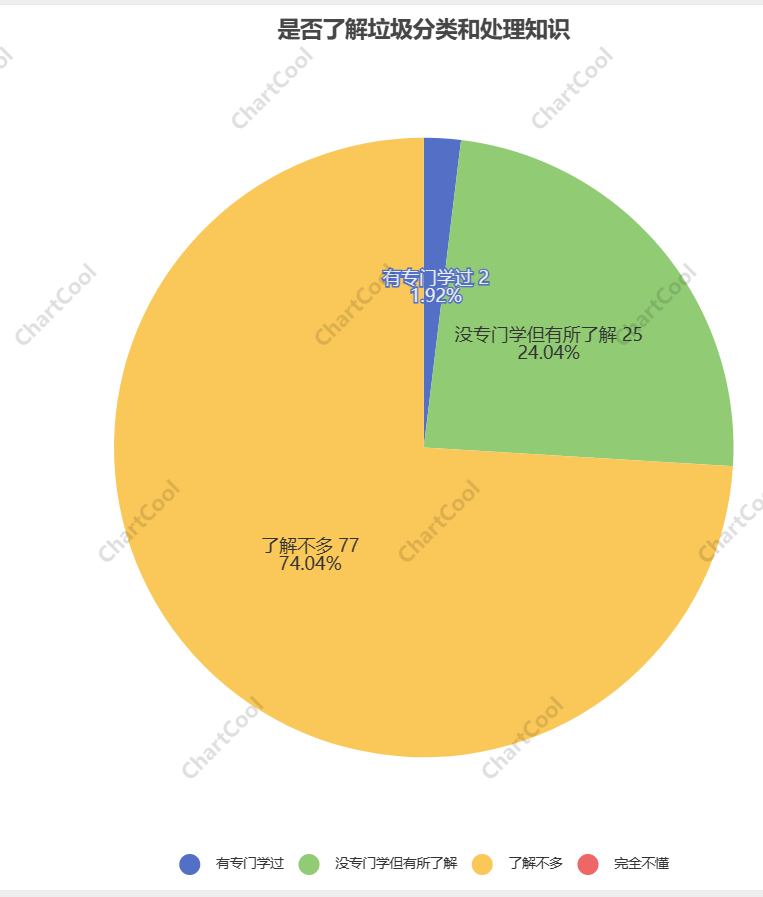
可以看到，绝大部分人是到指定点丢垃圾，占比高达90.48%，极少部分人选择了指定人员上门处理、家政人员处理或其他处理方式，说明大部分人都有自己处理垃圾的需求。

然后是是否觉得处理垃圾麻烦，如下图



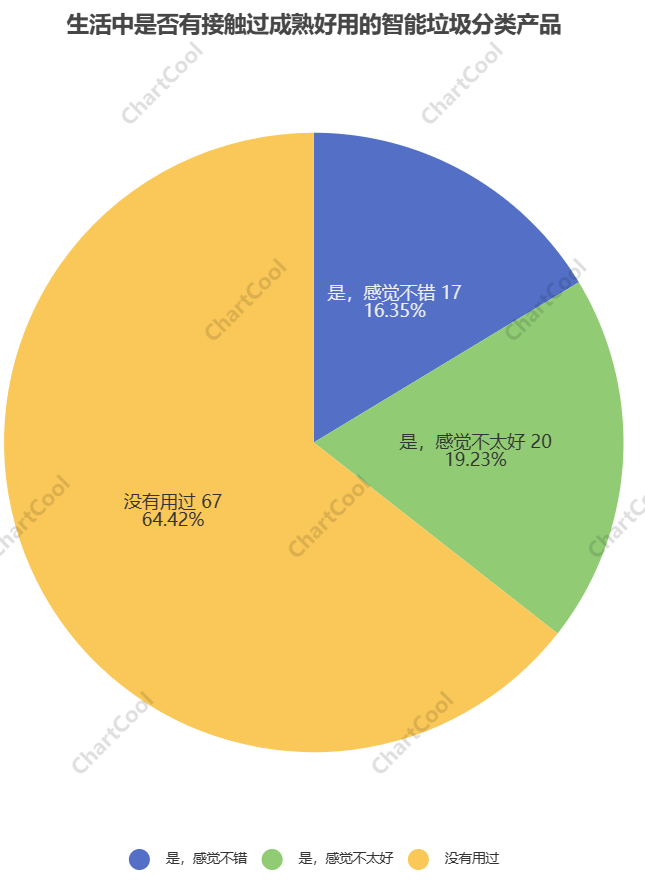
可以看到，大部分人觉得处理垃圾麻烦，占64.42%，还有28.85%的人觉得不麻烦，还有少部分人没有感觉。说明大部分人对现有的垃圾处理方式不太满意。

然后是对垃圾分类处理知识的了解程度，如下图



可以看到，绝大部分人对此了解并不多，还有一部分人有所了解，但没有专门学过，专门学过的占极少数，而没有人完全不懂。这说明大家其实都有一定的垃圾分类意识，但可能受限于多种原因，大部分人并没有专业、深入的了解。

最后是对智能垃圾分类产品的使用情况，如下图



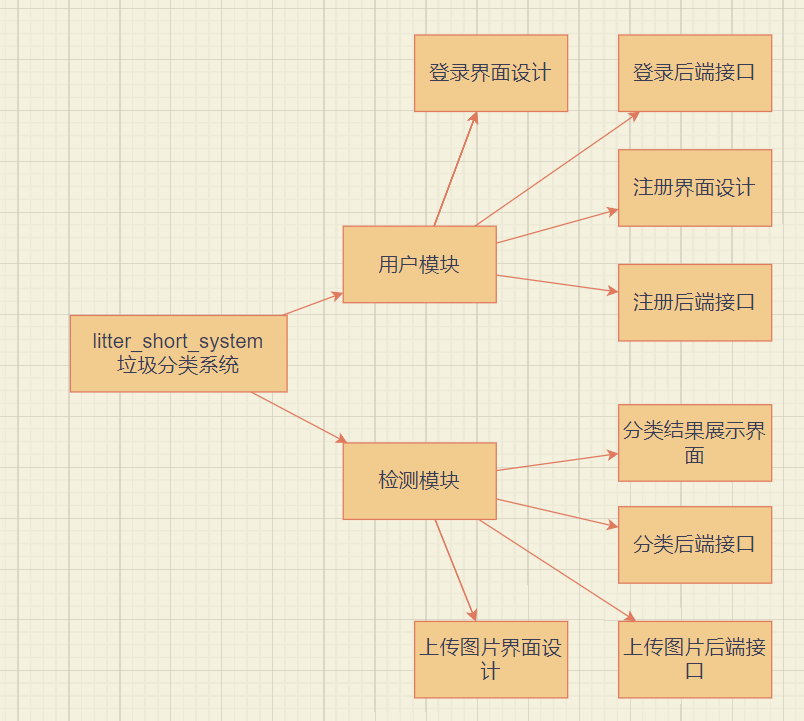
其中大部分人没有使用过智能垃圾分类产品，占64.42%，还有部分人使用过，感觉不好，占19.23%，还有小部分人使用过，感觉不错，占16.35%，这两部分人人数接近。可以看出，大部分人都没有接触过比较成熟的智能垃圾分类产品，这块的市场情况或者说推广使用情况还需要加强。

综上分析，我们可以得到这样的结论：大部分人都有比较积极的垃圾处理需求，同时更倾向于自己处理垃圾，但很多人都觉得处理垃圾麻烦，并且并不是很了解相关垃圾分类处理的知识，同时市面上智能垃圾分类产品在使用和推广方面存在一定问题。

这说明我们的产品确实是被大部分人群所需要的，我们需要去迎合大部分人的垃圾处理需求，让个人处理垃圾不再成为一件麻烦事，这能取代用户去自主学习垃圾分类知识的精力，我们在做的就是这样一个智能垃圾分类产品。

# 功能分析和任务分解WBS

功能：



象限分析

|  |  |  |
| --- | --- | --- |
|  | 外围功能 | 杀手功能 |
| 必要需求 | 第二象限：  上传图片界面设计，分类结果展示界面 | 第一象限：  上传图片后端接口，分类后端接口 |
| 辅助需求 | 第三象限：  用户模块 | 第四象限：  接入树莓派实时监测 |

所需时间

|  |  |  |  |
| --- | --- | --- | --- |
| 功能 | 子功能 | 预计完成时间/h | 拟定负责成员 |
| 用户模块 | 登录界面设计 | 10 | 郑博文 |
| 登录后端接口 | 5 | 陈曦 |
| 注册界面设计 | 5 | 陈曦 |
| 注册后端接口 | 10 | 陈曦 |
| 检测模块 | 上传图片界面设计 | 3 | 郑博文 |
| 上传图片后端接口 | 20 | 李睿 |
| 分类结果展示界面 | 3 | 李睿 |
| 分类后端接口 | 10 | 陈曦 |
| 测试模块 | 接口测试 | 5 | 范兴宇 |
| 界面测试 | 5 | 范兴宇 |
| 前期调查 | 需求分析 | 10 | 杨雨佳 |
| 数据库表设计 | 2 | 杨雨佳 |
| 代码结构 | 2 | 陈曦 |

# 六、编码规范

编码规范按照阿里巴巴代码规范来进行（详情请查看[alibaba/f2e-spec: Alibaba Front-end Coding Guidelines and Relevant Tools (github.com)](https://github.com/alibaba/f2e-spec)）

**(一) 命名规范**

1.1.1 项目命名

全部采用小写方式，以中线分隔。项目名称为:litter\_sort\_system

1.1.2 目录命名

全部采用小写方式， 以中划线分隔，有复数结构时，要采用复数命名法， 缩写不用复数。

1.1.3 JS、CSS、SCSS、HTML、PNG 文件命名

全部采用小写方式， 以中划线分隔。

1.1.4 命名严谨性

代码中的命名严禁使用拼音与英文混合的方式，更不允许直接使用中文的方式。 说明：正确的 英文拼写和语法可以让阅读者易于理解，避免歧义。注意，即使纯拼音命名方式也要避免采用

杜绝完全不规范的缩写，避免望文不知义：

1.1.5 使用微服务思想来管理项目

以sort加下划线加微服务模块

**(二) HTML 规范 （Vue Template 同样适用）**

1.2.1 HTML 类型

推荐使用 HTML5 的文档类型申明：

1.2.2 缩进

缩进使用 2 个空格（一个 tab）；

嵌套的节点应该缩进。

1.2.3 分块注释

在每一个块状元素，列表元素和表格元素后，加上一对 HTML 注释。

1.2.4 语义化标签

HTML5 中新增很多语义化标签，所以优先使用语义化标签，避免一个页面都是 div 或者 p 标 签。

**(三) CSS 规范**

1.3.1 命名

1.3.2 选择器

1) css 选择器中避免使用标签名

2) 使用直接子选择器

1.3.3 尽量使用缩写属性

1.3.4 每个选择器及属性独占一行

1.3.5 省略 0 后面的单位

1.3.6 避免使用 ID 选择器及全局标签选择器防止污染全局样式

**(四) Javascript 规范**

1.5.1 命名

1) 采用小写驼峰命名 lowerCamelCase，代码中的命名均不能以下划线， 也不能以下划线或美元符号结束

2) 方法名、参数名、成员变量、局部变量都统一使用 lowerCamelCase 风 格，必须遵从驼峰形式

3) 常量命名全部大写，单词间用下划线隔开，力求语义表达完整清楚， 不要嫌名字长

1.5.2 代码格式

1) 使用 2 个空格进行缩进

2) 不同逻辑、不同语义、不同业务的代码之间插入一个空行分隔开来以 提升可读性

说明：任何情形，没有必要插入多个空行进行隔开。

1.5.3 字符串

统一使用单引号(‘)，不使用双引号(“)。这在创建 HTML 字符串非常有好处：

1.5.4 对象声明

1) 使用字面值创建对象

2) 使用字面量来代替对象构造器

1.5.5 使用 ES6+

1.5.6 括号

1.5.7 undefined 判断

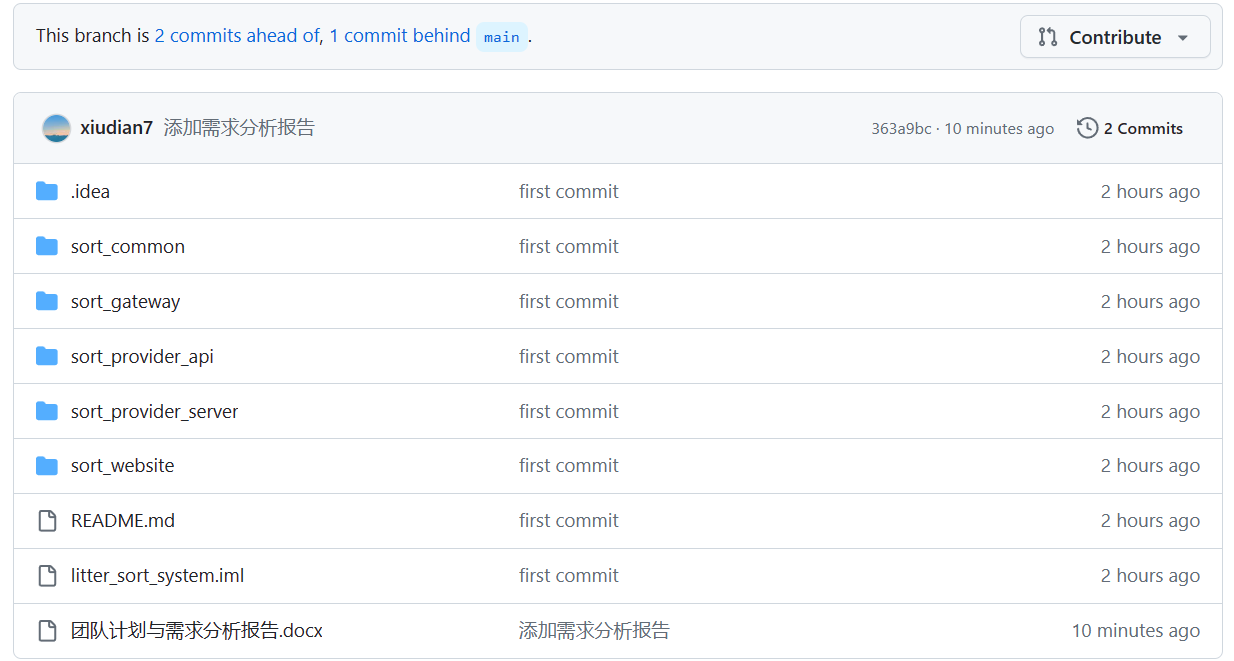
永远不要直接使用 undefined 进行变量判断；使用 typeof 和字符串’undefined’对变量进行判断。

1.5.8 条件判断和循环最多三层

1.5.9 this 的转换命名

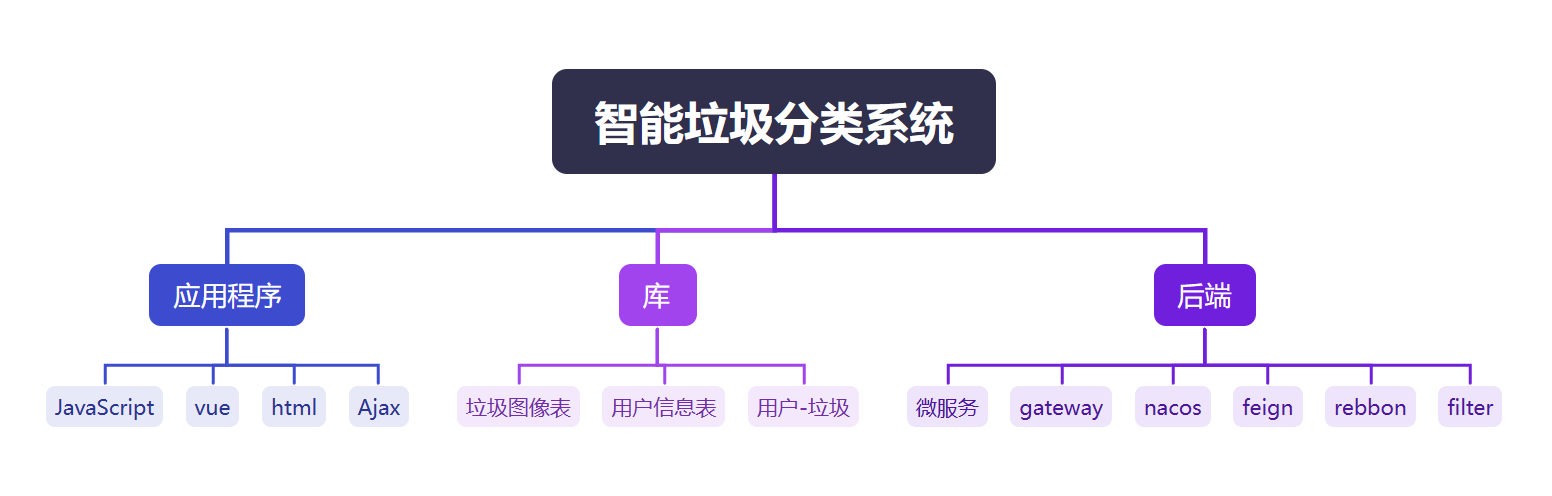
1.5.10 慎用 console.log

# 七、需求报告上传github



**八、系统设计**

1. **体系结构图**

****

1. **数据库设计**

**(1)垃圾图像表：包含垃圾图像的存储路径、标签信息等字段。**

**图像ID (Image ID)**

**图像路径 (Image Path)**

**垃圾类型 (Garbage Type)**

**(2)用户信息表：存储用户的登录信息以及个人信息。**

**用户ID (User ID)**

**用户名 (Username)**

**性别(Gender)**

**密码 (Password)**

**(3)用户-垃圾对应表：**

**用户ID (User ID)**

**图像ID (Image ID)**

**用户贡献时间 (Contribution Time)**

**九、审核标准**

|  |  |
| --- | --- |
| 成员 | 绩效考核意见(优、良、中等) |
| 陈曦 | 1.后端数据库设计  2.模块架构  3.系统安全性  4.代码可读性 |
| 李睿 | 1.数据集质量  2.模型准确率  3.模型性能 |
| 郑博文 | 1.界面ui  2.交互逻辑  3.代码可读性  4.安全性 |
| 杨雨佳 | 1.产品需求分析到位度  2.统筹规划全局开发进度 |
| 范兴宇 | 1.测试系统正常运行  2.检测系统的安全性 |