**图片分类系统测试方案**

**1. 测试概述**

**项目名称**：图片分类系统

**测试目标**：

（1）验证图片导入功能的正确性。

（2）验证图片管理功能的完整性和稳定性。

（3）测试图片分类功能的准确性，包括手动分类和AI自动分类。

（4）确保数据库操作（增删改查）的正确性和性能。

（5）前端与后端的集成测试，确保数据交互无误。

**测试范围**：前端（Vue.js）、后端（Node.js）、数据库（OpenGauss）、AI分类模块。

**2. 测试环境**

**硬件环境**：根据项目需求配置，推荐多核CPU、充足内存及高速存储。

**软件环境**：

（1）Node.js：根据项目依赖的Node.js版本安装。

（2）Vue.js：通过Vue CLI或类似工具搭建。

（3）数据库：OpenGauss，配置好连接和必要的权限。

（4）测试工具：

Jest（单元测试）、Cypress/Selenium（前端集成测试）、Postman/Newman（API测试）。

**3. 测试工具与方法**

**单元测试：**

后端：使用Jest测试Node.js的API、数据库交互。

前端：Vue组件的单元测试可以使用Jest结合Vue Test Utils。

**集成测试：**

前端集成：使用Cypress或Selenium进行端到端测试，模拟用户操作，验证前端与后端的交互。

API测试：使用Postman或Newman测试所有API接口，包括请求参数、响应状态码、响应内容等。

AI分类测试：

根据AI模型的具体实现，使用相应的测试框架或工具，验证模型对图片分类的准确性。

**4. 测试用例**

**4.1 单元测试**

**后端（Node.js）**

（1）图片上传API

测试用例1：正常上传图片x10，验证响应状态码和数据库记录。

测试用例2：上传非法格式文件x10，验证返回错误消息。

测试用例3：上传空文件x10，验证返回错误消息。

（2）图片管理API

测试用例1：获取图片列表，验证返回数据结构和数量。

测试用例2：删除指定图片，验证数据库中图片被删除且前端列表更新。

（3）图片分类API

测试用例1：图片添加分类，验证数据库记录。

测试用例2：删除图片，验证数据库记录更新。

**前端（Vue.js）**

（1）图片上传组件

测试用例1：用户选择文件并点击上传，验证请求发送及UI反馈。

（2）图片列表组件

测试用例1：加载图片列表，验证列表项与数据库数据一致。

**4.2 集成测试**

前端到后端集成

（1）测试用例1：用户上传图片，验证图片能在前端列表中显示且数据库有记录。

（2）测试用例2：用户删除图片，验证前端列表更新且数据库记录被删除。

API测试

使用Postman测试所有关键API，确保接口行为符合预期。

**4.3 AI分类测试**

AI模型测试

（1）测试用例1：对一系列已知分类的图片进行分类，验证分类准确率。

（2）测试用例2：对边界情况（模糊、低质量）图片进行分类，验证模型稳定性。

**4.4 性能测试**

（1）测试用例1：并发上传性能测试

测试目标：测试系统同时处理100个用户上传图片的能力。

测试步骤：

使用测试工具模拟100个用户同时上传图片。

记录上传时间、系统响应时间、CPU使用率、内存使用率等关键指标。

观察系统是否出现错误或异常。

预期结果：所有图片均成功上传，系统响应时间不超过预设阈值，资源使用率在合理范围内。

（2）测试用例2：批量上传性能测试

测试目标：测试系统同时处理100个上传图片的能力。

测试步骤：

同时上传100图片。

记录上传时间、系统响应时间、CPU使用率、内存使用率等关键指标。

观察系统是否出现错误或异常。

预期结果：所有图片均成功上传，系统响应时间不超过预设阈值，资源使用率在合理范围内。