

AI 云盘系统开发计划

1.项目背景与目标

本项目旨在为某企业设计一款支持视频、图像存储，具备数据上传、下载、共享等基础功能。企业期望实现以下智能功能：

- (1) 实时生成图片缩略图
- (2) 视频可选择性转码
- (3) 图片筛选与内容审核
- (4) 敏感信息检查与筛选
- (5) 实时图片分类（基于图像内容）
- (6) 图片文字识别（OCR）
- (7) 语音识别（ASR）

本项目通过快速原型法，快速验证需求并迭代完善系统设计与功能。

2.项目组织团队

团队依据职能模块与核心技术栈，设立“六边形”协作单元，覆盖需求、设计、开发、算法、质量到运维的全链路职责。各角色对各自专业成果负责，共同承担整体目标，确保产品功能、性能与稳定性达到预期。

表 1 项目核心职能分工表

角色	职责说明
项目负责人	项目规划、需求分解、技术架构、关键模块评审
前端工程师	WebUI、文件上传界面、预览界面、原型实现
后端工程师	API、存储模块、视频转码、图片处理接口
AI 工程师	OCR、ASR、内容审核、图像分类模型部署
测试工程师	功能测试、性能测试、安全性验证
运维工程师	服务器部署、微服务容器管理

3. 开发计划

(一) 阶段一：需求分析与团队启动

任务 1：项目启动会议

内容：介绍项目背景、目标、团队成员及分工，明确各阶段交付物要求。

任务 2：需求调研与分析

内容：需求分析师与企业方深入沟通，梳理云盘基础功能（上传、下载、共

享）及高级功能（图像缩略图、视频编码转换、内容审核、图片分类、OCR、语音识别）的具体需求细节。

任务 3：团队技术栈确认

内容：确定原型开发、UML 建模、微服务/无服务开发所使用的技术栈，如原型开发使用 Figma/Axure，后端开发暂用 Python+Flask。

（二）阶段二：原型设计与开发

任务 1：UI 原型设计

内容：前端工程师根据需求，设计云盘系统的界面原型，包括登录界面、文件管理界面、功能操作界面（上传、下载、共享、图片/视频处理功能入口）等，输出可交互的界面原型。

任务 2：原型功能开发

内容：后端工程师基于界面原型，实现基础功能的前端交互和后端简单逻辑，如模拟文件上传下载、共享链接生成等，对于高级功能，先实现功能入口和模拟效果，为后续细化做准备。

任务 3：原型评审

内容：组织团队内部评审，邀请企业方参与，对原型的界面、功能流程进行评估，收集改进意见。

（三）阶段三：需求细化与 UML 建模

任务 1：需求细化

内容：需求分析师结合原型评审结果，细化需求，明确每个功能的输入、输出、业务规则等。

任务 2：UML 图设计

内容：UML 建模师根据细化的需求，设计用例图（描述用户与系统的交互）、类图（描述系统类结构及关系）、序列图（描述功能实现的时序流程）。

（四）阶段四：设计模式重构

任务 1：设计模式选取与应用

内容：设计模式专家分析系统模块，选取 2-4 种合适的设计模式（如工厂模式、策略模式、观察者模式、适配器模式等），对系统设计进行重构，绘制重构后的 UML 类图。

（五）阶段五：微服务/无服务开发规划

任务 1：微服务/无服务架构设计

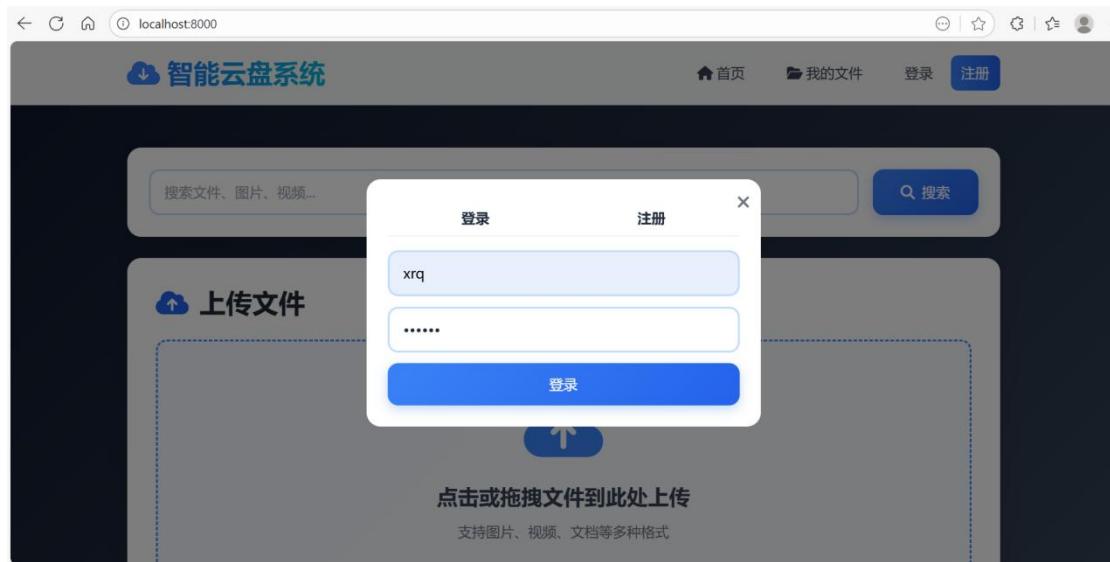
内容：根据系统功能，划分微服务模块（如文件存储服务、图像处理服务、视频处理服务、内容审核服务、AI 识别服务等）或设计无服务函数（如图片缩略图生成函数、OCR 识别函数等），明确各服务/函数的职责、接口及交互方式。

任务 2：开发计划收尾与总结

内容：整理各阶段文档，进行项目开发计划的总结，为后续正式开发提供指导。

4. 系统原型设计

4.1 登录与注册界面（二合一切换式）



布局结构（宽 400px 白色卡片，居中显示）

(1) 顶部切换栏

左侧“登录”按钮（黑色粗体字，激活状态）。

右侧“注册”按钮（灰色字，未激活）。

点击切换时，表单内容同步替换（登录/注册面板切换）。

(2) 登录表单区域（间距 16px）

账号输入框：左侧用户图标，提示文字“用户名”，边框聚焦时变蓝色。

密码输入框：左侧锁图标，提示文字“密码”。

(3) 底部操作区

登录按钮：蓝色渐变背景（#3b82f6 到 #2563eb），白色字，圆角-xl，点击后提交。

注册切换后，表单替换为“用户名→密码”，底部按钮变为“注册并登录”。

交互逻辑

输入为空时点击登录，显示错误提示。

用户名/密码不符合规则时（用户名至少 3 字符，密码至少 6 字符）显示对应提示。

4.2 云盘主界面（文件管理）

布局结构（响应式设计）

（1）顶部导航栏

左侧：Logo（“智能云盘系统”文字+云朵图标）。

中间：导航链接（首页、我的文件）。

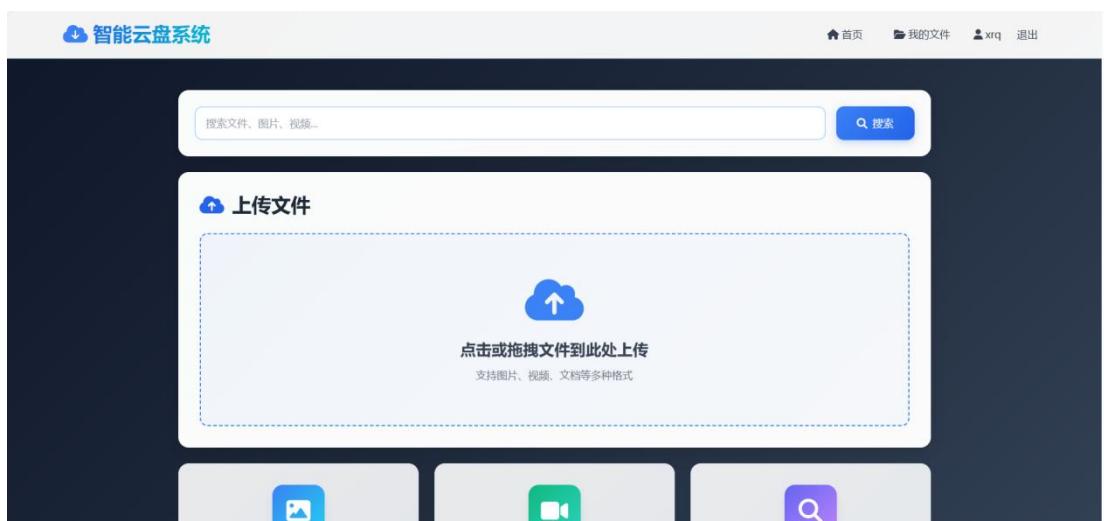
右侧：登录/注册按钮（未登录）或用户名+退出按钮（已登录）。

（2）搜索与筛选区

搜索框：提示“搜索文件名...”，支持搜索。

类型筛选：下拉选择（全部类型/图片/视频/其他）。

分类筛选：下拉选择（全部分类/动物/植物/风景等）。



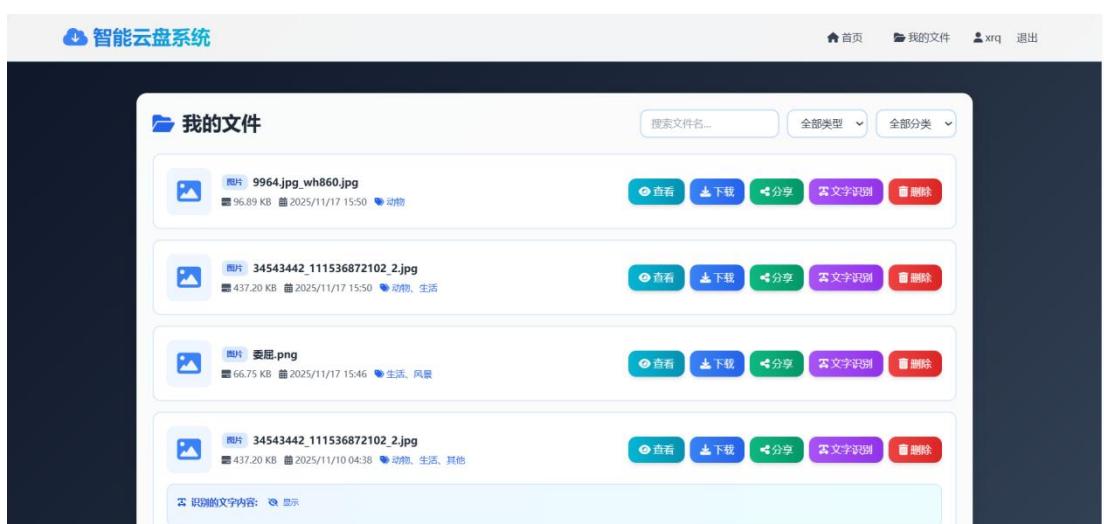
（3）文件展示区（响应式网格/列表）

图片文件卡片：图片图标+文件名+大小+上传时间+标签。

视频文件卡片：视频图标+文件名+大小+上传时间+标签。

操作按钮：查看/播放、下载、分享、OCR/语音识别、删除。

右侧无固定详情面板，操作通过弹窗实现。



4.3 文件上传界面（区域式）

布局结构（卡片式上传区域，非弹窗）

(1) 上传区域（虚线边框，大尺寸中央区域）

中间：云朵上传图标（蓝色）+提示文字“点击或拖拽文件到此处上传”。

下方：“支持图片、视频、文档等多种格式”。

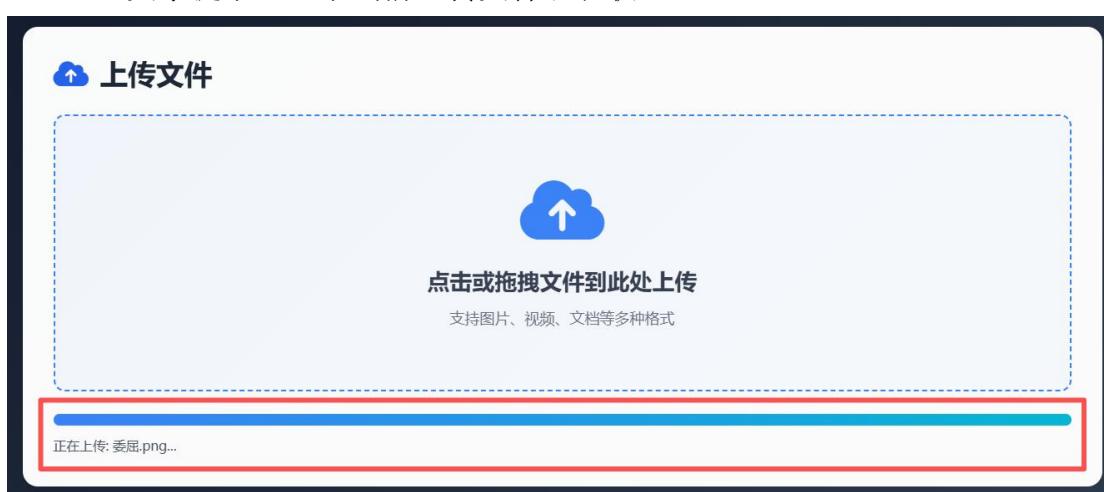
拖拽文件时，边框变蓝色，背景色加深。



(2) 上传进度区

进度条：蓝色渐变背景，显示上传进度。

文字提示：显示当前上传文件名和状态。



(3) 交互逻辑

选择文件后自动开始上传，实时更新进度条。

上传完成后显示成功提示，刷新最近文件列表。

支持多文件上传。

4.4 视频播放与转码状态界面

布局结构（弹窗式播放）

(1) 顶部操作栏

左侧：视频文件名。

右侧：关闭按钮。

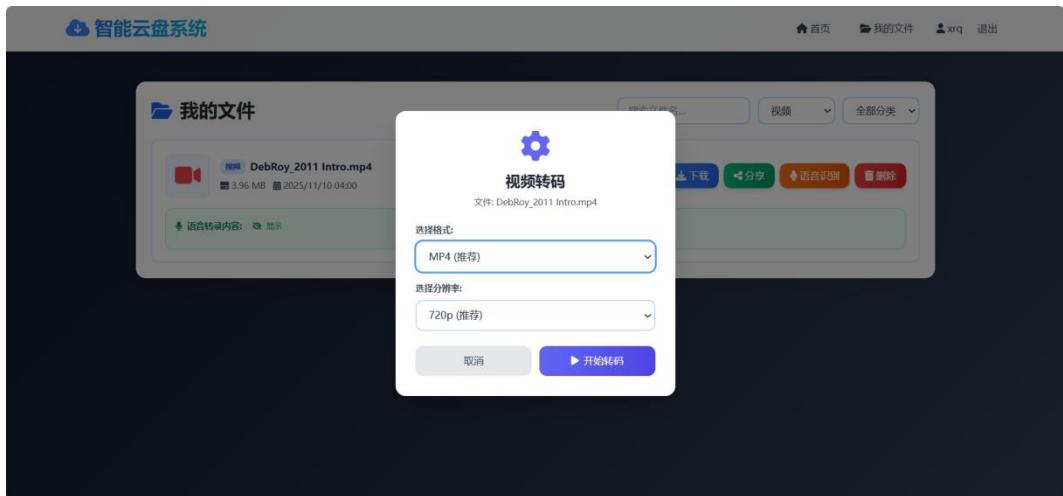
(2) 视频播放区

中央视频播放器（支持播放控制）。

(3) 转码状态区

转码中：显示进度条和提示文字“请稍候，转码可能需要较长时间...”。

转码完成：显示转码成功状态。



(4) 交互逻辑

转码过程中实时更新进度。

转码完成后可正常播放视频。

支持通过 ESC 键或点击背景关闭播放窗口。

4.5 智能分类与审核界面

布局结构（整合在文件管理界面中）

(1) 文件列表展示

每个文件项显示 AI 分类标签（如“动物”“植物”等）。

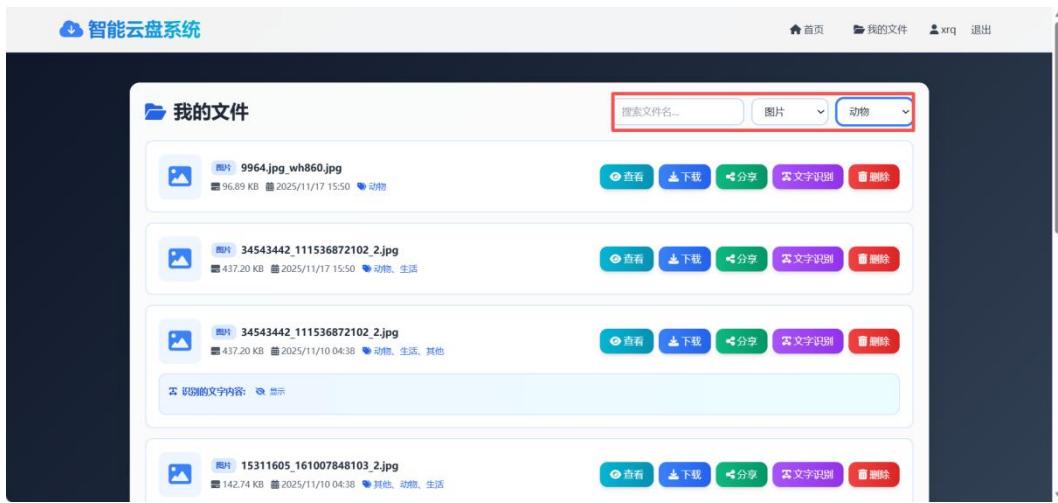
敏感内容标记：若有敏感内容显示红色提示。

(2) 操作功能

图片文件支持 OCR 文字识别，识别结果可查看。

视频文件支持语音识别，转录内容可查看。

所有文件显示 AI 自动分类标签。



4.6 额外补充：文件夹管理界面

布局结构（与文件管理界面整合）

(1) 文件展示区支持列表视图

表头：名称、大小、修改时间、AI 标签、操作。

(2) 每行文件项：

名称列：文件类型图标+文件名。

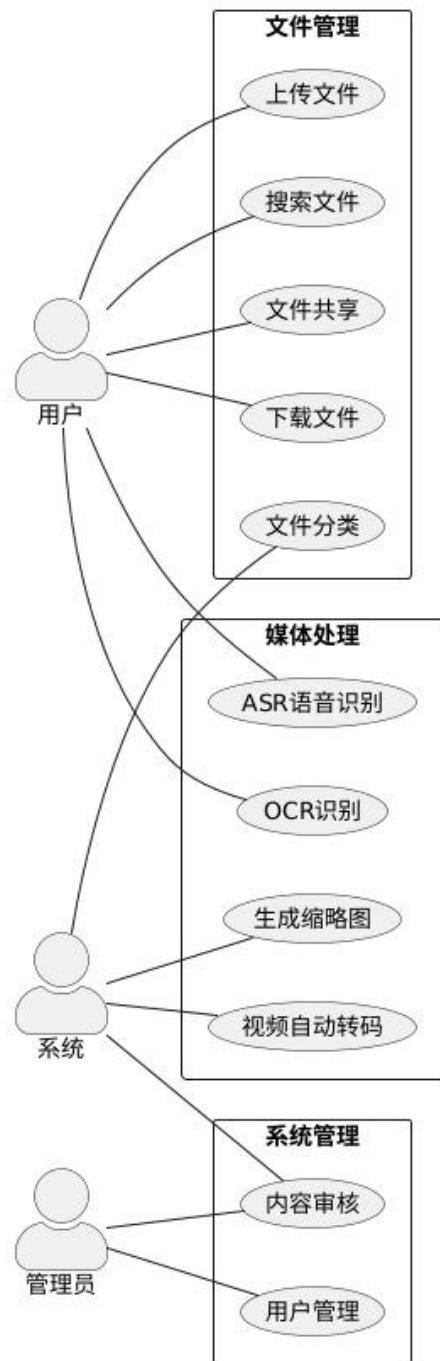
大小列：显示文件大小（文件夹显示空）。

AI 标签列：显示 1-3 个核心标签。

操作列：下载、分享、删除等按钮。

5. 需求细化与设计

5.1 用例图



上传/下载文件：支持多文件、大文件及拖拽上传，提供断点续传功能。

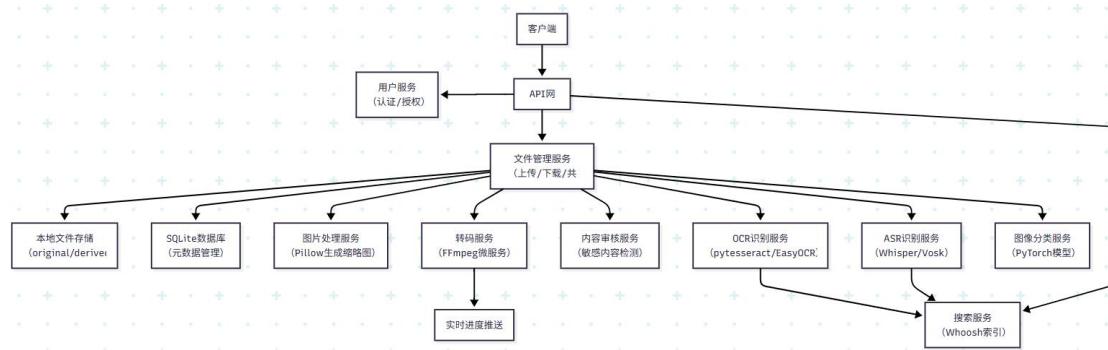
视频自动转码：基于 FFmpeg 将视频转为 MP4 格式（H.264/H.265），生成多码率版本。

内容审核：通过启发式算法检测裸露、暴力等敏感内容，违规文件拒绝上传。

AI 识别：OCR 支持图片文字提取（中英文），ASR 支持视频语音转文字。

文件分类: 可选深度学习模型自动分类图片内容(需安装 PyTorch 相关依赖)。
文件共享: 生成带有效期的加密短链, 支持直接预览或下载。

5.2 系统总体架构图



文件管理服务: 核心服务, 处理文件 CRUD、元数据管理及业务流程调度。
转码服务: 基于 ffmpeg-python 实现视频转码, 支持后台异步处理和进度推送。

图片处理服务: 通过 Pillow 生成多尺寸缩略图, 支持 GPU 加速 (可选)。

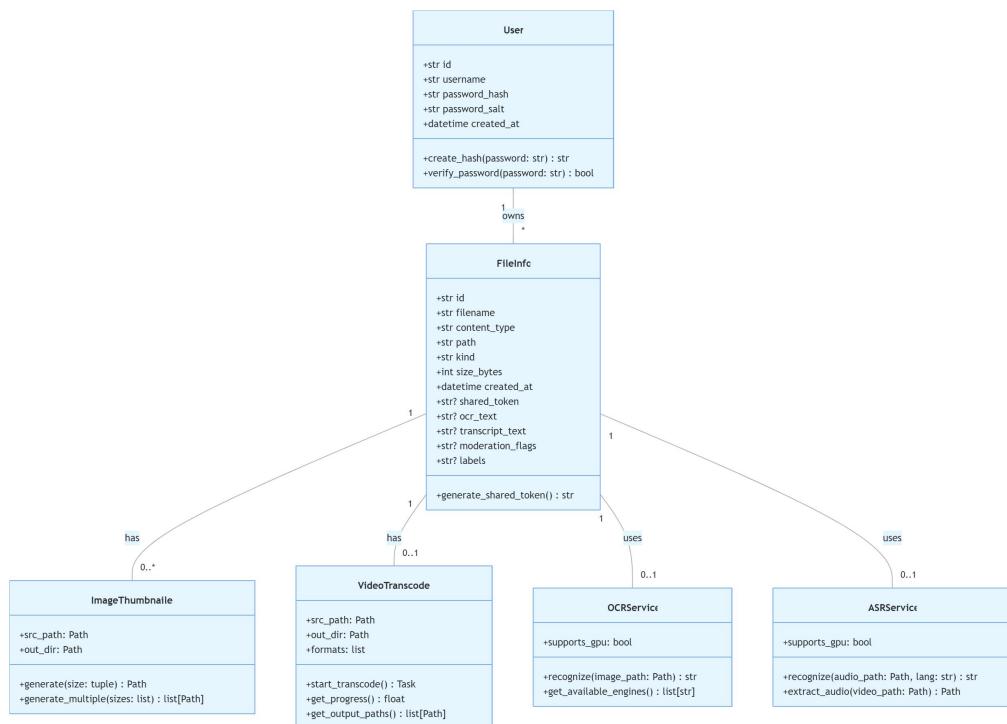
内容审核服务: 上传时同步检测敏感内容, 包括图像分析和文本识别审核。

OCR/ASR 识别服务: 提供文字提取功能, 结果同步至搜索索引。

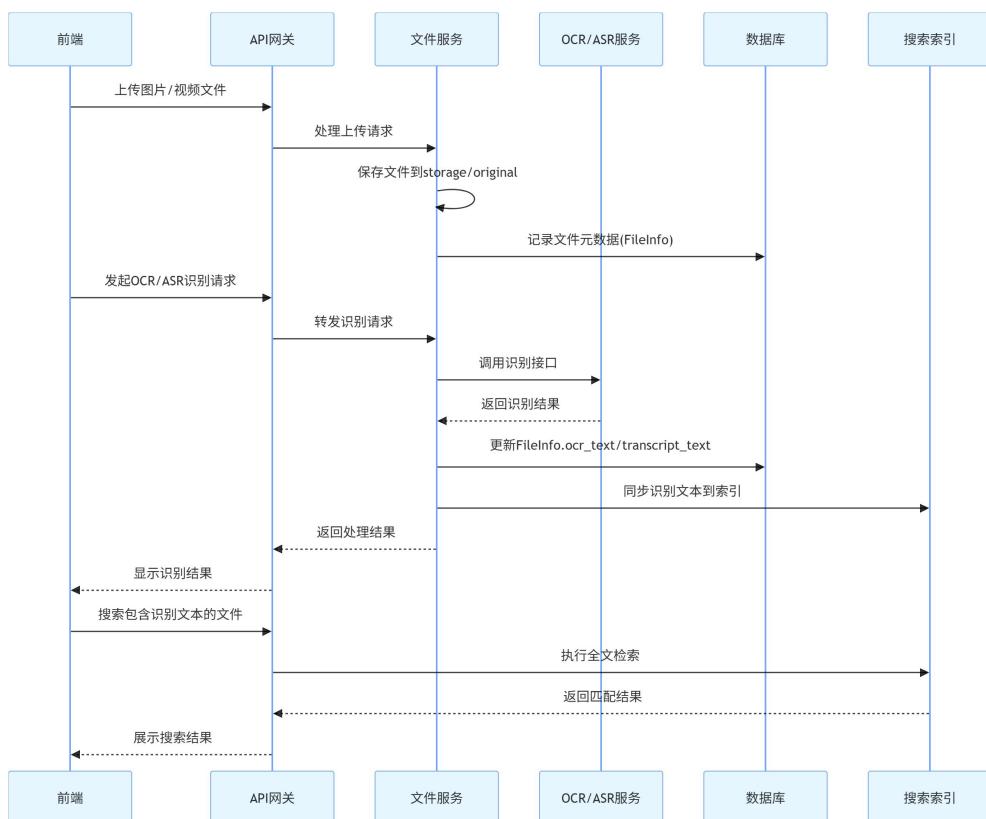
网关服务: 统一 API 入口, 处理路由、认证和请求转发。

用户服务: 管理用户账号、会话及权限控制。

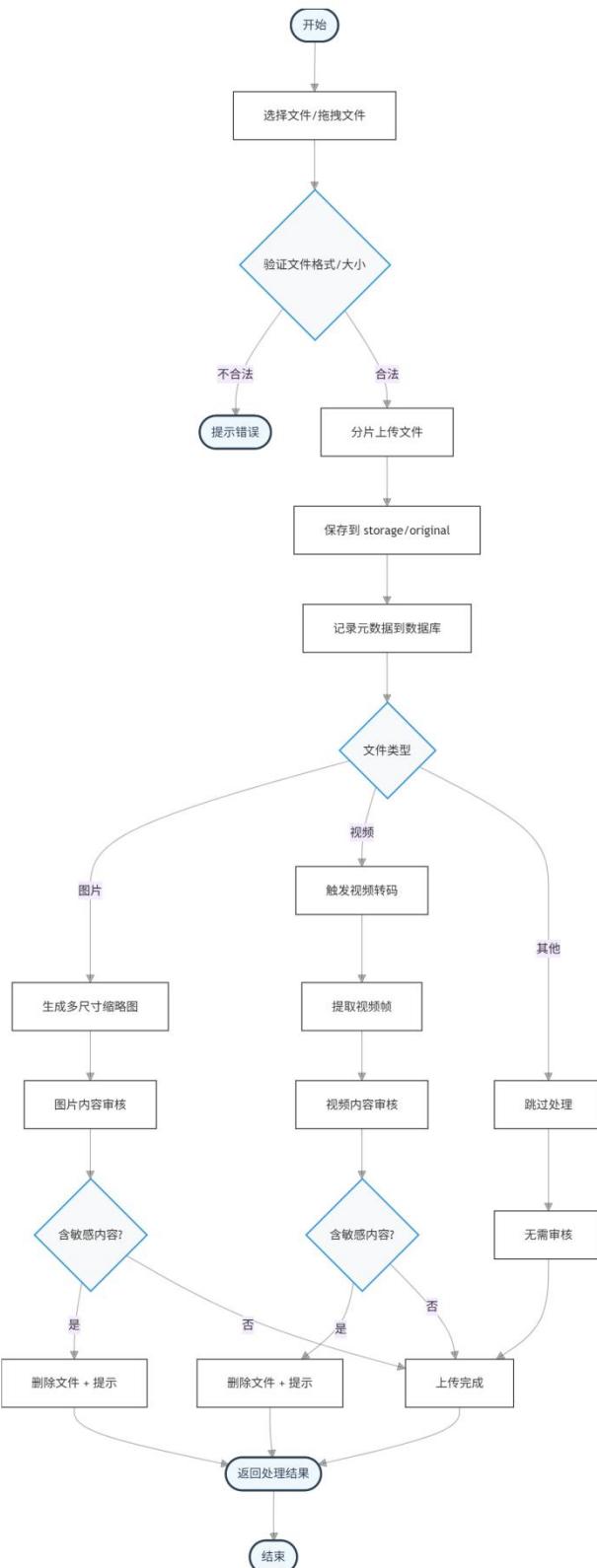
5.3 类图



5.4 时序图



5.5 活动图



6. 基于原型的进一步细化需求

6.1 文件管理需求

1. 多文件/大文件上传

支持同时选择多个文件批量上传，单个文件大小无硬性限制（通过分片上传机制适配大文件）。

上传过程中显示实时进度条，支持暂停/继续功能，网络中断后可自动恢复上传。

2. 拖拽上传

实现浏览器端拖拽功能，支持从本地文件管理器直接拖拽文件到网页上传区域触发上传。

拖拽时显示视觉反馈（如上传区域边框高亮、提示文字变化），并实时校验文件格式是否支持。

3. 文件搜索功能

支持文件名模糊搜索，匹配包含关键词的文件（基于 Whoosh 全文索引引擎）。

结合标签筛选（如文件类型、上传时间、识别结果标签），实现多维度精准搜索。

搜索结果实时展示，高亮匹配关键词，并支持按时间/大小排序。

4. 视频在线播放

基于 HLS 协议实现视频流式播放，支持断点续播和自适应码率（根据网络状况自动切换清晰度）。

播放器集成进度条、音量控制、全屏等基础功能，支持 MP4、WebM 等主流格式。

5. 文件分享功能

生成唯一加密短链（基于 UUID 和随机 token），支持设置分享有效期（如 24 小时、7 天）。

分享链接支持直接下载或在线预览（非登录用户可访问），并记录访问次数。

6.2 图片处理需求

1. 实时缩略图生成

图片上传完成后 1 秒内自动生成缩略图（基于 Pillow 库），存储于 storage/derived/ 目录。

缩略图生成失败时降级为默认占位图，并在后台日志记录错误信息。

2. 多尺寸支持

固定生成 128x128（列表展示用）和 256x256（预览用）两种尺寸，保持原

图比例避免拉伸。

支持通过 API 参数动态指定其他尺寸（如 /thumbnail/{asset_id}?size=512x512）。

3.GPU 加速（可选）

当系统检测到 NVIDIA GPU 时，可配置使用 PyTorchGPU 加速图像处理（需安装 torch 和 torchvision）。

未检测到 GPU 时自动降级为 CPU 处理，保证功能兼容性。

6.3 视频转码需求

1. 自动格式转码

上传后自动转码为 MP4 格式，编码格式支持 H.264（兼容性优先）和 H.265（高效压缩，可选）。

保留原始文件的同时，将转码后的视频存储于 storage/derived/ 目录，命名规则为 {原文件 ID}_transcoded_{格式}.mp4。

2. 多码率转码

支持生成多种码率版本（如 480p、720p、1080p），根据视频原始分辨率自适应最高支持的码率。

转码参数可配置（如帧率、比特率），默认使用 FFmpeg 的 -crf23 参数平衡画质和体积。

3. FFmpeg 微服务

基于 ffmpeg-python 库封装转码功能，作为独立后台任务运行（通过 FastAPI 的 BackgroundTasks 实现）。

转码失败时自动重试 1 次，仍失败则记录错误并通知用户。

4. 实时进度推送

通过 WebSocket 实时推送转码进度（如“分析视频”“转码视频”“处理音频”等阶段），前端动态更新进度条。

转码完成后触发通知（如页面弹窗提示“视频转码完成，可在线播放”）。

6.4 AI 内容审核需求

1. 审核类型及规则

裸露内容检测：通过分析图片/视频帧的 HSV 色彩特征（如皮肤色调占比超过 45%），标记可能的敏感内容。

暴力内容检测：基于 OCR 识别文本中的“暴力”“血腥”等关键词，或视频语音转文字中的敏感词汇，进行标记。

违法内容拦截：若检测到“政治敏感”“违法”等关键词，或视频帧包含违规画面，直接拒绝上传并返回原因。

2.OCR 文本提取与审核

对图片自动执行 OCR 识别（支持 pytesseract 或 EasyOCR），提取文字内容并存储于数据库（ocr_text 字段）。

对提取的文本进行关键词匹配，若包含敏感词则标记文件并限制分享。

3.ASR 语音转写与审核

对视频中的音频部分执行语音识别（支持 Vosk 或 Whisper），转写为文字后存储（transcript_text 字段）。

对转写文本进行敏感词检测，结果与视频文件关联，辅助内容审核决策。

4.审核结果处理

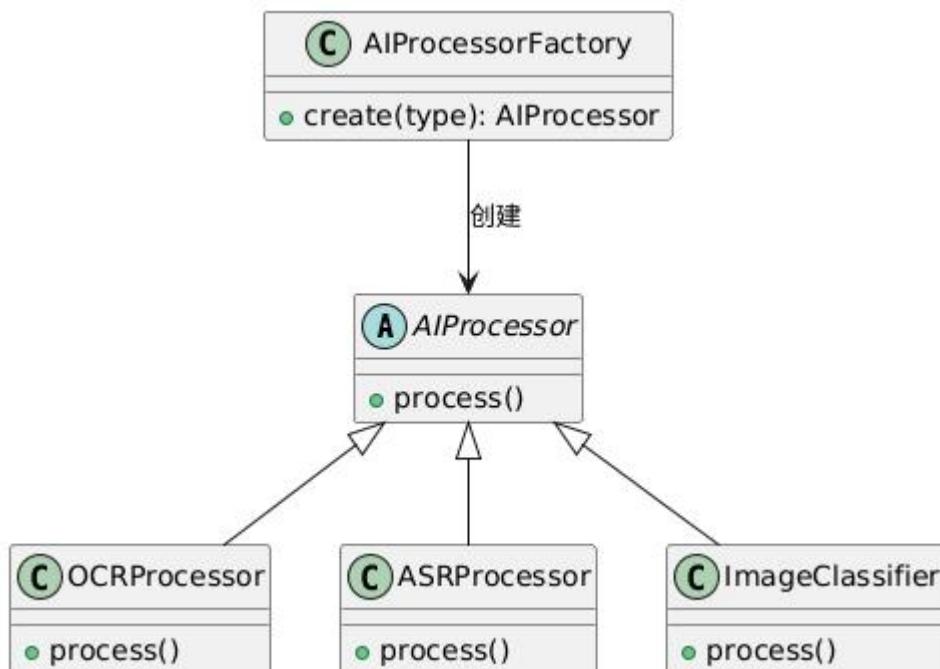
审核通过的文件正常展示；标记为“敏感”的文件仅本人可见，且无法分享；拒绝上传的文件自动删除并提示用户。

审核记录保存至数据库（moderation_flags 字段），支持管理员查看和人工复核。

7. 设计模式重构系统

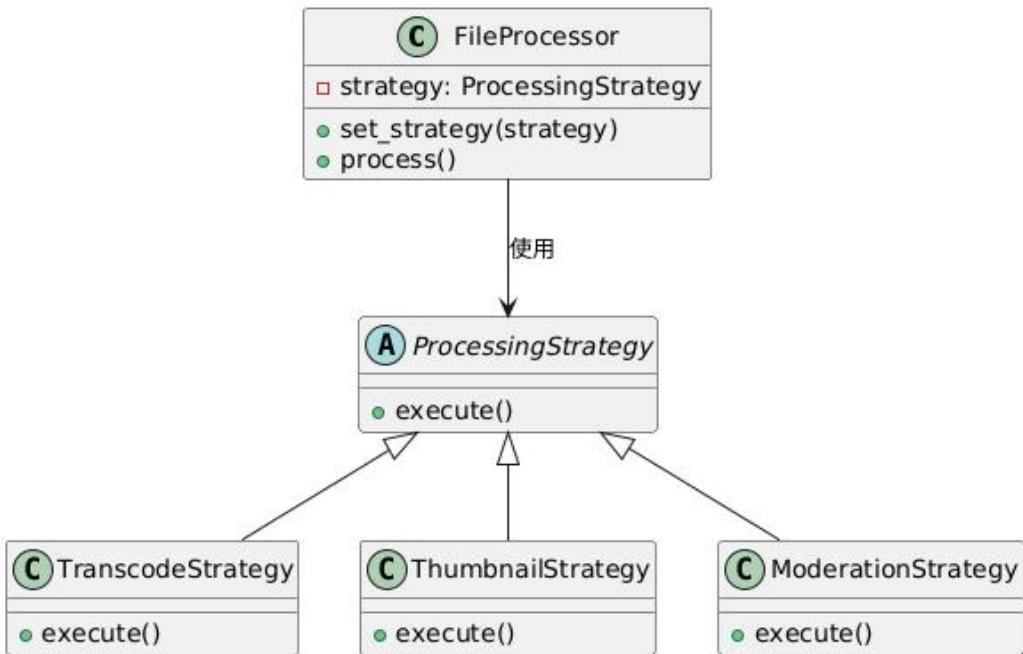
7.1 工厂模式（Factory Pattern）

创建不同 AI 处理器的统一接口，隐藏具体实现细节，使系统可灵活扩展新的 AI 处理能力。



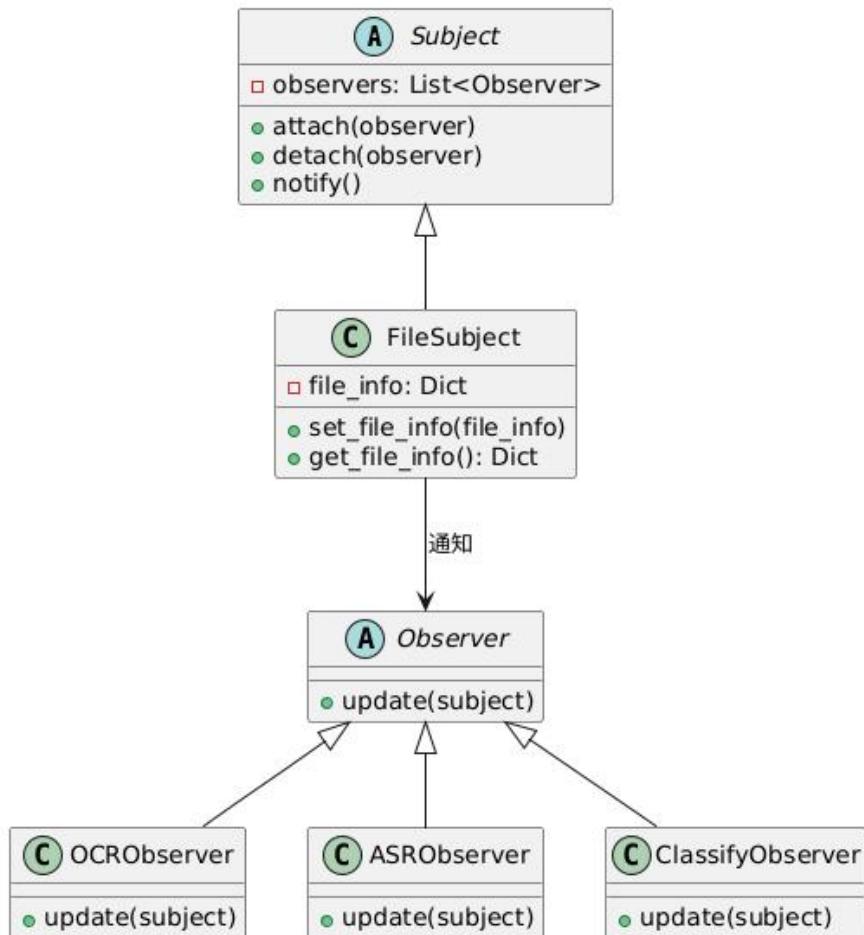
7.2 策略模式（Strategy Pattern）

定义一系列文件处理算法，将每种算法封装起来并使它们可互换，使算法可独立于使用它的客户端变化。



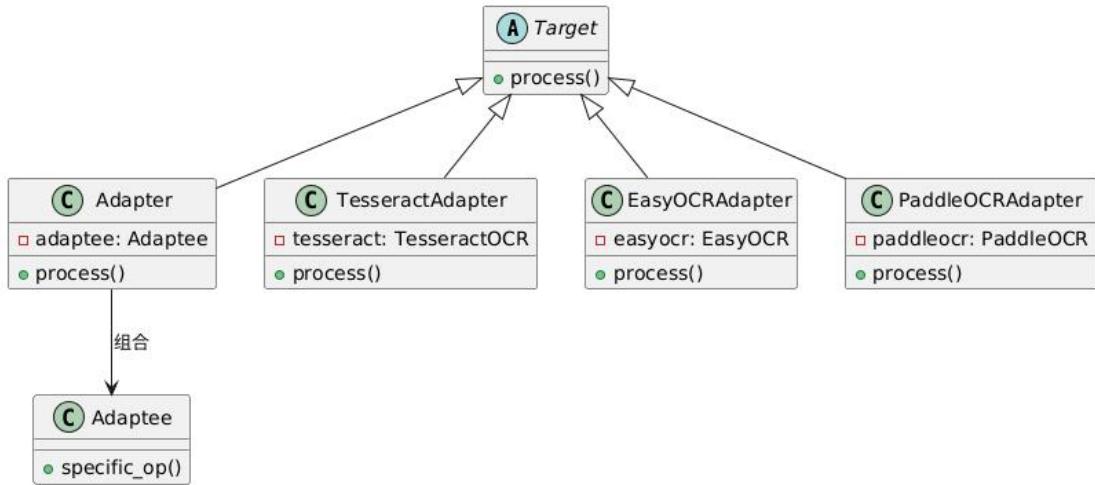
7.3 观察者模式 (Observer Pattern)

定义对象间的一对多依赖关系，当一个对象状态改变时，所有依赖它的对象都会收到通知并自动更新。用于文件上传后自动触发一系列 AI 任务。



7.4 适配器模式 (Adapter Pattern)

将一个类的接口转换成客户期望的另一个接口，使原本接口不兼容的类可以一起工作。用于统一不同 AI 模型的 API。



8. 总结

该开发计划针对企业视频、图像存储及智能处理需求，采用快速原型法设计并迭代系统，以实现文件上传、下载、共享等基础功能及实时图片缩略图生成、视频选择性转码、图片筛选与内容审核、敏感信息检查、实时图片分类、OCR 文字识别、ASR 语音识别七大智能功能为发展目标；团队构建“六边形”协作单元，明确项目负责人、前后端工程师等各角色职责，按需求分析与团队启动、原型设计与开发、需求细化与 UML 建模、设计模式重构、微服务/无服务开发规划五个阶段推进开发，同时完成登录注册、云盘主界面等核心界面设计，细化文件管理（如多文件分片上传）、图片处理（如多尺寸缩略图生成）、视频转码（如多码率转码）、AI 内容审核（如多维度敏感检测）等模块需求，设计用例图、系统架构图等，还通过工厂模式、策略模式、观察者模式、适配器模式优化系统，提升扩展性与灵活性，为系统开发提供全面指导。