# 🧠 Google\_Drive\_Manager

# Concept Analysis: Core Goals and Capabilities

**🎯 Primary Objective**

Build a **Windows-based Python GUI tool** that connects to Google Drive (via API), analyzes folder/file structure and metadata, identifies duplicates and naming inconsistencies, and provides a user interface for reviewing, approving, and executing cleanup actions.

**🧱 Infrastructure & Setup**

**✅ Best Practices for API Key and OAuth Management**

Based on current guidance:

* **Use environment variables** for all secrets (never hardcoded).
* **Label clearly**: e.g. GOOGLE\_OAUTH\_CLIENT\_ID\_SEARCHFILES, GOOGLE\_OAUTH\_CLIENT\_SECRET\_SEARCHFILES, GOOGLE\_OAUTH\_CLIENT\_ID\_X1, etc.
* **Use one OAuth client per app** (not per user).
* **Restrict API keys** to specific services and IPs if possible.
* **Avoid duplicate keys**—rotate and delete unused ones.

**🧩 Suggested Environment Variable Naming Convention**

| **Purpose** | **Variable Name** |
| --- | --- |
| Search & Find Google Files | GOOGLE\_OAUTH\_CLIENT\_ID\_SEARCHFILES, GOOGLE\_OAUTH\_CLIENT\_SECRET\_SEARCHFILES |
| Andrews X1 Thinkpad | GOOGLE\_OAUTH\_CLIENT\_ID\_X1, GOOGLE\_OAUTH\_CLIENT\_SECRET\_X1 |
| YouTube Data Pull | GOOGLE\_API\_KEY\_YOUTUBE |
| Google Maps Geocoding | GOOGLE\_API\_KEY\_GEOCODING |

**🖥️ Tool Design: Functional Modules**

**1. Google Drive Integration**

* Use **Google Drive API v3** with OAuth2 flow.
* Authenticate once, store token securely.
* Pull folder structure, metadata, file names, MIME types, modified dates, etc.

**2. Metadata & Content Analysis**

* Extract file metadata.
* Optionally download file previews (text, PDF, docx) for content-based classification.
* Use NLP or regex to infer purpose, category, or suggested folder.

**3. Duplicate Detection**

* Match by:
  + File name
  + File content hash (SHA256)
  + Metadata (modified date, size)
* Flag:
  + Exact duplicates
  + Same name, different content
  + Same content, different name

**4. GUI Review & Approval**

* Built with **PySide6** or **Tkinter** (PySide6 preferred for modern UI).
* Table view with:
  + Current file name
  + Suggested name
  + Editable field
  + Folder path
  + Parent folder
  + Action buttons (Rename, Move, Delete, Merge)
  + Filters and sort options

**5. Folder Reorganization**

* Suggest merges, renames, moves.
* Show preview of new structure.
* Allow override and manual edits.

**6. Logging & Audit Trail**

* Log all actions:
  + Variables used
  + API calls
  + File changes
  + User approvals
* Save logs to local file (e.g. drive\_cleanup\_log.json).

**7. User Experience & Feedback**

* Progress indicators
* Error messages
* Undo option
* Help menu
* Exit confirmation
* Configurable settings (e.g. ignore folders, recursive depth)

**🧭 Suggested Functional Structure**

| **Module** | **Functions** |
| --- | --- |
| Authentication | Connect to Google Drive, manage tokens |
| Data Retrieval | Pull folder/file metadata |
| Analysis | Detect duplicates, suggest names/folders |
| GUI | Display data, allow edits, approvals |
| Execution | Rename/move/delete files via API |
| Logging | Track all actions and variables |
| Settings | Configure behavior, filters, exclusions |

**❓ Clarifying Questions for Architecture & Design**

Here’s a structured set of questions to help finalize the concept:

**🔐 Authentication & API Setup**

| **Question** | **Valid Responses** | **Suggested Response** | **Impact** |
| --- | --- | --- | --- |
| Do you want to use one OAuth client for all functions or separate per function? | One shared client, Separate per function | One shared client | Simplifies setup and avoids duplication |
| Should the tool support multiple Google accounts? | Yes, No | Yes | Adds complexity but supports broader use |
| Should API keys be rotated periodically? | Yes, No | Yes | Improves security, requires key management |

**📁 Folder Selection & Scope**

| **Question** | **Valid Responses** | **Suggested Response** | **Impact** |
| --- | --- | --- | --- |
| Should the tool allow selecting any folder or only predefined ones? | Any folder, Predefined list | Any folder | More flexible, requires folder picker |
| Should subfolders be included recursively? | Yes, No, Configurable | Configurable | Adds control for large structures |
| Should certain folders be ignored? | Yes, No | Yes | Prevents clutter (e.g. system folders) |

**🧠 File Analysis**

| **Question** | **Valid Responses** | **Suggested Response** | **Impact** |
| --- | --- | --- | --- |
| Should the tool analyze file content or just metadata? | Metadata only, Metadata + content | Metadata + content | Enables smarter suggestions, but slower |
| Should NLP be used to infer file purpose? | Yes, No | Yes | Adds intelligence, requires model or library |
| Should file previews be downloaded? | Yes, No, Configurable | Configurable | Balances speed vs insight |

**🧹 Duplicate Handling**

| **Question** | **Valid Responses** | **Suggested Response** | **Impact** |
| --- | --- | --- | --- |
| What defines a duplicate? | Name only, Content only, Both | Both | Most accurate |
| Should duplicates be auto-deleted or user-approved? | Auto, User approval | User approval | Safer, more transparent |
| Should older versions be archived? | Yes, No | Yes | Preserves history, adds storage cost |

**🖼️ GUI Design**

| **Question** | **Valid Responses** | **Suggested Response** | **Impact** |
| --- | --- | --- | --- |
| Which GUI framework? | Tkinter, PySide6, PyQt | PySide6 | Modern, scalable |
| Should the interface be tabbed or single-page? | Tabbed, Single-page | Tabbed | Organizes functions cleanly |
| Should edits be inline or in popups? | Inline, Popup | Inline | Faster workflow |

**📝 Logging & Feedback**

| **Question** | **Valid Responses** | **Suggested Response** | **Impact** |
| --- | --- | --- | --- |
| Should all actions be logged? | Yes, No | Yes | Enables audit and rollback |
| Should logs be saved locally or to cloud? | Local, Cloud, Both | Local | Easier setup |
| Should verbose debugging be enabled? | Yes, No, Configurable | Configurable | Helps troubleshooting |

Would you like me to start drafting the Python architecture and GUI layout next? I can also help you define the folder picker, metadata puller, and duplicate checker as modular scripts. Let’s build this together—your vision is clear, and I’ll make sure the execution is just as sharp.