# 🧱 Google\_Drive\_Manager-Project Architecture Overview

Google\_Drive\_Manger/

├── Google\_Drive\_Manger/

│ ├── \_\_init\_\_.py

│ ├── main.py

│ ├── launcher.py

│ ├── config.py

│ ├── auth.py

│ ├── drive\_api.py

│ ├── analyzer.py

│ ├── gui.py

│ ├── logger.py

│ ├── utils.py

│ └── constants.py

├── docs/

│ ├── README.md

│ ├── CHANGELOG.md

│ ├── ARCHITECTURE.md

│ ├── DECISIONS.md

│ ├── ISSUES.md

│ └── LOGS/

│ ├── execution\_log\_YYYYMMDD\_HHMMSS.txt

│ ├── actions\_log\_YYYYMMDD\_HHMMSS.json

│ └── debug\_log\_YYYYMMDD\_HHMMSS.csv

├── data/

│ ├── raw/

│ ├── interim/

│ ├── processed/

│ └── outputs/

│ ├── html/

│ ├── csv/

│ ├── txt/

│ ├── json/

│ └── md/

├── run.bat

├── requirements.txt

└── setup.py

**🧠 Initial File: main.py**

"""

File: main.py

Version: 0.1.0

Author: Andrew Pearen

Created: 2025-09-23

Last Modified: 2025-09-23

Purpose: Entry point for Google Drive Manager tool.

Logging: All actions and errors are logged to /docs/LOGS with timestamped filenames.

"""

# [1] Imports

import sys

from logger import setup\_logging

from auth import authenticate\_google\_drive

from gui import launch\_gui

# [2] Setup Logging

logger = setup\_logging("main")

# [3] Main Execution

def main():

logger.info("Starting Google Drive Manager")

try:

service = authenticate\_google\_drive()

launch\_gui(service)

except Exception as e:

logger.exception("Fatal error in main execution")

sys.exit(1)

if \_\_name\_\_ == "\_\_main\_\_":

main()

**🖥️ GUI Layout: gui.py (PySide6)**

"""

File: gui.py

Version: 0.1.0

Author: Andrew Pearen

Created: 2025-09-23

Purpose: GUI interface for reviewing and managing Google Drive files.

"""

# [1] Imports

from PySide6.QtWidgets import (

QApplication, QMainWindow, QTableWidget, QTableWidgetItem,

QPushButton, QVBoxLayout, QWidget, QFileDialog, QLabel

)

from logger import setup\_logging

logger = setup\_logging("gui")

# [2] GUI Class

class DriveManagerGUI(QMainWindow):

def \_\_init\_\_(self, drive\_service):

super().\_\_init\_\_()

self.drive\_service = drive\_service

self.setWindowTitle("Google Drive Manager")

self.setGeometry(100, 100, 1200, 800)

self.init\_ui()

def init\_ui(self):

layout = QVBoxLayout()

self.status\_label = QLabel("Ready")

self.table = QTableWidget(0, 5)

self.table.setHorizontalHeaderLabels([

"Current Name", "Suggested Name", "Folder", "Parent Folder", "Action"

])

self.load\_button = QPushButton("Load Folder")

self.load\_button.clicked.connect(self.load\_folder)

layout.addWidget(self.status\_label)

layout.addWidget(self.table)

layout.addWidget(self.load\_button)

container = QWidget()

container.setLayout(layout)

self.setCentralWidget(container)

def load\_folder(self):

folder\_id = QFileDialog.getExistingDirectory()

logger.info(f"Selected folder: {folder\_id}")

# Placeholder: Load metadata and populate table

def launch\_gui(service):

app = QApplication([])

window = DriveManagerGUI(service)

window.show()

app.exec()

**🪵 Logging Setup: logger.py**

"""

File: logger.py

Version: 0.1.0

Author: Andrew Pearen

Created: 2025-09-23

Purpose: Centralized logging configuration.

"""

import logging

from datetime import datetime

import os

def setup\_logging(module\_name):

timestamp = datetime.now().strftime("%Y%m%d\_%H%M%S")

log\_dir = os.path.join("docs", "LOGS")

os.makedirs(log\_dir, exist\_ok=True)

log\_file = os.path.join(log\_dir, f"{module\_name}\_log\_{timestamp}.txt")

logger = logging.getLogger(module\_name)

logger.setLevel(logging.DEBUG)

fh = logging.FileHandler(log\_file)

fh.setLevel(logging.DEBUG)

ch = logging.StreamHandler()

ch.setLevel(logging.INFO)

formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')

fh.setFormatter(formatter)

ch.setFormatter(formatter)

logger.addHandler(fh)

logger.addHandler(ch)

return logger

**🧨 Launcher: run.bat**

@echo off

echo Launching Google Drive Manager...

python Google\_Drive\_Manger\main.py

echo.

echo Press any key to exit...

pause >nul

Would you like me to continue with auth.py, drive\_api.py, and the initial docs/ARCHITECTURE.md next? I’ll also begin the DECISIONS.md and ISSUES.md logs so we can track every choice and challenge. Let’s build this system with the rigor and clarity it deserves.