

# Projektbeschreibung: Master GUI mit PySide6

## 1. Einleitung

Diese Projektvorlage definiert eine **Master GUI**-Anwendung für Windows, basierend auf **PySide6**, die als wiederverwendbares Fundament für künftige Automatisierungs- und Management-Tools dient. Ziel ist eine **modulare, anpassbare** und **erweiterbare** Anwendung mit flexibler Buttonverwaltung, Skript- und Link-Integration sowie aussagekräftiger Statusrückmeldung.

## 2. Zielsetzung

- **Flexible Buttonverwaltung:** Hinzufügen, Entfernen, Bearbeiten und Positionieren (Drag & Drop) von Buttons in einem skalierbaren Grid.
- **Aktionstypen** für Buttons:
  - **SCRIPT:** Ausführung von Python-Skripten mit Fortschritts- und Fehler-Callbacks.
  - **LINK:** Öffnen externer URLs oder lokaler Pfade.
  - **GROUP:** Gruppieren von Buttons mit Breadcrumb-Navigation.
  - **FILE:** Öffnen von Dateien mit dem Standardprogramm.
  - **EXPLORER:** Öffnen von Verzeichnissen im Windows Explorer.
- **Task-Dashboard:** Zentrales Fenster zur Überwachung aller laufenden und abgeschlossenen Jobs inklusive Fortschritt, Status, Laufzeit und Log-Zugriff.
- **Konfigurations-Persistenz:** Speichern aller Button-Settings (Position, Größe, Icon-Pfad, Aktionstyp, Payload) in einer `config.json` per relativen Pfaden.
- **Theming:** Laden von QSS-Stylesheets und Background-Images vor Programmstart.
- **Packaging:** Erzeugung einer Windows-EXE via PyInstaller (non-console mode).

## 3. Projektstruktur

```
/project_root/
├── main.py           # Einstiegspunkt, Application Bootstrap
├── util/paths.py     # project_root() & relative-Pfad-Helper
├── core/             # Geschäftslogik & Datenmodelle
│   └── models.py     # ButtonModel @ dataclass, ButtonAction Enum
```

		storage.py	# config.json laden/speichern + Schema-Validation
		runner.py	# ScriptRunner (QRunnable) + Callback-Interface
		dispatcher.py	# TaskDashboard-Verbindung & Signale
		theming.py	# QSS-Loader, Background-Setter
		ui/	# Benutzeroberfläche
		master_window.py	# Hauptfenster mit QStackedWidget (Seiten)
		task_dashboard.py	# Singleton-Dialog für Job-Übersicht
		button_editor.py	# CRUD-Dialog für Button-Pflege
		assets/	# Icons, Default-Background
		gui_tools/	# Externer GUI-Editor (Mini-GUI)
		logs/	# RotatingFileHandler (max 5 MB)
		config.json	# Start-Konfiguration (Beispiel)
		tests/	# (optional) Smoke-Tests / PyTest

## 4. Technische Eckdaten

- **GUI-Toolkit:** PySide6
- **Threading:** QThreadPool für unbegrenzte parallele Jobs
- **Callback-Signatur:** `report(progress: int, msg: str) -> None` (0–100%, freier Text)
- **Grid-Layout:** Maximal 30 Buttons pro Seite, Pagination via QStackedWidget-Seiten ("Seite 1 / n").
- **Icon-Größe:** Default 64×64px, adaptierbar (Wert im ButtonModel).
- **Persistenz:** JSON-Datei (`config.json`) mit absolut portablen relativen Pfaden.
- **Theming:** QSS-Stylesheet und Hintergrundbild vor Start konfigurierbar.

## 5. Projektplan (Meilensteine)

Phase	Beschreibung	Ziel-Datum
P0	Spezifikation & Kick-off (alle Fragen geklärt)	09.07.2025
P1	Repository, Skeleton & Pfad-Helper	11.07.2025
P2	Domain-Model & Storage mit JSON-Schema	15.07.2025
P3	UI-Grundgerüst + Pagination-Seiten	22.07.2025
P4	Drag & Drop + Resize	26.07.2025

P5	Gruppen-Handling & Breadcrumb-Leiste	31.07.2025
P6	ScriptRunner + Task-Dashboard	07.08.2025
P7	Mini-GUI Button-Editor (CRUD)	14.08.2025
P8	Theming & Background Loader	20.08.2025
P9	Logging (RotatingFileHandler) & Progress UI	27.08.2025
P10	PyInstaller Packaging & Smoke-Tests	03.09.2025
P11	Polishing, README & Quick-Start-Guide	10.09.2025

## 6. To-Do-Liste (Auszug Sprint 00)

- **S00-1:** `git init`, Ordnerstruktur anlegen (👤 Du)
- **S00-2:** `requirements.txt` (PySide6==6.\*) erzeugen (🤖 GPT)
- **S00-3:** `util/paths.py` mit `project_root()` und `relative()` erstellen (🤖 GPT)
- **S00-4:** Leeres `main.py` mit QApplication-Bootstrap (🤖 GPT)
- **S00-5:** Initialer Commit & Push ins Remote-Repo (👤 Du)

## 7. Weiteres Vorgehen

Nach erfolgreichem Abschluss von Sprint 00 startet Sprint 01 mit Domain-Model und Storage. Du markierst jede erledigte Aufgabe durch Nennung der Nummer (z. B. „S00-1 done“), dann generiere ich die entsprechenden Code-Snippets, QSS-Vorlagen oder Tests.

---

**Damit liegt nun eine umfassende Projektbeschreibung vor, die alle bisherigen Anforderungen, Entscheidungen und Fahrpläne konsolidiert.**

Falls noch etwas fehlt oder unklar ist, melde dich bitte – ansonsten können wir direkt mit Sprint 00 starten.