oMioBio Pflanzendatenbank (PDB)

Pflichtenheft Version 1.0

Datum: 09. Januar 2025
Erstellt für: Peter Müller
Zweck: Open Source Tool für Bio-Gärtnereien, Gartenplaner und Gartenbauer

---

1. PROJEKTZIEL

1.1 Vision

Ein portables, benutzerfreundliches Open-Source-System zur Verwaltung von Pflanzendaten, das:

- Bio-Gärtnereien bei der Bestandsverwaltung unterstützt
- Gartenplanern bei der Pflanzenwahl hilft
- Den Datenaustausch zwischen Gärtnereien ermöglicht
- Auf USB-Stick lauffähig ist (XAMPP-basiert)
- Am Open Farming Hackday weiterentwickelt werden kann

1.2 Zielgruppen

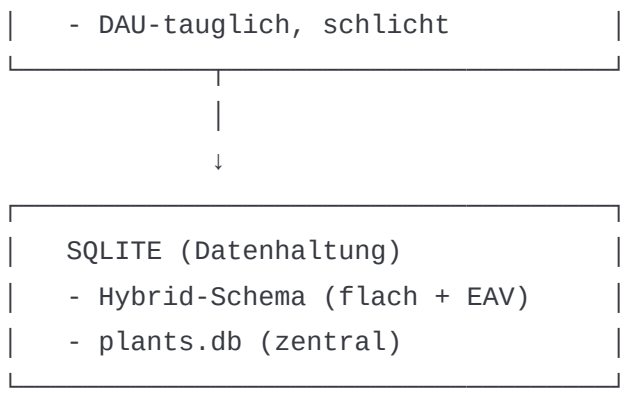
1. **Bio-Gärtnereien** (Primäre Nutzer)
    - Pflanzendaten erfassen und verwalten
    - Bestand tracken
    - Etiketten drucken
    - Export zu Webshops
2. **Gartenplaner & Gartenbauer** (Sekundäre Nutzer)
    - Pflanzenlisten erstellen
    - Verfügbarkeit prüfen
    - Anfragen an Gärtnereien senden
3. **Öffentliche User** (Tertiäre Nutzer)
    - Pflanzeninformationen abrufen
    - Gärtnereisuche

---

2. SYSTEM-ARCHITEKTUR

2.1 Technologie-Stack

```
┌─────────────────────────────────────┐
│   GRAV CMS (Frontend/UI)            │
│   - Liberation Sans Font            │
│   - Custom Design (#4B8223)         │
```



Platform: XAMPP (Windows/Linux/Mac)
Portabilität: USB-Stick-fähig
Lizenz: Open Source (MIT oder GPL)

2.2 Datenbank-Schema (Hybrid)

FLACHE TABELLE: `plants` (Stammdaten)

sql

```sql
CREATE TABLE plants (
    id INTEGER PRIMARY KEY,

    -- PFLICHTFELD!
    genus TEXT NOT NULL,

    -- Taxonomie
    species TEXT,
    subspecies TEXT,
    cultivar TEXT,
    ecotype TEXT,
    ecotype_origin TEXT,

    -- Generiert (automatisch)
    botanical_name TEXT UNIQUE NOT NULL,  -- EINDEUTIG!
    web_slug TEXT UNIQUE,
    matchcode TEXT,
    qr_code TEXT,

    -- Kategorisierung
    category TEXT,
    subcategory TEXT,

    -- Namen (mehrsprachig)
    name_de TEXT,
```

```
    name_en TEXT,
    name_fr TEXT,
    name_it TEXT,
    synonym TEXT,

    -- Meta
    created_at TIMESTAMP,
    updated_at TIMESTAMP,
    created_by TEXT
);
```

EAV-TABELLE: `plant_sections` (Aufklapp-Bereiche)

sql

```
CREATE TABLE plant_sections (
    id INTEGER PRIMARY KEY,
    plant_id INTEGER NOT NULL,

    section_name TEXT NOT NULL,  -- "bluete", "herkunft", "frucht"...
    field_name TEXT NOT NULL,    -- "bluetenfarbe", "zuechter_name"...

    value_text TEXT,
    value_numeric REAL,
    value_unit TEXT,
    value_date TEXT,

    data_source TEXT,
    notes TEXT,

    FOREIGN KEY (plant_id) REFERENCES plants(id) ON DELETE CASCADE
);
```

Bereiche (sections):

1. Herkunft (8 Felder)
2. Standort (6 Felder)
3. Blüte (6 Felder)
4. Blatt (4 Felder)
5. Frucht (7 Felder)
6. Wurzel (flexibel)
7. Wuchs (5 Felder)
8. Gärtnereidaten (13 Felder)
9. Verwendung (4 Felder)

10. Verfügbarkeit (6 Felder)
    (Alle erweiterbar)

---

3. FUNKTIONALE ANFORDERUNGEN

3.1 PRIO 1: Kern-Funktionen (MVP)

3.1.1 LISTE (Pflanzenliste anzeigen)

- Tabellenansicht mit Sortierung
- Live-Suche (während Tippen)
- Filter nach Kategorie
- Pagination (20/50/100/500 pro Seite)
- Spalten-Auswahl (User wählt, welche Felder angezeigt werden)
- Botanischer Name in erster Spalte (fett, eventuell condensed?)
- Export-Funktion (CSV/JSON/YAML)

3.1.2 EINGABE (Neue Pflanze erfassen)

- **Stammdaten immer sichtbar** (Gattung = PFLICHT!)
- **Aufklapp-Bereiche** für optionale Felder
    - Beispiel Rosen: Blüte aufklappen → Daten eintragen
    - Beispiel Gemüse: Frucht aufklappen → Daten eintragen
- **Botanischer Name automatisch generieren**
- **Dublettenprüfung** (Botanischer Name = eindeutig!)
- **Daten bleiben beim Zuklappen erhalten**
- **Badge zeigt ausgefüllte Felder** (z.B. "3/6" in Bereich Blüte)
- **Warnung bei ungespeicherten Änderungen**
- **"Möchten Sie speichern?"-Dialog beim Verlassen**

3.1.3 DATENBANKSTRUKTUR

- Hybrid-Schema implementiert (flach + EAV)
- Dublettenprüfung über `botanical_name UNIQUE`
- Indizes für Performance
- Migration von bestehenden Pflanzen (Satz von +1500 Pflanzen vorhanden)

3.2 PRIO 2: Erweiterte Funktionen

3.2.1 PORTRAIT (Detailansicht)

- Einzelne Pflanze vollständig anzeigen
- Alle Bereiche ausgeklappt
- Bilder einbinden (falls vorhanden)
- "Bearbeiten"-Button
- QR-Code anzeigen

3.2.2 BESTAND (Inventory-Management)

- Topfgrössen-Verwaltung (P9, C2, C10...)
- Jungpflanzen Produktionsverwaltung
- Anzahl pro Topfgrösse
- Bestandsverlauf
- "Nachproduzieren"-Funktion

3.2.3 TRANSFER (Datenaustausch)

- CSV-Import von Pflanzenlisten
- CSV-Export mit Feldauswahl
- API-Endpoints für Webshops
    - Bagisto-Integration
    - Shopware-Integration
    - Gambio
    - ...und andere Shopsysteme

3.2.4 HILFE

- Inline-Dokumentation
- Tooltips bei Feldern
- Video-Tutorials (später)

3.3 PRIO 3:

3.3.1 ETIKETTEN-DRUCK

- Verschiedene Drucker unterstützen
    - Stecketikettendrucker
    - Labeldrucker
    - Matrix-Drucker
    - Brother Drucker
    - Dymo LabelWriter
- Verschiedene Etikettenformate
- Template-System für Layouts
- QR-Code auf Etikett

3.3.2 NETZWERK-SYNC

- Öffentlicher Server für Pflanzendaten
- Gärtnereien können Daten teilen (opt-in)
- Verfügbarkeits-Sync zwischen Gärtnereien
- Gartenplaner können Anfragen stellen

3.3.3 MULTI-USER

- Benutzer-Verwaltung
- Rollen: Admin, Editor, User, Readonly
- Änderungs-Historie
- Jede Gärtnerei hat eigenes Login

---

4. NICHT-FUNKTIONALE ANFORDERUNGEN

4.1 Usability (DAU-Tauglich!)

- **Einfachheit über Features:** Keine unnötigen Funktionen
- **Selbsterklärend:** Kein Handbuch nötig für Grundfunktionen
- **Fehlervermeidung:** Dublettenprüfung, Pflichtfelder klar markiert
- **Konsistenz:** Gleiche UI-Elemente überall

4.2 Design-Prinzipien

- **Schrift Schwarz statt Grau:** Lesbarkeit > Ästhetik
- **Keine Kursivschrift:** Immer normal (nie italic)
- **Liberation Sans:** Überall die gleiche Schriftart
- **Grün:** `#4B8223` für Primär-Elemente
- **Minimalistisch:** Kein Schnickschnack, keine Animationen

4.3 Performance

- **<1 Sekunde:** Suche bei 1.500 Pflanzen
- **<3 Sekunden:** Seite laden
- **Skalierbarkeit:** Bis +50.000 Pflanzen ohne Probleme

4.4 Portabilität

- **USB-Stick:** Komplett lauffähig von externem Medium
- **Offline-First:** Funktioniert ohne Internet
- **Plattformen:** Windows, Linux, macOS (via XAMPP)

4.5 Wartbarkeit

- **Open Source:** Code auf GitHub
- **Dokumentiert:** Inline-Kommentare + README
- **Standard-Tools:** Grav CMS + SQLite (keine Exoten)
- **Erweiterbar:** Plugin-System (später)

---

5. SCHNITTSTELLEN

5.1 Import

- **CSV:** Pflanzenlisten importieren
- **JSON:** Aus bestehenden Systemen
- **YAML:** Aus Grav Collections

5.2 Export

- **CSV:** Für Excel/Shops

- **JSON:** Für APIs
- **YAML:** Für Backups
- **PDF:** Kataloge (später)

5.3 APIs (später)

- **REST API:** `/api/plants`, `/api/plants/{id}`
- **Bagisto:** Produkt-Sync
- **Shopware**: Shop-Integration
- andere Shopsysteme

---

6. DATENFELDER (Vollständige Liste)

Stammdaten (immer sichtbar)

- Gattung ⭐ PFLICHT
- Art
- Unterart
- Sorte
- Ökotyp
- Herkunft Ökotyp
- Botanischer Name (generiert) ⭐ EINDEUTIG Format "Gattung art 'Sorte'
- Webname (generiert) Format gattung-art-sorte
- Matchcode (generiert) Format GATARTSOR
- QR-Code (generiert)
- Kategorie
- Unterkategorie

Namen

- Name DE
- Name EN
- Name FR
- Name IT
- Synonym

Herkunft (aufklappbar)

- Züchter Name
- Züchter Land
- Züchter Jahr
- Sortenschutz
- Markenschutz
- Züchtername
- Ursprungsregion
- Züchtungsmethode

Standort (aufklappbar)

- Winterhärte
- Licht
- Feuchtigkeit
- Bodenart
- pH-Wert Boden
- Nährstoffbedarf

Blüte (aufklappbar)

- Blütenfarbe
- Duft
- Blütengrösse
- Blütenfüllung
- Blütenform
- Haltbarkeit

Blatt (aufklappbar)

- Farbe
- Form
- Aroma
- Struktur

Frucht (aufklappbar)

- Farbe
- Grösse
- Geschmack
- Saftigkeit
- Textur
- Erntezeit
- Entwicklungsdauer Blüte bis Ernte

Wurzel (aufklappbar)

- (noch zu definieren)

Wuchs (aufklappbar)

- Wuchszyklus
- Höhe
- Breite
- Form
- Stärke

Gärtnereidaten (aufklappbar)

- Topfgrösse

- Anzahl verkaufsfertig
- Vermehrungsart
- Jungpflanzengebinde
- Anzahl pro JPG
- Anzahl JPG
- Pflanzen pro m²
- Entwicklungsdauer bis JP
- Entwicklungsdauer JP – FP
- Keimdauer
- Keimart Licht
- Keimart Temperatur
- Bewurzelungsdauer

Verwendung Erntegut (aufklappbar)

- Pflanzenteile
- Zweck
- Verarbeitung
- Lagerung

Verfügbarkeit (aufklappbar)

- Gärtnereien
- Qualität
- Labels
- Menge
- Preis
- Vorbestellbar

Bei Bedarf noch weitere Felder.

---

7. BENUTZER-SZENARIEN

7.1 Szenario: User erfasst neue Rose

1. Öffnet `/pdb/eingabe`
2. Füllt **Stammdaten** aus:
    - Gattung: "Rosa"
    - Art: "damascena"
    - Sorte: "Rose de Resht"
    - Botanischer Name generiert sich → "Rosa damascena 'Rose de Resht'"
3. **Dublettenprüfung** läuft automatisch ✅
4. Klappt **Blüte** auf:
    - Blütenfarbe: "dunkelrot"
    - Duft: "stark"
5. Klappt **Herkunft** auf:

    - Züchter: "unbekannt"
    - Region: "Persien"
6. **Badge zeigt:** "2/6" bei Blüte, "2/8" bei Herkunft
7. Klickt **Speichern** → Pflanze in DB

7.2 Szenario: Gartenplaner sucht Stauden

1. Öffnet `/pdb/liste`
2. Sucht nach "Lavendel"
3. Filtert nach Kategorie "Kräuter"
4. Sortiert nach Winterhärte
5. Findet "Lavandula angustifolia 'Munstead'"
6. Klickt **Ansehen** → Details
7. Exportiert Pflanzenliste als CSV

7.3 Szenario: User exportiert zu Bagisto

1. Öffnet `/pdb/transfer`
2. Wählt **Export-Format:** "CSV (Bagisto)"
3. Wählt **Felder:**
    - Botanischer Name
    - Deutscher Name
    - Preis
    - Topfgrösse
4. Exportiert 200 Pflanzen
5. Importiert CSV in Bagisto

---

8. MIGRATION BESTEHENDER DATEN

8.1 Aktueller Stand

- **1.547 Pflanzen** in `pflanzen.db`
- EAV-Schema vorhanden

8.2 Migrations-Strategie für .json zu SQLite

sql

```sql
-- Schritt 1: Stammdaten übernehmen
INSERT INTO plants (genus, species, cultivar, botanical_name, name_de, category)
SELECT genus, species, cultivar, botanical_name, name_de,
       (SELECT category FROM plant_categories WHERE plant_id = plants.id LIMIT
1)
FROM old_plants;

-- Schritt 2: EAV-Daten konvertieren
INSERT INTO plant_sections (plant_id, section_name, field_name, value_text)
```

```
SELECT plant_id,
       CASE
           WHEN trait_name IN ('Bluetenfarbe', 'Duft') THEN 'bluete'
           WHEN trait_name IN ('Winterhaerte', 'Standort') THEN 'standort'
           -- etc.
       END,
       LOWER(trait_name),
       value_text
FROM old_plant_traits;
```

8.3 Import-Script

- `scripts/import_from_json.py` bereits vorhanden
- Erweitern für neues Schema
- Test mit 10 Pflanzen, dann vollständig

---

9. DEPLOYMENT

9.1 Lokale Installation (Gärtnerei)

bash

```
1. XAMPP herunterladen + installieren
2. Grav CMS in htdocs/pdb/ entpacken
3. plants.db nach user/data/ kopieren
4. Apache + PHP starten
5. http://localhost/pdb/ öffnen
```

9.2 USB-Stick-Installation

```
USB:\
├── xampp-portable\
│   ├── apache\
│   ├── php\
│   └── htdocs\pdb\
│       ├── user\data\plants.db
│       └── ...
└── START_PDB.bat  (startet XAMPP + öffnet Browser)
```

9.3 Öffentlicher Server

- Shared Hosting bei cyon
- HTTPS-Zertifikat von letsencrypt
- Regelmässige Backups

- Multi-Mandanten-Fähigkeit

---

10. TESTING

10.1 Manuelle Tests

- Pflanze erfassen (alle Bereiche)
- Dublettenprüfung
- Suche & Filter
- Sortierung
- Export CSV
- Import CSV
- Ungespeicherte-Änderungen-Warnung

10.2 Automatisierte Tests (später)

- Unit-Tests für DB-Queries
- Integration-Tests für API
- UI-Tests (Selenium)

---

11. DOKUMENTATION

11.1 User-Dokumentation

- README.md (GitHub)
- Inline-Hilfe in Grav
- Video-Tutorials (YouTube)
- FAQ

11.2 Developer-Dokumentation

- Datenbank-Schema (SQL + Diagramm)
- API-Endpoints (Swagger/OpenAPI)
- Plugin-Entwicklung (später)
- Code-Kommentare

---

12. ZEITPLAN (GROB)

Phase 1: MVP

- Datenbankschema implementieren + Migration
- LISTE + EINGABE UI entwickeln
- Testing + Bugfixes

Phase 2: Erweiterungen

- PORTRAIT + BESTAND
- TRANSFER (Import/Export)
- HILFE-System

Phase 3: Netzwerk

- Öffentlicher Server
- Sync-Mechanismus
- Multi-User

---

13. RISIKEN & MITIGATIONEN

Risiko	Wahrscheinlichkeit	Impact	Mitigation
Grav-SQLite-Integration	Niedrig	Hoch	Frühzeitiger Prototyp
Performance bei >50k Pflanzen	Niedrig	Mittel	Indizes + Tests
USB-Stick zu langsam?	Niedrig	Mittel	SSD-Stick empfehlen
User findet Aufklapp-Bereiche verwirrend	Mittel	Mittel	Usability-Tests mit Usern
Etikettendruck-Treiber-Probleme	Hoch	Mittel	Phase 3 verschieben

---

14. ERFOLGS-KRITERIEN

✅ **MVP ist erfolgreich, wenn:**

1. User kann neue Pflanze in <2 Minuten erfassen
2. Suche findet Pflanze in <1 Sekunde
3. Export zu Webshops funktioniert ohne Nachbearbeitung
4. System läuft auf USB-Stick ohne Installation
5. 5+ Gärtnereien interessiert

✅ **Langfristiger Erfolg:**

1. 10+ Gärtnereien nutzen das Tool
2. 50.000+ Pflanzen in zentraler DB
3. Aktive Open-Source-Community
4. Integration in bestehende Bio-Systeme (biomondo, etc.)

---

15. OPEN SOURCE STRATEGIE

15.1 Lizenz

- **MIT** oder **GPL v3** (noch zu entscheiden)
- Kommerzielle Nutzung erlaubt
- Namensnennung erforderlich?

15.2 GitHub Repository

```
omiobio/pflanzendatenbank
├── README.md
├── LICENSE
├── INSTALL.md
├── docs/
│   ├── SCHEMA.md
│   ├── API.md
│   └── CONTRIBUTING.md
├── grav/
│   └── user/
│       ├── themes/pdb/
│       └── plugins/pdb/
├── scripts/
│   ├── import_json.py
│   └── export_csv.py
└── tests/
```

15.3 Community

- Issues für Bug-Reports
- Pull Requests willkommen
- Monatliche Releases
- Changelog führen

---

16. KONTAKT & MAINTAINERS

Project Lead: Peter Müller
Email: peter.mueller@omiobio.ch
GitHub: https://github.com/sambarix/pflanzendatenbank
License: MIT / GPL v3 (TBD)

Contributors:

- Peter Müller (Challenge Owner)
- Claude AI, deepseek AI und andere
- Open Farming Hackday Community

ENDE PFLICHTENHEFT

Version: 1.0
Datum: 09. Januar 2025
Status: ✅ Bereit für Implementierung