**Ein Bild, das Text, Design, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.**

Inhalt

[1. Einleitung 3](#_Toc199917598)

[2. Produktübersicht 3](#_Toc199917599)

[3. Funktionale Anforderungen 4](#_Toc199917600)

[4. Nicht-funktionale Anforderungen 5](#_Toc199917601)

[5. Systemarchitektur 5](#_Toc199917602)

[6. Datenbankmodell 6](#_Toc199917603)

[7. Schnittstellen 6](#_Toc199917604)

[8. Testkonzept 6](#_Toc199917605)

[9. Datenschutz & Sicherheit 6](#_Toc199917606)

[10. Projektmanagement 7](#_Toc199917607)

# 1. Einleitung

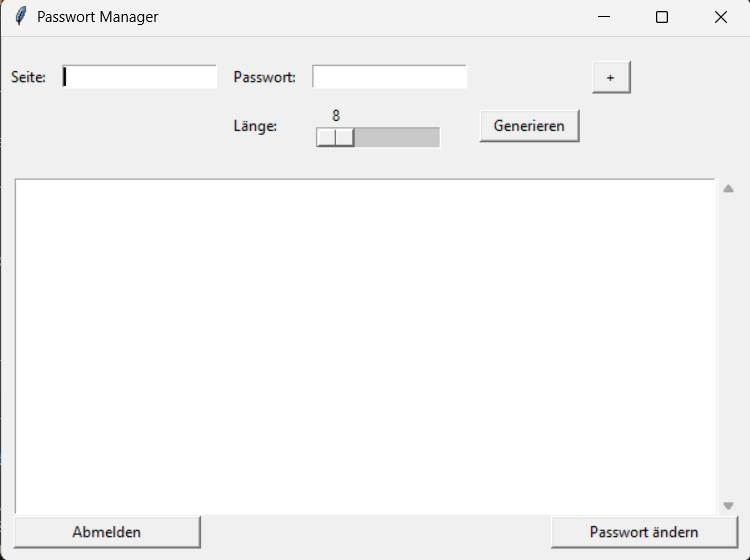
Dieses Pflichtenheft konkretisiert auf Basis des Lastenhefts die Umsetzung des lokalen Passwortmanagers in Python. Es legt fest, wie die Anforderungen technisch umgesetzt werden und bietet einen Plan für die Implementierung, Tests und Dokumentation.

**Projektstart & Kick-Off:**

*Datum Kick-Off: 08.05.2025*  
*Erste Ergebnisse: Pflichtenheft durchgesprochen und verschiedene Branches für die Aufgabenbereiche erstellt.*

# 2. Produktübersicht

* **Desktop-Anwendung (lokal ausführbar)**
* **Datenbankgestützte Verwaltung**
* **Benutzerfreundliche GUI für Passwortmanagement**
* **Keine Cloud-Anbindung**
* **Sicher durch Verschlüsselung & Login**



# 3. Funktionale Anforderungen

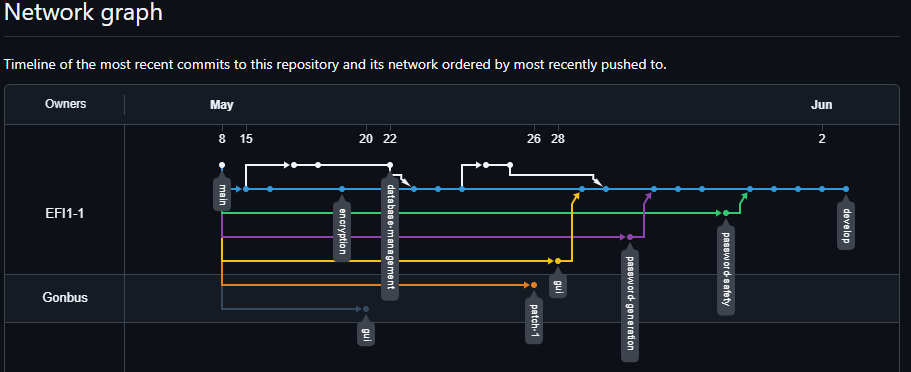
|  |  |  |
| --- | --- | --- |
| **Nr.** | **Anforderung** | **Beschreibung** |
| **F1** | **Login Funktion** | **Benutzer muss sich mit Passwort einloggen.**  **Passwort wird per Hash überprüft.** |
| **F2** | **Datenbankanbindung** | **Verbindung zu MySQL/SQLite, Daten verschlüsselt gespeichert.** |
| **F3** | **Passwortverwaltung** | **Create, Read, Update, Delete von Passworteinträgen.** |
| **F4** | **Passwortgenerator** | **Automatische Erstellung starker Passwörter.** |
| **F5** | **Passwortstärkeprüfung** | **Einstufung der Passwortsicherheit mit Visualisierung.** |

**Umsetzungsstatus (✓/✗):**

* **✓ F1 Login-Funktion:** Benutzer-Login mit Passwort funktioniert. Passwörter werden gehasht (SHA-256) und korrekt geprüft. Login über GUI getestet.
* **✓ F2 DB-Anbindung:** Verbindung zu SQLite erfolgreich. Daten werden AES-verschlüsselt gespeichert und ausgelesen. Lokale Tests ohne Fehler.
* **✓ F3 Verwaltung:** CRUD-Funktionen umgesetzt. Einträge können erstellt, angezeigt, bearbeitet und gelöscht werden. Funktioniert über GUI.
* **✓ F4 Generator:** Passwortgenerator erstellt sichere Passwörter nach wählbaren Kriterien (Länge, Sonderzeichen etc.).
* **✓ F5 Stärke-Check:** Passwortsicherheit wird geprüft (Länge, Zeichenvielfalt) und farblich in der GUI dargestellt.

# 4. Nicht-funktionale Anforderungen

* GUI mit **Tkinter**
* Verschlüsselung mit **AES (cryptography)**
* Testkonzept: **Unittest**
* Codequalität: Dokumentiert, modular, PEP8-konform
* Versionsverwaltung: **GitHub**, mit Branches, Issues, Pull Requests



# 5. Systemarchitektur

[ GUI-Schicht (Tkinter) ]

↓

[ Logik-Schicht (Passwortlogik, Validierung) ]

↓

[ Datenbank-Schicht (pymysql / sqlite3) ]

# 6. Datenbankmodell

****

**Notizfeld – Datenbank-Status:**

* Tabellen erstellt: [**✓** - Erfolgt automatisch]
* Testdatensätze eingefügt? [**✓** - Erfolgt automatisch]
* Zugriff mit Python erfolgreich getestet? [**✓** - Getestet]

# 7. Schnittstellen

|  |  |
| --- | --- |
| Schnittstelle | Beschreibung |
| MySQL/SQLite | Speicherung der Benutzer- und Passwortdaten |
| GUI (Tkinter) | Darstellung und Eingabe über Benutzeroberfläche |
| Dateiexport (JSON) | Export-Funktion verschlüsselter Daten |

# 8. Testkonzept

**Testarten:**

* Unit-Tests: Für Logikfunktionen (z. B. Passwortverschlüsselung)
* Manuelle GUI-Tests: Bedienbarkeit, Fehlerbehandlung
* Integrationstests: Verbindung von GUI ↔ Logik ↔ DB

# 9. Datenschutz & Sicherheit

* Passwörter in der Datenbank sind **AES-verschlüsselt**
* **Keine unverschlüsselte Datenhaltung**
* Lokale Speicherung, keine Internet-/Cloud-Schnittstellen

# 10. Projektmanagement

**Tools & Prozesse:**

* **GitHub Projektverwaltung**
* Branching-Modell: *main, feature/*, *bugfix/*, *release/*
* Aufgabenverteilung nach Rollen
* Agile Arbeitsweise (Sprints oder wöchentliche Statusupdates)

**Meilensteine:**

|  |  |  |  |
| --- | --- | --- | --- |
| Meilenstein | Geplantes Datum | Erreicht am | Notizen |
| Projektstart | 08.05.2025 | 08.05.2025 | Kickoff-„Meeting“ |
| Datenbankmodell fertig | 15.05.2025 | 15.05.2025 | Entwurf & SQL-Datei |
| GUI-Prototyp | 22.05.2025 | 23.05.2025 | Screenshot HERE |
| Funktionale Integration | 22.05.2025 | 03.05.2025 | Erste Testversion |
| Testphase | 29.05.2025 | 03.06.2025 | Protokoll Testfälle |
| Präsentation / Abgabe | 11.06.2025 | 11.06.2025 | PowerPoint-Datei, Lastenheft & Pflichtenheft |