

Pflichtenheft "JustForYou"

Version	Datum	Änderung	Name
0.0	23. Okt. 2024	Inhalt	Paul Dorn
0.1	24. Okt. 2024	Inhalt	Paul Dorn
0.2	5. Dez. 2024	Inhalt	Paul Dorn
0.3	27. Jan. 2025	Inhalt	Paul Dorn

Inhaltsverzeichnis:

1. Einleitung	2
2. Zielsetzung	2
3. Systembeschreibung	2
3.1 Allgemeine Anforderungen	2
3.2 Funktionale Anforderungen	3
3.3 Datenspeicherung und -verarbeitung	3
3.4 Bedienung und Benutzerführung	4
4. Technische Anforderungen	4
4.1 Schnittstellen	4
4.2 Systemleistung	4
5. Entwicklungsumgebung und Werkzeuge	4
6. Systemtechnische Lösungen	5
6.1 Dynamisches Laden von Modulen	5
7. Qualitätssicherung	7
8. Projektmanagement	7
8.1 Projektorganisation	7
8.2 Zeitplan und Kosten	7
8.3 Dokumentation	7

1. Einleitung

Die **Kleinstweich Deutschland GmbH (KWD)** plant die Entwicklung eines modularen Rechnermoduls für das Windows-Betriebssystem, das für verschiedene Branchen anpassbar ist. Ziel ist es, eine Lösung zu entwickeln, die den klassischen Taschenrechner ersetzt und es den Kunden ermöglicht, spezifische Module in ihre Anwendung zu integrieren.

2. Zielsetzung

Das Ziel des Projekts ist die Entwicklung einer **modularisierten Softwarelösung** mit einem konsistenten Bedienkonzept, das die Kombination von maximal drei **branchenbezogenen Funktionsmodulen** ermöglicht. Das Programm soll effizient arbeiten und wenig redundanten Code enthalten.

3. Systembeschreibung

3.1 Allgemeine Anforderungen

- **Betriebssystem:** Die Anwendung ist für Windows zu entwickeln.
- **Modularität:** Die Funktionsmodule werden als unabhängige Python-Module bereitgestellt.
- **Programmiersprache:** Die Entwicklung erfolgt in **Python** (GUI-Framework PySide 6).
- **Benutzeroberfläche:** Die Oberfläche muss nutzerfreundlich und vollständig mit der Maus bedienbar sein. Schriftgröße, -art und Hintergrundfarbe sollen anpassbar sein.
- **Auslieferung:** Als ".exe" zur einfachen Nutzung und Verwendung.

3.2 Funktionale Anforderungen

Die Anwendung besteht aus einem **Grundrechenmodul** und wählbaren **branchen abhängigen Modulen**:

1. **Startmodul**
 - Listet alle durchgeführten Berechnungen mit Eingabewerten und Ergebnissen.
 - Funktionen: Programmstart, Auswahl von Funktionsmodellen, Programmbeendigung, Löschen von Daten und AES-verschlüsselte Speicherung.
2. **Grundrechenarten Modul**
 - Enthält Grundrechenarten (Addition, Subtraktion, Multiplikation, Division).
3. **Prozentrechnung Modul**
 - Funktionen wie: „% dazu“, „% weg“, „% davon“, Brutto-/Nettopreis Berechnungen.
4. **Kredit Rechnungsmodule**
 - Berechnet Kreditraten und Zinsen für verschiedene Kreditmodelle.
5. **Geometrie Modul**
 - Berechnet Umfang und Flächeninhalt von Dreiecken, Kreisen und Parallelogrammen.
6. **Schulmodul**
 - Dient der Zeugnisnoten Berechnung und gibt Empfehlungen basierend auf Notendurchschnitt.
7. **Informationstechnik Modul**
 - Grafikspeicher und Video Dateigröße berechnen
 - Zahlensystem Umrechnungen
 - Umrechnung von Datenmengen
8. **Mathematische Funktionen**
 - Fakultät
 - Quadratwurzel
 - Potenzfunktionen
 - Primzahlen zwischen Grenzwerten
 - Umwandlung Dezimal Bruch

3.3 Datenspeicherung und -verarbeitung

- **Protokollierung:** Alle Berechnungen werden mit Eingabewerten, Ergebnissen und Nebenrechnungen gespeichert.

- **Rundung:** Ergebnisse werden auf sechs signifikante Ziffern gerundet. Finanzmathematische Berechnungen müssen währungs richtig gerundet werden.

3.4 Bedienung und Benutzerführung

- Die Bedienung erfolgt zwangsgeführt durch eine sequentielle Eingabe der Parameter.
- Die Benutzeroberfläche ist so zu gestalten, dass sie mit der Maus vollständig bedienbar ist. Dabei wird eine **universelle Eingabeoberfläche** verwendet, die eine schnelle und fehlerfreie Eingabe sicherstellt.
- **Anpassung:** Der Nutzer kann die Darstellung der Benutzeroberfläche nach seinen Bedürfnissen anpassen.

4. Technische Anforderungen

4.1 Schnittstellen

- **Betriebssystem:** Die Anwendung interagiert über grafische Benutzeroberflächen (GUI) mit dem Betriebssystem.
- **Laufzeitmodule:** Die Module werden als Python Bibliotheken eingebunden. Eine dynamische Anbindung externer Module ist vorgesehen.

4.2 Systemleistung

Die Anwendung muss auf handelsüblichen Office-PCs lauffähig sein. Der **Ressourcenbedarf** ist minimal zu halten.

5. Entwicklungsumgebung und Werkzeuge

- **Entwicklungsbasis:** Die Anwendung wird mit einer **JetBrains-Software** (PyCharm) entwickelt.
- **Versionskontrolle:** Alle Dateien werden auf GitHub verwaltet.

6. Systemtechnische Lösungen

6.1 Dynamisches Laden von Modulen

- Ein Modul besteht aus einer Python-Datei, die eine Haupt-Widget-Klasse implementiert und bestimmte Metadaten exportiert. Diese Metadaten dienen dem Laufzeitsystem zur korrekten Identifikation und Initialisierung des Moduls.
- **Erforderliche Metadaten:**
 - **MODULE_NAME:** Der Name des Moduls, z. B. "Hello World".
 - **MODULE_DESCRIPTION:** Eine Beschreibung des Moduls, z. B. "Just a simple hello world module".
 - **MODULE_MAIN_WINDOW:** Eine Funktion mit dem Namen `create_main_window`, die das Haupt-Widget des Moduls erzeugt.
- Das folgende Beispiel zeigt den Aufbau eines einfachen Moduls:

```
from PySide6.QtWidgets import QWidget, QVBoxLayout, QLabel

class HelloWorldWidget(QWidget):
    def __init__(self):
        super().__init__()

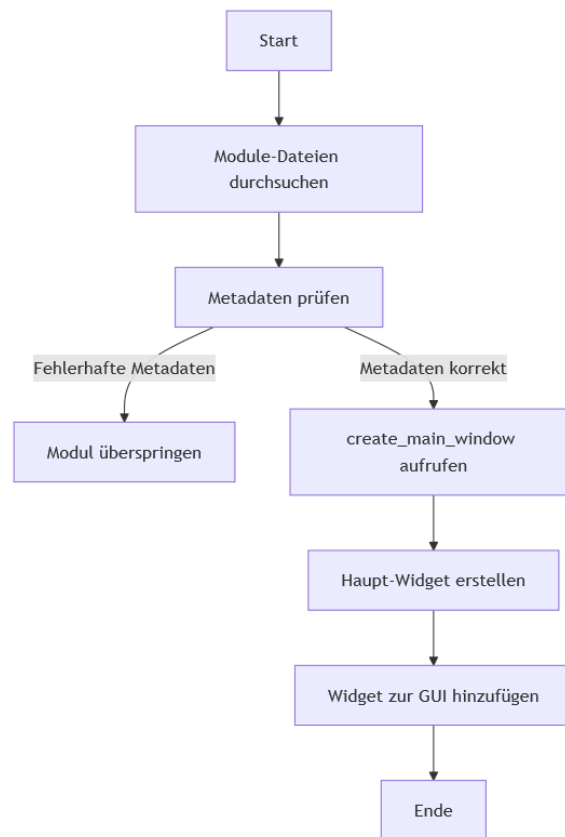
        layout = QVBoxLayout(self)
        label = QLabel('Hello World', self)
        layout.addWidget(label)

# Metadaten des Moduls
MODULE_NAME = 'Hello World'
MODULE_DESCRIPTION = 'Just a simple hello world module'

# Funktion, die das Haupt-Widget erstellt
def create_main_window():
    return HelloWorldWidget()

MODULE_MAIN_WINDOW = create_main_window
```

- Ablauf der Modul Initialisierung:



7. Qualitätssicherung

- **Testmethoden:** Für alle Module sind **Black Box-Tests** zu implementieren. Die Funktionsbibliotheken sind modular aufgebaut und proprietär, um eine optimale Performance zu gewährleisten.
- **Ergonomie:** Die Benutzeroberfläche muss den Richtlinien der **DIN ISO 9241** folgen.

8. Projektmanagement

8.1 Projektorganisation

- **Projektleiter:** Gabriel (im GitHub genannt als "gabrielbzt1")
- **Product Owner:** Paul (im GitHub genannt als "Katze719")
- **Entwickler:** Hugo (im GitHub genannt als "Spaex01"), Till (im GitHub genannt als "TilGP")
- Die Entwicklung erfolgt nach agilen Methoden gemäß **DIN ISO 21500**.

8.2 Zeitplan und Kosten

- **Realisierungszeitraum:** 10/2024 – 03/2025
- **Kostenrahmen:** 30.000 € (Personalkosten)

8.3 Dokumentation

Die Projektdokumentation umfasst den **Projektstrukturplan**, ein **Projekttagbuch** (Siehe GitHub), **Klassendiagramme**, den **Testplan** und das finale **Pflichtenheft**.