

IU Internationale Hochschule – Fernstudium  
Angewandte Künstliche Intelligenz

**PORTFOLIO**  
**Objektorientierte und funktionale Programmierung mit Python**  
**DLBDSOOFPP\_D**

**PHASE 1 - KONZEPTIONSPHASE**

Martin Pichler  
Matrikelnummer: 9212724

## Inhaltsverzeichnis

1.	Abbildungsverzeichnis.....	3
2.	Zielsetzung / Kennzahlen .....	4
2.1.	Zeitliche Zielsetzung.....	4
2.2.	Zu erreichender Notendurchschnitt.....	4
2.3.	Start der Bachelorarbeit.....	4
3.	Dashboard.....	4
3.1.	Grundaufbau des UIs.....	4
3.2.	Inhalte des Dashboards .....	5
4.	UML-Klassendiagramm .....	6
5.	Machbarkeitsüberprüfung .....	7
5.1.	Python Entwicklungsumgebung und Bibliotheken .....	7
5.2.	Datenspeicherung.....	7
5.3.	Verwaltung Quellcode .....	7
5.4.	Benutzer-Interaktion .....	7

## 1. Abbildungsverzeichnis

Abbildung 1: Grundaufbau mit Frames.....	5
Abbildung 2: Dashboard Skizze .....	5
Abbildung 3: UML-Diagramm .....	6

## **2. Zielsetzung / Kennzahlen**

### **2.1. Zeitliche Zielsetzung**

Das erste Ziel, welches mit dem Dashboard überwacht werden soll, ist die zeitliche Zielsetzung. Hier soll das vorgegebene Ende des Studiums mit dem berechneten Ende des Studiums anhand der offenen Module mittels Balkendiagramm verglichen werden. Es wird davon ausgegangen, dass jedes Modul innerhalb von zwei Monaten abgeschlossen wird. Falls das Studium laut Vergleich nicht im vorgegebenen Zeitraum beendet werden wird, soll der Überziehungszeitraum dargestellt werden. Die zeitliche Darstellung erfolgt in Monatsschritten.

### **2.2. Zu erreichender Notendurchschnitt**

Ein weiteres zu überwachendes Ziel ist das Erreichen eines Notendurchschnitts von 2,0. Dies soll in einem Liniendiagramm dargestellt werden. Durch eine grafische Darstellung kann man auf einen Blick erkennen wie sich der Notendurchschnitt über die Zeit entwickeln.

### **2.3. Start der Bachelorarbeit**

Eine für das Studium relevante Kennzahl ist der Zeitpunkt, ab welchem der Student mit der Bachelorarbeit starten darf. Die IU-Internationale Hochschule setzt diesen Zeitpunkt ab dem Erreichen von mindestens 120 Ects Punkten. Diese Kennzahl soll nur als Datum dargestellt werden.

## **3. Dashboard**

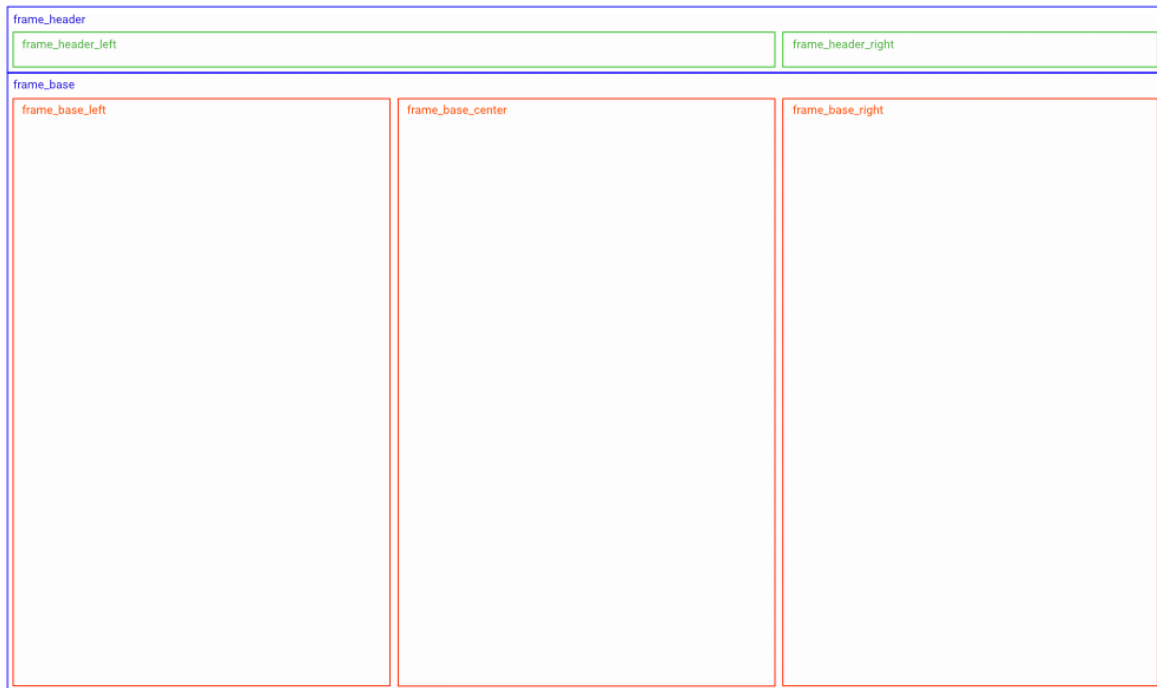
### **3.1. Grundaufbau des UIs**

Die Frames werden verwendet, um als Layouthilfe die Oberfläche in logische Bereiche zu unterteilen und so Widgets wie Tabellen und Diagramme übersichtlich anzuordnen.

Folgende Frames werden verwendet, um Widgets darzustellen:

- `frame_header`: Informationen zum Studenten und Studiengang. Diese Daten können über den Menu-Button eingestellt werden.
- `frame_base_left` und `frame_base_right`: Hier werden die Semester mit den Modulen angezeigt. Diese Daten können über die Auswahl der Zeile durch einen Doppelklick angepasst werden.
- `frame_base_right`: In diesem Bereich werden die Ziele und Kennzahlen dargestellt.

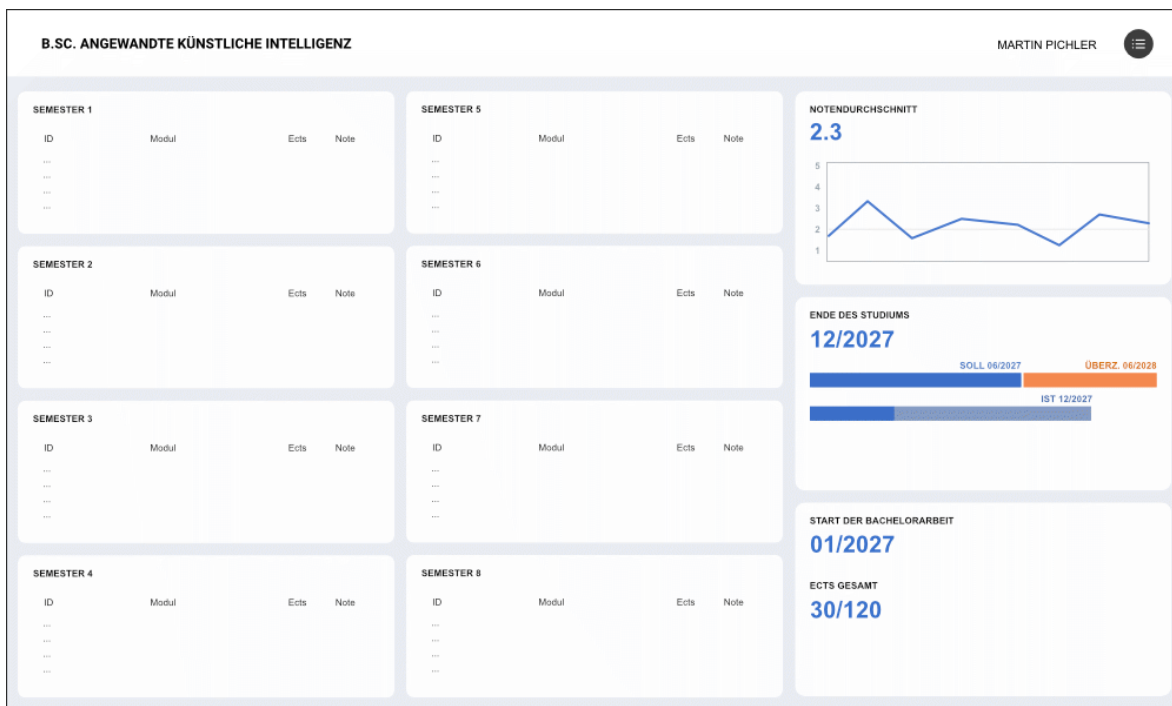
Abbildung 1: Grundaufbau mit Frames



Quelle: Eigene Darstellung, 2025.

### 3.2. Inhalte des Dashboards

Abbildung 2: Dashboard Skizze

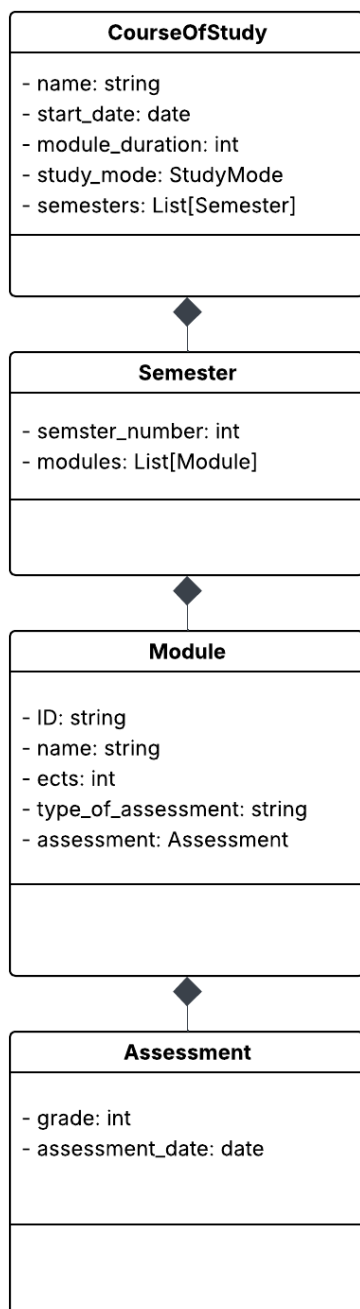


Quelle: Eigene Darstellung, 2025.

## 4. UML-Klassendiagramm

Das UML-Diagramm enthält die Entity-Klassen CourseOfStudy, Semester, Module und Assessment. Diese Klassen haben ein „has-a“ Beziehungsverhältnis, sind somit klassische Beispiele für Komposition. Beispielsweise ist ein Modul Teil eines Semesters und ergibt ohne der Basis Semester keinen Sinn. Komposition ist eine gute Wahl um flexible, wartbare und erweiterbare Software zu entwickeln und fördert eine modulare und entkoppelte Architektur.

Abbildung 3: UML-Diagramm



Quelle: Eigene Darstellung, 2025.

## 5. Machbarkeitsüberprüfung

### 5.1. Python Entwicklungsumgebung und Bibliotheken

Für die Umsetzung des Dashboards wird Visual Studio Code (VS Code) als Entwicklungsumgebung verwendet. VS Code ist ein moderner und erweiterbarer Editor, der sich besonders gut für die Python-Entwicklung eignet und zusätzlich die Möglichkeit bietet JSON-Files zu bearbeiten. Ein weiterer Vorteil ist die direkte Git-Integration zur Verwaltung des Quellcodes. Python wird in der aktuellen Version 3.13.3 angewendet (keine pre-release Version).

Folgende zentrale Python Bibliotheken kommen zum Einsatz:

- **Custom Tkinter:** Ist die erweiterte Standardbibliothek für die Erstellung grafischer Benutzeroberflächen in Python. Besonders gut geeignet ist Custom Tkinter für kleine bis mittlere Desktop-Anwendungen, ist ein Leichtgewicht und lässt sich schnell erlernen.
- **Matplotlib:** Erweitert grafische Benutzeroberflächen um statische, animierte und interaktive Diagramme. Matplotlib lässt sich sehr gute in Tkinter integrieren.
- **Pandas:** Bietet als leistungsfähige Python-Bibliothek die Möglichkeit Daten zu analysieren und zu verarbeiten. Sie bietet Dataframes, die das effiziente Arbeiten mit tabellarischen Daten ermöglicht.

### 5.2. Datenspeicherung

Da die zu speichernde Datenmenge des Dashboards überschaubar ist bietet sich die Speicherung in ein JSON-Format an. Hier können die Daten in Form von Objekten einfach, sicher und strukturiert abgelegt werden. Um Datenverlust zu vermeiden, wird bei jedem Speichervorgang eine Kopie der Daten in die OneDrive-Cloud gespeichert.

### 5.3. Verwaltung Quellcode

Zur Verwaltung des Quellcodes wird das Versionskontrollsystem Git, in Verbindung mit der Hosting-Plattform GitHub, verwendet. Dadurch können Änderungen im Entwicklungsverlauf nachvollzogen und frühere Versionen wiederhergestellt werden.

### 5.4. Benutzer-Interaktion

Die Interaktion mit dem Benutzer erfolgt über eine grafische Benutzeroberfläche, da neue Prüfungsleistungen deutlich einfacher eingefügt und die Ziele mittels Diagramme besser interpretiert werden können.