

# POS+DBI Projekt: Anforderungen

- 1. Organisatorisches
- 2. Technischer Inhalt
  - 2.1. POS (Frontend)
  - 2.2. DBI (Backend)
- 3. Benotung
- 4. Ablauf
  - 4.1. Team + Projektidee
  - 4.2. Planungsphase
  - 4.3. Umsetzungsphase
  - 4.4. Präsentation/Abgabe

Anbei findet ihr die Anforderungen an das Projekt. Die genaue Timeline werden wir noch klären.

## 1. Organisatorisches

---

- Idealerweise 2er Teams
- Arbeit teils im Unterricht, teils zuhause

### Timeline

- 25.04. - Kick off
- 19.05. - Zwischenpräsentation (5 min)
- 16.06. - Endpräsentation (10 min)
- 18.06. - Endpräsentation + Endabgabe

## 2. Technischer Inhalt

---

Nachfolgend findet ihr Mindestanforderungen für das Projekt. Bei Unklarheiten, wendet euch bitte immer an die entsprechenden Lehrpersonen.

### 2.1. POS (Frontend)

---

- aktive Verwendung von GIT
- Projektplanung per Github
  - Github zum Tracken der Arbeitspakete
  - Kanban Board mit Tickets -> Zuweisung der Arbeiten
- Klassendiagramme **vor** dem Start der Programmierung
- Grafische Anwendung
  - mind. 3 Fenster/Pages oÄ
- Vererbung; abstrakte Klassen
- Interfaces
- API Dokumentation
- Unit Tests
- Logging
- Bonus: Parallele Programmierung (Threads, Tasks, ...)

## 2.2. DBI (Backend)

---

- Mehrere Tabellen (min. 3)
  - mind. 3. Normalform
  - Queries mit Select, Joins, Aggregation
  - Schreibender und lesender Zugriff
- mind. 2 Rollen (admin, user) mit entsprechenden Rechten
- SQL Datenbank (kein SQLite) - MariaDB
  - min. lokale Datenbank
  - Optional: Datenbank in der Cloud (zB [Cloudflare](#))
- REST Interface (Backend: Python/C#, Swagger)
- API Dokumentation (Swagger) + JSON Schema für Objekte
- Unit Tests
- Logging

## 3. Benotung

---

- Erfüllung der Grundanforderungen => max. 3er möglich
- Einflussfaktoren für bessere Noten
  - Komplexität des Projekts
  - Umsetzung des Projekts (Feinschliff, keine Bugs, ist die Anwendung intuitiv)
  - Wie gut wurde die Dokumentation umgesetzt
  - Gesamteindruck

### 📌 Wichtig

**KI Richtlinie:** In den Projekten könnt und werdet ihr KI Hilfstools einsetzen. Es gelten aber folgende Regeln.

- Erstellt nicht komplette Klassen mit KI Hilfe. Lasst euch nur bei Teilen helfen
- Methoden oder Teilalgorithmen die mit KI erstellt wurden, müssen entsprechend gekennzeichnet (Schlagwort: **prompt**) werden:

```
// prompt: Schreibe mir eine Funktion um zwei Zahlen zu summieren  
public void Sum(int a, int b) { ... }
```

- KI erstellte Methoden müsst ihr erklären können.
  - Es kann sein, dass wir euch hierzu entsprechend Fragen stellen.
  - Könnt ihr diese teile nicht ausreichend erklären, fließt dies negativ in die Benotung ein

## 4. Ablauf

---

- Team suchen und Projektidee ausarbeiten
- Planungsphase
- Umsetzungsphase
- Präsentation und Abgabe

### ① Hinweis

Wenn wir in der Unterrichtszeit an den Projekten arbeiten, könnt ihr entsprechend Fragen stellen.

### 📌 Wichtig

Wenn ihr Grafiken, Sounds, etc. verwendet, achtet darauf, dass sie frei verwendbar sind (z.B.: Creative Commons Lizenz). In der Dokumentation sollen auch die Quellen (Referenz + Lizenz) genannt werden. Oder erstellt die Grafiken eigenständig.

### 4.1. Team + Projektidee

---

Jedes Team überlegt sich eine Projektidee. Dabei sollen bereits folgende Überlegungen in einem Dokument zusammengefasst werden.

- Wie werden die Mindestanforderungen umgesetzt?
- Welche Features sind ein muss
- Welche Features sind Erweiterungen (nice-to-have), wenn genügend Zeit bleibt
- Wie möchten wir das Ganze grob umsetzen

Abgabeform ist hierbei ein kurzes Dokument. Nach der Freigabe könnt ihr mit der nächsten Phase starten.

### 4.2. Planungsphase

---

Ihr dürft mit der Umsetzung erst beginnen, wenn wir eure Planung durchbesprochen haben. Plant hier folgende Dinge (beispielsweise als Skizze am Blatt):

- Aufbau der GUI (Skizzen):
  - Welche Fenster wird es geben?
  - Wie sehen diese grob aus?
  - Wie ist die Benutzernavigation geplant?
- Aufbau des Programms/der Datenbank
  - Entsprechende UML Diagramme sind verpflichtend
  - Wie arbeiten die Systeme miteinander
- Ablauf planen
  - Erstellt mit Hilfe von Github ein [Projekt](#) mit Kanban Board
  - Erstellt entsprechende Tasks für die Aufgaben
    - Diese werdet ihr dann im Laufe des Projekts noch verfeinern
    - Weißt die Tasks zu (Verantwortlichkeiten)
    - Plant die Tasks auch zeitlich ein

### 4.3. Umsetzungsphase

---

In dieser Zeit arbeitet ihr an euren Projekten. Erweitert regelmäßig die Dokumentation und führt ein Projekttagbuch.

Mögliche Stolpersteine und Lösungen dazu sollen dokumentiert werden.

### 4.4. Präsentation/Abgabe

---

Den genauen Ablauf der Präsentationen werden wir kurz vor Ende besprechen. Die Abgabeform ist wie folgt.

- Dokumentation als Markdown und PDF im Repository
  - Projekttagbuch: Wer hat wann an was gearbeitet
  - Projektplanung (Lastenheft)
  - Umsetzungsdetails (Pflichtenheft)
    - Welche Softwarevoraussetzungen werden benötigt (mit Versionen)
    - Funktionsblöcke bzw. Architektur
    - Detaillierte Beschreibung der Umsetzung
    - Mögliche Probleme und ihre Lösung
  - Wie wurde die Software getestet?
  - Bedienungsanleitung mit Screenshots
  - Quellen für verwendete Bilder, oder andere Medien
- Projektordner als -.zip-Archiv

Abgabe per Repository

- `POS1_3HIF_[Projektname]`
  - `doc` ... Dokumentation (pdf, md)
  - `src` ... Komplettes VisualStudio Projekt
  - `bin` ... Kompiliertes Programm (`*.exe`) mit allen benötigten Abhängigkeiten (Bilder, andere `*.dlls`)