



# Alternativprojekt

Softwaredesign (MECH-B-3-MDT-MET-ILV)

MCI 4

Bachelor - Mechatronics, Design and Innovation

3<sup>rd</sup> semester

Lektor: Dr. rer. pol. Julian Huber

Gruppe: MECH-B-3-SWD-SWD-ILV

Autor: Josef Obwaller, Sandro Streicher

29. Februar 2024

# Inhaltsverzeichnis

<b>1 Idee</b>	<b>1</b>
1.1 Basisanforderungen . . . . .	1
1.2 Erweiterungsmöglichkeiten . . . . .	1
<b>2 Dokumentation</b>	<b>2</b>
2.1 Login . . . . .	2
2.2 Defaultpage . . . . .	2
2.3 Dokumentation . . . . .	2
2.4 Bücher . . . . .	2
2.5 Schüler:innen . . . . .	2
2.6 Gruppen . . . . .	3
2.7 Benotung . . . . .	3
2.8 Einstellungen . . . . .	4
<b>3 Key-Points</b>	<b>5</b>
3.1 Aufbau Datenbank . . . . .	5
3.2 Product-Keys . . . . .	6
3.3 Benutzererstellung . . . . .	6
3.4 Exporte . . . . .	6
3.5 Entscheidung gegen Hosting . . . . .	6
3.6 Verschlüsselung . . . . .	6
3.7 Backup . . . . .	6
3.8 Reload einer Datenbank . . . . .	6
3.9 Benotungsdarstellungen . . . . .	7
3.10 Endnotenberechnung . . . . .	7
3.11 Installation . . . . .	7
<b>4 Fazit</b>	<b>8</b>
<b>5 Quellen zum Programmieren</b>	<b>8</b>
<b>Abbildungsverzeichnis</b>	<b>III</b>

# 1 Idee

## 1.1 BASISANFORDERUNGEN

- Software für Schüler-Notenverwaltung für eine Volksschule
- mit Python und Streamlit
- pro Software nur eine zugriffsberechtigte Person, wie bei der Case Study
- Lehrer:innen können Schüler anlegen mit Name, Geb.-Datum, Schulstufe, etc. Schulstufe wird automatisch jedes Jahr erhöht
- Schüler werden in Datenbank gespeichert
- Lehrer:innen können Schüler:innen zu Gruppen zuteilen
- Gruppen können erstellt und verändert werden
- Hausaufgaben können erstellt werden (Aufgabenname, Kommentar, Gewichtung) und Gruppen oder Schülern zugeteilt werden
- HÜs können benotet werden (Note und Kommentar, Datum und Zeitpunkt)
- Noten und Kommentare sind in der SQL Datenbank gespeichert
- Übersichtsseite, wo Notenverlauf von jedem Schüler individuell angezeigt wird, Möglichkeit zum Export evt. als Excel
- Lehrer:innen können verschiedene Fächer anlegen
- Hosten der Website und damit Zugang ermöglichen, Idee: Install.bat (klont Repository, erstellt venv ...), Start.bat (startet Anwendung)
- Visualisierung der HÜ Ergebnisse als Gauss

## 1.2 ERWEITERUNGSMÖGLICHKEITEN

- Admin script die Endjahresnoten zusammenführen
- Verschlüsselung der Daten und damit sichere Ablage auf der Datenbank
- Zugang nur mit Benutzername und Passwort
- Endjahresnoten als Export
- Datenbank kann als gesamtes in geeignetem Datenformat exportiert und separat abgespeichert werden. Dient der Datensicherung. Kann auch wieder hineingeladen werden.
- Automatische Backups auf Server oder per Mail
- Deployment in Docker Container

## 2 Dokumentation

### 2.1 LOGIN

- Beim Erststart wird zur Freischaltung ein Product-Key benötigt. Außerdem wird der eingegebene Benutzer und Passwort in der Datenbank verschlüsselt abgelegt. Bei künftigen Anmeldungen ist der Benutzername und das Passwort einzugeben.

### 2.2 DEFAULTPAGE

Diese wird nach dem erfolgreichen Login angezeigt. Es sind einige Information zur Software zu sehen.

### 2.3 DOKUMENTATION

Folgende Punkte sind hier aufgelistet:

- Haftungsausschluss
- Allgemeine Informationen zum Handling

### 2.4 BÜCHER

An der Zielschule werden anstatt herkömmlicher Fächer Bücher benotet. In diesem Reiter können Bücher angelegt und bearbeitet werden. Folgende Anwendungen wurden implementiert:

#### Anlegen von Büchern:

- Name
- Seitenanzahl
- Autonummerierung
- Speichern in der Datenbank

#### Bearbeiten von Büchern:

- Auswahl über Dropdown
- Änderung des Namens, der Seitenanzahl und der Autonummerierung
- Speicherung der Änderung
- Löschen des Buches

### 2.5 SCHÜLER:INNEN

In diesem Reiter können Notenübersichten angezeigt werden, Schüler:innen angelegt und bearbeitet werden.

#### Übersicht

- Auswahl des Kindes über Dropdown
- Visualisierung durch Linechart der vergebenen Noten jedes zugewiesenen Buches

- Export des Linecharts
- Notizen zum Schüler
- Speicherung der Notizen

#### **Anlegen**

- Eingabe von Vor- Nachname mit anschließender Speicherung

#### **Bearbeiten**

- Dropdown zur Auswahl des Kindes
- Änderung von Vor- und Nachname
- Speichern bzw. Löschen des Kindes

## **2.6 GRUPPEN**

In diesem Reiter werden die Gruppen angezeigt, angelegt und bearbeitet.

#### **Teilnehmer**

- Gruppe kann mittels Dropdown ausgewählt werden.
- Die zugeteilten Kinder der Gruppe und die Kinder ohne Gruppe werden darunter angezeigt.
- Durch Checkboxes können die Kinder entfernt bzw. hinzugefügt werden.
- Die Auswahl kann gespeichert werden.

#### **Anlegen**

- Gruppenname wird gespeichert.

#### **Bearbeiten**

- Durch Dropdown kann die Gruppe ausgewählt werden.
- Gruppennamen kann geändert werden..
- Speichern bzw. Löschen der Gruppe ist möglich wenn kein Kind mehr in der Gruppe ist.

## **2.7 BENOTUNG**

In diesem Reiter werden die Noten eingetragen und können angesehen werden.

#### **Übersicht**

- Buchauswahl über Dropdownmenü
- Für das ausgewählte Buch wird eine Benotungsübersicht angezeigt:  
Für jede Gruppe wird eine Tabelle geladen. In dieser Tabelle sind die Noten der Kinder für jede Seite zu sehen. Jede Reihe ist ein Kind der Gruppe und jede Spalte ist eine Seite des Buches.
- Von jeder Gruppentabelle kann ein Export in csv-Format durchgeführt werden.

- Am Seitenende wird eine Gesamtübersicht des Buches angezeigt. Hier sind alle Kinder in einer Tabelle eingetragen, jedoch symbolisiert die erste Spalte den Gruppennamen.
- Diese Gesamtübersicht kann anschließend als csv- Datei oder im pdf-Format exportiert werden.
- Die Endnoten der Schüler und Schülerinnen können als Pdf exportiert werden.

### **Detailansicht**

- Es wird zunächst das Buch, die Gruppe und das Kind ausgewählt.
- Als Seitenzahl kann entweder die Zahl hingeschrieben werden oder ein eigener Aufgabentitel.
- Wird ein eigener Titel vergeben muss dieser per Entertaste und Button der Datenbank hinzugefügt werden.
- Es werden alle Details zum Assignment geladen.
- Es können nun alle Eingaben zur Note getätigt werden.
- Folgende Dinge können eingegeben werden: Note, Kommentar, Gewichtung, Aufgabenbeschreibung und Datum.
- Wird der Krankheits-toggle betätigt wird ein K als Note und als Gewichtung 999 gespeichert.
- Die Note kann entweder gespeichert oder gelöscht werden.
- Wurde die Note bearbeitet wird die Eingabe nur aktualisiert.

## **2.8 EINSTELLUNGEN**

In diesem Reiter können Einstellungen bezüglich des Backups und des Ladens der Datenbank getätigt werden.

### **Backups erstellen**

- Hier kann der Ort für das Datenbank-Backup geändert werden.
- Default-Ordner ist der Desktop.
- Es kann jederzeit ein manuelles Backup der Datenbank erstellt werden.
- Es werden immer nur die letzten 10 Backups behalten.

### **Datenbank laden**

- Hier kann eine schon bestehende Datenbank geladen werden.
- Diese Datenbank überschreibt die bestehende.
- Wird eine Datenbank hinein geladen wird von der aktuellen Version noch ein Backup erstellt, so dass keine Daten verloren gehen.

## 3 Key-Points

### 3.1 AUFBAU DATENBANK

Bei der Erarbeitung des Programms wurde der Aufbau der Datenbank mehrmals stark modifiziert. Es wurden Tabellen umgeschrieben, andere Datentypen vergeben, Tabellen hinzugefügt und komplette Teile wieder entfernt. Der finale Aufbau der Datenbank sieht wie folgt aus:

Name	Typ	Schema
<b>Assignment</b>		CREATE TABLE "Assignment" ( "bookID" INTEGER, "assignmentID" INTEGER NOT NULL, "name" TEXT NOT NULL, "description" TEXT, PRIMARY KEY("assignmentID" AUTOINCREMENT) )
bookID	INTEGER	"bookID" INTEGER
assignmentID	INTEGER	"assignmentID" INTEGER NOT NULL
name	TEXT	"name" TEXT NOT NULL
description	TEXT	"description" TEXT
<b>Book</b>		CREATE TABLE Book ( bookID INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT NOT NULL, pages INTEGER NOT NULL, autonumbering INTEGER NOT NULL CHECK (autonumbering IN {0, 1}) )
bookID	INTEGER	"bookID" INTEGER
name	TEXT	"name" TEXT NOT NULL
pages	INTEGER	"pages" INTEGER NOT NULL
autonumbering	INTEGER	"autonumbering" INTEGER NOT NULL CHECK("autonumbering" IN {0, 1})
<b>Grade</b>		CREATE TABLE "Grade" ( "gradeID" INTEGER, "kidID" INTEGER NOT NULL, "bookID" INTEGER NOT NULL, "assignmentID" INTEGER NOT NULL, "grade" TEXT NOT NULL, "weight" INTEGER NOT NULL, "comment" TEXT NOT NULL, "date" DATE NOT NULL, PRIMARY KEY("gradeID" AUTOINCREMENT) )
gradeID	INTEGER	"gradeID" INTEGER
kidID	INTEGER	"kidID" INTEGER NOT NULL
bookID	INTEGER	"bookID" INTEGER NOT NULL
assignmentID	INTEGER	"assignmentID" INTEGER NOT NULL
grade	TEXT	"grade" TEXT NOT NULL
weight	INTEGER	"weight" INTEGER NOT NULL
comment	TEXT	"comment" TEXT NOT NULL
date	DATE	"date" DATE NOT NULL
<b>Group</b>		CREATE TABLE "Group" ( "groupID" INTEGER PRIMARY KEY AUTOINCREMENT, "groupname" TEXT NOT NULL )
groupID	INTEGER	"groupID" INTEGER
groupname	TEXT	"groupname" TEXT NOT NULL
<b>Kid</b>		CREATE TABLE "Kid" ( "kidID" INTEGER, "firstname" TEXT NOT NULL, "lastname" TEXT NOT NULL, "groupID" INTEGER, "textfield" TEXT, PRIMARY KEY("kidID" AUTOINCREMENT) )
kidID	INTEGER	"kidID" INTEGER
firstname	TEXT	"firstname" TEXT NOT NULL
lastname	TEXT	"lastname" TEXT NOT NULL
groupID	INTEGER	"groupID" INTEGER
textfield	TEXT	"textfield" TEXT
<b>Session</b>		CREATE TABLE "Session" ( "activeUser" TEXT, "token" TEXT )
activeUser	TEXT	"activeUser" TEXT
token	TEXT	"token" TEXT
<b>TokenHash</b>		CREATE TABLE "TokenHash" ( "hashedToken" TEXT )
hashedToken	TEXT	"hashedToken" TEXT
<b>sqlite_sequence</b>		CREATE TABLE sqlite_sequence(name,seq)
name		"name"
seq		"seq"
<b>userData</b>		CREATE TABLE "userData" ( "username" TEXT NOT NULL UNIQUE, "password" TEXT NOT NULL, "storeBackup" TEXT, PRIMARY KEY("username") )
username	TEXT	"username" TEXT NOT NULL UNIQUE
password	TEXT	"password" TEXT NOT NULL
storeBackup	TEXT	"storeBackup" TEXT

Abbildung 3.1: Überblick der Datenbank

## 3.2 PRODUCT-KEYS

Um zu garantieren, dass die Software nicht unbefugt benutzt wird, wurden Product-Keys implementiert. Es wurden 100 Product-Keys erstellt, welche lokal bei uns am Rechner gespeichert sind. Beim Erststart muss ein gültiger Product-Key eingegeben werden, um die Software zu starten. Wurde das erfolgreich gemacht wird der Key in der Datenbank gespeichert, so dass bei erneutem Starten kein Key mehr nötig ist.

Die Lektoren erhalten folgende Product-Keys:

Julian Huber: SEStp7UTPN

Matthias Panny: caMVpP0hV0

## 3.3 BENUTZERERSTELLUNG

Beim Erststart wird der Benutzer und das Passwort selber generiert. Neben der Eingabe des Product-Keys erfolgt die Ersteingabe des Users. Bei erfolgreichem Eingeben des Product-Keys wird der gewählte Benutzer und das zugehörige Passwort gehasht in der Datenbank abgespeichert. Dies ist nun der Benutzer dieser Anwendung.

## 3.4 EXPORTE

Exporte sind an mehreren verschiedenen Knotenpunkten der Software möglich. Die Exporte können im CSV- und PDF-Format gemacht werden.

## 3.5 ENTSCHEIDUNG GEGEN HOSTING

Um sicherzustellen, dass die Software in der Praxis effektiv eingesetzt werden kann, wurde eine enge Kooperation mit der Zielschule angestrebt. Aufgrund der spezifischen Anforderungen vor Ort erschien es unnötig und zu komplex, die Datenbank zu hosten. Stattdessen erhält jeder Lehrer sein eigenes virtuelles Notenbuch, das einem physischen Notenbuch entspricht. Ein Zugriff auf die Notizen anderer Lehrer ist nicht erforderlich, weshalb die Möglichkeit des Hostings ausgeschlossen wurde. Bisher führt auch jede Lehrperson seine eigenen Aufzeichnungen und speichern diese Lokal auf ihrem PC.

## 3.6 VERSCHLÜSSELUNG

Prinzipiell wurden die Daten nicht verschlüsselt. Passwörter oder Tokens werden gehasht abgespeichert. Derzeit werden die Noten auch nicht verschlüsselt und da sie nur lokal am PC gespeichert bleiben wurde auf eine verschlüsselte Datenablage verzichtet.

## 3.7 BACKUP

Es wurde eine Backup-Funktion integriert, die auf zwei Arten arbeitet. Zum einen werden automatisch Backups in einen dedizierten Ordner generiert. Zum anderen besteht die Möglichkeit, manuelle Backups zu erstellen. Die Art des Backups wird im Dateinamen durch ein großes "A" für automatisch oder ein großes "M" für manuell gekennzeichnet.

## 3.8 RELOAD EINER DATENBANK

Die Software ermöglicht das Laden einer Backup-Version der Datenbank. Bei diesem Vorgang wird automatisch eine Sicherungskopie der aktuellen Version erstellt, bevor diese überschrieben wird.



## 3.9 BENOTUNGSDARSTELLUNGEN

Es wurde in mehreren Bereichen der Software Visualisierungen erstellt. Einerseits wird wie in herkömmlichen Notenbüchern eine Tabelle der Noten angezeigt. Es ist jedoch auch möglich alle Noten eines Schülers mittels Tabelle bzw. Linecharts zu visualisieren und im Anschluss auch zu exportieren.

## 3.10 ENDNOTENBERECHNUNG

In verschiedenen Bereichen der Software wurden Visualisierungen implementiert. Einerseits wird eine Tabelle der Noten dargestellt, die der Struktur herkömmlicher Notenbücher folgt. Andererseits besteht die Möglichkeit, sämtliche Noten eines Schülers mithilfe von Tabellen und Liniendiagrammen zu visualisieren. Zusätzlich kann diese visualisierte Datenansicht exportiert werden.

## 3.11 INSTALLATION

Es wurde eine einfache Installation durch ein executable angestrebt. Es wird nur die Installation.exe Datei benötigt, um die Anwendung auf den PC zu installieren. Diese muss unbedingt als Admin ausgeführt werden. Es wird lediglich nach einem Pfad gefragt an dem das Programm installiert werden soll - dies ist die einzige Eingabe, welche die Person tätigen muss. Folgende Schritte werden von der Anwendung durchgeführt:

- Die ExecutionPolicy wird auf RemoteSigned gesetzt.
- Python wird installiert.
- Git wird installiert.
- Das Repository wird geklont.
- Es wird ein virtuelles Environment erstellt.
- Das virtuelle Environment wird aktiviert.
- Es werden alle benötigten Pakete installiert.
- Es wird ein Symlink namens ObSt.lnk auf den Desktop von der Run.exe erstellt.
- Es wird eine config.txt erstellt. In diesem Textfile sind alle durchgeführten Änderungen dokumentiert.

Die Installation ist nun komplett. Wird nun die Verknüpfung auf dem Desktop als Admin ausgeführt startet die Anwendung:

- Es wird zunächst der Verknüpfungspfad geladen, um im richtigen Working Directory zu sein.
- Es wird die venv aktiviert.
- Es wird streamlit gestartet.
- Das Programm öffnet sich im Browser und ist vollkommen funktionsfähig.

## 4 Fazit

Die Erarbeitung dieser Software war in vielerlei Hinsicht sehr lehrreich:

- Jede Iteration der Software hatte neue Challenges geboten. Je komplexer das Programm wurde desto detaillierter wurden die Fehlermeldungen, welche wiederum gefunden und ausgemärzt werden mussten.
- Die Abstimmung des UI war in enger Zusammenarbeit mit den Lehrpersonen der Zielschule, so dass wir das bestmögliche Erlebnis bieten können. Damit es zu keiner Verwirrung beim Programm kommt wurde das Handling so einfach wie möglich gestaltet.
- Da die Software und die Datenbank sehr eng miteinander verstrickt sind erforderten gewisse Probleme lange Abstimmungen, um auf eine geeignete Lösung zu kommen.
- Dieses Benotungssystem wirkte zunächst sehr irritierend und unkonventionell.
- Die Anforderungen an die Software wurden laufend angepasst und entsprechen somit nicht mehr denen, die mit den Lektoren besprochen wurden. Manche Implementierungen (wie bsp. Schüler automatisch einlesen) wurden nicht durchgeführt, da diese von den Lehrpersonen (= unseren Kunden) nicht gewünscht wurden. Unser Ziel war es, eine möglichst gute Lösung für unseren Kund\*innen zu erstellen.

## 5 Quellen zum Programmieren

Folgende Litertur und Tools wurden zur Erarbeitung des Programms verwendet:

- Stack Overflow
- ChatGPT
- Github
- GitHub CoPilot
- Diverse Dokumentationen von Paketen

# Abbildungsverzeichnis

3.1 Überblick der Datenbank . . . . .	5
---------------------------------------	---