

Berufliche Schule Elmshorn Europaschule

Schuljahr: 2024/06 (2 Halbjahr)

Lernfeld 10 Benutzerschnittstellen gestalten und entwickeln

## **Entwicklung einer SQL-Lern-Anwendung mit direktem Feedback**

Bearbeitungszeitraum: 10.05.2024 bis 30.06.2024

Abgabedatum: 30.06.2024

Verfasser:

Koschinski, Ken Gerrit (ITA22 /Fachinformatiker für Anwendungsentwicklung)

Bull, Léon (ITA22 /Fachinformatiker für Anwendungsentwicklung)

Opitz, Mika (ITA22 /Fachinformatiker für Anwendungsentwicklung)

Autor	Abschnitte	Seiten
Koschinski, Ken Gerrit	Einleitung,, Grundlegender Aufbau, Schluss	S.1-4  S.I - IV
Bull, Léon	Einrichtung Setup	S.1
Opitz, Mika	Datenbank	S.3

## Inhalt

1.	Einleitung .....	1
2.	Hauptteil.....	1
2.1.	Grundlegender Aufbau .....	1
2.2.	Einrichtung (SETUP) .....	1
2.2.1.	Abhängigkeiten.....	1
2.2.2.	Flutter Installation .....	2
2.3.	Erweiterbarkeit .....	2
2.4.	Erstellen neuer Seiten .....	2
2.5.	Datenbank.....	2
2.5.1.	Datenbank Zugriff .....	2
2.5.2.	Datenbank wiederherstellen .....	3
2.5.3.	User Verwaltung .....	3
2.5.4.	MySQL Rechte Konfiguration .....	3
2.5.5.	Datenbank Verbindung Programm .....	4
3.	Schluss .....	4
I.	Abbildungsverzeichnis .....	IV

## **1. Einleitung**

In Rahmen des Schulischen Ausbildungsteiles, wurde ein Auftrag erteilt eine Webanwendung zu schreiben, die es Nutzern erlaubt SQL-anweisungen zu üben. Die Nutzergruppe, die mit diesem Programm erreicht werden soll, sind Schüler, Lehrende und alle, die einen blick in SQL wagen möchten. Fokus in diesem Teil ist es, das User sich anmelden und beginnen können SQL abfragen zu schreiben. Für die zeitliche Einteilung wurde ein Gant-Diagramm erstellt. (Siehe Abbildung 3)

## **2. Hauptteil**

### **2.1. Grundlegender Aufbau**

Das Tool basiert auf dem Flutter-Framework und wurde in der Programmiersprache Dart entwickelt. Für die Datenerhaltung kommt eine MySQL-Datenbank zum Einsatz. Dabei wird die Sakila Datenbank, für die Übungen verwendet. Account Daten werden in einer separaten Datenbank gespeichert. Zwar bietet Flutter die Möglichkeit auf verschiedenen Betriebssystemen zu laufen, jedoch wurde sich hier auf Windows spezialisiert.

### **2.2. Einrichtung (SETUP)**

Damit User und Entwickler mit diesem Programm arbeiten können, wird das Programm MySQL vorausgesetzt. Zum Einfügen der Sakila Datenbank und der Users Datenbank, die im Projekt angehängen Projektdartein Importieren. Dafür kann die MySQL workbranch (siehe Abbildung 4) oder die Konsole benutzt werden. Die Verbindung erfolgt im Localhoast: 3306.

#### **2.2.1. Abhängigkeiten**

Das Framework Flutter muss heruntergeladen und in der Entwicklungsumgebung der SDK-Pfad angegeben werden. In Visual Studio Code müssen die Erweiterungen Dart und Flutter installiert werden.

### **2.2.2. Flutter Installation**

Google, die das Framework zu Verfügung stellen, bieten eine umfassende Anleitung zur Installation von Flutter. Zum bisherigen stand wurde Flutter für Web heruntergeladen<sup>1</sup>.

### **2.3. Erweiterbarkeit**

Das Programm wurde flexible geschrieben, sodass eine einfache Erweiterung möglich ist. Um eine neue Seite hinzuzufügen reicht es, eine neue Dart Datei hinzuzufügen und über in der side\_bar.dart eine Verlinkung herzustellen.

### **2.4. Erstellen neuer Seiten**

Um Bei den Neuen Seiten in der Aktuellen Strucktur zu bleiben, sollte für jedes Widget eine neue dart angelegt werden. Diese muss lediglich in die Angezeigte dart Datei eingeführt werden. (Siehe Abbildung 1). Die Angezeigte dart wird dann in die sidebar gepackt und von dort aus erreicht (siehe Abbildung 2).

### **2.5. Datenbank**

Als Oberfläche um die Datenbank zu Verwalten wurde die MySQL-Workbench, welche unter anderem auch die Rechte Verwaltung sowie das Anlegen neuer User oder Schemata erleichtert genutzt.

#### **2.5.1. Datenbank Zugriff**

Um auf die Datenbank zuzugreifen haben wird eine Methode verwendet, die mithilfe der mitgegebenen Parameter (z.b. Host, User und Passwort) Eine Verbindung zur Datenbank herstellt. Diese Methode kann jederzeit aufgerufen werden, und die resultierende Verbindung kann dann genutzt werden.

---

<sup>1</sup> <https://docs.flutter.dev/get-started/install>

### 2.5.2. Datenbank wiederherstellen

Es gibt eine Methode, um die Datenbank zu zurückzusetzen. Dies ist erforderlich, nachdem Ein User die Übungen abgeschlossen hat. Dabei werden neue Datensätze hinzugefügt oder alte gelöscht. Diese Methode leert alle Tabellen und füllt sie anschließend wieder auf.

### 2.5.3. User Verwaltung

Für die Datenbank wurden zwei User erstellt. **root** (Benutzername: root, Passwort: root) – Dieser Benutzer hat alle Rechte und kann daher die Struktur der Datenbank verändern sowie Daten oder Tabellen entfernen und hinzufügen. **ProgrammUser** (Benutzername: ProgrammUser, Passwort: ProgrammUser) – Dieser Benutzer darf nur einfache Abfragen ausführen, die die Struktur der Datenbank nicht signifikant verändern oder Tabellen vollständig löschen.

### 2.5.4. MySQL Rechte Konfiguration

Um in MySQL Nutzer rechte zu Vergeben müssen einige Schritte erfolgen:

1. Mit Root User über die Oberfläche auf die Local Instance von MySQL zugreifen
2. Unter Server – Users und Privileges auswählen
3. Nun können wir neue User anlegen und Ihnen eine Rechtekategorie zuweisen oder eine Customisierte erstellen.
4. Dort können nun befehle ausgewählt werden, die der User benutzen darf (z.B. DELET, SELECT, oder INSERT)

Über Den Menüpunkt Felder -> Open SQL-Script können außerdem Dateien mit SQL-Befehlen ausgeführt werden, indem sie in der Workbench geöffnet werden und daraufhin per Knopfdruck ausgeführt werden. So wird auch die Sakila Datenbank eingelesen. Die User die für den Server bestehen werden auf die Installierten Datenbanken angewandt. Jeder User der neu erstellt wird kann auch mit den entsprechenden Daten (Username und Passwort) Im Programm genutzt werden und mit den zugewiesenen rechten auf die Datenbank zugreifen.

### **2.5.5. Datenbank Verbindung Programm**

Damit die User Anmeldung und die SQL abfragen ausgeführt werden können, muss zunächst **eine** Datenbank Verbindung hergestellt werden. Um eine Datenbank Verbindung herzustellen werden die Methoden `connectToDatabaseFuture` für asynchrone Aufrufe und `connectToDatabase` für Synchrone aufrufe in der `database.dart` ausgeführt. Daraufhin können SQL abfragen ausgeführt werden.

## **3. Schluss**

Mit diesem Programm ist der Grundstein für das Projekt gelegt, und bereit weiter bearbeitet zu werden, um ein voll umfängliches Lern-Programm zu werden.

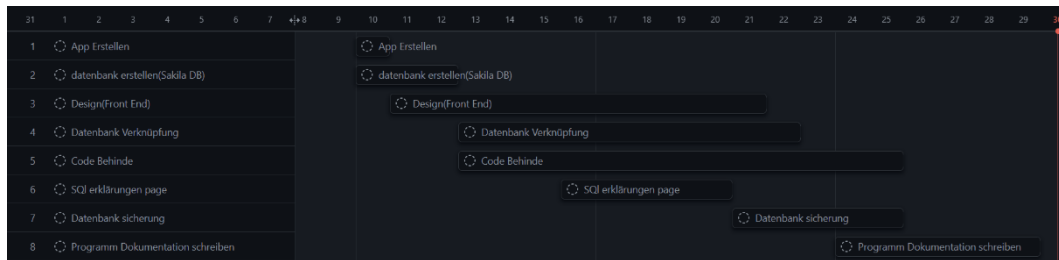


Abbildung 3 Gant Diagramm

```

ListTile(
  leading: Icon(Icons.pageview),
  title: Text('Seite 2'),
  onTap: () {
    Navigator.pushNamed(context, '/second');
  }
)

```

Abbildung 2 Die Sidbar Bearbeiten, sodass eine weitere Seite verlinkt wird

```

class MyHomePage extends StatelessWidget {
  const MyHomePage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Main Page'),
        actions: [UserProfileButton()],
      ),
      drawer: AppDrawer(),
      body: TextInputList(
        correctAnswers: correctAnswersFirstPage,
        fixedTexts: fixedTextsFirstPage,
      ),
    );
  }
}

```

Abbildung 1 eine Neue Seite

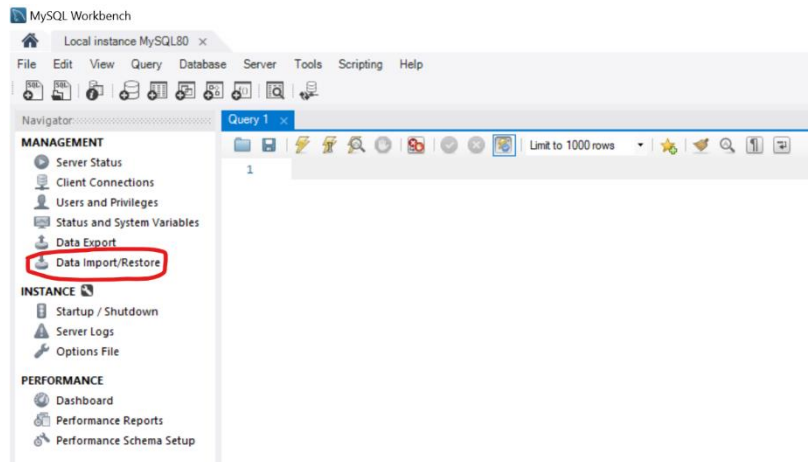


Abbildung 4 MYSQL WorkBench

## I. Abbildungsverzeichnis

Abbildung 3 eine Neue Seite .....	III
Abbildung 2 Die Sidbar Bearbeiten, sodass eine weitere Seite verlinkt wird .....	III
Abbildung 1 Gant Diagramm.....	III
Abbildung 4 MYSQL WorkBench .....	IV