Flutter and MySQL for Hochschulsport App HSP-Freiplatzbörse

Kristijan Lazeski
Angewandte Informatik
Hochschule Fulda
Fulda, Germany
kristijan.lazeski@cs.hs-fulda.de

Abstract—**This paper introduces a comprehensive solution for Hochschulsport, utilizing the power of Flutter for mobile app development and MySQL for data management. The mobile application simplifies venue reservations, empowering users to seamlessly share unutilized slots, thus optimizing resource allocation. By enhancing user flexibility and administrative efficiency, this approach not only improves the Hochschulsport experience but also sets a precedent for addressing similar resource management challenges in educational institutions, ensuring equitable access and fostering collaborative sporting communities.**

Index Terms—**Flutter, MySql**

I. INTRODUCTION

In the modern era, mobile applications have become integral tools for organizations and institutions, facilitating seamless communication, engagement, and accessibility for their users. Hochschul Sport, as a vital component of higher education institutions, plays a pivotal role in promoting physical activity and well-being among students and faculty. To enhance its reach, effectiveness, and user experience, the integration of cutting-edge technologies into Hochschul Sport services is imperative.

This paper presents a comprehensive exploration of the rationale behind the strategic choice to employ Flutter, a versatile cross-platform framework, and MySQL, a robust relational database management system, in the development of a specialized mobile application tailored for Hochschul Sport. The collaborative synergy between Flutter and MySQL holds the potential to address the unique requirements and challenges faced by Hochschul Sport, ultimately resulting in a more dynamic, efficient, and user-friendly app.

In this paper, we delve into the distinctive features and advantages of both Flutter and MySQL, elucidating why these technologies were deemed ideal for the realization of the Hochschul Sport mobile application. Through a careful analysis of their respective strengths, we aim to shed light on how this strategic technological pairing can revolutionize the way Hochschul Sport operates, engages its users, and manages its data.

The subsequent sections will provide a detailed examination of the reasons behind choosing Flutter as the development framework, emphasizing its cross-platform capabilities, rich user interface (UI) capabilities, and rapid development cycle. Additionally, we will explore the rationale for selecting MySQL as the backend database system, emphasizing its reliability, scalability, and robust data management capabilities. By juxtaposing these technologies with the specific requirements and challenges faced by Hochschul Sport, we intend to underscore the soundness of our technological choices and the potential impact on the institution's sports and fitness ecosystem.

This paper serves as a valuable resource for institutions contemplating the adoption of modern technologies to enhance their services. It offers insights into the decision-making process that guided the selection of Flutter and MySQL for the Hochschul Sport app, with the overarching goal of inspiring innovative solutions and fostering a deeper understanding of the symbiotic relationship between technology and sports in higher education institutions.

II. STATE OF THE TECHNOLOGY

A. Flutter

Flutter, an open-source UI software development framework developed by Google, has rapidly gained prominence in the realm of app development due to its unique capabilities and versatility. One of its standout features is its ability to create natively compiled applications for multiple platforms from a single codebase. This cross-platform compatibility enables developers to write code once and deploy it on both Android and iOS devices, significantly reducing development time and costs.

Traditionally, building applications for multiple platforms required distinct codebases for Android and iOS, leading to duplicated efforts, increased maintenance overhead, and potential discrepancies in user experiences. Flutter addresses these challenges by providing a unified framework that ensures consistent UI and behavior across platforms. This streamlined approach not only accelerates development but also simplifies updates and bug fixes, making it an ideal choice for institutions like Hochschul Sport looking to reach a broad audience efficiently.

Furthermore, Flutter's extensive package ecosystem provides access to a wealth of pre-built solutions and third-party integrations, further expediting development. This availability of resources simplifies the integration of features such as geolocation services, notifications, and social media sharing,

enriching the Hochschul Sport app's functionality and user engagement.

In conclusion, Flutter's cross-platform capabilities, customizable UI, and rapid development cycle make it a compelling choice for the Hochschul Sport app. Its ability to streamline development efforts, maintain consistency across platforms, and deliver an engaging user experience positions it as a powerful tool for modern app development. The subsequent sections of this paper will delve into the selection of MySQL as the database system, emphasizing how this choice complements Flutter and contributes to the holistic success of the Hochschul Sport app.

B. MySQL

MySQL, a popular open-source relational database management system (RDBMS), has garnered widespread acclaim for its reliability and robust data management capabilities. When developing the Hochschul Sport app, ensuring the integrity and security of user data, event schedules, fitness records, and other critical information is paramount. MySQL's proven track record in handling large datasets with ACID (Atomicity, Consistency, Isolation, Durability) compliance instills confidence that user data is stored and processed in a secure and dependable manner.

The RDBMS architecture of MySQL also facilitates the organization of data into structured tables with well-defined relationships, making it an ideal choice for Hochschul Sport's complex data needs. This structured approach simplifies data retrieval, manipulation, and reporting, ensuring that the app can efficiently manage and present essential information to users. As Hochschul Sport anticipates growth and an expanding user base, the ability to scale the app's backend infrastructure is a crucial consideration. MySQL excels in this regard, offering various options for horizontal and vertical scaling. This scalability ensures that the app can accommodate an increasing number of users, events, and fitness-related data without compromising performance.

Additionally, MySQL's query optimization capabilities contribute to faster data retrieval and processing, ensuring that users experience minimal latency when accessing information or interacting with the app. Hochschul Sport can thus deliver a responsive and efficient experience to its users, whether they are checking sports schedules, registering for events, or tracking their fitness progress.

Hochschul Sport's diverse array of activities generates a wealth of data that can be harnessed for strategic planning, decision-making, and user engagement. MySQL's robust data management features facilitate efficient storage, retrieval, and analysis of this data. The ability to perform complex queries and generate customized reports empowers Hochschul Sport administrators to gain valuable insights into user behavior, event attendance, and fitness trends.

Moreover, MySQL's compatibility with various reporting and business intelligence tools allows for seamless integration with analytics dashboards. This integration can enhance Hochschul Sport's ability to monitor app performance, track key performance indicators (KPIs), and make data-driven improvements to the app over time.

In summary, MySQL's reputation for reliability, scalability, and data management capabilities aligns with the core requirements of the Hochschul Sport app. It ensures the secure storage and efficient retrieval of user and event data, while also providing the foundation for future growth and data-driven decision-making. The symbiotic relationship between MySQL as the database system and Flutter as the app development framework positions Hochschul Sport for success in delivering a dynamic, responsive, and user-centric mobile application. The subsequent sections of this paper will explore the implementation and integration of these technologies within the context of Hochschul Sport's app development.

III. INTEGRATION OF COMPONENTS FOR THE HOCHSCHUL SPORT APP

The Hochschul Sport app seamlessly integrates two core components: Flutter for app development, MySQL as the backend database system. This cohesive integration forms the foundation for a user-friendly and efficient application that empowers users to manage their attendance in sports classes. Here's how these components work together:

A. Frontend

In a Flutter app, the frontend is based on a widget-based structure where you can create various pages or screens that correspond to the functionalities you described, such as "See Courses," "Add Course," "Select Course to Apply," "Delete Course," and "Attendees for Course." The structure of the frontend for these functionalities is as follows:

1) Widgets and Screens:
   - Flutter is built around the concept of widgets, which are building blocks for the app's user interface. In this case, widgets are created on the screen for each of the functionalities mentioned above.

2) Navigation:
   - To navigate between different screens, the App uses Flutter's built-in navigation system. The 'Navigator' widget allows to push and pop screens on the navigation stack, making it easy to move between pages.

3) Buttons:
   - To create buttons, one can use the 'ElevatedButton', 'TextButton' or 'OutlinedButton' widgets. These buttons in this case are OutlinedButton. As shown in the next code sample. This type of code would create a Button for each needed Funcitonality that would leed you to the next Page.

```
Padding(
    padding: const EdgeInsets.all(8.0),
    child: OutlinedButton(
        onPressed: () {
            Navigator.push(
                context,
                MaterialPageRoute(
```

```
        builder: (context) =>
          const
            DeletePage(title:
              'Flutter MySQL
              Delete Sample'),
        ),
    );
  },
  child: const Text(
    "Delete Course",
  ),
  ),
),
```

4) **Routing:**
   - Each button, when pressed, triggers navigation to a different screen or page. By importing the koherent dart files of the given pages you can navigate to the next sscreen:

```
import 'package:fluter/material.dart';
import 'delete_page.dart';
import 'insert_page.dart';
import 'read_all_page.dart';
import 'select_page.dart';
import 'attendee_page.dart';
```

5) **Screen Implementation:**
   - Each screen corresponds to a specific functionality. For example, the "See Courses" screen would display a list of available courses, and the "Add Course" screen would allow users to add a new course.

6) **Button Action:**
   - With the usage of the 'onPressed' property of the buttons to define what happens when a button is pressed. For example, for the "See Courses" button, you'd use the "read all" button to get the list of the free places in the given courses:

```
body: SingleChildScrollView(
    child: Column(
      children:
        displayList.map<Widget>((data)
        {
      return Card(
        child: ListTile(
        leading:
          Text(data['selectedYear']
          ?? ""),
        title:
          Text(data['selectedName']
          ?? ""),
        subtitle:
          Text(data['selectedCountry']
          ?? ""),
        trailing: TextButton(
          child: const Text("Apply"),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(
```

```
        builder: (context) =>
            UpdatePage(title:
              data['selectedId']
              ?? "")),
        );
      },
    ),
  ));
}).toList()),
),
```

7) **User Interaction:**
   - For user interaction, the App uses input widgets like text fields. These widgets help users perform actions like selecting a course or entering course information.

B. Backend

To facilitate access and interaction with the MySQL database from within the system's codebase, the mysql_client.dart package is employed. This Dart package serves as a bridge between the sports management system's code and the MySQL database, providing a means for data retrieval, storage, and manipulation. The mysql_client.dart package offers a wide range of functions and methods for connecting to the database, executing SQL queries, and managing the retrieved data, thus enhancing the system's capabilities in data handling.

```
Future<void> _delete() async {
  print("Connecting to mysql server...");
  // create connection
  final conn = await
    MySQLConnection.createConnection(
    //host: "127.0.0.1", //when you use
      simulator
    host: "10.0.2.2", // when you use emulator
    //host: "localhost"
    port: 3306,
    userName: "root",
    password: "root", // you need to replace
      with your password
    databaseName: "sportsmanagement", // you
      need to replace with your db name
  );
  await conn.connect();
  print("Connected");
  // update some rows
  // SQL Call
  print(res.affectedRows);
  // close all connections
  await conn.close();
}
```

Each SQL call is based upon how the Database is built. This exists upon three Tables that are connected to each other:

1) **User Table:**

```
CREATE TABLE User (
    UserID INT AUTO_INCREMENT PRIMARY KEY,
```

```
      Name VARCHAR(255) NOT NULL,
      Lastname VARCHAR(255) NOT NULL,
      MatrNumber VARCHAR(10) UNIQUE NOT NULL
);
```

- UserID: This is an auto-incremented primary key, ensuring the uniqueness of each user within the system.
- Name and Lastname: These fields store the user's personal information, which can be used for identification and communication purposes.
- MatrNumber: This field is designed for a unique identifier, typically associated with educational institutions, and should be used for enrollment and registration.

2) Sport Table

```
CREATE TABLE Sport (
    SportID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Description TEXT,
    Date VARCHAR(255),
    Time VARCHAR(255),
    Place VARCHAR(255),
    Price INT,
    TeacherID INT,
    AppliedForPlaces bool,
    FOREIGN KEY (TeacherID) REFERENCES
        User(UserID)
);
```

- SportID: This auto-incremented primary key is crucial for uniquely identifying each sports event within the system.
- Name: The name of the sport event is a key element for user recognition and event categorization.
- Description: This text field provides detailed information about the sport event, including its purpose and content.
- Date and Time: These fields specify when the sport event will take place, offering crucial scheduling information.
- Place: Describes the physical location of the event, which helps users locate and attend the event.
- Price: This field includes the cost of participation. This field is not used and is always set at 1 but could be helpful if sometimes down the line Payment should be implemented.
- TeacherID: This foreign key connects to the "User" table, indicating the instructor or teacher responsible for the event. This is important for the system's educational aspect.
- AppliedForPlaces: Tracks the number of participants who have applied to the event, helping to manage and allocate available slots.

3) UserAppliedSport Table:

```
CREATE TABLE UserAppliedSport (
```

```
    ApplicationID INT AUTO_INCREMENT
        PRIMARY KEY,
    UserID INT,
    SportID INT,
    MatrNumber VARCHAR(255),
    AppliedDate TIMESTAMP DEFAULT
        CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES
        User(UserID),
    FOREIGN KEY (SportID) REFERENCES
        sport(SportID),
    UNIQUE KEY (UserID, SportID)
);
```

- ApplicationID: This field serves as a unique identifier for each user's application to a sports event.
- UserID: A foreign key that links to the "User" table, indicating which user has applied for a particular sport event.
- SportID: Another foreign key, connecting to the "Sport" table and signifying the specific sport event to which a user has applied.
- AppliedDate: Captures the timestamp when the application was made, assisting with the chronological organization of applications.

C. Conclusion

In this paper, we have presented a comprehensive solution for Hochschulsport, harnessing the power of Flutter for mobile app development and MySQL for efficient data management. This mobile application redefines the user experience by simplifying venue reservations, offering a dynamic platform for users to seamlessly share unutilized time slots, ultimately optimizing resource allocation. The user-centric approach not only enhances flexibility but also streamlines administrative efficiency within the Hochschulsport ecosystem.

This resulting application provides a blueprint for addressing similar resource management challenges in educational institutions. By prioritizing equitable access to sporting facilities and fostering collaborative sporting communities, we introduce a new standard for efficient resource allocation in academic sports programs. The seamless integration of the mobile app and MySQL database forms a foundation for future developments and extensions, setting the stage for the evolution of Hochschulsport management.

In conclusion, this paper demonstrates the potential of cutting-edge technologies in shaping the future of sports management, ensuring that it remains accessible, efficient, and collaborative. Our solution is not just a research project; it is a practical application that can serve as a basic structure to be built upon, improving the Hochschulsport experience and inspiring similar innovations in resource management across educational institutions.

D. References

- Github Project
- Dart Documentation
- MySQL Documentation