**SPOR SALONU YÖNETİM SİSTEMİ (ZEUS GYM)**

**Veritabanı Yönetim Sistemleri Dersi - Final Projesi - 2025 Bahar Dönemi**

EMİRHAN OKTAY - 231307074  
EFEKAN DEMİR - 231307054  
OĞUZHAN ERBİL - 231307021

Bilgisayar Mühendisliği Bölümü  
Teknoloji Fakültesi  
Türkiye, 2025  
emiroktay09@gmail.com, efekan@gmail.com, oguzhan@gmail.com

**Öğretim Görevlisi:** Dr. Öğr. Üyesi SAMET DİRİ  
**Rapor Tarihi:** Mayıs 2025

**ÖZET**

Bu çalışmada, modern bir spor salonu işletmesinin yönetiminde karşılaşılan temel sorunlara çözüm getiren kapsamlı bir veritabanı yönetim sistemi geliştirilmiştir. ZEUS GYM adı verilen bu sistem, üyelik yönetimi, ders programlama, eğitmen atamaları, ödeme takibi ve giriş-çıkış kontrolü gibi işlemleri otomatize etmeyi hedeflemektedir. MySQL veritabanı ve Flutter mobil uygulama çerçevesi kullanılarak geliştirilen sistem, spor salonu işletmecileri, eğitmenler ve üyeler olmak üzere üç farklı kullanıcı rolüne hizmet vermektedir. Sistem, yetkilendirme mekanizmaları, ilişkisel veri modeli ve optimizasyon teknikleri kullanılarak verimli bir şekilde tasarlanmıştır. Bu rapor, sistemin tasarım aşamalarını, veritabanı yapısını, uygulama mimarisini ve kullanıcı arayüzünü ayrıntılı olarak incelemektedir.

**Anahtar Kelimeler—** Veritabanı Yönetim Sistemleri, Spor Salonu Yönetimi, Flutter, MySQL, Mobil Uygulama

**I. GİRİŞ**

Günümüzde spor salonları, artan sağlık bilinci ile birlikte toplumun önemli bir parçası haline gelmiştir. Bu salonların etkin yönetimi, hem işletme sahipleri hem de kullanıcılar için kritik öneme sahiptir. Geleneksel yöntemlerle yönetilen spor salonlarında üye kaydı, ders programları, eğitmen atama, giriş-çıkış takibi ve ödeme işlemleri gibi süreçler genellikle manuel olarak yürütülmekte, bu da zaman kaybı, veri tutarsızlığı ve hata riskini artırmaktadır.

Bu proje, bu sorunları çözmek amacıyla kapsamlı bir veritabanı yönetim sistemi geliştirmeyi hedeflemektedir. ZEUS GYM adını verdiğimiz bu sistem, spor salonu işletmelerinin tüm operasyonel süreçlerini verimli bir şekilde yönetmelerine olanak tanıyan mobil bir çözüm sunmaktadır.

Sistemimiz, üç temel kullanıcı grubu için özelleştirilmiş fonksiyonlar içermektedir:

* **Üyeler**: Üyelik durumlarını görüntüleme, ders programlarına erişim ve rezervasyon yapma
* **Eğitmenler**: Ders programlarını yönetme, katılımcı listelerini görüntüleme
* **Yöneticiler (Admin)**: Tüm sistem verilerine erişim, kullanıcı ve paket yönetimi

Bu rapor, ZEUS GYM sisteminin tasarım sürecini, veritabanı yapısını, uygulama mimarisini ve geliştirme sürecini detaylı olarak açıklamaktadır.

**II. PROBLEM TANIMI**

Modern spor salonu işletmeleri, çok sayıda üye, eğitmen ve çeşitli aktivitelerin yönetimini gerektiren karmaşık organizasyonlardır. Bu karmaşıklık, aşağıdaki sorunları beraberinde getirmektedir:

1. **Üyelik Yönetimi**: Üye kayıtları, üyelik süreleri, paket bilgileri ve ödemelerin manuel takibi zor ve hataya açıktır.
2. **Ders Programlaması**: Farklı eğitmenlerin ders programlarının çakışmadan planlanması ve üyelerin bu derslere katılımının takibi önemli bir zorluktur.
3. **Giriş-Çıkış Kontrolü**: Spor salonunun sadece aktif üyeler tarafından kullanılmasını sağlamak için etkili bir kontrol mekanizması gerekmektedir.
4. **Veri Entegrasyonu**: Üyelik, ödeme, ders ve katılım verilerinin entegre bir sistemde tutularak tutarlılığının sağlanması önemlidir.
5. **Raporlama ve Analiz**: İşletme sahiplerinin karar verme süreçlerini destekleyecek veri analizlerinin yapılması gerekmektedir.

Bu problemleri çözmek için, verilerin güvenli ve tutarlı bir şekilde saklanmasını, işlenmesini ve erişilmesini sağlayan kapsamlı bir veritabanı yönetim sistemi tasarlanmıştır.

**III. YAPILAN ARAŞTIRMALAR**

Projemizin başlangıç aşamasında, mevcut spor salonu yönetim sistemlerini inceleyerek güçlü ve zayıf yönlerini belirledik. Ayrıca, spor salonu işletmecileri ve üyeleriyle görüşmeler yaparak gerçek ihtiyaçları tespit ettik.

**A. Benzer Sistemlerin İncelenmesi**

Piyasada bulunan spor salonu yönetim yazılımları (GymMaster, Mindbody, Virtuagym vb.) incelenerek aşağıdaki özellikler analiz edilmiştir:

* Üyelik yönetimi mekanizmaları
* Ödeme sistemleri entegrasyonu
* Ders programı ve rezervasyon yönetimi
* Kullanıcı arayüzü tasarımları
* Mobil uygulama özellikleri

**B. Veritabanı Teknolojileri Karşılaştırması**

Projemizde kullanılacak veritabanı yönetim sistemi için aşağıdaki alternatifler değerlendirilmiştir:

* MySQL
* PostgreSQL
* SQLite
* MongoDB

Değerlendirme sonucunda, aşağıdaki nedenlerle MySQL'in projemiz için en uygun çözüm olduğuna karar verilmiştir:

* İlişkisel veri modeline uygunluk
* Performans ve ölçeklenebilirlik
* Flutter ile kolay entegrasyon
* Açık kaynak kodlu olması
* Yaygın kullanım ve topluluk desteği

**C. Mobil Uygulama Geliştirme Platformları**

Uygulama geliştirme için aşağıdaki platformlar incelenmiştir:

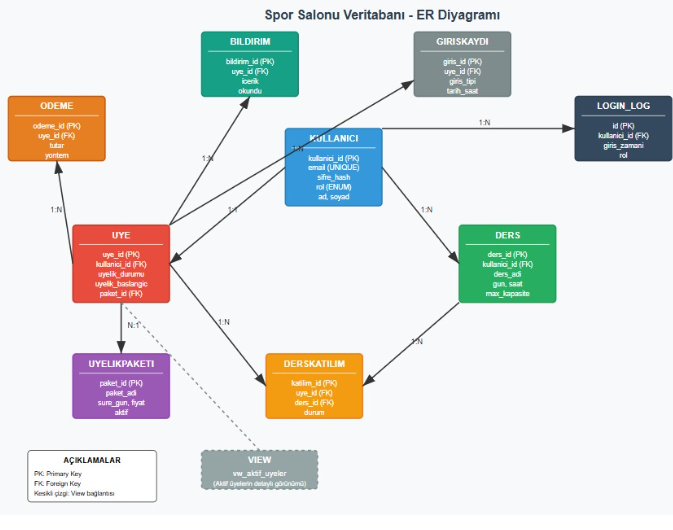
* Native Android (Java/Kotlin)
* Native iOS (Swift)
* React Native
* Flutter
* Xamarin

Flutter'ın seçilme nedenleri:

* Çapraz platform desteği (iOS ve Android)
* Hızlı geliştirme süreci
* Zengin widget kütüphanesi
* Performans ve kullanıcı deneyimi
* Veritabanı entegrasyonu için geniş paket desteği

**IV. VERİTABANI TASARIMI**

**A. ER Diyagramı**

****

Sistemimizin veritabanı, 9 ana tablo ve 1 view içermektedir. Tablolar arasındaki ilişkiler aşağıdaki ER diyagramında gösterilmiştir:

**B. Veri Tabloları ve İlişkileri**

**1. Ana Tablolar**

* **kullanici**: Tüm kullanıcıların temel bilgilerini içerir (ID, email, şifre, rol vb.)
* **uye**: Üyelere özel bilgileri ve üyelik durumunu içerir (kullanıcı\_id ile ilişkili)
* **uyelikpaketi**: Farklı üyelik paketlerinin tanımlarını içerir
* **ders**: Eğitmen derslerinin program bilgilerini içerir
* **derskatilim**: Üyelerin derslere katılım kayıtlarını tutar
* **odeme**: Ödeme işlemlerinin kayıtlarını içerir
* **bildirim**: Sistem bildirimlerini saklar
* **giriskaydi**: Spor salonuna giriş kayıtlarını tutar
* **login\_log**: Sisteme giriş loglarını kaydeder

**2. View**

* **vw\_aktif\_uyeler**: Aktif üyelerin detaylı bilgilerini içeren görünüm

**3. Tablo İlişkileri**

* **kullanici-uye**: Bire-bir ilişki (Her kullanıcı en fazla bir üye kaydına sahip olabilir)
* **uyelikpaketi-uye**: Bire-çok ilişki (Bir paket birden fazla üyeye atanabilir)
* **kullanici-ders**: Bire-çok ilişki (Bir eğitmen birden fazla ders verebilir)
* **uye-derskatilim**: Bire-çok ilişki (Bir üye birden fazla derse katılabilir)
* **ders-derskatilim**: Bire-çok ilişki (Bir derse birden fazla üye katılabilir)
* **uye-giriskaydi**: Bire-çok ilişki (Bir üye birden fazla giriş kaydına sahip olabilir)
* **uye-odeme**: Bire-çok ilişki (Bir üye birden fazla ödeme yapabilir)
* **uye-bildirim**: Bire-çok ilişki (Bir üye birden fazla bildirim alabilir)

**C. Normalizasyon**

Veritabanımız, veri tutarlılığını ve bütünlüğünü sağlamak için 3. normal forma (3NF) uygun olarak tasarlanmıştır:

1. **1NF**: Tüm tablolardaki alanlar atomiktir, tekrar eden gruplar bulunmamaktadır.
2. **2NF**: Tüm tablolar 1NF'dir ve anahtar olmayan tüm alanlar, birincil anahtara tam fonksiyonel bağımlıdır.
3. **3NF**: Tüm tablolar 2NF'dir ve anahtar olmayan hiçbir alan, birincil anahtara geçişli olarak bağımlı değildir.

**D. Kısıtlamalar ve Tetikleyiciler**

Veritabanımızda veri bütünlüğünü sağlamak için aşağıdaki kısıtlamalar ve tetikleyiciler kullanılmıştır:

1. **Birincil Anahtar Kısıtlamaları**: Tüm tablolarda benzersiz tanımlayıcılar için otomatik artan ID'ler
2. **Yabancı Anahtar Kısıtlamaları**: Tablolar arasındaki ilişkileri korumak için
3. **NOT NULL Kısıtlamaları**: Zorunlu alanlar için
4. **UNIQUE Kısıtlamaları**: Email gibi benzersiz olması gereken alanlar için
5. **DEFAULT Değerler**: Belirli alanlarda varsayılan değerler tanımlanmıştır
6. **CHECK Kısıtlamaları**: Veri değerlerinin belirli koşulları sağlamasını garantilemek için
7. **Tetikleyiciler**: Kullanıcı girişlerinin otomatik olarak login\_log tablosuna kaydedilmesi için

**V. AKIŞ ŞEMASI**

Sistemin genel iş akışı aşağıdaki bileşenlerden oluşmaktadır:

**A. Kullanıcı Kimlik Doğrulama Akışı**

1. Kullanıcı email ve şifre girer
2. Sistem verileri kullanici tablosunda kontrol eder
3. Kimlik doğrulama başarılı ise:
   * Kullanıcı rolüne göre ilgili panele yönlendirilir
   * Giriş bilgileri login\_log tablosuna kaydedilir
4. Kimlik doğrulama başarısız ise:
   * Hata mesajı gösterilir

**B. Üye İşlemleri Akışı**

1. Üye kaydı:
   * Admin kullanıcı bilgilerini girer
   * Sistem kullanici ve uye tablolarına kayıt oluşturur
2. Üyelik paketi atama:
   * Admin üye için paket seçer
   * Sistem uye tablosunda paket\_id güncellemesi yapar
3. Derslere katılım:
   * Üye katılmak istediği dersi seçer
   * Sistem derskatilim tablosuna kayıt ekler

**C. Ders Yönetimi Akışı**

1. Ders oluşturma:
   * Admin ders bilgilerini ve eğitmen atamasını yapar
   * Sistem ders tablosuna kayıt ekler
2. Ders katılım kontrolü:
   * Eğitmen derse katılım durumunu kontrol eder
   * Sistem derskatilim tablosundaki durum alanını günceller

**D. Spor Salonu Giriş Kontrolü**

1. Üye spor salonuna giriş yapar
2. Sistem üyelik durumunu kontrol eder
3. Üyelik aktif ise:
   * Giriş onaylanır ve giriskaydi tablosuna kayıt eklenir
4. Üyelik aktif değil ise:
   * Giriş reddedilir ve kullanıcıya bilgi verilir

**VI. YAZILIM MİMARİSİ**

**A. Genel Mimari Yapı**

Uygulamamız, aşağıdaki katmanlı mimari yapısına sahiptir:

1. **Sunum Katmanı (UI)**: Flutter widget'ları kullanılarak oluşturulan kullanıcı arayüzleri
2. **İş Mantığı Katmanı**: Provider yapısı üzerinden yönetilen uygulama durumu ve iş mantığı
3. **Veri Erişim Katmanı**: MySQL veritabanı ile etkileşimi sağlayan servisler
4. **Veritabanı Katmanı**: MySQL tabloları ve ilişkileri

**B. Klasör Yapısı**

Projemizin klasör yapısı aşağıdaki gibidir:

lib/

├── config/ # Konfigürasyon ve sabitler

├── models/ # Veri modelleri

├── providers/ # State yönetimi (Provider yapısı)

├── screens/ # Uygulama ekranları (UI)

├── services/ # Veritabanı servisleri ve API çağrıları

└── main.dart # Uygulama başlangıç noktası

**C. Kullanılan Teknolojiler**

**1. Flutter & Dart**

Uygulamamız, Flutter framework'ü ve Dart programlama dili kullanılarak geliştirilmiştir. Flutter'ın çapraz platform özelliği sayesinde, aynı kod tabanı ile hem iOS hem de Android platformlarında çalışabilen bir uygulama elde edilmiştir.

**2. State Management: Provider**

Uygulama genelinde durum yönetimi için Provider paketi kullanılmıştır. Bu yapı, verilerin uygulamanın farklı bölümleri arasında kolayca paylaşılmasını ve senkronize edilmesini sağlamaktadır.

final userProvider = Provider<UserProvider>(

create: (\_) => UserProvider(),

);

**3. MySQL & mysql\_utils**

Veritabanı işlemleri için MySQL kullanılmış olup, uygulama ile veritabanı arasındaki bağlantı mysql\_utils Dart paketi üzerinden sağlanmıştır.

class DatabaseService {

MySqlUtils conn = new MySqlUtils(

settings: {

'host': 'localhost',

'port': 3306,

'user': 'root',

'password': 'password',

'db': 'spor\_salonu',

'charset': 'utf8',

},

);

Future<List<Map<String, dynamic>>> getActiveMembers() async {

var result = await conn.select(

'SELECT \* FROM vw\_aktif\_uyeler'

);

return result;

}

}

**VII. VERİTABANI SORGULARI VE İMPLEMENTASYONU**

Sistemimizde kullanıcı rollerine göre özelleştirilmiş sorgular bulunmaktadır:

**A. Üye Paneli Sorguları**

**1. Üyelik Bilgilerini Getirme**

SELECT u.\*, up.paket\_adi, up.sure\_gun, up.fiyat,

DATEDIFF(u.uyelik\_bitis, CURDATE()) as kalan\_gun

FROM uye u

JOIN uyelikpaketi up ON u.paket\_id = up.paket\_id

WHERE u.kullanici\_id = ?

**2. Ders Programını Getirme**

SELECT d.\*, k.ad as egitmen\_adi, k.soyad as egitmen\_soyadi

FROM ders d

JOIN kullanici k ON d.kullanici\_id = k.kullanici\_id

WHERE d.aktiflik = 'Aktif'

ORDER BY FIELD(d.gun, 'Pazartesi', 'Salı', 'Çarşamba', 'Perşembe', 'Cuma', 'Cumartesi', 'Pazar'),

d.baslangic\_saati

**B. Eğitmen Paneli Sorguları**

**1. Eğitmenin Verdiği Dersleri Listeleme**

SELECT \* FROM ders

WHERE kullanici\_id = ?

ORDER BY FIELD(gun, 'Pazartesi', 'Salı', 'Çarşamba', 'Perşembe', 'Cuma', 'Cumartesi', 'Pazar'),

baslangic\_saati

**2. Derse Katılım Kontrolü**

SELECT dk.\*, k.ad, k.soyad

FROM derskatilim dk

JOIN uye u ON dk.uye\_id = u.uye\_id

JOIN kullanici k ON u.kullanici\_id = k.kullanici\_id

WHERE dk.ders\_id = ? AND dk.tarih = CURDATE()

**C. Admin Paneli Sorguları**

**1. Kullanıcı Listesi**

SELECT \* FROM kullanici

**2. Üyeleri ve Paketlerini Listeleme**

SELECT u.\*, k.ad, k.soyad, k.email, up.paket\_adi, up.fiyat

FROM uye u

JOIN kullanici k ON u.kullanici\_id = k.kullanici\_id

LEFT JOIN uyelikpaketi up ON u.paket\_id = up.paket\_id

**3. Giriş Logları**

SELECT \* FROM login\_log ORDER BY giris\_zamani DESC

**D. Veritabanı Optimizasyonu**

Sistemin performansını artırmak için aşağıdaki optimizasyonlar yapılmıştır:

1. **İndeksler**: Sık sorgulanan alanlar için indeksler oluşturulmuştur
2. CREATE INDEX idx\_email ON kullanici(email);
3. **View Kullanımı**: Sık kullanılan karmaşık sorgular için view oluşturulmuştur
4. CREATE VIEW vw\_aktif\_uyeler AS
5. SELECT u.uye\_id, u.kullanici\_id, k.email, k.ad, k.soyad,
6. u.uyelik\_durumu, u.uyelik\_baslangic, u.uyelik\_bitis,
7. p.paket\_adi, p.fiyat
8. FROM uye u
9. JOIN kullanici k ON u.kullanici\_id = k.kullanici\_id
10. LEFT JOIN uyelikpaketi p ON u.paket\_id = p.paket\_id
11. WHERE u.uyelik\_durumu = 'Aktif'
12. AND (u.uyelik\_bitis IS NULL OR u.uyelik\_bitis >= CURDATE());

**VIII. KULLANICI ARAYÜZÜ TASARIMI**

Uygulamamız, kullanıcı rollerine göre özelleştirilmiş üç farklı arayüz sunmaktadır:

**A. Giriş Ekranı**

Tüm kullanıcılar için ortak olan giriş ekranı, email ve şifre alanları içermektedir. Kullanıcı bilgileri doğrulandıktan sonra, rol bilgisine göre ilgili panele yönlendirilmektedir.

**B. Üye Paneli**

Üye paneli aşağıdaki özellikleri içermektedir:

* Üyelik durumu ve kalan gün sayısı görüntüleme
* Üyelik paketi bilgileri
* Haftalık ders programı
* Derse kayıt olma ve katılım durumu

**C. Eğitmen Paneli**

Eğitmen paneli aşağıdaki özellikleri içermektedir:

* Sorumlu olunan derslerin listesi
* Ders katılım durumlarını kontrol etme
* Derse katılan üyeleri görüntüleme

**D. Admin Paneli**

Admin paneli aşağıdaki özellikleri içermektedir:

* Kullanıcı yönetimi (ekleme, düzenleme, silme)
* Üyelik paketi yönetimi
* Ders programı düzenleme
* Eğitmen atama
* Sistem loglarını görüntüleme

**IX. GÜVENLİK ÖNLEMLERİ**

Sistemimizde aşağıdaki güvenlik önlemleri alınmıştır:

**A. Kimlik Doğrulama**

* Şifrelerin veritabanında hash'lenerek saklanması
* Oturum yönetimi ve token tabanlı kimlik doğrulama

**B. Yetkilendirme**

* Rol tabanlı erişim kontrolü (RBAC)
* Her kullanıcının yalnızca yetkisi olan işlemleri gerçekleştirebilmesi

**C. Veri Güvenliği**

* SQL injection saldırılarına karşı parametreli sorgular kullanımı
* Hassas verilerin şifrelenmesi
* Veritabanı bağlantılarının güvenli bir şekilde yönetilmesi

**X. PERFORMANS DEĞERLENDİRMESİ**

Sistemimizin performansı aşağıdaki yöntemlerle optimize edilmiştir:

**A. Veritabanı Optimizasyonu**

* İndeksler kullanılarak sorgu performansı artırılmıştır
* Karmaşık sorgular için view'lar oluşturulmuştur
* Normalizasyon ile veri tutarlılığı sağlanmıştır

**B. Uygulama Performansı**

* Lazy loading teknikleri kullanılarak uygulama başlatma süresi kısaltılmıştır
* Widget yeniden oluşturmaları minimize edilerek UI performansı artırılmıştır
* Provider yapısı ile durum yönetimi optimize edilmiştir

**XI. TEST VE DOĞRULAMA**

Sistemimiz aşağıdaki test aşamalarından geçirilmiştir:

**A. Veritabanı Testleri**

* SQL sorguları performans testleri
* Veri bütünlüğü testleri
* İlişki kısıtlamalarının doğrulanması

**B. Fonksiyonel Testler**

* Kullanıcı giriş senaryoları
* Üyelik işlemleri
* Ders yönetimi işlemleri

**C. Arayüz Testleri**

* Farklı ekran boyutlarında uyumluluk
* Kullanıcı deneyimi testleri

**XII. SONUÇ VE GELECEK ÇALIŞMALAR**

**A. Sonuç**

Bu projede, modern bir spor salonu işletmesinin ihtiyaçlarını karşılayan kapsamlı bir veritabanı yönetim sistemi geliştirilmiştir. Sistem, üyelik yönetimi, ders programlama, kullanıcı yetkilendirme ve raporlama gibi temel işlevleri başarıyla yerine getirmektedir.

MySQL veritabanı yönetim sistemi ve Flutter framework'ü kullanılarak geliştirilen uygulama, hem iOS hem de Android platformlarında çalışabilmektedir. Rol tabanlı yetkilendirme sistemi sayesinde, farklı kullanıcı grupları için özelleştirilmiş arayüzler sunulmaktadır.

Veritabanı tasarımı, normalizasyon kurallarına uygun olarak gerçekleştirilmiş ve performans optimizasyonları yapılmıştır. Sistem, gerçek bir spor salonu işletmesinin ihtiyaçlarını karşılayacak şekilde tasarlanmıştır.

**B. Gelecek Çalışmalar**

Sistemin gelecekteki versiyonları için aşağıdaki geliştirmeler planlanmaktadır:

1. **Ödeme Sistemi Entegrasyonu**: Çevrimiçi ödeme sistemleri ile entegrasyon
2. **Gelişmiş Raporlama**: Daha kapsamlı istatistikler ve grafiksel raporlama
3. **Bildirim Sistemi**: Push notification desteği ile üyelere otomatik bildirimler
4. **Eğitmen-Üye Mesajlaşma**: Doğrudan iletişim için mesajlaşma sistemi
5. **QR Kod ile Giriş**: Temassız giriş-çıkış sistemi
6. **Performans İzleme**: Üyelerin antrenman performansını izleme ve analiz etme

**KAYNAKLAR**

[1] MySQL 8.0 Reference Manual, "The MySQL Database Management System". [Online]. Available: https://dev.mysql.com/doc/refman/8.0/en/

[2] Flutter Documentation, "Flutter: The UI toolkit for building beautiful, natively compiled applications". [Online]. Available: https://flutter.dev/docs

[3] P. Dubois, "MySQL Cookbook: Solutions for Database Developers and Administrators", 3rd ed., O'Reilly Media, 2014.

[4] A. Biessek, "Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart", Packt Publishing, 2019.

[5] C. J. Date, "An Introduction to Database Systems", 8th ed., Pearson, 2003.

[6] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley, 1994.

[7] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems", 7th ed., Pearson, 2015.

[8] Provider Package Documentation, "A mixture between dependency injection and state management". [Online]. Available: https://pub.dev/packages/provider

[9] mysql\_utils Package Documentation, "A Flutter package for MySQL database operations". [Online]. Available: https://pub.dev/packages/mysql\_utils