ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

IOS İle Mobil Uygulama Geliştirme Proje Raporu

UNINOTES MOBILE

Cross-platform Mobile Note Sharing Application (Flutter)

Elem Yalçınkaya - 22290419

10.01.2026

ÖZET

Bu rapor; IOS İle Mobil Uygulama Geliştirme dersi kapsamında geliştirilen UniNotes projesinin Flutter tabanlı mobil uygulamasını (UniNotes Mobile Frontend) tanıtmak amacıyla hazırlanmıştır. UniNotes Mobile, üniversite öğrencilerinin ders notlarını mobil ve web platformlarında paylaşabilmesini, filtreleyebilmesini ve cihaza indirebilmesini sağlayan çapraz platform bir istemci uygulamasıdır. Uygulama Material Design 3 prensipleriyle tasarlanmış, mor tema paleti (#7C3AED) ile görsel tutarlılık hedeflenmiş ve kullanıcı akışları; giriş/kayıt, paylaşılan notları görüntüleme, not ekleme, profil yönetimi, iletişim ve raporlama senaryolarını kapsayacak şekilde kurgulanmıştır.

Geliştirme sürecinde Flutter 3.4.0+ ve Dart 3.4.0+ kullanılmıştır. Durum yönetimi Provider (ChangeNotifier) deseni ile çözülmüş; JWT tabanlı kimlik doğrulama akışında token saklama ve oturum sürekliliği için `flutter_secure_storage` ve `shared_preferences` çözümlerinden yararlanılmıştır. Backend ile haberleşme `http` paketi üzerinden RESTful API yaklaşımıyla sağlanmış, isteklerde `Authorization` başlığına JWT eklenerek yetkilendirme akışı yürütülmüştür. Dosya seçimi ve yükleme işlemlerinde `file_picker`; indirme işlemlerinde ise platforma özgü yaklaşımı soyutlayan `file_downloader` yardımcı bileşeni kullanılmıştır. Web ortamında tarayıcı indirme davranışı, mobil ortamda ise cihaz dosya sistemi erişimi dikkate alınarak platform tespiti yapılmıştır.

UniNotes Mobile; notları sınıf ve dönem bazında filtreleyerek sunan paylaşılan notlar ekranı, dinamik ders listesi ile kontrollü not ekleme ekranı, kullanıcı bilgilerini ve yüklenen notları gösteren profil ekranı ve uygunsuz içerikler için raporlama diyaloğu ile bütünleşik bir paylaşım deneyimi sağlar. Bu çalışma, Flutter ile geliştirilen, temiz mimari prensiplerine göre katmanlı şekilde organize edilmiş ve farklı platformlarda çalışabilen bir mobil uygulama örneği sunmaktadır.

İçindekiler

1 GİRİŞ

Üniversite öğrencileri ders sürecinde farklı kaynaklardan notlar üretmekte ve bu notları çoğu zaman kişisel depolama alanlarında dağınık biçimde saklamaktadır. Bu durum, özellikle sınav dönemlerinde veya ders tekrarlarında içeriklere hızlı erişimi zorlaştırmakta; aynı ders için benzer notların tekrar tekrar hazırlanmasına yol açmaktadır. Bu bağlamda geliştirilen UniNotes Mobile, öğrencilerin ders notlarını merkezi bir platformda görüntüleyip paylaşmasına ve paylaşılan notları filtreleyerek indirebilmesine odaklanan mobil istemci uygulamasıdır.

Uygulama, Flutter ile geliştirilmiş cross-platform bir yapıdadır ve Material Design 3 bileşenleri ile kullanıcı akışlarının hızlı ve anlaşılır şekilde yürütülmesi hedeflenmiştir. Sistem, backend servisleriyle RESTful API üzerinden haberleşir; kimlik doğrulama JWT ile sağlanır ve yetkilendirme gerektiren ekranlar yalnızca giriş yapan kullanıcılar tarafından erişilebilir olacak şekilde kurgulanır. Notların sınıf seviyesi ve dönem bilgisiyle ilişkilendirilmesi, kullanıcıların aradıkları içeriğe hızlı ulaşmasını sağlar.

1.1 UniNotes Nedir

UniNotes; öğrencilerin ders notlarını yükleyip paylaşabildiği, notları sınıf ve dönem bazında filtreleyerek görüntüleyebildiği ve indirebildiği bir akademik paylaşım platformudur. UniNotes Mobile ise bu platformun mobil uygulama arayüzünü sunar ve kullanıcıların aynı işlemleri mobil cihazlar üzerinden gerçekleştirmesini hedefler. Uygulama, notların sadece dosya olarak değil; ders kodu, sınıf seviyesi, dönem, açıklama gibi metaverilerle birlikte yönetilmesini sağlayan bir istemci deneyimi sunar.

1.2 Amaç ve Kapsam

UniNotes Mobile uygulamasının temel amacı, ders notu paylaşımını mobil cihazlar üzerinden güvenli ve organize bir yapı altında mümkün kılarak öğrencilerin akademik kaynaklara erişimini kolaylaştırmaktır. Projenin kapsamı aşağıdaki bileşenleri içermektedir:

- Not görüntüleme, filtreleme ve indirme akışları
- Not ekleme, dosya seçme ve dosya yükleme süreçleri

- Kullanıcı kayıt/giriş işlemleri ve oturum yönetimi
- Profil ekranı üzerinden kullanıcı bilgileri ve yüklenen notlar
- İletişim formu ile geri bildirim gönderimi
- Raporlama diyaloğu ile uygunsuz içerik bildirimi
- Web ve mobil platformlar için farklı ağ/dosya davranışlarının yönetimi

1.3 Çalışmanın Önemi

UniNotes Mobile, mobil cihazların erişilebilirliğini kullanarak akademik içerik paylaşımını pratik hale getirir. Flutter ile geliştirilen tek kod tabanının Android, iOS ve web üzerinde çalıştırılabilmesi, geliştirme ve bakım maliyetini düşürürken uygulamanın yaygınlaşmasını destekler. Ayrıca Provider tabanlı durum yönetimi, servis katmanı ile API çağrılarının merkezi yönetimi ve güvenli token saklama yaklaşımı; gerçek hayatta kullanılabilir, sürdürülebilir bir istemci mimarisi sunar.

1.4 Raporun Yapısı

Bu rapor altı ana bölümden oluşmaktadır. Birinci bölümde projenin amacı, kapsamı ve genel tanımı sunulmuştur. İkinci bölümde geliştirme ortamı, kullanılan teknolojiler ve uygulamanın mimari yaklaşımı açıklanmıştır. Üçüncü bölümde Flutter ön yüz bileşenleri, ekranlar, durum yönetimi ve API entegrasyonu ayrıntılı olarak incelenmiştir. Dördüncü bölümde .NET tabanlı backend mimarisi; güvenlik (JWT), veritabanı (MS-SQL/EF Core) ve API servis altyapısı üzerinden detaylandırılmıştır. Beşinci bölümde uygulama; kullanılabilirlik, performans, güvenlik ve platform uyumluluğu açısından değerlendirilmiştir. Altıncı bölümde sonuç ve gelecekteki geliştirme önerileri sunulmuştur.

2 PROJE ORTAMI VE KULLANILAN TEKNOLOJİLER

Bu bölümde UniNotes Mobile projesinde kullanılan geliştirme araçları, teknoloji yığını ve mobil uygulamanın genel mimari yaklaşımı açıklanmaktadır.

2.1 Geliştirme Ortamı

Proje geliştirme sürecinde kaynak kod yönetimi Git üzerinden yürütülmüş; Flutter projesi modüler bir yapıda organize edilmiştir. Geliştirme ve hata ayıklama süreçlerinde Android Studio veya Visual Studio Code kullanılmıştır. Flutter SDK ve Dart SDK sürümleri proje bağımlılıklarıyla uyumlu olacak şekilde yapılandırılmıştır. Uygulama, emülatör ve gerçek cihaz üzerinde test edilmiştir. Backend iletişimi geliştirme sırasında yerel ortamda yürütülmüş; web platformunda `localhost`, Android emülatörde ise `10.0.2.2` üzerinden erişim sağlanmıştır.

2.2 Projenin Genel Teknik Yapısı

UniNotes Mobile, istemci-sunucu mimarisinde çalışan bir mobil istemci uygulamasıdır. Uygulama tarafında kullanıcı arayüzü Flutter widget ağacı ile oluşturulur; API istekleri `http` paketi üzerinden backend servislere gönderilir. Kimlik doğrulama durumsuz bir token yaklaşımı ile yürütülür ve token her istek için `Authorization` başlığı üzerinden iletilir. Uygulama içerisinde durum yönetimi Provider ile sağlanır; kullanıcı kimliği, giriş durumu ve oturum sürekliliği merkezi bir sağlayıcı üzerinden yönetilir.

2.3 Kullanılan Yazılım Teknolojileri

- Flutter 3.4.0+ ve Dart 3.4.0+
- Material Design 3
- Provider 6.1.1 ve ChangeNotifier
- http 1.2.0
- shared_preferences 2.2.2 ve flutter_secure_storage 9.0.0
- file_picker 8.0.5, path_provider 2.1.1

- intl 0.19.0, url_launcher 6.3.0
- flutter_lints 4.0.0, flutter_test

2.4 Sistem Mimarisi

Uygulama, clean architecture yaklaşımını takip eden katmanlı bir dizin yapısına sahiptir. Ekranlar ve widget'lar sunum katmanında; servisler API iletişimini ve iş kurallarını kapsayan katmanda; modeller veri temsili ve serileştirme işlemlerinde; config ve utils bileşenleri ise ortam bağımlı yapılandırma ve platforma özel yardımcı fonksiyonları üstlenir. Bu ayrım, kodun okunabilirliğini artırır ve yeni ekran/özellik eklerken bağımlılıkların kontrol edilebilir kalmasını sağlar.

3 YAZILIM GERÇEKLEME SÜRECİ

Bu bölümde UniNotes Mobile uygulamasının Flutter tabanlı ön yüz katmanı, ekranlar arası akışlar, durum yönetimi, servis katmanı ve dosya işlemleri ayrıntılı olarak açıklanmaktadır. Uygulama; giriş ve kayıt, paylaşılan notları filtreleyip listeleme, not yükleme, profil yönetimi, iletişim ve raporlama işlevleri etrafında modüler biçimde geliştirilmiştir.

3.1 Frontend Gerçekleme Süreci (Flutter)

3.1.1 Geliştirilen Modüller

UniNotes Mobile sistemi aşağıdaki temel modüller etrafında geliştirilmiştir:

- Kimlik doğrulama ve oturum yönetimi modülü
- Paylaşılan notlar listeleme ve filtreleme modülü
- Not ekleme ve dosya yükleme modülü
- Profil ekranı ve kullanıcı içerikleri modülü
- İletişim formu modülü
- Raporlama diyaloğu ve rapor nedenleri modülü
- Platforma özgü indirme ve dosya erişim yardımcıları

3.1.2 Ön Yüz Tasarımı

UniNotes Mobile ön yüzü, Flutter widget ekosistemi kullanılarak bileşen tabanlı bir yaklaşımla geliştirilmiştir. Uygulama akışı, sayfa temelli ekranlar ve ortak yeniden kullanılabilir widget'lar üzerinden kurgulanmış; Material Design 3 prensipleriyle tutarlı bir arayüz oluşturulmuştur. Mor tema paleti uygulama genelinde merkezi tema dosyası üzerinden yönetilerek AppBar, butonlar, kartlar, input alanları ve ikincil aksiyonlar arasında görsel süreklilik sağlanmıştır.

Uygulamada temel hedefler; kullanıcıların paylaşılan notlara hızlı erişmesi, filtreleme ile listeyi daraltabilmesi, dosya indirme ve yükleme işlemlerini hata toleranslı şekilde tamamlayabilmesi ve oturum yönetiminin kullanıcıya yük bindirmeden sürdürülebilmesidir. Bu hedefler doğrultusunda ekranlar arası geçişlerde sade ve öğrenilebilir

bir gezinim yapısı benimsenmiş; bottom navigation ile ana fonksiyonlara tek dokunuşla erişim sağlanmıştır.

3.1.3 Ön Yüz Proje Yapısı ve Dizin Organizasyonu

Flutter projesi, sorumluluk ayrımı prensibine uygun olarak `lib` dizini altında katmanlı şekilde organize edilmiştir:

- `lib/screens/`: Tam sayfa ekranlar (ör. `landing_page.dart`, `login_page.dart`, `register_page.dart`, `shared_notes_page.dart`, `add_notes_page.dart`, `profile_page.dart`, `contact_page.dart`, `about_page.dart`). Ayrıca `main_scaffold.dart` bottom navigation ve ortak layout düzenini yönetir.
- `lib/widgets/`: Ortak bileşenler (ör. `report_dialog.dart`).
- `lib/providers/`: Durum yönetimi sağlayıcıları (ör. `auth_provider.dart`).
- `lib/services/`: API servisleri (ör. `api_service.dart`, `auth_service.dart`).
- `lib/config/`: Ortam bağımlı ayarlar (ör. `api_config.dart`).
- `lib/models/`: Veri modelleri ve JSON serileştirme (ör. `user.dart`).
- `lib/utils/`: Platforma özel yardımcılar (ör. `file_downloader.dart`).
- `lib/theme.dart`: Renk, tipografi ve bileşen stilleri.
- `lib/main.dart`: Uygulama giriş noktası (tema, Provider kayıtları, başlangıç routing).

3.1.4 Ekranlar Arası Gezinim ve Ana Scaffold Yapısı

Ana gezinim, `main_scaffold.dart` üzerinden sağlanan bottom navigation ile yürütülür. Ana sekmeler:

- Home
- Shared Notes
- Add Notes

- Profile

Ekran geçişleri Flutter Navigator yapısı ile yönetilir. Kimlik doğrulama durumuna göre başlangıç ekranı dinamik belirlenir; kullanıcı giriş yapmamışsa login akışına, giriş yapmışsa main scaffold akışına yönlendirilir.

3.1.5 Kimlik Doğrulama Akışı ve Oturum Yönetimi

Kullanıcı giriş yaptığında backend tarafından üretilen JWT token, güvenli saklama katmanına kaydedilir:

- Giriş/kayıt ekranlarında form doğrulama tamamlanır.
- `auth_service.dart` ile ilgili endpoint'e istek atılır.
- Yanıttan kullanıcı bilgisi ve token ayrıştırılır.
- Token, hassas veri olduğu için `flutter_secure_storage` üzerinde saklanır.
- Uygulama açılışında `AuthProvider` token varlığını kontrol eder ve oturum durumunu günceller.
- Token gerektiren tüm isteklerde `api_service.dart` `Authorization` başlığına token ekler.

3.1.6 Paylaşılan Notlar Ekranı Tasarımı ve Listeleme Modeli

Paylaşılan notlar ekranı `shared_notes_page.dart` ile sunulur. Üst bölümde filtre alanları, alt bölümde kart tabanlı liste bulunur. Kartlarda ders/not bilgisi, paylaşan kullanıcı, tarih gibi metaveriler gösterilir; indirme ve raporlama aksiyonları aynı kart üzerinde sunulur.

3.1.7 Not Ekleme Ekranı Tasarımı ve Kontrollü Veri Girişi

Not ekleme ekranı `add_notes_page.dart` üzerinde kurgulanmıştır. Kullanıcı; başlık/açıklama, sınıf/dönem, dinamik ders listesi seçimi ve dosya seçimi adımlarını tamamladıktan sonra yükleme işlemini başlatır. Dosya seçimi `file_picker` ile yapılır. PDF, DOCX, PPTX ve TXT gibi formatlar desteklenir. Yükleme isteği `multipart/form-data` şeklinde backend'e gönderilir.

3.1.8 Profil Ekranı ve Kullanıcı İçerikleri

Profil ekranı `profile_page.dart` üzerinden sunulur. Kullanıcı bilgileri, kullanıcının yüklediği notların listesi ve temel istatistikler yer alır. Çıkış işleminde token temizlenir ve kullanıcı login akışına yönlendirilir.

3.1.9 Raporlama Diyaloğu ve İçerik Denetimi Arayüzü

Raporlama işlevi, `report_dialog.dart` ile standartlaştırılmış bir dialog bileşenidir. Backend'den alınan rapor nedenleri kontrollü seçenekler halinde sunulur ve seçime göre rapor endpoint'ine gönderim yapılır.

3.1.10 Mobil Tarafta API Entegrasyonu

UniNotes Mobile, backend servisleriyle RESTful API üzerinden haberleşir. Mobil tarafta entegrasyonun karşılığı; API entegrasyon mimarisi, servis katmanı, token yönetimi, hata yakalama stratejisi ve platforma özgü ağ yapılandırmalarıdır.

3.1.10.1 API Konfigürasyonu ve Platforma Göre Base URL Seçimi

Flutter uygulaması hem web hem mobil hedeflediği için base URL seçimi platforma göre farklılık göstermektedir. Android emülatörde host makineye erişim için `10.0.2.2` IP'si kullanılırken; web platformunda `localhost` kullanılmaktadır. Bu ayrım, `lib/config/api_config.` dosyasında merkezi şekilde yönetilir.

3.1.10.2 Servis Katmanı Yaklaşımı

API çağrıları doğrudan ekran dosyalarında yapılmak yerine `lib/services/` altında yer alan servisler üzerinden yürütülür. `api_service.dart` ortak header yönetimi, token ekleme, metod standardizasyonu, durum kodu ayrıştırma ve hata yönetimi görevlerini üstlenir. `auth_service.dart` ise login/register çağrılarını kapsar.

3.1.10.3 Endpoint Haritalaması

- Authentication: register ve login
- Notes: listeleme, kullanıcının notları, oluşturma, silme

- Files: yükleme ve indirme
- Courses: sınıf/döneme göre ders listesi
- Reports: rapor nedenleri ve rapor gönderimi
- Contact: iletişim mesajı gönderimi

3.1.10.4 Token Yönetimi ve Hata Yakalama

JWT token `flutter_secure_storage` üzerinde saklanır ve her istekte `Authorization` başlığına eklenir. Ağ hataları try-catch ile yakalanır; 401 gibi durumlarda kullanıcı yeniden girişe yönlendirilebilir. Uzun süren işlemlerde loading göstergeleri kullanılır.

3.2 Backend Gerçekleme Süreci (.NET / ASP.NET Core Web API)

UniNotes sisteminin sunucu tarafı ASP.NET Core Web API ile geliştirilmiştir. Backend; mobil istemci (Flutter) ve web istemcilerin kullandığı REST uçlarını sunar, kimlik doğrulama ve yetkilendirmeyi JWT tabanlı olarak yönetir ve veriyi MSSQL üzerinde Entity Framework Core aracılığıyla kalıcı hale getirir. Bu bölümde uygulama başlangıç konfigürasyonu (Program.cs), katmanlı mimari, veri sözleşmesi (DTO), güvenlik yaklaşımı, dosya yönetimi ve mobil istemci entegrasyonunu doğrudan etkileyen performans/kararlılık bileşenleri ayrıntılı olarak açıklanmaktadır.

3.2.1 Arka Uç Mimari Yaklaşımı ve Mobil Entegrasyon Prensipleri

UniNotes API, RESTful servis tasarımına uygun olacak şekilde **durumsuz (stateless)** istek/yanıt modeli ile çalışır. Mobil istemci tarafında her istek, gerekli kimlik bilgisini `Authorization: Bearer <token>` başlığı ile taşır. Bu sayede:

- Sunucu tarafında oturum saklama ihtiyacı ortadan kalkar,
- Yatay ölçeklenebilirlik artar,
- Mobil istemci için ağ kopmaları/yeniden denemeler daha yönetilebilir hale gelir.

Mobil kullanım senaryoları (düşük bant genişliği, dalgalı bağlantı, sınırlı pil/CPU) göz önüne alınarak API cevapları **gereksiz alanlar taşımayan DTO'lar** üzerinden minimize edilir; listeleme uçlarında **filtreleme ve sayfalama (pagination)** yaklaşımı benimsenir. Dosya indirme/yükleme süreçleri ise **stream tabanlı** ele alınarak yüksek bellek tüketiminin önüne geçilir.

3.2.2 Uygulama Başlangıcı ve Middleware Zinciri (Program.cs)

Backend'in çalışma zamanı konfigürasyonu `Program.cs` dosyasında oluşturulan `WebApplication` hattı üzerinden yürütülür. Uygulama; servis kayıtları (Dependency Injection), veritabanı bağlantısı, güvenlik (JWT Authentication/Authorization), CORS politikası ve Swagger gibi geliştirme yardımcılarını bu dosyada merkezi olarak yapılandırır.

3.2.2.1 Controller Altyapısı ve Endpoint Yayınlama

Projede REST API uçlarının çalışması için controller altyapısı aktiftir:

- `AddControllers()`: API controller'larının (JSON tabanlı) yayınlanması.
- `app.MapControllers()`: `[ApiController]` ile işaretli controller'ların route edilmesi.

3.2.2.2 Veritabanı Bağlantısı, DbContext ve EF Core

Backend, MSSQL veritabanına Entity Framework Core aracılığıyla bağlanır. `UniNotesContext` sınıfı DbContext olarak projeye eklenmiş ve `appsettings.json` içindeki `UniNotesConnection` connection string üzerinden yapılandırılmıştır:

- `AddDbContext<UniNotesContext>(...UseSqlServer(...))` ile SQL Server sağlayıcısı kullanılır.
- Varlıklar ve ilişkiler EF Core modelleme ile tanımlanır; şema değişiklikleri migration ile versiyonlanır.

3.2.2.3 Dependency Injection (DI) ile Servis Kayıtları

- `JwtService`: token üretimi, claim yönetimi.
- `IEmailService` → `EmailService`: iletişim/geri bildirim.

3.2.3 Arka Yüz Proje Yapısı ve Katmanlaşma

Backend dizin yapısı; controller, servis, veri erişim ve DTO katmanları ile modüler hale getirilmiştir:

- `Controllers/Api/`: HTTP isteklerini karşılar.
- `DTOs/`: İstemci-sunucu veri sözleşmesi.
- `Models/`: Domain varlıkları.
- `Data/`: DbContext ve konfigürasyonlar.
- `Services/`: İş kuralları.
- `Migrations/`: Şema versiyonlama.

3.2.4 Güvenlik: JWT Authentication ve Authorization

JWT doğrulama; issuer, audience, imza anahtarı ve süre kontrolleri ile yürütülür. `[Authorize]` ile işaretli uçlar korunur. Kaynak sahipliği gerektiren işlemlerde token içindeki kullanıcı kimliği ile kaydın `UserId` alanı eşleştirilir; uyuşmazlık durumunda `403 Forbidden` döndürülür.

3.2.5 CORS Yapılandırması

Geliştirme sürecinde `"AllowAll"` politikası ile `AllowAnyOrigin`, `AllowAnyMethod`, `AllowAnyHeader` uygulanır. Üretimde whitelist tabanlı yaklaşım tercih edilmelidir.

3.2.6 Swagger / OpenAPI

Swagger geliştirme ortamında aktiftir. Swagger UI üzerinde Bearer token tanımı ile JWT'li test imkânı sağlanır.

3.2.7 Controller Sorumlulukları

Auth/User, Notes, Files, Courses, Tags, Reports/ReportReasons, Favorites/Downloads ve Contact controller'ları; mobil istemciye yönelik tutarlı HTTP kodları ve öngörülebilir response sözleşmeleri ile tasarlanmıştır.

3.2.8 DTO Katmanı, Migration ve Dosya Doğrulama

DTO yaklaşımı hassas verileri sızıntıya karşı korur ve payload'ı azaltır. Migration ile şema izlenebilir hale gelir. Dosya yüklemede uzantı/MIME, boyut limitleri ve content-type doğrulaması uygulanır.

3.2.9 Hata Yönetimi ve Durum Kodları

API; `401`, `403`, `400`, `404` gibi durum kodlarını tutarlı biçimde kullanarak Flutter tarafında hata yakalama ve kullanıcıya anlamlı mesaj üretme süreçlerini standartlaştırır.

3.2.10 Routing ve Endpoint Haritalama

- `UseRouting()`
- `UseCors("AllowAll")`

- `UseAuthentication()` ve `UseAuthorization()`
- `MapControllers()`

3.3 Veri Tabanı Tasarımı

UniNotes projesinde veri tabanı tasarımı; kullanıcı hesap yönetimi, ders-not eşleştirme, dosya saklama metaverileri, kullanıcı etkileşimleri (favori/indirme geçmişi) ve içerik denetimi (raporlama) gibi platformun tüm işlevsel gereksinimlerini karşılayacak şekilde kurgulanmıştır. Sistem, ilişkisel veri modelinin sağladığı tutarlılık ve bütünlük avantajları nedeniyle Microsoft SQL Server üzerinde tasarlanmış ve Entity Framework Core aracılığıyla Code First yaklaşımı ile yönetilmiştir. Bu yaklaşım sayesinde tablo şemaları, ilişkiler ve kısıtlar C# entity modelleri üzerinden oluşturulmuş; şema değişiklikleri ise `__EFMigrationsHistory` tablosu ile izlenebilir hale getirilmiştir.

UniNotes veri tabanı şeması; birincil anahtarlar (Primary Key), yabancı anahtarlar (Foreign Key) ve bire-çok (One-to-Many) ilişkiler üzerinden veri tutarlılığını garanti altına alacak şekilde tasarlanmıştır. Şemada sistemin merkezinde `Users` ve `Notes` tabloları yer almakta; diğer tablolar bu çekirdek varlıkların etrafında konumlanmaktadır: not dosyaları (`Files`), indirme geçmişi (`Downloads`), favoriler (`Favorites`), raporlar (`Reports` + `ReportReasons`), ders kataloğu (`Courses`), etiketleme (`Tags`) ve kullanıcı geri bildirimi (`ContactMessages`).

3.3.1 Şema Yönetimi ve Migration Stratejisi

Veri tabanı şeması, Entity Framework Core migration mekanizmasıyla yönetilmektedir. `__EFMigrationsHistory` tablosu, uygulanan migration kayıtlarını ve EF sürüm bilgisini tutarak şema değişimlerinin izlenebilirliğini sağlar. Bu yaklaşım; farklı geliştirme ortamlarında aynı veritabanı yapısının hızlı ve hatasız şekilde kurulmasını destekler ve manuel SQL ihtiyacını azaltır.

3.3.2 Temel Varlıklar ve Alanların Gerekçelendirilmesi

3.3.2.1 Users (Kullanıcılar)

`Users` tablosu kimlik ve sahiplik merkezidir. `Id` birincil anahtar olup `Name`, `Email` gibi alanlar arayüzde profil ve paylaşım gösterimlerinde kullanılır. `Password` alanı düz metin değil hashlenmiş biçimde saklanacak şekilde tasarlanmıştır. `CreatedAt` ve `UpdatedAt` zaman damgaları denetim ve yaşam döngüsü takibi için eklenmiştir. Users tablosu; `Notes`, `Files`, `Downloads`, `Favorites` ve `Reports` tabloları ile ilişkili çalışır.

3.3.2.2 Courses (Dersler)

`Courses` tablosu notların ders bağlamında standardize edilmesi için kullanılır. `CourseCode` ve `CourseName` temel tanımlayıcı alanlardır. `ClassLevel` ve `Semester` alanları, ön yüzde filtreleme ve seçim süreçlerini destekler. (Varsa) `IsElective` alanı ders kataloğunu daha doğru modellemeyi sağlar.

3.3.2.3 Notes (Notlar)

`Notes` UniNotes'in çekirdek içerik varlığıdır. Not; `UserId` ile sahibine bağlanır, `Title` ve `Summary` ile açıklanır, `CourseCode` ile ders bağlamına yerleştirilir. `IsDeleted` alanı ile soft delete yaklaşımı uygulanır; bu sayede yanlış silmeler ve moderasyon/denetim izleri kaybolmadan yönetilebilir.

3.3.2.4 Files (Dosyalar)

`Files` tablosu notlara ait dosya metaverilerini temsil eder. `NoteId` ile nota bağlanır, `UserId` ile sahiplik kontrolü desteklenir. `FileUrl` fiziksel saklama konumuna referans verir. Not–dosya ilişkisinde yetim dosya oluşumunu önlemek için yabancı anahtar bütünlüğü kritik rol oynar.

3.3.2.5 Downloads (İndirme Geçmişi)

`Downloads`; `UserId`, `NoteId` ve `FileId` alanları üzerinden indirme olayını kaydeder. Böylece kullanıcı tarafında indirme geçmişi gösterimi, kullanım analitiği ve denetim senaryoları mümkün hale gelir.

3.3.2.6 ReportReasons ve Reports

`ReportReasons` raporlama nedenlerini sözlük mantığında tutar ve rapor girişlerini standardize eder. `Reports` tablosu, raporlanan not (`NoteId`), raporlayan kullanıcı (`UserId`) ve seçilen neden (`ReportReasonId`) ilişkisini kaydeder. Bu yapı, moderasyon için işlenebilir rapor verisi üretir.

3.3.2.7 ContactMessages (İletişim Mesajları)

`ContactMessages` tablosu, kullanıcı geri bildirimlerini saklar. `Name`, `Email`, `Subject`, `Message` alanları içeriği taşır; (varsa) `IsRead` alanı basit iş akışı takibi sağlar.

3.3.3 Tablolar Arası İlişkiler ve Kardinalite

- **Users** $\rightarrow$ **Notes (1-N):** Bir kullanıcı birden fazla not oluşturabilir.
- **Notes** $\rightarrow$ **Files (1-N):** Bir nota birden fazla dosya bağlanabilir.
- **Users** $\rightarrow$ **Files (1-N):** Kullanıcı birden fazla dosya yükleyebilir.
- **Users**/**Notes** $\rightarrow$ **Favorites (1-N):** Favoriler kullanıcı-not ilişkisini kurar.
- **Users**/**Notes**/**Files** $\rightarrow$ **Downloads (1-N):** İndirme geçmişi çok boyutlu izlenebilir.
- **Notes**/**Users** $\rightarrow$ **Reports (1-N):** Notlar raporlanabilir, kullanıcılar rapor oluşturabilir.
- **ReportReasons** $\rightarrow$ **Reports (1-N):** Nedenler raporlarda tekrar kullanılabilir.

3.3.4 Veri Tutarlılığı, Kısıtlar ve Performans Perspektifi

Yabancı anahtar kısıtları, yetim kayıt oluşumunu önler ve doğal iş akışına uygun veri bütünlüğü sağlar. Sık sorgulanan alanlar (örn. `UserId`, `NoteId`, `CourseCode`) için indeksleme yaklaşımı performans açısından uygundur. Bu rapor kapsamında indekslerin fiziksel tanımı ayrıntılandırılmasa da ilişkisel tasarım performans optimizasyonuna açık bir zemin sunar.

4 DEĞERLENDİRME

Bu bölümde UniNotes Mobile uygulaması; kullanılabilirlik, arayüz tutarlılığı, performans, güvenlik ve platform uyumluluğu açısından değerlendirilmiştir. Değerlendirmede temel kullanıcı akışları (giriş/kayıt, not listeleme/filtreleme, not yükleme, dosya indirme, profil yönetimi, raporlama ve iletişim) esas alınmış; Flutter'ın cross-platform yapısının sağladığı avantajlar ve mobil uygulama geliştirme sürecinde ortaya çıkan pratik kısıtlar ele alınmıştır.

4.1 Kullanılabilirlik Değerlendirmesi

UniNotes Mobile'da kullanıcıların en sık kullanacağı ekranlara hızlı erişim sağlamak için bottom navigation yaklaşımı tercih edilmiştir. Bu sayede kullanıcı, paylaşılan notlar ve not ekleme gibi ana işlevlere minimum adım ile ulaşabilir. Paylaşılan notlar ekranında sınıf ve dönem filtrelerinin üstte konumlandırılması, listeyi daraltma davranışını görünür ve anlaşılır hale getirir. Not ekleme ekranında dinamik ders listesi kullanılması, serbest metin kaynaklı ders kodu hatalarını azaltır ve veri tutarlılığına katkı sağlar.

Form tabanlı ekranlarda istemci tarafı validasyon, boş alan veya hatalı email formatı gibi sorunları kullanıcıya işlem öncesinde gösterir. Dosya seçimi ve yükleme adımlarında dosya adı ve uzantı bilgisinin gösterilmesi, kullanıcı davranışlarını daha kontrollü hale getirir. Raporlama diyaloğunun kart üzerinde hızlı erişilebilir olması, içerik denetimini kullanıcı açısından kolaylaştırır.

4.2 Arayüz ve Duyarlı Tasarım Değerlendirmesi

Uygulamada Material Design 3 bileşenlerinin kullanılması, platformlar arası görsel ve davranış tutarlılığı sağlar. Mor tema paletinin merkezi temadan yönetilmesi, ekranlar arası tasarım farklılıklarını azaltır. Kart tabanlı listeleme yaklaşımı, metin yoğun içeriklerin okunabilirliğini artırır. Responsive tasarım için esnek layout bileşenleri kullanılması, farklı ekran boyutlarında bileşenlerin taşmasını ve hizalama problemlerini azaltır.

4.3 Güvenlik Değerlendirmesi

Oturum yönetiminin JWT token yaklaşımı ile yürütülmesi, durumsuz mimariyi destekler. Token'ın güvenli saklama alanında tutulması, istemci tarafı güvenliği açısından önemli bir önlemdir. API servis katmanında token'ın otomatik eklenmesi, geliştirici hatalarını azaltır ve her istekte aynı güvenlik standardının uygulanmasını sağlar. Logout işleminde token temizliği ve provider state sıfırlaması, oturumun güvenli biçimde sonlandırılmasına katkı sunar.

4.4 Performans ve Ölçeklenebilirlik Değerlendirmesi

Flutter'ın widget tabanlı render yaklaşımı, doğru state yönetimi ile birlikte performanslı bir kullanıcı deneyimi sağlayabilir. Liste ekranlarında yükleme durumlarının yönetilmesi ve gereksiz rebuild'lerin azaltılması, akıcılığı artırır. Servis katmanında gereksiz veri taşınmaması ve yanıtların kontrollü biçimde işlenmesi, ağ maliyetini düşürür. İleride not sayısı arttığında pagination veya lazy loading yaklaşımı performansı daha sürdürülebilir hale getirebilir.

4.5 Platform Uyumluluğu ve Teknik Kısıtlar

Cross-platform geliştirme sayesinde aynı kod tabanıyla Android, iOS ve web hedeflenebilmiştir. Bununla birlikte dosya indirme ve ağ erişimi gibi alanlarda platforma özgü farklılıklar bulunmaktadır. Web ortamında CORS ve tarayıcı indirme davranışları; Android emülatörde localhost erişimi gibi konular konfigürasyon katmanında çözülmüştür. Desktop hedefleri deneysel olduğundan dosya yolları ve pencere davranışlarında ek test ihtiyacı doğabilir.

4.6 İyileştirme Önerileri

Uygulama temel senaryoları karşılasa da ileride geliştirilebilecek alanlar bulunmaktadır:

- Anahtar kelime bazlı arama ve gelişmiş filtreleme
- PDF için uygulama içi önizleme

- Karanlık tema desteği
- Favoriler ve çevrimdışı erişim
- Raporlama ve iletişim mesajları için yönetici arayüzü ile moderasyon süreçleri
- Not verilerini dinamik map yerine tip güvenli model yapısına dönüştürme

5 SONUÇ

Bu çalışma kapsamında, üniversite öğrencilerinin akademik içerikleri mobil cihazlar üzerinden yapılandırılmış biçimde paylaşabilmesi amacıyla geliştirilen UniNotes Mobile uygulaması tasarlanmış ve uygulanmıştır. Proje, Flutter tabanlı tek kod tabanı ile Android, iOS ve web platformlarını hedefleyen bir istemci çözümü sunmuştur. Provider tabanlı durum yönetimi, servis katmanı ile merkezi API entegrasyonu ve güvenli token saklama yaklaşımı ile sürdürülebilir bir mobil mimari elde edilmiştir.

Uygulama; paylaşılan notları filtreleyerek listeleme, dosya indirme, not yükleme, profil yönetimi, raporlama ve iletişim akışlarını destekleyerek gerçek kullanım senaryolarına uygun bir temel ortaya koymaktadır. İleride gelişmiş arama, önizleme, karanlık tema ve kişiselleştirme özellikleriyle genişletilmesi halinde, UniNotes Mobile daha kapsamlı bir akademik paylaşım ekosistemine dönüşebilecek niteliktedir.