

DERS KİTABI SATIŞ UYGULAMASI :

NOCTUA

Elif Sena Polat

Kocaeli Üniversitesi
Bilişim Sistemleri Mühendisliği, Teknoloji Fakültesi

Kocaeli, Türkiye
elifspolatt@gmail.com

ÖZET

Bu mobil uygulamada, kullanıcıların fiziksel kitapçılara gitme zorunluluğu olmadan, mobil cihazları üzerinden kitapları inceleyebilecekleri ve satın alma süreçlerini gerçekleştirebilecekleri Android tabanlı bir Mobil Kitap Satış Uygulaması geliştirilmiştir. Mobil uygulama, güncel yazılım geliştirme standartlarına uygun olarak Kotlin programlama dili ile geliştirilmiş olup, veri depolama ve gerçek zamanlı senkronizasyon işlemleri için bulut tabanlı Firebase Realtime Database teknolojisi kullanılmıştır. Uygulama mimarisi, kullanıcı dostu bir arayüz ve yüksek performanslı veri akışı sağlayacak şekilde kurgulanmıştır. Geliştirilen projede; kullanıcı tarafında kimlik doğrulama, ürün listeleme, favorileme ve kişiselleştirilmiş sepet yönetimi modülleri kullanılmıştır. Admin kısmında ise ürünler ve siparişleri takip etme, ürünlerin görsellerini yüklemeye, kullanıcıları listeleme gibi bölümlerden oluşmuştur. Deneysel sonuçlar, uygulamanın farklı Android cihazlarda kararlılığını ve NoSQL veri yapısı sayesinde veri erişim hızının optimize edildiği görülmektedir. Bu proje, modern mobil uygulama geliştirme süreçlerine ve bulut tabanlı veri yönetimine dair kapsamlı bir örnek teşkil etmektedir.

Anahtar Kelimeler—Android, Kotlin, Mobil Uygulama, E-ticaret uygulaması, Kitap Uygulaması

I. GİRİŞ

Mobil teknolojilerin son on yılda gösterdiği hızlı gelişim, bilgiye erişim algısını ve e-ticaret alışkanlıklarını köklü bir şekilde değiştirmiştir. Günümüzde akıllı telefonlar, sadece iletişim aracı olmaktan çıkmış, eğitimden alışverişe kadar birçok günlük aktivitenin merkezi haline gelmiştir. Özellikle kitap gibi fiziksel veya dijital materyallere erişimin kolaylaştırılması, eğitim ve akademik açıdan önemlidir. Geleneksel kitap satış yöntemleri, zaman ve mekân kısıtlamaları nedeniyle kullanıcıların isteklerini karşılamakta yetersiz kalabilmektedir. Bundan dolayı, kullanıcıların diledikleri yerden kitapları inceleyebilecekleri, fiyat ve içerik bilgisine ulaşabilecekleri ve güvenli bir şekilde alışveriş yapabilecekleri mobil platformlara ve sistemlere olan ihtiyaç artmıştır. Bu projede, Android işletim sistemi üzerinde çalışan, bulut sistemi tabanlı bir Mobil Kitap Satış Uygulaması tasarlanmıştır. Uygulamanın temel amacı, kullanıcılarla kullanıcı dostu bir arayüz sunarak istedikleri test kitaplarını inceleme ve aradıkları kitapları satın alma süreçlerini kolaylaştırmaktır. Uygulama, mobil uygulama geliştirme standartlarına uygun olarak Kotlin programlama

dili ile Android Studio platformunda geliştirilmiştir. Veri yönetimi ve depolama işlemleri için ise Google'in sunduğu Firebase Realtime Database teknolojisi kullanılmıştır. Bu sayede, uygulamadaki kitap adetleri, kullanıcı bilgileri ve sepet hareketleri sunucu tarafında anlık olarak senkronize şekilde görülebilmektedir.

Bu uygulama kullanıcı ve admin şeklinde iki bölüm etrafında tasarlanmıştır. Kullanıcı kısmında kullanıcıların uygulamaya güvenli bir şekilde kayıt olması, giriş yapması, ürünleri favorilere ve septe ekleyip çıkarabilme gibi özellikler bulunmaktadır. Yine kullanıcı tarafında hesap bilgilerini güncelleyebilecekleri ve şifre ya da e-posta bilgilerini değiştirebilecekleri bölümler vardır. Kullanıcılar, uygulama içerisindeki RecyclerView bileşeni sayesinde yüzlerce kitabı performans kaybı yaşamadan listeleyebilmekte ve Glide gibi kütüphaneler aracılığıyla kitap kapak görsellerini internet üzerinden hızla yükleyebilmektedir. Admin kısmında ise adminin ürün ekleyip çıkarabileceği, ürün bilgilerini güncelleyebileceği ekranlar olup, sipariş ve stok takibi yapabildiği özellikler mevcuttur.

Bu raporun ilerleyen bölümlerinde; uygulamanın geliştirilmesinde kullanılan yöntem ve teknolojiler, sistemin mimari tasarımını, veritabanı şeması ve akış diyagramları, uygulamanın çalışma performansını gösteren deneysel sonuçlar ve ekran görüntüleri ele alınmıştır. Sonuç bölümünde ise projenin kazanımları konuşulmuştur. Bu çalışma, Android tabanlı modern bir e-ticaret uygulamasının uçtan uca nasıl geliştirileceğine dair bir örnek olmuştur.

II. YÖNTEM

A. Yazılım Geliştirme Ortamı ve Dili

Uygulamanın geliştirilmesinde Entegre Geliştirme Ortamı (IDE) olarak Android Studio programı kullanılmıştır. Programlama dili olarak ise Google tarafından Android geliştirme için önerilen Kotlin dili tercih edilmiştir. Kotlin'in kullanılmasındaki temel nedenlerden bazıları çalışma zamanı hatalarını minimize etmesi ve modern fonksiyonel programlama özelliklerini desteklemesidir. Proje yapısı incelediğinde, nesne yönelimli programlama prensiplerinin uygulandığı görülmektedir. Data class yapısı buna bir örnek olarak verilebilir. Kitap sınıfı; yazar, kitap adı, resim URL'si, fiyat ve stok bilgisi gibi özellikleri barındırırken, Kullanıcı sınıfı; e-posta, şifre ve kullanıcı adı içeriği gibi verileri modellemektedir. Arayüz tasarımda ise XML tabanlı layout dosyaları kullanılmış, ConstraintLayout ve LinearLayout gibi görünüm grupları ile farklı ekran boyutlarına uyumlu ve anlaşılabılır bir tasarım elde edilmiştir.

B. Veri Tabanı Mimarisi (Firebase Realtime Database)

Uygulamanın backend kısmında, gerçek zamanlı veri senkronizasyonu sağlayan bir NoSQL çözümü olan Firebase Realtime Database kullanılmıştır. Geleneksel ilişkisel veritabanlarının (RDBMS) aksine, bu projede veriler JSON

formatında, bir ağaç yapısı şeklinde saklanmaktadır. Projenin veritabanı şeması iki ana kök düğümden oluşmaktadır:

1. Kitaplar: Uygulamada satış yapılan tüm kitapların bilgilerini tutar. Her kitap, benzersiz bir ID kodu altında saklanır. Bu yapı, uygulamanın MainActivity, AdminActivity veya Kitaplar ile Kullanıcılar sınıflarında verilerin asenkron olarak çekiliş kullanıcıya anlık olarak gösterilmesini sağlar.

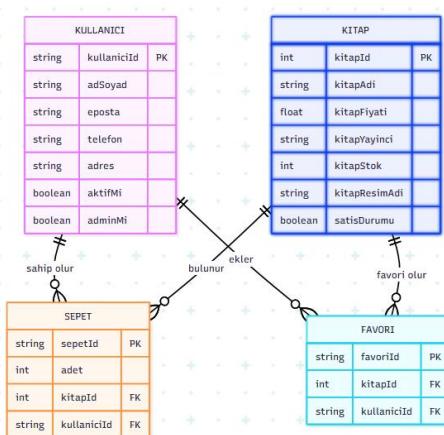
2. Kullanıcılar: Sisteme kayıt olan kullanıcıların profil bilgilerini ve kişisel sepetlerini sisteme tutar. Kullanıcı kayıt işlemi (KayıtFragment), Firebase Authentication kullanarak, proje gereksinimleri doğrultusunda bir güvenlik katmanı sağlar. Kullanıcı giriş yaptığından (GirişFragment), girilen e-posta ve şifre bilgilerini veritabanındaki kayıtlarla karşılaştırarak oturum açma işlemi doğrulanır. Bunun yanında admin ve standart rolleri ile farklı activity sayfalarına gönderilir.

C. Kullanıcı Arayüzü ve Veri Bağlama

Verilerin kullanıcıya sunulmasında Android'in RecyclerView özelliğinden faydalanylmıştır. RecyclerView, çok sayıda kitap verisinin listelendiği durumlarda bellek yönetimini optimize eder, uygulamanın performansını artırır. Her bir liste elemanı (Kitaplar ve kullanıcılar), kendilerine ait Adapter sınıfları ile veritabanından gelen veri modeline bağlanmıştır. Kitap kapak görsellerinin URL bağlantısıyla kolaylıkla yüklenmesi ve önbelleğe alınması işlemi için ise görüntü işleme kütüphanesi Glide kullanılmıştır. Bu sayede kullanıcı deneyimi optimize edilmiştir.

D. Varlık İlişki (ERD) Diyagramı

Uygulamanın veri tabanı mimarisini tasarlarken, sistemdeki varlıklar arasındaki ilişkilerin bütünlüğünü korumak amacıyla bir Varlık İlişki modeli oluşturulmuştur. Aşağıda Şekil 1'de gösterilen diyagramda sistem; Kitaplar, Kullanıcılar, Sepet ve Favoriler olmak üzere dört temel veri üzerine kurulmuştur. Burada Kitaplar ve Kullanıcılar ana veri kaynaklarını oluşturken, Sepet ve Favoriler ise kullanıcıların kitaplara olan etkileşimlerini (satın alma isteği veya beğenme) tutan ilişki tabloları olarak görev almaktadır.



Şekil 1 - Varlık İlişki Diyagramı

Mesela, kitapId ve kullaniciId kitaplar ve kullanıcılar verilerinin birincil anahtarları iken sepet içinde yabancı anahtar haline gelmiştir.

E. Versiyon Kontrol Sistemi (Git & GitHub)

Yazılım geliştirme sürecinin takibi, kod güvenliği ve yedekleme işlemleri için Git versiyon kontrol sistemi kullanılmıştır. Uygulama geliştirme sona erken projenin uzun vadeli kolay saklanabilmesi için projenin uzak sunucusu olarak GitHub platformundan yararlanılmıştır. Çalışmada kaynak kodların yanı sıra veritabanı yedekleri de JSON formatında yedeklenmiştir. Özellikle veritabanı şemasında yapılan değişikliklerin kaybolmaması adına, databaseBackup klasörü oluşturulmuş ve Kitaplar.json ile Kullanıcılar.json dosyaları düzenli olarak depoya gönderilmiştir. Bu yöntem, projenin taşınabilirliğini artırmış ve olası veri kayıplarına karşı bir güvenlik mekanizması sağlamıştır.

III. DENEYSEL BULGULAR

Yapılan mobil test kitabı satış uygulamasının kolaylığı, güvenliği ve performansı, Android emülatörler (mesela Pixel 7) ve fiziksel cihazlar üzerinde gerçekleştirilen çoklu testler ile doğrulanmıştır. Bu bölümde, uygulamanın yapı taşları, çalışma mantığı, elde edilen arayüz görselleri ve veritabanları gösterilmektedir.

A. Kullanıcı Kimlik Doğrulama ve Giriş Modülü

Uygulamanın başlangıç aşamasında, kullanıcı kayıt ve giriş süreçleri incelenmiş, sonra admin girişi eklenmiştir. KayıtFragment üzerinden sisteme yeni bir kullanıcı kayıt edildiğinde, verilerin Firebase Authentication üzerine anlık olarak girildiği gözlemlenmiştir. Ayrıca kayıt olunurken kullanıcılarından belirli bir şifre güvenlik seviyesi talep edilmiştir.

Şekil 2 - Firebase Authentication Sayfası

Kullanıcı şifreleri ve hassas bilgileri (telefon, adres vs.) güvenli bir şekilde veritabanına kaydedilmiştir. Giriş ekranında (GirişFragment), kayıtlı olmayan bir e-posta adresi veya yanlış şifre ile giriş yapılmaya çalışıldığında, uygulama kullanıcıya "Giriş başarılı değil, tekrar deneyiniz" şeklinde bir Toast mesajı vererek güvenlik kontrolünü başarıyla sağlamıştır. Doğru bilgilerle giriş yapıldığında ise sistem, kullanıcının benzersiz ID'sini (UID) oturum boyunca saklamış ve kullanıcıyı ana sayfaya yönlendirmiştir. Eğer kullanıcı admin rolünde ise onu tamami ile farklı bir activity'e yönlendirmiştir.

BAŞLA

“Kullanıcılar” koleksiyonuna bağlan.

EĞER (Kullanıcılar koleksiyonuna bağlanabiliyor ise)

EĞER (kullanıcı admin ise)

‘Admin Giriş’ mesajı gönder

AdminActivity sayfasına GİT

DEĞİLSE

‘Giriş Yapıldı’ mesajı gönder

MainActivity sayfasına GİT

DEĞİLSE

‘Hata’ mesajı gönder

BİTİR

Pseudo Kod 1

B. Kitap Listeleme ve Veri Çekme Performansı

Uygulamanın ana ekranı olan MainActivity veya AdminActivity yüklenliğinde, Kitaplar koleksiyonundaki veriler eşzamanlı olarak çekilmiştir. Kitaplar ve kullanıcılardaki veriler ekleme çıkarma güncelleme gibi işlevlerle eşzamanlı olarak güncellenmiştir. Mesela, kullanıcı telefon ve adres bilgilerini ProfilDegistirFragment üzerinden güncellediğinde başka bir cihazdaki admin bu bilgilerin değiştiğini görmüştür. Kitap kapak resimleri, URL adresleri üzerinden çekilerek KitapDetayFragment'a konulmuştur. Recycler View yapısının kolay kullanımı sayesinde çoklu sayfalar yerine tek sayfada kaydırma özelliğini de kullanarak tüm kitaplara erişilmiştir.



Şekil 3 - RecyclerView kullanılan bir activity

C. Sepet Yönetimi ve Favoriler

Uygulamanın en önemli işlevlerinden biri sepet yönetimidir. Bir kullanıcı, ana sayfadan bir kitabı seçip detay sayfasına gittiğinde ve "Sepete Ekle" butonuna bastığında, seçilen kitabın bilgileri (no, ad, yayıncı ve fiyat) o an giriş yapmış kullanıcının veritabanındaki Sepet koleksiyonunda depolanmıştır. Farklı kullanıcıların sepetleri birbirine karışmamış, her kullanıcının sepet verisi kendi kullanıcıId'si kullanılarak korunmuştur. Aynı zamanda kullanıcıya beğendiği kitapları favoriye ekleyerek uygulamayı kişiselleştirme şansı verilmiştir. Bununla birlikte kullanıcı uygulamaya tekrar girdiğinde istediği kitabı kolaylıkla bulması sağlanmıştır.

BAŞLA

EĞER (İşlem Tipi EKLE ise)

FavoriKitap listesi oluştur

Firestore'a bu listeyi ekle

‘Favorilere Eklendi’ mesajı gönder

DEĞİLSE (İşlem Tipi ÇIKAR ise)

Firestore'dan bu listeyi çıkar

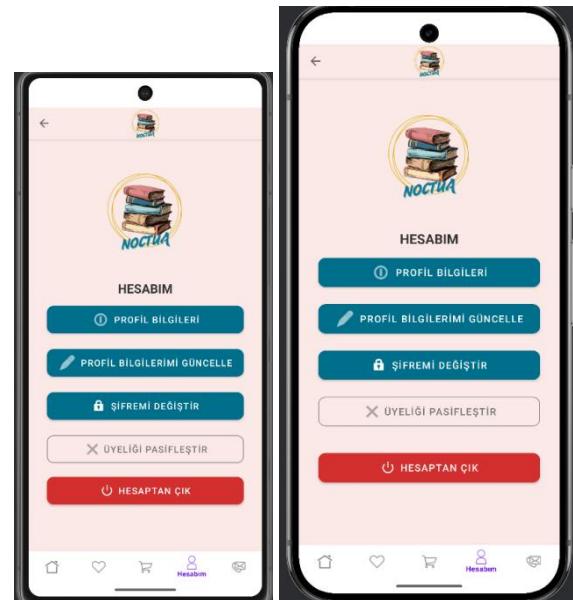
‘Favorilerden Çıkarıldı’ mesajı gönder

BİTİR

Pseudo Kod 2

D. Arayüz Uyumluluğu ve API Kullanımı

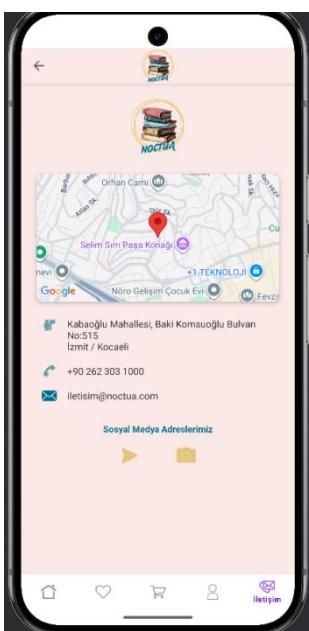
Uygulama arayüzü, farklı ekran çözünürlüklerine sahip cihazlarda test edilmiştir. XML tasarımlarında kullanılan ConstraintLayout yapısı kullanılarak, butonların, metinlerin ve görsellerin kayma gerçekleşmeden, her cihazda doğru konumda aynı şekilde görüntülenmiştir.



Şekil 4 - Pixel 7 ve Pixel 9 Pro emülatörlerinde görülen hesabım ekranı

Uygulamada iki adet farklı API kullanılmıştır. IletisimFragment sayfasına bir konum API'si eklenmiştir, bu sayede kullanıcı adminin belirlediği konumu iletişim sayfasında görebilmüştür. SepetFragment'a bir döviz API'si

eklenmiş olup kullanıcıların alışveriş yaparken sepette kurları karşılaştırma fırsatı vermiştir.



Şekil 5 - İletişim sayfasındaki konum API'si

IV. SONUÇLAR

Bu projede, mobil uygulamaların günlük hayatı artan rolü dikkate alınarak, Android tabanlı Kotlin dilinde bir test kitabı satış ve yönetim uygulaması geliştirilmiş ve test edilmiştir. Proje süresince, mühendislik prensiplerine uygun olarak ilk önce bir proje fikri bulunmuş sonra projenin tasarımları yapılmış daha sonra da test aşamasına geçilmiştir. Elde edilen uygulama, kullanıcıların her yerde ve istedikleri zaman test kitaplarını satın almasını sağlayan bir platform olmuştur. Projede yazılım dili olarak Kotlin dilinin tercih edilme sebebi, kod okunabilirliği ve kolaylığı açısından diğer programlama dillerine göre avantajlı olmasıdır. Özellikle Java'ya kıyasla daha az kod satırı ile daha işlevsel blokların kullanılabilmesi, geliştirme sürecini kolaylaştırmış ve hızlandırmıştır.

Veri katmanında tercih edilen Firebase Realtime Database, sunucu kurulumu gerektirmemesi ile avantaj sağlamıştır. Analiz ve testler sonucunda, uygulamanın temel e-ticaret işlevlerini gerçekleştirebildiği görülmüştür. Mesela kimlik doğrulama, ürün listeleme, favorilere ekleme, kitap detaylarını görüntüleme, sepete ekleme ve benzeri sayfalar hatasız bir şekilde çalışmıştır. NoSQL veritabanı yapısı kullanılarak veri okuma hızı artırılmış, aynı zamanda projenin geliştirilmesi de hızlanmıştır.

Gelecek çalışmalarında, bu proje temel alınarak uygulamanın arayüz tarafında daha işlevsel yeni özelliklerin eklenmesi planlanmaktadır. Örnek olarak sayfalar arası geçişler daha kolaylaştırılabilir, veya sipariş ekranında daha fazla seçenek sunulabilir. Bunun yanı sıra, kullanıcıların kitaplara yorum yapabileceği bir yer ayarlanabilmelidir.

Bu proje kapsamında mobil uygulamanın temellerini öğrenmiş olup frontend ve backend kısımlarının nasıl geliştirildiği görülmüştür.

Proje dosyalarına ve GitHub bağlantısına şu linklerden ulaşabilirsiniz:

https://drive.google.com/file/d/1sIF3tBG_ZXVs1tvJAAw_4ZsnaD8xRU/view?usp=sharing

<https://github.com/elifspolatt/MobilUygulamaProjesi>

KAYNAKÇA

- [1] Kotlin Foundation, "Kotlin Programming Language Documentation," [Kotlinlang.org](https://kotlinlang.org/docs/home.html), 2024. [Online]. Available: <https://kotlinlang.org/docs/home.html>.
- [2] Google Firebase, "Firebase Realtime Database," Firebase Documentation, 2024. [Online]. Available: <https://firebase.google.com/docs/database>.
- [3] D. Spilker et al., "Glide: An image loading and caching library for Android," GitHub Repository, 2024. [Online]. Available: <https://github.com/bumptech/glide>.
- [4] Square Inc., "Retrofit: A type-safe HTTP client for Android and Java," Square Open Source, 2024. [Online]. Available: <https://square.github.io/retrofit/>.
- [5] Google Developers, "Android Jetpack Navigation Component," Android Developers, 2024. [Online]. Available: <https://developer.android.com/guide/navigation>.
- [6] Google Developers, "Android Studio User Guide," Android Developers Documentation, 2024. [Online]. Available: <https://developer.android.com/studio/intro>.