

Envanter Takip Uygulaması(Mobil)

Mehmet Erdem MORGÜL 231307099
Talha Fırat MEŞE 231307029
Metehan YÜKSEK 241307137
Bilişim Sistemleri Mühendisliği
Veritabanı Yönetim Sistemleri
Kocaeli Üniversitesi Kocaeli/Türkiye
Grup:93

mehmet.morgul14@gmail.com
tfmese@gmail.com
mthnyksk@gmail.com

Geliştirme Dili: Kotlin

Veri Tabanı Katmanı: Room (SQLite abstraction)

Geliştirme Ortamı: Android Studio

Mimari Yaklaşım: MVVM

Giriş:

Günümüz iş dünyasında dijitalleşmenin etkisiyle birlikte, işletmelerin süreçlerini dijital araçlarla yönetme ihtiyacı her geçen gün artmaktadır. Bu dönüşümden en çok etkilenen alanlardan biri de envanter ve stok yönetimidir. Özellikle küçük ve orta ölçekli işletmeler, manuel olarak yürütülen envanter takip işlemlerinde zaman kaybı, veri tutarsızlığı ve operasyonel hatalar gibi birçok sorunla karşı karşıya kalmaktadır. Bu bağlamda, mobil cihazlar üzerinden çalışan, kolay erişilebilir ve kullanıcı dostu envanter yönetim sistemlerinin geliştirilmesi büyük önem taşımaktadır.

Bu proje, işletmelerin ürün, stok, sipariş, sevkiyat ve müşteri ilişkilerini mobil ortamda etkili bir şekilde yönetmelerini sağlayacak bir **Mobil Envanter Takip Sistemi** geliştirmeyi amaçlamaktadır. Uygulama, Android işletim sisteminde çalışacak şekilde **Kotlin** programlama diliyle geliştirilmiş, veritabanı yönetimi için ise Android Jetpack bileşenlerinden biri olan **Room** kullanılmıştır. Room, SQLite üzerine inşa edilmiş, tip güvenliği sunan ve veri erişim katmanını kolaylaştıran modern bir veritabanı çözümüdür.

Proje kapsamında; bir ürünün kategoriye ait olması, birden fazla depoda farklı miktarlarda bulunabilmesi, sipariş oluşturulabilmesi, sevkiyat işlemlerinin planlanabilmesi gibi temel ticari operasyonlar sistematik bir yapıya oturtularak uygulama içerisinde modellenmiştir. Her bir işlem,

arka planda veri tutarlılığı sağlayacak şekilde kurgulanmış; uygulama mimarisi ise MVVM (Model-View-ViewModel) yaklaşımı ile düzenlenerek sürdürülebilir, okunabilir ve genişletilebilir bir yapı sağlanmıştır.

Geliştirilen bu uygulama, temel olarak aşağıdaki sorunlara çözüm sunmayı hedeflemektedir:

- Stokların güncel ve doğru şekilde takip edilmesi,
- Müşteri siparişlerinin hızlı ve eksiksiz işlenmesi,
- Ürünlerin depo bazlı sevkiyat bilgilerinin izlenebilir olması,
- Ürün ve kategori bilgilerinin merkezi bir yapı üzerinden yönetilmesi,
- Envanter hareketlerinin kayıt altına alınması ve geriye dönük izlenebilirliğinin sağlanması.

Kullanılan Teknolojiler ve Yapılar

- **Kotlin:** Android geliştirme için modern, açık kaynaklı bir dil.
- **Room:** SQLite üzerine kurulu, tip güvenli ve LiveData uyumlu veri tabanı kütüphanesi. Room kütüphanesini SQLite Database nesneleri ile daha kolay çalışmamıza olanak sağlıyor ve aynı zamanda kod yükünden kurtarıyor.
- **MVVM Mimari Yapısı:** Yazılım geliştirme mimarilerinden biridir. Kodun daha test edilebilir ve sürdürülebilir olması için ViewModel, Repository ve LiveData kullanılmıştır.
- **LiveData:** Verilerin UI ile senkronize olmasını sağlar.

- **ViewModel:** UI'dan bağımsız veri yönetimi sağlar.
- **DAO :** (Data Access Object — Veri Erişim Nesnesi), Interface olarak kullanılır ve bütün database işlemleri (ekleme, silme, güncelleme, sorgu) bu interface içerisinde yapılır.

Veri Tabanı Modeli (Entity Yapısı)

Uygulamada toplam **11 tablo (Entity)** bulunmaktadır. Bu tablolar arası ilişkiler bire-bir veya bire-çok olarak Room'da @ForeignKey anotasyonlarıyla tanımlanmıştır. Aşağıda temel yapılar açıklanmıştır:

Kategoriler ve Ürünler (Category - Product)

- Her ürün bir kategoriye aittir.
- Ürünlerin fiyatı, stoğu, açıklaması ve ait olduğu tedarikçi bilgileri tutulur.

Depolar ve Stoklar (Warehouse - WarehouseStock)

- Depolar fiziksel olarak şehir ve adres bilgileri ile tanımlanır.
- Her ürün birden fazla depoda farklı miktarlarda bulunabilir.
- Bu ilişkiyi temsil eden WarehouseStock tablosu ara tablo olarak görev yapar.

Sipariş ve Detayları (Order - OrderDetails)

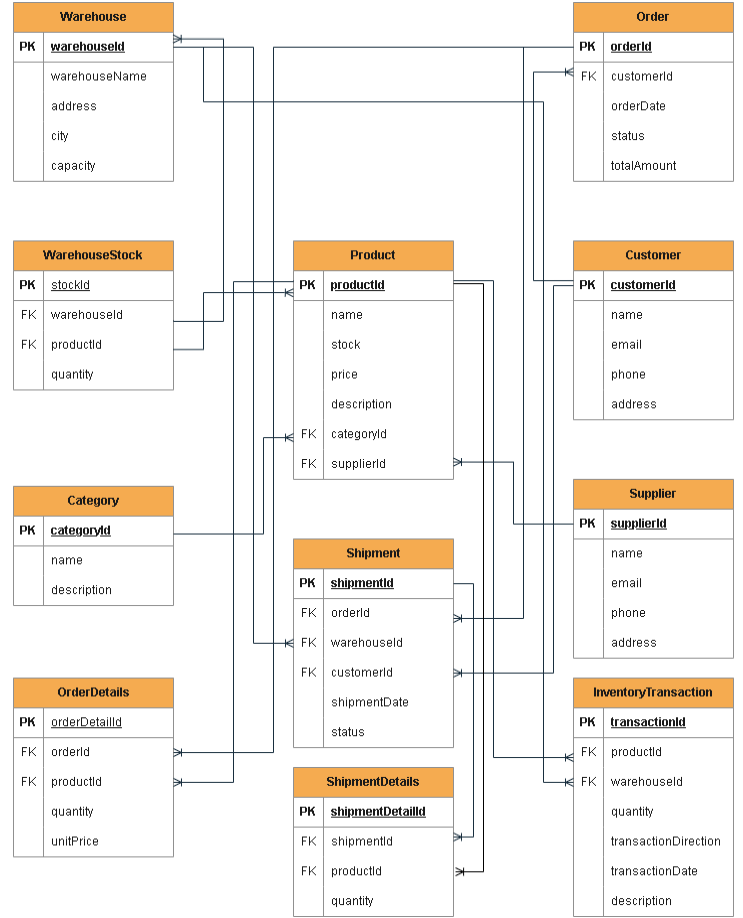
- Her müşteri sipariş verebilir. Her siparişin birden fazla ürünü olabilir.
- Siparişler duruma göre "Hazırlanıyor", "Kargoya Verildi", "Teslim Edildi" gibi statülere sahip olabilir.
- Sipariş detay tablosunda her bir ürünün sipariş miktarı ve birim fiyatı yer alır.

Sevkiyatlar ve Detaylar (Shipment - ShipmentDetails)

- Siparişler bir veya birden fazla sevkiyata konu olabilir.
- Sevkiyatlar hangi depodan, hangi müşteriye ve ne zaman yapıldığı gibi bilgileri içerir.
- ShipmentDetails tablosu sevk edilen ürünleri listeler.

Envanter Hareketleri (InventoryTransaction)

- Stok giriş-çıkış işlemleri burada kaydedilir.



Tasarım Yaklaşımı

Bu uygulama Kotlin ve Android Studio kullanarak MVVM mimarisiyle geliştirildi. Kullanıcının kolay ve anlaşılır kullanımına yoğunlaşıp tasarım katmanlarında aşağıdaki yapı izlendi:

■ UI (Kullanıcı Arayüzü)

- XML dosyalarında modern ve sade bir tasarım tercih edilmiştir.
- Veri gözlemi için LiveData kullanılarak anlık güncellemeler sağlanmıştır.
- Kullanıcıların ürünleri, siparişleri ve depo bilgilerini kolayca görüntüleyebileceği sade bir menü yapısı oluşturulmuştur.

■ ViewModel

- Her ekran için ayrı ViewModel sınıfı tanımlandı.
- Bu sınıflar, Repository ile bağlantı kurarak UI'ya temiz veri akışı sağladı.

■ Repository

- DAO sınıflarından veri olarak ViewModel'a aktarıldı.
- Veri kaynaklarını soyutlayarak kodun yeniden kullanılabilirliği artırıldı.

■ Room Database

- AppDatabase sınıfı ile tüm DAO'lar tek merkezde birleştirildi.
- @Database anotasyonu kullanılarak tüm Entity'ler tanımlandı.

Müşteri Girişi ve Sipariş Oluşturma:

Müşteriler uygulama üzerinden kendi profilleriyle giriş yapabilir. Bu giriş işlemi sırasında isim, e-posta, telefon ve adres bilgileri alınır ve Room veritabanına kaydedilir. Kayıt olan müşteri, daha sonra uygulama içerisinde ürünleri görüntüleyerek sipariş oluşturabilir.

Yeni Ürün Ekleme ve Kategoriye Atanması:

Yetkili kullanıcı, yeni bir ürün tanımlamak istediğinde ürün adı, açıklaması, fiyatı, stok durumu, kategori ve tedarikçi bilgilerini girerek ürünü sisteme ekleyebilir. Her ürün bir kategoriye ait olmak zorundadır.

Depo Bazlı Stok Takibi:

Ürünlerin farklı depolarda farklı miktarlarda bulunabileceği göz önüne alınarak, sistemde her deponun adresi, kapasitesi ve içindeki ürünlerin miktar bilgisi ayrı olarak tutulmaktadır.

Siparişe Ait Sevkiyat Planlama

Bir sipariş oluşturulduktan sonra, bu siparişin hangi depodan gönderileceği belirlenerek sevkiyat süreci başlatılır. Sevkiyat bilgisi sistemde ayrı bir yapıda tutulur.

Envanter Hareketlerinin İzlenmesi

Her ürün için yapılan stok giriş ve çıkış işlemleri sistemde "Envanter Hareketi" olarak kaydedilir. Böylece geriye dönük izleme ve stok kontrolü sağlanmış olur.

Bu senaryolar sayesinde uygulama, ürünlerin tanımlanmasından sevkiyat ve stok takibine kadar uçtan uca bir envanter sürecini dijital olarak yönetilebilir hale getirmiştir. Kullanıcı dostu arayüzlerle desteklenen bu senaryolar, işletmelerin stok kayıplarını azaltmasına, sipariş takibini kolaylaştırmasına ve müşteri memnuniyetini artırmasına olanak sağlar.

Süreç

Bu çalışma, **Kocaeli Üniversitesi Bilişim Sistemleri Mühendisliği** lisans programı kapsamında verilen **Veritabanı Yönetim Sistemleri** dersi için bir dönem projesi olarak gerçekleştirilmiştir. Proje, hem veritabanı tasarımı hem de mobil uygulama geliştirme yönünden işlevsel bir envanter yönetim sistemi ortaya koymayı hedeflemiştir. Kotlin üzerinde ROOM kütüphanesinden yararlanılarak SQLite çalışmaları sürdürülmüş ve ders kapsamında bulunan konulara bağlı kalınmıştır. ER Diyagramı, veri tabanı ve data kullanımı, kullanıcı dostu arayüz, kullanım kolaylığı gibi konulara önem verilmiştir. günün sonunda mobil bir **Envanter Takip Sistemi** geliştirilmiştir.

Proje bir grup çalışması şeklinde gerçekleştirilmiş olup, her grup üyesi farklı aşamalarda aktif roller üstlenmiştir. Görev dağılımı yapılmış, kod ve tasarım kararları ortaklaşa alınmış, düzenli olarak iletişim kurulmuş ve proje boyunca iş birliği içinde çalışılmıştır. Bu süreç, hem teknik bilgi birikimini artırmak hem de ekip çalışması disiplini edinmek açısından tüm grup üyeleri için verimli ve öğretici olmuştur.

Kaynakça

<https://www.w3schools.com>

<https://developer.android.com/kotlin/learn?hl=tr>

<https://developer.android.com/jetpack/androidx/releases/room>

Chatgpt, Github

Proje Github Linki:

<https://github.com/tfmese/InventoryManagement-master>