



**T.C.**

**GAZİ ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BMT-311**

**Veritabanı Yönetim Sistemleri Dersi**

**Proje Raporu**

**Stok Takip Sistemi**

**31.012.2019**

**171816063 Sadık Barkın GÜLER**

**161816059 Muhsin Kılıç**

## İçindekiler

1.GİRİŞ.....	1
Amaç .....	1
Kapsam .....	1
Referans .....	1
2.PROJEDE KULLANILAN TEKNOLOJİLER .....	1
3.YERLEŞİM ŞEMASI .....	2
4. MİMARİ TASARIM .....	2
5. YETKİLER .....	3
6. MANTIKSAL VERİ MODELİ.....	3
7. ER DİYAGRAMI .....	4
8. UML DİYAGRAMI.....	5
9. NORMALİZASYON .....	6
10. VERİ SÖZLÜĞÜ .....	7
11.GENEL KISITLAMALAR .....	8
12. EKRAN GÖRÜNTÜLERİ .....	9
13. KULLANILAN TRİGGERLAR .....	11
14. ÖNEMLİ GÖRÜLEN PROSEDÜRLER.....	13
15. ÖNEMLİ GÖRÜLEN JAVA KODLARI .....	14

# 1.GİRİŞ

## Amaç

Stok otomasyonu ve takip sisteminin amacı , bir stok şirketinin envanterinde olan ürünler hakkındaki bilgileri sistem aracılığıyla görmek bu Bilgileri ekleyip , değiştirip ve silebilmek ürünlerin kargolanmasını ve teslimini sistem aracılığı ile takip etmektir.Ayrıca ürünlerin üzerindeki hareketleri müşteriler takip edebilmektedir ve hareketlerin geçmişi yedeklenmektedir.

## Kapsam

Bu proje, ilerleyen teknolojinin ticari hayata uyarlanması, ürün bilgilerinin güvenli bir biçimde tutulmasını, çalışanların ve müşterinin işlemlerde kaybedeceği zamanı en aza indirme faaliyetini kapsamaktadır.

Sistemde 3 adet kullanıcı tanımlanmıştır:

- **Admin:** Giriş ekranı için kullanıcı bilgilerini tutar, bilgiler üzerinde ekleme, silme ve güncelleme işlemleri yapılabilir. Ürünlerin hareketlerinin geçmişini görebilir ama ürün bilgilerini göremez hareket bilgilerinde beklenmedik bir durum olunursa Admin ile birlikte yedek hareketler tablosuna bakılır.
- **Kullanıcı:** Depo bilgileri, ürün bilgileri ve hareket bilgilerinin eklenmesi silinmesi ve güncellenmesinde rol oynarlar. Ayrıca ürünlerin hareketleri bilgilerini Kullanıcı girer.
- **Müşteri:** Kendisine verilen hareketno ile hareket içerisinde olan ürününün durumunu takip edebilir.

## Referans

<https://www.javatpoint.com/example-to-connect-to-the-oracle-database>

<https://medium.com/gokhanyavas/oracle-database-sql-nedir-sql-temelleri-1-select-komutu-fc8af5714793>

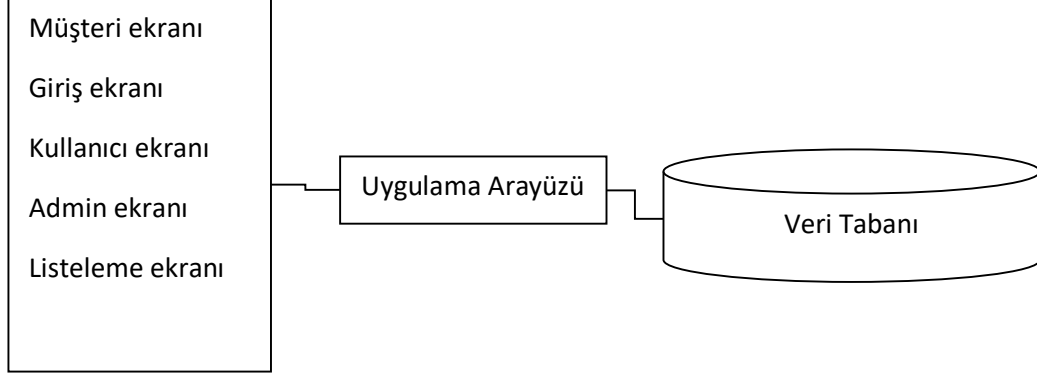
## 2.PROJEDE KULLANILAN TEKNOLOJİLER

Proje kapsamında kullanılan teknolojiler aşağıda sıralanmıştır:

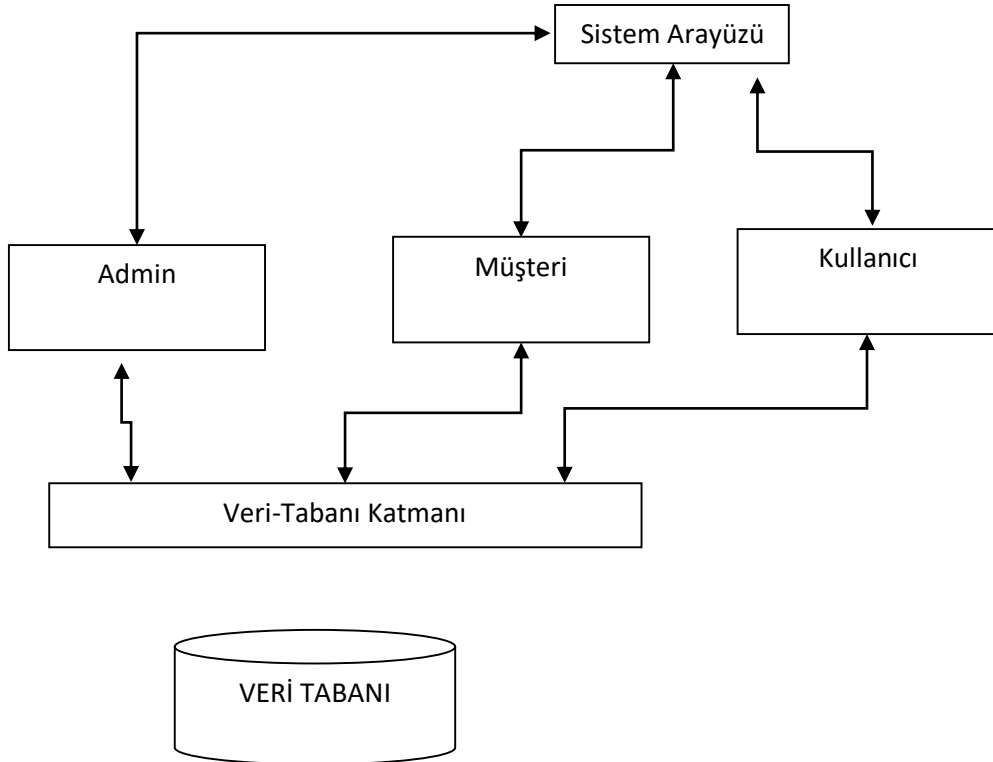
- JAVA dili.
- NetBeans IDE java kodlaması bu platform üzerinden yapıldı.
- Oracle sql dili database işlemleri için kullanılacak dil.
- Oracle sql developer database işlemleri için.
- Oracle SQL Developer Data Modeler diyagramlar için.

### 3.YERLEŞİM ŞEMASI

Aşağıdaki uygulama şeması, şirket bilgi sistemine fiziksel bir görünüm kazandırmaktadır.



### 4. MİMARİ TASARIM



## 5. YETKİLER

E:ekleme

S:Silme

G:Güncelleme

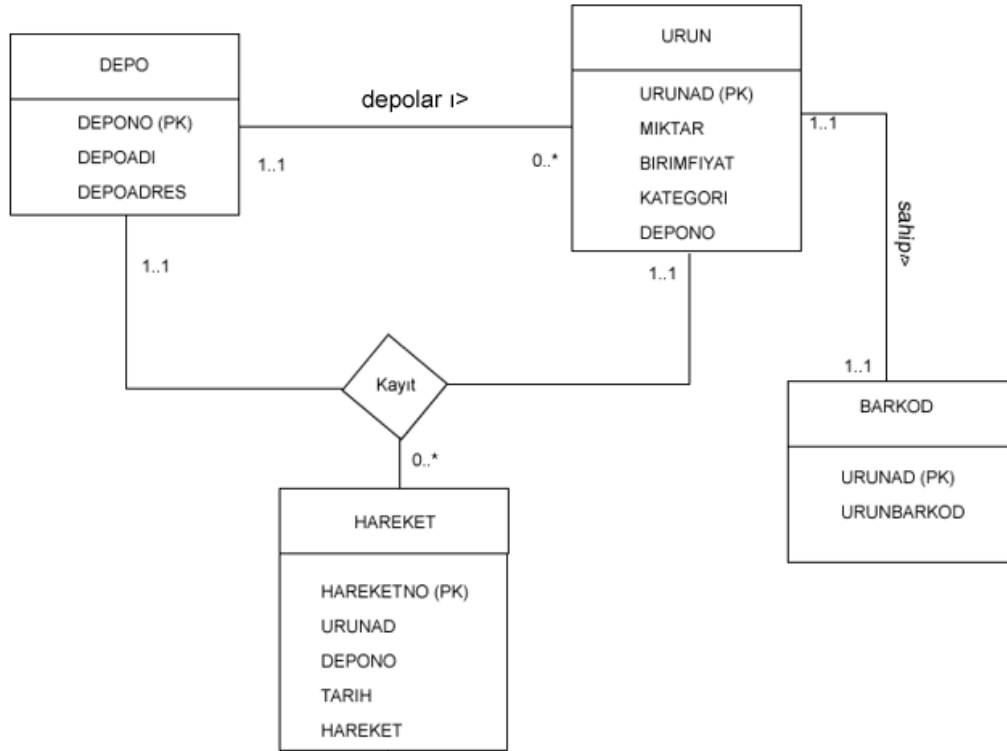
L:Listeleme

TABLolar	DEPO				URUN				HAREKET				BARKOD				HAREKET_YEDEK				BİLGİLER				MUSTERİ			
KULLANICILAR	E	S	G	L	E	S	G	L	E	S	G	L	E	S	G	L	E	S	G	L	E	S	G	L	E	S	G	L
Kullanıcı	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	-	-	-	-	-	X	X	X	
Müşteri	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
Admin	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	X	X	X	-	-	-	-	-

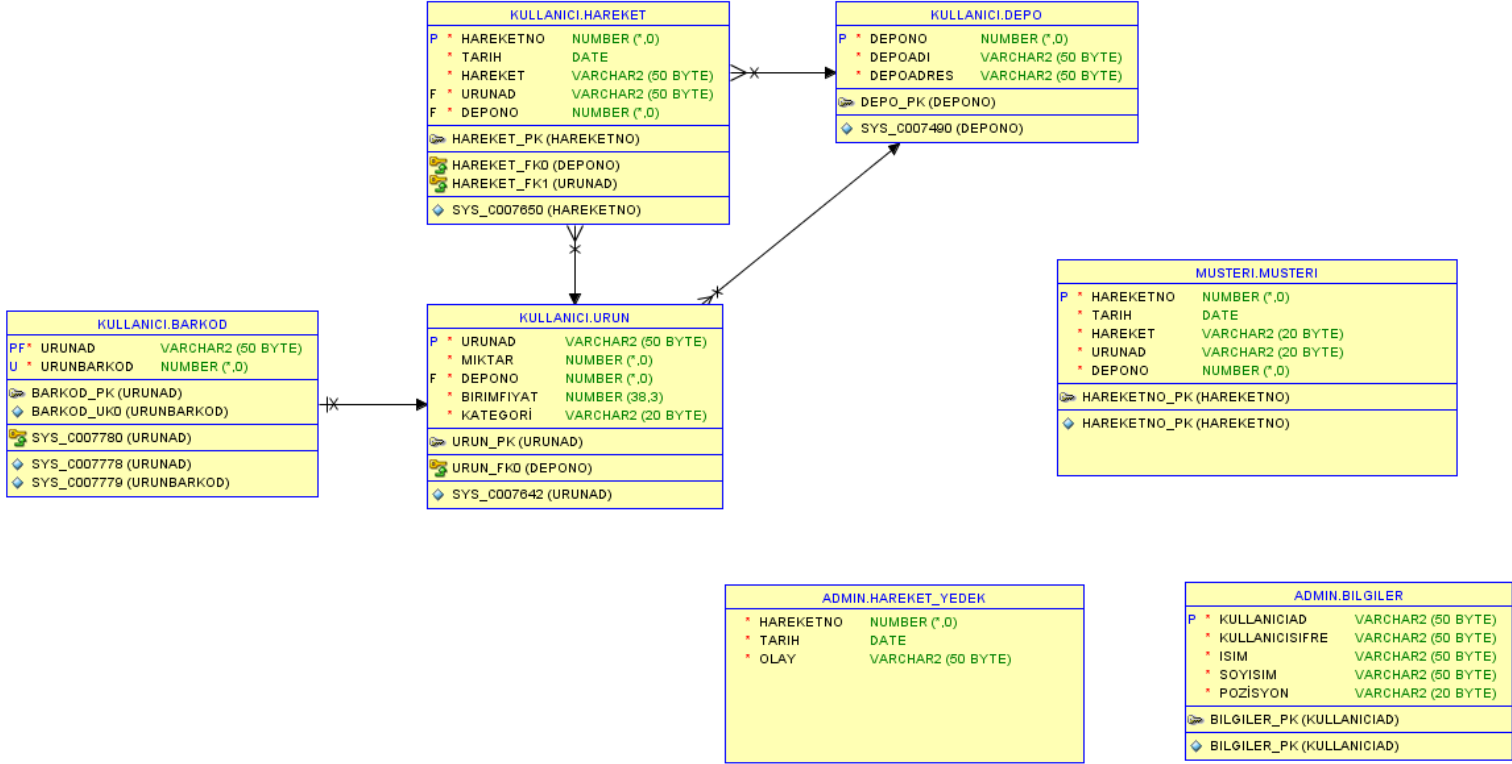
## 6. MANTIKSAL VERİ MODELİ

DEPO(DEPONO, DEPOAD,DEPOADRES) Birincil Anahtar DEPONO	URUN(URUNAD,MIKTAR,BİRİMFİYAT,KATEGORİ,DEPO NO) Birincil Anahtar URUNAD Yabancı Anahtar DEPONO references DEPO
HAREKET(HAREKETNO,URUNAD,DEPONO, TARİH,HAREKET) Birincil Anahtar HAREKETNO Yabancı Anahtar DEPONO references DEPO	BARKOD(URUNAD,URUNBARKOD) Birincil Anahtar URUNAD
HAREKET_YEDEK(HAREKET_NO,TARİH,OLA Y) Birincil Anahtar HAREKET_NO	ADMİN_BİLGİLER(KULLANICIAD,KULLANICISİFRE,İSİM, SOYİSİM,POZİSYON) Birincil Anahtar KULLANICIAD
MUSTERİ(HAREKETNO,TARİH,HAREKET,UR UNAD,DEPONO) Birincil Anahtar HAREKET_NO	

## 7. ER DİYAGRAMI



## 8. UML DİYAGRAMI



## 9. NORMALİZASYON

### Birinci Normal Form:

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID
1	KULLANICIAD	VARCHAR2 (50 BYTE)	No	(null)	1
2	KULLANICISIFRE	VARCHAR2 (50 BYTE)	No	(null)	2
3	ISIMSOYISIM	VARCHAR2 (50 BYTE)	No	(null)	3
4	POZİSYON	VARCHAR2 (50 BYTE)	No	(null)	4

Tablolarda her satır ve sütunun kesişimi tek değer içermelidir. İsim ve soy isim farklı sütunlara konuldu.

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID
1	KULLANICIAD	VARCHAR2 (50 BYTE)	No	(null)	1
2	KULLANICISIFRE	VARCHAR2 (50 BYTE)	No	(null)	2
3	ISIM	VARCHAR2 (50 BYTE)	No	(null)	3
4	SOYISIM	VARCHAR2 (50 BYTE)	No	(null)	4
5	POZİSYON	VARCHAR2 (20 BYTE)	No	(null)	5

### İkinci Normal Form:

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID
1	URUNAD	VARCHAR2 (50 BYTE)	No	(null)	1
2	URUNBARKOD	NUMBER (38, 0)	No	(null)	2
3	MIKTAR	NUMBER (38, 0)	No	(null)	3
4	DEPONO	NUMBER (38, 0)	No	(null)	4
5	BİRİMFİYAT	NUMBER (38, 3)	No	(null)	5
6	KATEGORİ	VARCHAR2 (50 BYTE)	No	(null)	6

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID
1	URUNAD	VARCHAR2 (50 BYTE)	No	(null)	1
2	URUNBARKOD	NUMBER (38, 0)	No	(null)	2

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID
1	URUNAD	VARCHAR2 (50 BYTE)	No	(null)	1
2	MIKTAR	NUMBER (38, 0)	No	(null)	2
3	DEPONO	NUMBER (38, 0)	No	(null)	3
4	BİRİMFİYAT	NUMBER (38, 3)	No	(null)	4
5	KATEGORİ	VARCHAR2 (20 BYTE)	No	(null)	5

Yukardaki şekilde urun tablosu gözükmemektedir. Urun tablosunda ilk başta unique olan URUNBARKOD sütunu tam fonksiyonel bağımlılığı bozmaktadır ve kısmi bağımlılık oluşturmaktadır. Bu yüzden BARKOD diye ayrı bir tabloya alındı şekilde görüldüğü gibi. Kısmi bağımlılık ortadan kaldırılıp tam fonksiyonel bağımlılık elde edildi.

### Üçüncü Normal Form:

Tüm tablolarda birincil anahtarla geçişli bağımlı olan bir ilişki bulunmamaktadır. Dolayısıyla tablolar üçüncü normal forma uygundur.



## 10. VERİ SÖZLÜĞÜ

### Tablolar

1. URUN
2. BARKOD
3. DEPO
4. HAREKET
5. HAREKET\_YEDEK
6. BILGILER
7. MUSTERI

#### BILGILER Tablosu:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	KULLANICIAD	VARCHAR2 (50 BYTE)	No	(null)	1	Giriş için kullanılan kullanıcı adı
2	KULLANICISIFRE	VARCHAR2 (50 BYTE)	No	(null)	2	Giriş için şifre
3	ISIM	VARCHAR2 (50 BYTE)	No	(null)	3	Kullanıcının kendi ismi
4	SOYISIM	VARCHAR2 (50 BYTE)	No	(null)	4	Kullanıcının soy ismi
5	POZİSYON	VARCHAR2 (20 BYTE)	No	(null)	5	Kullanıcının pozisyonu

#### HAREKET\_YEDEK Tablosu:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	HAREKETNO	NUMBER (38, 0)	No	(null)	1	Yedeklenen hareketin numarası
2	TARİH	DATE	No	(null)	2	Güncel hareket tarihi
3	OLAY	VARCHAR2 (50 BYTE)	No	(null)	3	Hareketin içeriği

#### MUSTERI Tablosu:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	HAREKETNO	NUMBER (38, 0)	No	(null)	1	Musterinin işlemini içeren hareketno
2	TARİH	DATE	No	(null)	2	Son işlemin güncel tarihi
3	HAREKET	VARCHAR2 (20 BYTE)	No	(null)	3	Hareket içeriği
4	URUNAD	VARCHAR2 (20 BYTE)	No	(null)	4	Hangi ürün üzerinde hareket var
5	DEPONO	NUMBER (38, 0)	No	(null)	5	Hangi depodan işlem görüyor

#### DEPO Tablosu:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	DEPONO	NUMBER (38, 0)	No	(null)	1	Deponun numarası
2	DEPOADI	VARCHAR2 (50 BYTE)	No	(null)	2	Deponun adı
3	DEPOADRES	VARCHAR2 (50 BYTE)	No	(null)	3	Deponun adresi

### URUN Tablosu:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	URUNAD	VARCHAR2(50 BYTE)	No	(null)	1	Eklenecek olan ürünün adı
2	MIKTAR	NUMBER(38,0)	No	(null)	2	Ürünün miktarı
3	DEPONO	NUMBER(38,0)	No	(null)	3	Ürünün depolandığı deponun numarası
4	BIRIMFIYAT	NUMBER(38,3)	No	(null)	4	Ürünün birim fiyatı
5	KATEGORİ	VARCHAR2(20 BYTE)	No	(null)	5	Ürünün kategorisi

### BARKOD Tablosu:

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	URUNAD	VARCHAR2(50 BYTE)	No	(null)	1	Ürünün adı
2	URUNBARKOD	NUMBER(38,0)	No	(null)	2	Ürünün barkodu

### HAREKET Tablosu:

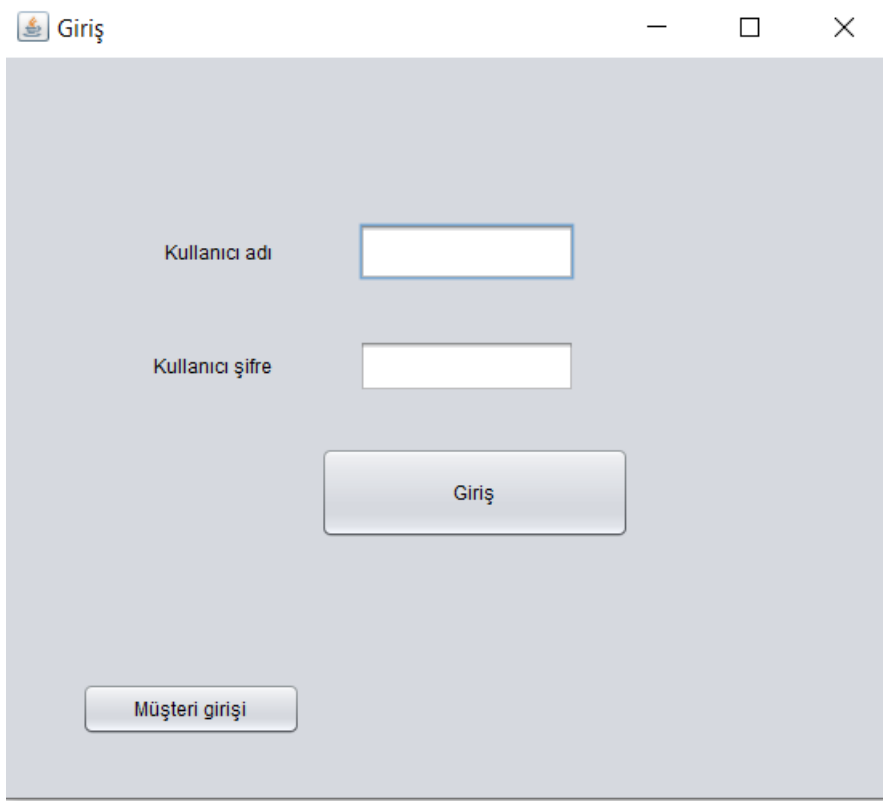
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	HAREKETNO	NUMBER(38,0)	No	(null)	1	Hareketin numarası
2	TARİH	DATE	No	(null)	2	Hareketin tarihi
3	HAREKET	VARCHAR2(50 BYTE)	No	(null)	3	Hareketin içeriği
4	URUNAD	VARCHAR2(50 BYTE)	No	(null)	4	Hareketi yapılan ürünün adı
5	DEPONO	NUMBER(38,0)	No	(null)	5	Ürünün bulunduğu deponun numarası

## 11.GENEL KISITLAMALAR

Sistemde görülen eksik olarak, sistemin arayüz görünümü denilebilir. Swing kısmının görselliği geliştirilebilir. Indexleme yapılabilir sistem performansı için.

## 12. EKRAN GÖRÜNTÜLERİ

**Giriş ekranı görünümü:** Sistemi Kullanacak olanlar müşteri dışında sisteme giriş yapmalılar.



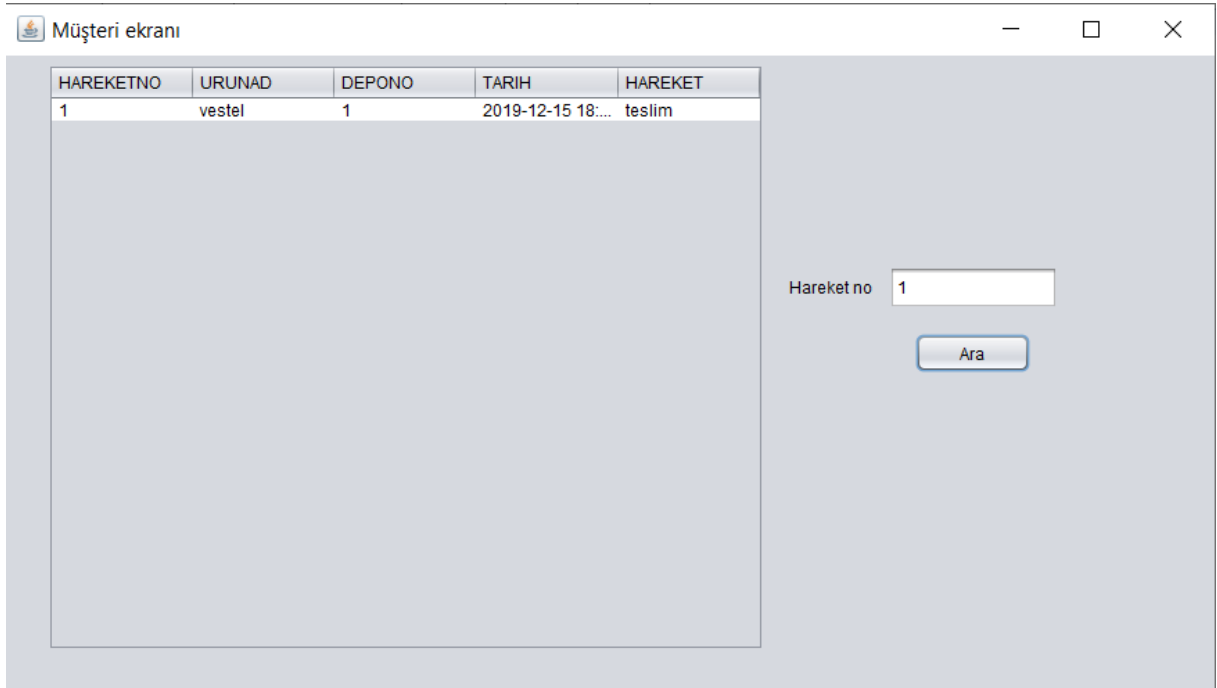
Kullanıcı adı

Kullanıcı şifre

Giriş

Müşteri girişi

**Müşteri ekranı görünümü:** Müşteriler verilen hareketno ile ürününün takibini yapabilir.



HAREKETNO	URUNAD	DEPONO	TARİH	HAREKET
1	vestel	1	2019-12-15 18:...	teslim

Hareket no 1

Ara

**Admin paneli:** Bu ekrana sadece Adminler girebilir kullanıcı ekleme, silme, güncelleme ve yedek hareketleri görme yeteneğine sahiptirler.

Admin paneli

HAREKETNO	TARİH	OLAY
1	2019-11-30 17:56:28.0	EKLEME
2	2019-11-30 22:22:42.0	EKLEME
5	2019-12-01 20:09:41.0	EKLEME
5	2019-12-01 20:10:35.0	GÜNCELLEME
5	2019-12-01 20:16:51.0	SİLME
2	2019-12-01 21:44:34.0	EKLEME
3	2019-12-01 21:46:23.0	EKLEME
2	2019-12-10 15:20:05.0	SİLME
1	2019-12-10 15:20:05.0	SİLME
3	2019-12-10 15:20:05.0	SİLME
123	2019-12-10 15:21:10.0	EKLEME
124	2019-12-10 15:22:55.0	EKLEME
124	2019-12-15 00:50:12.0	SİLME
123	2019-12-15 00:50:12.0	SİLME
1	2019-12-15 00:52:12.0	EKLEME
1	2019-12-15 02:29:22.0	SİLME
1	2019-12-15 14:22:15.0	EKLEME
2	2019-12-15 16:43:31.0	EKLEME
1	2019-12-15 17:37:13.0	GÜNCELLEME
2	2019-12-15 17:37:13.0	GÜNCELLEME
1	2019-12-15 18:08:08.0	GÜNCELLEME
1	2019-12-15 18:08:30.0	SİLME
2	2019-12-15 18:08:39.0	SİLME
1	2019-12-15 18:13:52.0	EKLEME

Ara

Yedek Hareket

Kullanıcı Ad

Kullanıcı Şifre

Kullanıcı Listele

İsim

Soy isim

Pozisyon

Ara

Sil

Ekle

Kullanıcı ad

Yeni şifre

Şifre değiştir

**Kullanıcı ekranı:** Bu ekrana sadece kullanıcı pozisyonundaki bireyler girebilir. Ürün, depo ve hareket bilgilerini ekleyebilir, silebilir ve güncelleyebilirler.

Kullanıcı ekranı

Depo adres güncelleme

Depono

Sil

DepoNo

Depo Ad

Depo Adress

Barkod

Sil

Güncelle

Ürün güncelleme

Barkod

Miktar güncelle

Hareketno

Sil

Güncelle

Hareket Güncelleme

Hareket no

Yeni hareket

Güncelle

Depo ekle

Urun ekle

Hareket No

Hareket

Urun Ad

Depo No

Ekleme

Ekle

Ekle

Listeleme ekranı

**Listeleme ekranı:** Listeleme ekranına Kullanıcı pozisyonuna sahip olanlar girebilir.Ürün, depo ve Hareket tablolarını görebilir aralarındaki ilişkileri görebilirler.

Listeleme ekranı

HAREKETNO	URUNAD	DEPONO	TARİH	HAREKET
2	vestel1	1	2019-12-15 18:14:05.0	kargo
3	vestel1	1	2019-12-15 18:15:50.0	kargo
1	vestel	1	2019-12-15 18:13:52.0	teslim

Ürün barkod

Depo adres

Ara

Ara

Depo no

Hareket no

Ara

Ara

Hepsini listele

Depoları listele

Ürünleri listele

Hareketleri listele

☐ Birim fiyatından büyük

☐ Miktarından büyük

Birimfiyat/miktar

Ara

☐ Ortalama miktardan fazla ürünler

☐ Ortalama birim fiyattan fazla ürünler

☐ Toplam ürün bilgileri

☐ Ürünlerin hareket sayıları

☐ Birim fiyata göre sırala

☐ Miktarla göre sırala

☐ Tarihe göre sırala

### 13. KULLANILAN TRİGGERLAR

Hareketi yapılacak ürünün bulunduğu deponun numarası doğru girilmelidir.

```
1 create or replace trigger hareketdepono BEFORE INSERT on hareket
2 FOR each ROW
3 DECLARE
4 DEPONUMARA INTEGER;
5
6 BEGIN
7 SELECT DEPONO INTO DEPONUMARA FROM URUN WHERE URUNAD = :NEW.URUNAD;
8 IF DEPONUMARA != :NEW.DEPONO THEN
9 RAISE_APPLICATION_ERROR( -20001,
10 'Hareketi yapılacak ürünün doğru depo numarasını giriniz' );
11 END IF;
12
13 END;
```

Hareket tablosuna eklenen bilgiler Musteri tablosuna ekleniyor.

```
1 create or replace trigger musteritabloekleme after insert on HAREKET
2 FOR each ROW
3 BEGIN
4
5 INSERT INTO MUSTERI.MUSTERI VALUES (:NEW.HAREKETNO,SYSDATE, :NEW.HAREKET, :NEW.URUNAD, :NEW.DEPONO);
6 END;
```

Hareket tablosunda yapılan güncellemeler Musteri tablosunda da güncelleniyor.

```
1 create or replace trigger musteritablogüncelleme after update on HAREKET
2 FOR each ROW WHEN (NEW.HAREKET!=OLD.HAREKET)
3 BEGIN
4
5 UPDATE MUSTERI.MUSTERI SET HAREKET=:NEW.HAREKET,TARİH=SYSDATE;
6 end;
```

Hareket tablosunda yapılan silme işlemi Musteri tablosunda da siliniyor.

```
1 create or replace trigger musteritablosilme after delete on HAREKET
2 FOR each ROW
3 BEGIN
4
5 DELETE from MUSTERI.MUSTERI where hareketno=:old.hareketno;
6 END;
```

Ürünün bulunduğu depo no değiştirildiğinde, ona bağlı olarak hareketlerin depo numarası değişiyor.

```
1 create or replace trigger uruncascadegüncelleme after update of DEPONO on URUN
2 FOR each ROW WHEN (NEW.DEPONO!=OLD.DEPONO)
3 BEGIN
4
5 UPDATE HAREKET SET DEPONO=:NEW.DEPONO WHERE DEPONO=:OLD.DEPONO;
6 END;
```

Urun tablosunda miktar negative olamaz.

```
1 create or replace trigger urunsıfırgüncelleme BEFORE update of MIKTAR on URUN
2 FOR each ROW WHEN (NEW.MIKTAR<0)
3
4 BEGIN
5 RAISE_APPLICATION_ERROR( -20001,
6 'NEGATİF SAYI OLAMAZ' );
7 end;
```

Urun tablosunda birim fiyat negative olamaz.

```
1 create or replace trigger urunsıfırgüncelleme2 AFTER update of BIRIMFIYAT on URUN
2 FOR each ROW WHEN (NEW.BIRIMFIYAT<0)
3 BEGIN
4 RAISE_APPLICATION_ERROR( -20001,
5 'NEGATİF SAYI OLAMAZ' );
6 END;
```

Hareket tablosuna eklenen bilgiler, Hareket\_yedek tablosuna eklendi yazar.

```
1 create or replace trigger yedektabloekle BEFORE insert on HAREKET
2 FOR each ROW
3 BEGIN
4 INSERT INTO ADMIN.HAREKET_YEDEK VALUES (:NEW.HAREKETNO,SYSDATE,'EKLEME');
5 END;
```

Hareket tablosunda güncellenen bilgiler, Hareket\_yedek tablosunda güncellendi yazar.

```
1 create or replace trigger yedektablogüncelle BEFORE update on HAREKET
2 FOR each ROW
3 BEGIN
4 INSERT INTO ADMIN.HAREKET_YEDEK VALUES (:NEW.HAREKETNO,SYSDATE,'GÜNCELLEME');
5 END;
```

Hareket tablosunda silinen bilgiler, Hareket\_yedek tablosunda silindi yazar.

```
1 create or replace trigger yedektablosil BEFORE delete on HAREKET
2 FOR each ROW
3 BEGIN
4 INSERT INTO ADMIN.HAREKET_YEDEK VALUES (:OLD.HAREKETNO,SYSDATE,'SİLME');
5 end;
```

## 14. ÖNEMLİ GÖRÜLEN PROSEDÜRLER

Depo tablosuna bilgi ekleme işlemi.

```
1 create or replace procedure depo_ekle(DEPO_NO IN INTEGER,DEPO_ADI IN VARCHAR2,DEPOADRES IN VARCHAR2)
2 AS
3 BEGIN
4 INSERT INTO DEPO VALUES (DEPO_NO,DEPO_ADI,DEPOADRES);
5 END;
```

Depo adı güncelleme işlemi.

```
1 create or replace procedure DEPO_GUNCELLE_AD(DEPO_NO IN NUMBER,DEPO_AD IN VARCHAR2)
2 AS
3 BEGIN
4 UPDATE DEPO SET DEPOADI=DEPO_AD WHERE DEPONO=DEPO_NO;
5 END;
```

Hareket ekleme işlemi.

```
1 create or replace procedure hareket_ekle(HAREKET_NO IN INTEGER,HAREKET IN VARCHAR2,URUNAD IN VARCHAR2,DEPO_NO IN integer)
2 AS
3 BEGIN
4 INSERT INTO HAREKET VALUES (HAREKET_NO,sysdate,HAREKET,URUNAD,DEPO_NO);
5 END;
```

Urun tablosuna ürün bilgisi girme işlemi.

```
1 create or replace procedure urun_ekle(URUN_AD IN VARCHAR2,BARKOD IN NUMBER,MIKTARR IN INTEGER,BIRIM_FIYAT IN decimal,DEPO_NO IN INTEGER,KATEG
2 AS
3 BEGIN
4 INSERT INTO URUN VALUES (URUN_AD,MIKTARR,DEPO_NO,BIRIM_FIYAT,KATEGORİİ);
5 INSERT INTO BARKOD VALUES (URUN_AD,BARKOD);
6 END;
```

Urun tablosunda ürün silme işlemi.

```
1 create or replace procedure URUN_SİL(URUN_BARKOD IN NUMBER)
2 AS
3 BEGIN
4 DELETE URUN WHERE URUNAD=(SELECT U.URUNAD FROM URUN U,BARKOD B WHERE URUNBARKOD=URUN_BARKOD AND B.URUNAD=U.URUNAD);
5 END;
```

Urun birim fiyat arttırma işlemi.

```
1 create or replace procedure URUN_GUNCELLE_BRM_ART(BARKOD IN NUMBER,FIYAT IN NUMBER)
2 AS
3
4 BEGIN
5 UPDATE URUN SET BIRIMFIYAT=BIRIMFIYAT+FIYAT WHERE URUNAD=(SELECT U.URUNAD FROM URUN U,BARKOD B WHERE URUNBARKOD=BARKOD AND B.URUNAD=U.URUNAD);
6
7 END;
```

## 15. ÖNEMLİ GÖRÜLEN JAVA KODLARI

```
import java.sql.CallableStatement;

import java.sql.Connection;
import java.sql.PreparedStatement;
import javax.swing.JOptionPane;
```

Depo tablosuna bilgi ekleme kodu.

```
Connection conn=null;

    CallableStatement cStmt = null;
    try{
        if(jTextField3.getText().equals("")||jTextField1.getText().equals("")||jTextField2.getText().equals(""))
            throw new Exception ("Hata");
        conn=connect.getcon();
        cStmt=conn.prepareCall("{call DEPO_EKLE(?,?,?)}");
        cStmt.setInt(1,Integer.parseInt(jTextField1.getText()));
        cStmt.setString(2,jTextField2.getText());
        cStmt.setString(3,jTextField3.getText());
        cStmt.execute();
    }
    conn.close();
    cStmt.close();
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
// TODO add your handling code here:
```

Hareket tablosuna bilgi ekleme kodu.

```
// TODO add your handling code here:
Connection conn=null;

    CallableStatement cStmt = null;
    try{
        if(jTextField9.getText().equals("")||jTextField11.getText().equals("")||jTextField12.getText().equals("")||jTextField13.getText().equals(""))
            throw new Exception ("Hata");
        conn=connect.getcon();
        cStmt=conn.prepareCall("{call HAREKET_EKLE(?,?,?,?)}");
        cStmt.setInt(1,Integer.parseInt(jTextField9.getText()));
        cStmt.setString(2,jTextField11.getText());
        cStmt.setString(3,jTextField12.getText());
        cStmt.setInt(4,Integer.parseInt(jTextField13.getText()));

        cStmt.execute();
    }
    conn.close();
    cStmt.close();
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
```



Veritabanı – Uygulama bağlantı kodu. Farklı kullanıcılar için farklı kod girişleri bulunmaktadır.

```
public class connect {

    public static Connection getcon() {
        Connection conl=null;

        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            conl=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","Kullanıcı","123");
            //conl = DriverManager.getConnection("jdbc:mysql://localhost:3306/stok?useUnicode=true&char

            if (conl != null) System.out.print("calıstı");

        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }
    }
    return conl;
}

public static Connection getcon1() {
    Connection conl=null;

    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        conl=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","Admin","123");
        //conl = DriverManager.getConnection("jdbc:mysql://localhost:3306/stok?useUnicode=true&char

        if (conl != null) System.out.print("calıstı");

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);

    }
    return conl;
}
```

Hareket güncelleme.

```
Connection conn=null;

CallableStatement cStmt = null;
try{
    if(jTextField17.getText().equals("")||jTextField18.getText().equals(""))
        throw new Exception ("Hata");
    conn=connect.getcon();
    cStmt=conn.prepareCall("{call HAREKET_GUNCELLE(?,?)}");
    cStmt.setInt(1,Integer.parseInt(jTextField17.getText()));
    cStmt.setString(2,jTextField18.getText());

    cStmt.execute();
    conn.close();
    cStmt.close();
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
```

Depo tablosunda depo adı ve depo adresi güncelleme.

```
Connection conn=null;

CallableStatement cStmt = null;
try{
    if(jTextField10.getText().equals("")||jTextField14.getText().equals(""))
        throw new Exception ("Hata");
    conn=connect.getcon();
    if(jCheckBox1.isSelected()){
        cStmt=conn.prepareCall("{call DEPO_GUNCELLE_ADRES(?,?)}");
        cStmt.setInt(1,Integer.parseInt(jTextField10.getText()));
        cStmt.setString(2,jTextField14.getText());
    }
    else if(jCheckBox2.isSelected()){
        cStmt=conn.prepareCall("{call DEPO_GUNCELLE_AD(?,?)}");
        cStmt.setInt(1,Integer.parseInt(jTextField10.getText()));
        cStmt.setString(2,jTextField14.getText());
    }

    cStmt.execute();

    conn.close();
    cStmt.close();
}
catch(Exception e){
    JOptionPane.showMessageDialog(null, e);
}
```

Depo tablosunu listeleme.

```
public ArrayList<Stokdto> Depoliste (){
    ArrayList<Stokdto> stoklist=new ArrayList<Stokdto>();
    Statement st=null;
    ResultSet rs=null;
    Connection con=null;
    try{
        con=connect.getcon();

        st=con.createStatement();

        rs=st.executeQuery("select * from depo");

        Stokdto stok;

        while(rs.next()){
            stok=new Stokdto(rs.getInt("DEPONO"),rs.getString("DEPOADI"),rs.getString("DEPOADRES"));
            stoklist.add(stok);
        }
        con.close();
        st.close();
    }
    catch(Exception e){
        System.out.print("hataaaaaaa");
    }
    return stoklist;
}
```

Hareket tablosunu listeleme.

```
public ArrayList<Stokdto> hareketliste () {
    ArrayList<Stokdto> stoklist=new ArrayList<Stokdto>();
    Statement st=null;
    ResultSet rs=null;
    Connection con=null;
    try{
        con=connect.getcon();

        st=con.createStatement();

        rs=st.executeQuery("select * from hareket");

        Stokdto stok;

        //int depono, int hareketno, String tarih, String hareket, String urunad

        while(rs.next()){

            stok=new Stokdto(rs.getInt("DEPONO"),rs.getInt("HAREKETNO"),rs.getString("TARİH"),rs.getString("HAREKET"),rs.getString("URUNAD")
            );
            stoklist.add(stok);

        }
        con.close();
        st.close();
    }
    catch(Exception e){
        System.out.print("hataaaaaaa");
    }
    return stoklist;
}
```

Hangi ürünün kaçtane hareketi olduğunu listeler.

```
public ArrayList<Stokdto> Listeurunhareket () {
    ArrayList<Stokdto> stoklist=new ArrayList<Stokdto>();
    Statement st=null;
    ResultSet rs=null;
    Connection con=null;
    try{
        con=connect.getcon();

        st=con.createStatement();

        rs=st.executeQuery("SELECT U.URUNAD,count(U.URUNAD) AS HAREKETSAYISI FROM URUN U,HAREKET H WHERE U.URUNAD=H.URUNAD GROUP BY U.URUNAD");

        Stokdto stok;

        while(rs.next()){
            stok=new Stokdto(rs.getString("URUNAD"),rs.getInt("HAREKETSAYISI")
            );
            stoklist.add(stok);
        }
        con.close();
        st.close();
    }
    catch(Exception e){
        System.out.print(e);
    }
    return stoklist;
}
```