

**T.C.
KIRKLARELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ**



SATIŞ İŞLETMESİ OTOMASYONU RAPORU

**Ahmet
ÇELİK
1180505021**

**Gökay
AYYILDIZ
1180505601**

**Murat
DÖĞÜM
1180505032**

**Yusuf Baha
USTA
1180505033**

YAZILIM MÜH. 2.SINIF

20.12.2019

**Fatih AYDIN
NESNEYE YÖNELİK PROGRAMLAMA**

İçindekiler

1.GİRİŞ	3
2.YÖNTEM	3
2.1. Paketleme	3
2.2. Dosya Okuma İşlemi	3
2.3. Dosya Yazma İşlemi	4
2.4. Verilerin Listelenmesi	4
2.5. Giriş Yap Sistemi	5
2.6. Ara yüz	6
2.7. Görev Dağılımı	7
2.8. Sınıf Diyagramları	8
3.SONUÇ	8

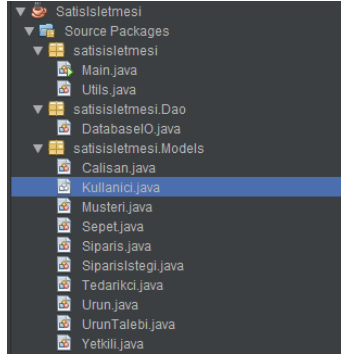
1.GİRİŞ

Dünya nüfusunun artmasıyla birlikte otomasyona duyulan ihtiyaç giderek artmaktadır. İnsanlar artık işlerini makinelerle paylaşmakta, makine gücü giderek artmaktadır. Otomasyona duyulan gereksinimin asıl nedeni hayatımızı kolaylaştırmakla beraber daha pratik hale getirmesi olarak görülmektedir. Bu amaç doğrultusunda proje bir satış otomasyonu olarak hayata geçirilmiştir.

2.YÖNTEM

2.1. Paketleme

Projenin daha kolay okunabilmesi ve karışıklığın giderilmesi açısından ara yüz, model sınıfları ve veri tabanı işlemleri ayrı sınıflarda yazılmış ve sınıflar arasındaki ilişkiye göre paketlemeye gidilmiştir.



Şekil 1: Paketleme

Şekil-1’deki paketlemeye göre proje içerisinde kullanılacak olan nesneler “Models” paketinin altında yer almaktadır. Veri tabanı ile ilgili işlemleri ise Dao (Database Access Object) paketinin altında yer almaktadır.

2.2. Dosya Okuma İşlemi

Dosya okuma işlemini müşteri girişi yaparken, tüm ürünleri listelerken vb. kullandık. Her işlemde öncelikle bir “ArrayList<>” oluşturduk, daha sonra okunacak dosyaların adreslerini belirttik. Okumak için File nesnesi verilen yeni bir FileReader nesnesi oluşturur. Daha sonra BufferedReader yapısını kullanarak dosyadan alacağımız verileri While döngüsü ile line.split(“;”) metodunu kullanarak dosyasından kelime kelime verileri alıp ArrayList ‘in içine depoluyoruz. Metodu main de çağıldığımızda artık istediğimiz hedef dosyayı okuyabilecektir.

2.3. Dosya Yazma İşlemi

Dosya yazma işlemini müşteri eklerken, ürün eklerken vb. kullandık.”
urunEkle() “fonksiyonuna gelen parametreleri kullanıcıdan aldıktan sonra: Kullanıcı numarası, ürün numarası vb. için “Random()” metodunu bir numara verdik. File nesnesi verilen txt dosyasına yazmak için yeni bir FileReadar nesnesi oluşturuyoruz. ”urunEkle()“ fonksiyonuna gelen parametreleri BufferedReader ile oluşturduğumuz nesnenin içine atıyoruz. String Lin kullanarak parametrelerin yazılma sıralarıyla birlikte tanımlıyoruz. Oluşturulan nesne ile belirtilen txt dosyasına yazıyoruz. Main’den ”urunEkle()“ metodunun içine girdiğimiz parametrelerini yazdıracaktır.

```
public void urunEkle(String isim, String fiyat, String stok_adeti, String tedarikci_id) throws IOException {  
    Random rd = new Random();  
    String barkod_no = String.format("%06d%07d", rd.nextInt(999999), rd.nextInt(9999999));  
    File f = new File(Utils.urunPath);  
    if(!f.exists())  
        f.createNewFile();  
    FileWriter fw = new FileWriter(f,true);  
    BufferedWriter bw = new BufferedWriter(fw);  
    String line = "\n" + barkod_no + ";" + isim + ";" + fiyat + ";" +  
        stok_adeti + ";" + tedarikci_id;  
    bw.write(line);  
    bw.flush();  
    bw.close();  
    System.out.println("-----");  
    System.out.println("Yeni ürün oluşturuldu");  
}
```

Şekil 2: Örnek ürün ekleme fonksiyonu

2.4. Verilerin Listelenmesi

Projede, aynı yapıya ait nesneleri bir arada tutmak için Array List yapısı kullanılmaktadır. arraylist yapısını bunlardan ayıran özellikleri nesneleri birden fazla liste şeklinde sıralı olarak tutması ve bir nesneden birden fazla bulundurabilmesi diyebiliriz.

```

public ArrayList<Urun> tumUrunler() throws IOException {
    ArrayList<Urun> results = new ArrayList<>();
    File f = new File(Utils.urunPath);
    FileReader fr = new FileReader(f);
    BufferedReader br = new BufferedReader(fr);
    String line = "";
    while((line = br.readLine()) != null) {
        if(line.isEmpty()) continue;
        Urun urun = new Urun();
        String[] data = line.split(";");
        urun.setBarkod_no(data[0]);
        urun.setIsim(data[1]);
        urun.setFiyat(Integer.valueOf(data[2]));
        urun.setStok_adeti(data[3]);
        urun.setTedarikci_id(data[4]);
        results.add(urun);
    }
    br.close();
    return results;
}

```

Şekil 3: Örnek bir ürün listeleme fonksiyonu

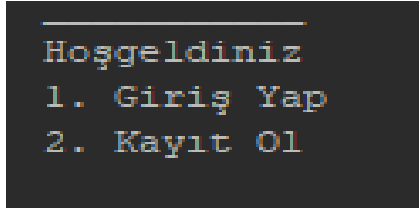
Şekil-3’de ürün sınıfından ürün arraylist oluşturulmuştur. Daha sonra yeni bir dosya nesnesi oluşturulmuş yolu Utils.UrunPath şeklinde tanımlanmış bir dosyayı çağırmıştır. Dosya Okuma (File Reader) ve bufferedreader oluşturuldu. Dosya içerisindeki her satır line adındaki bir değişkene atılır ve dosyanın sonuna gelene kadar bütün satırlar değişkene atılır ve dosyanın sonuna gelene kadar bütün satırlar konsola yazdırılır. bufferedreader yüksek tampon bellek sağlar. Oluşturulan yeni ürün nesnesine ürün var olan ürün sınıfının özelliklerine urun.set şeklinde tanımlayarak hangi ürünün barkod no isim vb. gibi özellikleri ile bulabilerek ürün listesini oluşturmuş olduk en son işlem bittikten sonra bufferedreader.close ile kapatmış oluruz.

2.5. Giriş Yap Sistemi

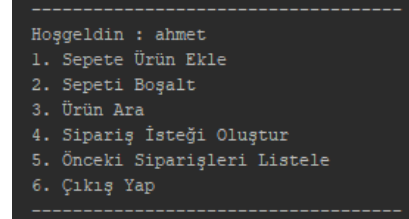
Kullanıcı tarafından giriş yapma işlemi başlatıldığında e-posta ve parola istenir. İşlem sonucunda bu bilgiler DatabaseIO sınıfında girisYap fonksiyonuna parametre olarak iletilir. Bu fonksiyon, kullanıcı bilgilerin bulunduğu dosyalar (Musteri.txt, Calisan.txt ve Yetkili.txt) bir Array List yapısında tutar ve for each döngüsüyle dosyaları okur. E-Posta ve parolası eşleşen hesaba göre (müşteri, çalışan veya yetkili) bir nesne oluşturup Object tipinde döndürür. Main sınıfında ise da gelen nesnenin sınıf adına göre tip dönüşü işlemi yapıp ilgili sayfaya yönlendirir.

2.6. Ara yüz

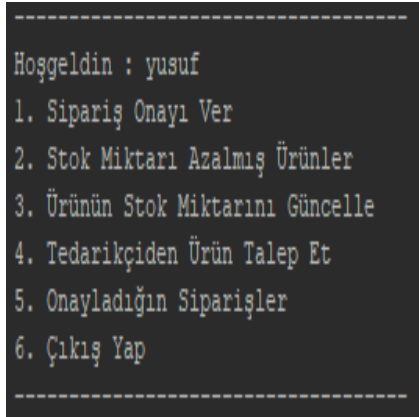
Projenin ara yüzü konsol ara yüzü şeklinde geliştirilmiştir. Kullanıcı uygulamayı ilk açtığında “Giriş Yap” ve “Kayıt Ol” seçeneklerini kullanabilmektedir (Şekil-4). Kayıt ol sekmesinde müşteri kaydı için gerekli bilgiler kullanıcıdan temin edilip giriş yap ekranına yönlendirilir. E-Posta ve parola girildikten sonra kullanıcının yetkisine göre seçenekler kullanıcıya sunulur (Şekil-5, Şekil-6, Şekil-7).



Şekil 4: Ana ekran



Şekil 5: Müşteri Ekranı



Şekil 6: Çalışan ekranı



Şekil 7: Yetkili Ekranı

2.7. Görev Dağılımı

Görev dağılımı, Tablo-1’deki gibidir.

Tablo 1:Görev dağılımı

	Görevlendirme 1	Görevlendirme 2	Görevlendirme 3
Yusuf Baha USTA	<ul style="list-style-type: none">• Müşteri sınıfı• Çalışan sınıfı• Yetkili sınıfı	<ul style="list-style-type: none">• Ürün arama	<ul style="list-style-type: none">• Stok miktarı azalmış ürünleri listeleme• Ürün kaydı• Siparişleri Listeleme
Murat DÖĞÜM	<ul style="list-style-type: none">• Kullanıcı sınıfı• Tedarikçi sınıfı	<ul style="list-style-type: none">• Önceki siparişleri görüntüleme• Sipariş isteği oluşturma	<ul style="list-style-type: none">• Tedarikçide n ürün talebi• Tedarikçi kaydı• Sipariş onayı verme
Gökay AYYILDIZ	<ul style="list-style-type: none">• Ürün sınıfı• Ürün Talebi sınıfı	<ul style="list-style-type: none">• Ürünleri listeleme	<ul style="list-style-type: none">• Onaylanan siparişler• Çalışan kaydı• Sipariş onayı verme
Ahmet ÇELİK	<ul style="list-style-type: none">• Sipariş İsteği sınıfı• Sipariş sınıfı	<ul style="list-style-type: none">• Giriş yap• Müşteri kaydı• Sepete ürün ekleme-kaldırma	<ul style="list-style-type: none">• Stok miktarı güncelleme• Sipariş onayı verme• Yetkili kaydı• Tedarikçi Listeleme• Konsol ara yüzü

2.8. Sınıf Diyagramları



3.SONUÇ

Geliştirilen proje; müşteriye rahat alışveriş yapma ve siparişleri takip etme kolaylığı sağlamaktadır. Yetkiliye ise çalışan, müşteri ve tedarikçileri kolay yönetebilme imkânı sunmaktadır. Sonuç olarak bu otomasyon sayesinde birçok kişi kullanım kolaylığına ve çeşitli faydalara sahip olmaktadır.

4.KAYNAKÇA

1. <https://www.geeksforgeeks.org/abstract-classes-in-java/>
2. <https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri>
3. <http://mesutpiskin.com/blog/jsf-datatable-kullanimi-veri-listeleme.html>

