

STOK TAKİP UYGULAMASI RAPORU

ÖZET

Giyim sektöründeki işletmeler için geliştirilen stok takip otomasyon projemiz, iş süreçlerini daha sistemli ve verimli bir şekilde yönetmeyi hedeflemektedir. Bu program, mağazaların ürünleri sistemimize kolayca kaydedip sayılarını güncellemelerine olanak tanıyarak, zaman ve maliyet tasarrufu sağlamayı amaçlamaktadır. İş yeri sahiplerine uzaktan erişim imkânı sunarak, mağazalarını her an kontrol etme şansı verirken, manuel sayım ve takip işlemlerini ortadan kaldırarak daha fazla iş yapılmış olacaktır. Güncel altyapısı ve hızlı işlem yeteneği ile bu otomasyon projesi, giyim mağazalarına, iş süreçlerini daha etkili bir şekilde yönetmelerine olanak tanımaktadır.

İçindekiler

ÖN SÖZ	I
ÖZET	II
KONUSU, AMACI VE BEKLENEN SONUCU	1
HEDEF KİTLE	2
PROJENİN İÇERİĞİ	3
İLK SAYFA İÇERİĞİ	3
İKİNCİ SAYFA İÇERİĞİ	3
YETKİLİ GİRİŞ SAYFASI.....	4
YETKİLİ GİRİŞ KODLARI	5
STOK SORGU SAYFASI.....	7
STOK SORGU KODLARI.....	11
BAĞLANTI KODLARI	12
PROJE ZAMAN ÇİZELGESİ.....	15
PROJE EKİBİ VE YÖNETİMİ	15
Ekibimiz.....	15
PROJE SONUÇLARI.....	15
Başarılar	15
Zorluklar	15
Gelecek İyileştirmeler.....	15
Proje Değerlendirmesi.....	15
KAYNAKÇA.....	16

ŞEKİLLER TABLOSU

Şekil 1: Yetkili Tablo	4
Şekil 2: İptal Butonu	4
Şekil 3: Giriş Butonu	4
Şekil 4: Giriş Hata	4
Şekil 5: Yetkili Giriş	4
Şekil 6: Yetkili Verileri	4
Şekil 7: Kod İçerikleri	5
Şekil 8: Kod İçerikleri	5
Şekil 9: Kod İçerikleri	5
Şekil 10: Kod İçerikleri	5
Şekil 11: Kod İçerikleri	5
Şekil 12: Kod İçerikleri	6
Şekil 13: Kod İçerikleri	6
Şekil 14: Kod İçerikleri	6
Şekil 15: Sorgu Ekranı	7
Şekil 16: Ürünler Verisi	7
Şekil 17: Ürünler Tabloları	7
Şekil 18: Listele Butonu	8
Şekil 19: Listele Sonucu Vt	8
Şekil 20: Kaydet Butonu	8
Şekil 21: Kaydet Butonu Sonucu	8
Şekil 22: Kaydet Sonucu Vt	8
Şekil 23: Güncelle Butonu	9
Şekil 24: Güncelle Sonucu	9
Şekil 25: Silme Sonucu	9
Şekil 26: Sil Butonu	9
Şekil 27: Sorgula Butonu	10
Şekil 28: Alana Göre Sorgu	10
Şekil 29: Kod İçerikleri	11
Şekil 30: Kod İçerikleri	11
Şekil 31: Kod İçerikleri	11
Şekil 32: Kod İçerikleri	12
Şekil 33: Kod İçerikleri	12
Şekil 34: Kod İçerikleri	13
Şekil 35: Kod İçerikleri	13
Şekil 36: Kod İçerikleri	13
Şekil 37: Kod İçerikleri	13
Şekil 38: Kod İçerikleri	14
Şekil 39: Kod İçerikleri	14
Şekil 40: Kod İçerikleri	14
Şekil 41: Proje Zaman Çizelgesi	15

KONUSU, AMACI VE BEKLENEN SONUCU

Giyim sektöründeki işletmeler için tasarlanan stok takip otomasyon projemiz, iş süreçlerini düzenli ve verimli bir şekilde yönetmeyi hedefler. Proje, giyim mağazalarının günlük işlerini kolaylaştırarak zaman ve maliyet tasarrufu sağlamayı amaçlar. Ana hedef, mağazalardaki ürünleri sistemimize kaydetme ve stokları güncelleme kolaylığı sunarak işletme sahiplerine anlık stok takibi imkânı sağlamaktır. Otomasyon sistemi, manuel stok sayımını ortadan kaldırarak işletmelere daha fazla iş yapma olanı sunar.

Projemiz ayrıca iş yeri sahiplerine uzaktan erişim imkânı tanıyarak, mağazalarını her an kontrol etme şansı verir. Bu özellik, işletme sahiplerine esneklik sağlayarak işlerini istedikleri yerden yönetme imkânı sunar, fiziksel olarak mağazada bulunma zorunluluğunu ortadan kaldırır ve zaman yönetimini artırır.

Güncel veritabanı ve hızlı işlem yapabileceğimiz tek sayfalık bu otomasyon projesi ile giyim mağazalarına iş süreçlerini etkili bir şekilde yönetirler. Projemizin beklenen sonucu, işletmelerin stok yönetimi ve takibi konusundaki sorunlarını minimuma indirerek iş verimliliklerini artırmaktır. Bu sayede işletmeler, stok süreçlerini daha etkili bir şekilde yöneterek zaman ve maliyet tasarrufu sağlayabilir, müşteri memnuniyetini artırabilir ve rekabet avantajıyla da bir adım önde olabilirler.

HEDEF KİTLE

Stok takip uygulamamızın hedef kitlesi, özellikle küçük ve orta ölçekli işletmelerin stok yönetim süreçlerini kolaylaştırmayı amaçlamaktadır. Özellikle giyim, bakkal ve benzeri sektörlerde faaliyet gösteren işletmeler için tasarlanmış olan bu uygulama, stok süreçlerinden sorumlu olan işletme sahipleri ve stok yöneticileri gibi yetkilileri hedeflemektedir.

Bu uygulama, küçük işletmelerin stok takibi konusundaki zorlukları en aza indirerek zaman ve maliyet tasarrufu sağlamayı amaçlamaktadır. İşletme sahiplerine ve stok yöneticilerine, ürünleri sisteme kolayca entegre edebilme, stok durumlarını anlık olarak takip edebilme ve gerekirse müdahale edebilme imkanı sunarak operasyonel süreçleri daha etkili bir şekilde yönetmelerine yardımcı olacaktır. Bu sayede, stok süreçleri üzerinde daha fazla kontrol sağlanacak ve işletmeler günlük operasyonlarını daha verimli bir şekilde yönetebileceklerdir.

Böylelikle, stok takip uygulaması, hedef kitlesi olan küçük ve orta ölçekli işletmeler için birçok avantaj sunarak, işletmelerin daha etkili, düzenli ve karlı bir stok yönetimine ulaşmalarına destek olmayı amaçlamaktadır.

PROJENİN İÇERİĞİ

Projenin temel yapısı, iki sayfalık bir otomasyon sistemini içermektedir. İlk sayfa, yetkili kullanıcıların sisteme giriş yapabildiği ana sayfadır. Başarılı bir girişin ardından kullanıcı, ikinci sayfaya yönlendirilmektedir. İkinci sayfa, stok takip uygulamasının ana işlevlerini barındıran bir kontrol panelini içermekte ve ayrıca giriş hatası durumunda hata mesajı gösterip, iptal butonuna basıldığında yazılanları sıfırlama özelliğini taşımaktadır.

İLK SAYFA İÇERİĞİ:

- 1.1. Giriş Butonu:** Yetkili kullanıcılar, kullanıcı numarası ve şifre bilgilerini girerek sisteme giriş yapabilirler. Yanlış bir giriş durumunda sistem, hata mesajı ile kullanıcıyı bilgilendirir.
- 1.2. Hata Mesajı:** Yanlış giriş durumunda kullanıcıya anında bir hata mesajı gösterilir. Mesaj, kullanıcının hatalı giriş yaptığını belirten mesaj içerir.
- 1.3. İptal Butonu:** İptal butonuna basıldığında, girilen bilgilerin sıfırlanması sağlanır. Bu özellik, kullanıcının hatalı giriş sonrasında kolayca yeni giriş denemesi yapabilmesini sağlar.

İKİNCİ SAYFA İÇERİĞİ:

- 1.4. Kaydet Butonu:** Bu buton, kullanıcıların yeni stok girişlerini sisteme eklemelerini sağlar. Eklenen veriler, veri tabanına anında kaydedilir, böylece güncel stok bilgileri tutulur.
- 1.5. Sil Butonu:** Sil butonu, belirli bir stok girişini sistemden kaldırmak için kullanılır. Silinen veri, veri tabanındaki kayıtlardan anında çıkarılır.
- 1.6. Güncelle Butonu:** Güncelle butonu, mevcut stok bilgilerini düzenleme olanağı sunar. Yapılan güncellemeler, veri tabanında anlık olarak kaydedilir.
- 1.7. Sorgulama Butonu:** Sorgulama butonu, belirli kriterlere göre stok verilerini filtreleme imkanı sağlar. Bu sayede kullanıcılar, istedikleri kriterlere uygun olarak verileri sorgulayabilir.
- 1.8. Listele Butonu:** Listele butonu, tüm stok verilerini görüntüleme yetkisi sağlar. Bu işlem, veri tabanındaki tüm stok kayıtlarını anlık olarak listeleyerek kullanıcıya genel bir bakış sunar.

Veri tabanında yapılan tüm işlemler, kullanıcının uygulama üzerinden gerçekleştirdiği herhangi bir hareketle anlık olarak güncellenir. Bu sayede, stok takip uygulaması her zaman güncel ve doğru verilere erişim sağlar. Bu otomasyon sistemi, stok yönetimi süreçlerini kullanıcı dostu bir arayüzle etkili bir şekilde yönetmeyi amaçlamaktadır.

YETKİLİ GİRİŞ SAYFASI



YETKİLİ GİRİŞ

YETKİLİ ID

PAROLA

GİRİŞ **İPTAL**

NERCŞ STOK TAKİP

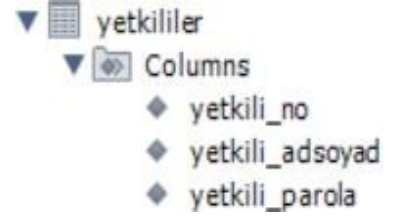
Şekil 5: yetkili giriş



Şekil 3: giriş butonu



Şekil 2: iptal butonu



yetkililer
Columns
yetkili_no
yetkili_adsoyad
yetkili_parola

Şekil 1: yetkili tablo



Şekil 4: giriş hata

Başlangıçta, içeriği belirleyip renkleriyle dikkat çeken bir tasarım oluşturmayı hedefledik. İlk olarak giriş ekranının içeriğini düzenledik ve ardından paneller, butonlar ve labellar ile tasarımı renklendirdik. Arka yüzüne geçtiğimizde ise, özellikle yetkililerin kullanacağı bir tablo oluşturduk. Her yetkili için benzersiz bir kimlik (ID) ve parola belirledik. Doğru parola girildiğinde, kullanıcıyı stok ile ilgili sayfaya yönlendirdik. İptal butonuna basılması durumunda, yazılanları direkt olarak sıfırladık. Eğer yanlış bir parola girilirse, hata uyarısı içeren bir mesaj gönderdik.

	yetkili_no	yetkili_adsoyad	yetkili_parola
▶	12345	Nisa Usta	12345
	23456	Şimal Bülbül	12345
	34567	Eray Güngör	12345
	45678	Öznur Ayazoğlu	12345
	56789	Rıdvan Beyiş	12345
	67890	Ceydanur Özgür	12345
•	NULL	NULL	NULL

Şekil 6: yetkili verileri

Giriş kod içeriğine gelirse bu şekildedir:

YETKİLİ GİRİŞ KODLARI

Şekil 7: kod içerikleri

```
yetkili_giris.java x baglanti.java stok_sorgu.java
1 package stoktakipprojesi;
2 //Gerekli Java sınıflarını içe aktarıyoruz
3 import java.awt.Color;
```

Şekil 8: kod içerikleri

```
//Giris penceresi için ana sınıf
public class yetkili_giris extends JFrame {
    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField txt_id;
    private JLabel yetkili_id;
    private JLabel parola;
    private JPasswordField txt_parola;
    private JButton girisyap_butonu;
    private JButton iptal_butonu;
    static String no;
    static String sifre;
    private JLabel lblNewLabel_2;
```

Şekil 9: kod içerikleri

```
// Uygulamayı başlatmak için ana metod
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                // Giris penceresi olusturuluyor ve görünür yapılıyor
                yetkili_giris frame = new yetkili_giris();
                frame.setVisible(true);
            } catch (Exception e) {
                // Hata durumunda hata izleme çıktısını yazdırma
                e.printStackTrace();
            }
        }
    });
}
```

Şekil 10: kod içerikleri

```
// Kullanıcı girişi için etiket ve metin alanları oluşturma
JLabel yetkili_giris = new JLabel("YETKİLİ GİRİŞ");
yetkili_giris.setHorizontalAlignment(SwingConstants.CENTER);
yetkili_giris.setBackground(new Color(255, 255, 255));
yetkili_giris.setForeground(new Color(255, 128, 0));
yetkili_giris.setFont(new Font("Bookman Old Style", Font.BOLD, 35));
yetkili_giris.setBounds(477, 64, 410, 102);
contentPane.add(yetkili_giris);
```

Şekil 11: kod içerikleri

```
// Giriş ve İptal butonları oluşturma
girisyap_butonu = new JButton("GİRİŞ");
girisyap_butonu.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseExited(MouseEvent e) {
        girisyap_butonu.setBackground(getBackground());
    }
    @Override
    public void mouseEntered(MouseEvent e) {
        girisyap_butonu.setBackground(Color.ORANGE);
    }
});
```

Yetkili giriş adında bir sayfa açarak giriş ekranını bu sayfa üzerinden iletlettik. Yandaki görselde klasör adı ve import ettiklerimiz görünmektedir.

Bu sınıf bir kullanıcı giriş penceresi tasarlamak ve bu pencerede giriş bilgilerini almak için kullanılır. Etiketler, metin alanları ve düğmeler gibi Swing bileşenleri kullanarak bir kullanıcı arayüzü oluşturulmuştur.

Bu Java kodu, bir Swing uygulamasında kullanıcı giriş penceresini oluşturarak uygulamayı başlatır ve görünür hale getirir.

Etiketin görüntüsü ve stil özellikleri belirli renk, yazı tipi, boyut ve hizalamayı içerir.

Fare buton üzerine gelindiğinde (mouseEntered), düğmenin arka plan rengini değiştirir ve fare buton üzerinden çıktığında (mouseExited), düğmenin arka plan rengini varsayılan arka plan rengine geri döndürür. Bu, kullanıcı arayüzünde butonun görsel geribildirimini sağlamak için kullanılır.

```

giris yap_botonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Veritabanında yetkilinin doğruluğunu kontrol etme
        no = txt_id.getText();
        sifre = txt_parola.getText();
        String sql_sorgu = "SELECT count(yetkili_no) as giris FROM yetkililer where yetkili_no=" + no
            + " and yetkili_parola=" + sifre + " ";

        ResultSet myRs = baglanti.yap();
        myRs = baglanti.sorgula(sql_sorgu);

        try {
            while (myRs.next()) {
                if (myRs.getInt("giris") == 1) {
                    // Giriş başarılı ise stok_sorgu penceresini aç
                    stok_sorgu srg = new stok_sorgu();
                    srg.setVisible(true);
                    setVisible(false);
                } else {
                    // Hatalı giriş durumu
                    JOptionPane.showMessageDialog(null, "Hatalı Giriş! Yetkili ID veya Şifre hatalı.",
                        "Hata", JOptionPane.ERROR_MESSAGE);
                }
            }
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
    }
}

```

Şekil 12: kod içerikleri

Bu Java kodu, "GİRİŞ" butonuna bir ActionListener ekler. Bu ActionListener, butona tıklandığında gerçekleşen olayları kontrol eder. Butona tıklandığında, kullanıcının girdiği yetkili ID ve şifre bilgilerini alır, bu bilgilerle bir SQL sorgusu oluşturarak veritabanında yetkilinin doğruluğunu kontrol eder. Eğer doğrulama başarılıysa, "stok_sorgu" adlı bir pencereyi görünür yapar ve mevcut pencereyi gizler. Doğrulama başarısızsa bir hata iletişim kutusu görüntüler.

```

iptal_botonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Metin alanlarını temizleme
        txt_id.setText("");
        txt_parola.setText("");
    }
}

```

Şekil 13: kod içerikleri

Butona tıkanıldığında, kullanıcı giriş penceresinde bulunan yetkili ID ve şifre metin alanlarını temizler, yani içerdikleri veriyi siler. Bu, kullanıcının giriş bilgilerini silmek ve yeni giriş yapmak veya işlemi iptal etmek istediğinde kullanılır.

```

// Logonun bulunduğu etiket ve yazılı logonun adını içeren etiketler
JLabel lblNewLabel = new JLabel("");
lblNewLabel.setFont(new Font("Bookman Old Style", Font.BOLD, 25));
lblNewLabel.setIcon(new ImageIcon("C:\\Users\\000008\\Desktop\\NESNE TABANLI\\depo-stok-programi.png"));
lblNewLabel.setBounds(87, 0, 274, 333);
panel.add(lblNewLabel);

```

Şekil 14: kod içerikleri

Bu etiket, bir panel içinde belirli bir konum ve boyutla yerleştirilmiştir. Bu genellikle kullanıcı arayüzüne estetik veya tanıtım amacıyla bir logo eklemek için kullanılır.

STOK SORGU SAYFASI

STOK SORGU

ÜRÜN No

ÜRÜN Adı

ÜRÜN Adedi

ÜRÜN Kategorisi

ÜRÜN Bedeni

ÜRÜN Rengi

ÜRÜN Fiyat

KAYDET

GÜNCELLE

SİL

STOKTAKİLER

LİSTELE

Ürün No	Ürün Adı	Ürün Adedi	Ürün Kategorisi	Ürün Bedeni	Ürün Rengi	Ürün Fiyat
1	Pantolon	99	Erkek	M	Mavi	629.99
2	Ceket	15	Kadın	S	Siyah	1299.99
3	Mont	56	Erkek	L	Beyaz	1299.99
4	Pijama	78	Kadın	XL	Pembe	279.99
5	Etek	26	Kadın	S	Kırmızı	159.99
6	Şort	200	Çocuk	XS	Turuncu	254.9
7	Pijama	79	Kadın	XL	Pembe	289.99
8	Gömlek	256	Erkek	M	Lacivert	356.99
9	Body	45	Çocuk	XS	Mor	74.9
10	Hırka	257	Kadın	L	Yeşil	399.9

ALAN: Ürün No

SORGULA

ÇIKIŞ YAP

Şekil 15: sorgu ekranı

Doğru giriş yapıldıktan sonra, stok sorgularını gerçekleştirebileceğimiz sayfaya yönlendirildik. Bu sayfanın tasarımını planlamadan önce, tablo içerikleri için bir hazırlık yaptık. Ürünler tablosunu oluşturarak içeriğini belirledik ve ardından kodlarla birlikte veritabanını oluşturduk. Bu sayede, butonlar aracılığıyla stok işlemlerini gerçekleştirebilecek bir platform oluşturduk.

Bu platformda stokla ilgili çeşitli işlemler yapılabilir. Kaydet butonu ile yeni stok bilgileri ekleyebilir, güncelle butonu ile mevcut stokları güncelleyebilir, sil butonu ile stokları kaldırabilir, listele butonu ile mevcut stokları görebilir, sorgula butonu ile özel sorgular yapabilir ve çıkış butonu ile platformdan çıkış yapabilirsiniz.

urunler

Columns

- urun_no
- urun_adi
- urun_adet
- urun_kategori
- urun_beden
- urun_renk
- urun_fiyat

Şekil 17: ürünler tabloları

urun_no	urun_adi	urun_adet	urun_kategori	urun_beden	urun_renk	urun_fiyat
1	Pantolon	99	Erkek	M	Mavi	629.99
2	Ceket	15	Kadın	S	Siyah	1299.99
3	Mont	56	Erkek	L	Beyaz	1299.99
4	Pijama	78	Kadın	XL	Pembe	279.99
5	Etek	26	Kadın	S	Kırmızı	159.99
6	Şort	200	Çocuk	XS	Turuncu	254.9
7	Pijama	79	Kadın	XL	Pembe	289.99
8	Gömlek	256	Erkek	M	Lacivert	356.99
9	Body	45	Çocuk	XS	Mor	74.9
10	Hırka	257	Kadın	L	Yeşil	399.9

Şekil 16: ürünler verisi

BUTONLAR

STOKTAKİLER							LİSTELE
Ürün No	Ürün Adı	Ürün Adedi	Ürün Kategorisi	Ürün Bedeni	Ürün Rengi	Ürün Fiyat	
17	Elbise	74	Kadın	XL	Pembe	119.9	
1	Pantolon	99	Erkek	M	Mavi	629.99	
2	Ceket	15	Kadın	S	Siyah	1299.99	
3	Mont	56	Erkek	L	Beyaz	1299.99	
4	Pijama	78	Kadın	XL	Pembe	279.99	
5	Etek	26	Kadın	S	Kırmızı	159.99	
6	Şort	200	Çocuk	XS	Turuncu	254.9	
7	Pijama	79	Kadın	XL	Pembe	289.99	
8	Gömlek	256	Erkek	M	Lacivert	356.99	
9	Body	45	Çocuk	XS	Mor	74.9	
10	Hırka	257	Kadın	L	Yeşil	399.9	

1. LİSTELE BUTONU: Veri tabanında olan tüm stoktaki ürünleri güncel olarak listeler.

Şekil 18: listele butonu

urun_no	urun_adi	urun_adet	urun_kategori	urun_beden	urun_renk	urun_fiyat
1	Pantolon	99	Erkek	M	Mavi	629.99
2	Ceket	15	Kadın	S	Siyah	1299.99
3	Mont	56	Erkek	L	Beyaz	1299.99
4	Pijama	78	Kadın	XL	Pembe	279.99
5	Etek	26	Kadın	S	Kırmızı	159.99

Şekil 19: listele sonucu vt.

Ürün No	25
Ürün Adı	Mont
Ürün Adedi	45
Ürün Kategorisi	Kadın
Ürün Bedeni	M
Ürün Rengi	Kırmızı
Ürün Fiyat	1259.99
KAYDET	
GÜNCELLE	
SİL	

2. KAYDET BUTONU: Ürünleri listeledikten sonra olan ID yazmamak şartıyla tek tek istenilen temel bilgiler girilir.

Şekil 20: kaydet butonu

23	Hırka	251	Kadın	L	Kahverengi	399.9
24	Kaban	80	Kadın	M	Kahverengi	1459.99
25	Mont	45	Kadın	M	Kırmızı	1259.99

Şekil 21: kaydet butonu sonucu

Kaydettikten sonraki listele butonuna basınca gelen güncel eklenen ürün.

24	Kaban	80	Kadın	M	Kahverengi	1459.99
25	Mont	45	Kadın	M	Kırmızı	1259.99
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 22: kaydet sonucu vt.

Aynı şekilde kaydet butonuna bastıktan sonra eklenen ürünün veri tabanındaki görüntüsü

Ürün No: 24
 Ürün Adı: Kaban
 Ürün Adedi: 80
 Ürün Kategorisi: Kadın
 Ürün Bedeni: M
 Ürün Rengi: Kahverengi
 Ürün Fiyat: 1459.99

KAYDET GÜNCELLE

SİL

Şekil 23: güncelle butonu

3. GÜNCELLE BUTONU: Güncellemek istediğimiz ürünü ister tek tek bilgilerini listeden bakıp tekrar yazarak güncelle istersen listelenen stoktaki ürünlerden hangisini güncellemek istiyorsan listeden üzerine gelip basıp direkt olarak yandaki tabloda yazmaktadır. Onun üzerinden otomatik olarak güncelleme yapabilirsiniz.

23	Hırka	251	Kadın	L	Kahverengi	399.9
24	Kaban	80	Kadın	M	Kahverengi	1459.99

Şekil 24: güncelle sonucu

Güncelleme yaptıktan sonra listele butonuna basınca güncel veriyi görebilirsiniz.

Ürün No: 25
 Ürün Adı: Mont
 Ürün Adedi: 45
 Ürün Kategorisi: Kadın
 Ürün Bedeni: M
 Ürün Rengi: Kırmızı
 Ürün Fiyat: 1259.99

KAYDET GÜNCELLE

SİL

STOKT

Ürün No

15
16
17
18
19
20
21
22
23
24
25

Şekil 26: sil butonu

4. SİL BUTONU: Silmek istediğimiz ürünü ister tek tek bilgilerini listeden bakıp tekrar yazarak silebiliriz ya da listelenen stoktaki ürünlerden hangisini silmek istiyorsak listeden üzerine gelip basıp direkt olarak yandaki tabloda yazmaktadır. Onun üzerinden otomatik olarak silme işlemi yapabilirsiniz.

23	Hırka	251	Kadın	L	Kahverengi	399.9
24	Kaban	80	Kadın	M	Kahverengi	1459.99
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Şekil 25: silme sonucu

STOKTAKİLER

LiSTELE

Ürün No	Ürün Adı	Ürün Adedi	Ürün Kategori	Ürün Bedeni	Ürün Rengi	Ürün Fiyat
5	Etek	26	Kadın	S	Kırmızı	159.99
12	Elbise	82	Çocuk	XS	Pembe	119.9
15	Etek	14	Kadın	XS	Pembe	179.99
17	Elbise	74	Kadın	XL	Pembe	119.9

ALAN:

SORGULA

ÇIKIŞ YAP

Şekil 27:sorgula butonu

5. **SORGULA BUTONU:** Sorgula butonunda alan kısmından seçim yapıp sorgulamak istediğin ürünü kısaca yazıp sorgulasan bile direkt ilişkili ürünleri listeler.

ALAN:

SORGULA

Ürün No

Ürün Adı

Ürün Adedi

Ürün Kategoris

Ürün Bedeni

Ürün Rengi

Ürün Fiyatı

Şekil 28: alana göre sorgu

STOK SORGU KODLARI

```
// Tablo modeli
DefaultTableModel modelim= new DefaultTableModel();//tabloyu dinamik hale getiriyoruz
Object[] kolonlar={"Ürün No","Ürün Adı","Ürün Adedi","Ürün Kategorisi","Ürün Bedeni",
"Ürün Rengi","Ürün Fiyat"};
Object[] satirlar= new Object[7];
private JTextField txt_adsorgu;
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                stok_sorgu frame = new stok_sorgu();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

Şekil 29: kod içerikleri

Bu Java programı, bir stok sorgulama uygulamasının grafiksel kullanıcı arayüzünü oluşturur ve dinamik bir tablo kullanarak ürün bilgilerini gösterir.

```
JButton kaydet_butonu = new JButton("KAYDET");
// "KAYDET" butonu için action listener
kaydet_butonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Metin alanlarından girdi değerlerini al
        String no,adi,adet,beden,kategori,renk,fiyat,sql_sorgu;
        no = txt_no.getText();
        adi = txt_adi.getText();
        adet = txt_adet.getText();
        kategori = txt_kategori.getText();
        beden = txt_beden.getText();
        renk = txt_renk.getText();
        fiyat = txt_fiyat.getText();
        // Veritabanına ekleme işlemini buraya ekleyebilirsiniz
        sql_sorgu = "INSERT INTO urunler(urun_no, urun_adi, urun_adet, urun_kategori, urun_beden, urun_renk, urun_fiyat) "
        + "VALUES (" + no + ", " + adi + ", " + adet + ", " + kategori + ", " +
        + beden + ", " + renk + ", " + fiyat + ")";
        // System.out.println(sql_sorgu);
        baglanti.ekle(sql_sorgu);
    }
});
```

Şekil 30: kod içerikleri

Bu Java kodu, "KAYDET" adlı bir JButton'e bir ActionListener ekleyerek, kullanıcının girdiği ürün bilgilerini alıp bir SQL sorgusu oluşturarak veritabanına eklemek için bir buton işlevselliği sağlar.

```
// (Ürün bilgilerini al, SQL sorgusu oluştur, güncelleme işlemi yap)
JButton guncelle_butonu = new JButton("GÜNCELLE");
guncelle_butonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Kullanıcının girdiği ürün bilgilerini al
        String no,adi,adet,beden,kategori,renk,fiyat,sql_sorgu;
        no = txt_no.getText();
        adi = txt_adi.getText();
        adet = txt_adet.getText();
        kategori = txt_kategori.getText();
        beden = txt_beden.getText();
        renk = txt_renk.getText();
        fiyat = txt_fiyat.getText();
        // Güncelleme için SQL sorgusunu oluştur
        sql_sorgu="UPDATE urunler SET urun_no="+no+", "+
        "urun_adi="+adi+", urun_adet="+adet+", urun_kategori="+kategori+", "
        + "urun_beden="+beden+", urun_renk="+renk+", urun_fiyat="+fiyat+" WHERE urun_no="+no;
        //System.out.println(sql_sorgu);
        baglanti.update(sql_sorgu);// Veritabanında güncelleme yap
    }
});
```

Şekil 31: kod içerikleri

Bu Java kodu, "GÜNCELLE" adlı bir JButton'e bir ActionListener ekleyerek, kullanıcının girdiği ürün bilgilerini alıp bir SQL sorgusu oluşturarak veritabanında güncelleme işlemi yapmak için bir buton işlevselliği sağlar.

```
// (Ürün numarasını al, SQL sorgusu oluştur, silme işlemi yap)
JButton sil_buttonu = new JButton("Sil");
sil_buttonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String no,sql_sorgu; // Kullanıcının girdiği ürün numarasını al
        no= txt_no.getText();
        // Silme için SQL sorgusunu oluştur
        sql_sorgu="DELETE FROM urunler WHERE urun_no="+no;
        baglanti.sil(sql_sorgu); // Veritabanından ürünü sil
    }
});
```

Şekil 32: kod içerikleri

Bu Java kodu, "SİL" adlı bir JButton'e bir ActionListener ekleyerek, kullanıcının girdiği ürün numarasını alıp bir SQL sorgusu oluşturarak veritabanında silme işlemi yapmak için bir buton işlevselliği sağlar.

BAĞLANTI KODLARI

```
listele_butonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        modelim.setRowCount(0);
        ResultSet myRs = baglanti.yap(); // Veritabanından tüm ürünleri çek
        modelim.addRow(satirlar); // Tabloyu güncelle
        try {
            while(myRs.next()) {
                // Veritabanından çekilen bilgilerle tabloyu güncelle
                satirlar[0]=myRs.getString("urun_no");
                satirlar[1]=myRs.getString("urun_adi");
                satirlar[2]=myRs.getString("urun_adet");
                satirlar[3]=myRs.getString("urun_kategori");
                satirlar[4]=myRs.getString("urun_beden");
                satirlar[5]=myRs.getString("urun_renk");
                satirlar[6]=myRs.getString("urun_fiyat");
                modelim.addRow(satirlar);
            }
        } catch (SQLException e1) {
            e1.printStackTrace();
        }
        stoktakiler_table.setModel(modelim);
    }
});
```

Şekil 33: kod içerikleri

Bu Java kodu, "listele_butonu" adlı bir JButton'e bir ActionListener ekleyerek, veritabanındaki tüm ürünleri çekip bir JTable'da görüntülemek için bir buton işlevselliği sağlar. Veritabanından çekilen her ürün bilgisini, tabloyu temsil eden DefaultTableModel (modelim) üzerinden alınan satırlara ekleyerek tabloyu günceller ve ardından bu güncellenmiş modeli JTable'a uygular.


```

sorgula_buttonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        modelim.setRowCount(0);
        String sql_sorgu=null;
        String alan=txt_adsorgu.getText();
        ResultSet myRs = null;
        int secilen = comboBox.getSelectedIndex();
        // Kullanıcının girdiği değere göre veritabanında sorgulama yap
        if(secilen == 0) {
            sql_sorgu= "select * from urunler where urun_no="+ Integer.parseInt(alan);
        }
        else if (secilen == 1) {
            sql_sorgu= "select * from urunler where urun_adi like '"+alan+"%";
        }
    }
}

```

Şekil 34: kod içerikleri

Bu Java kodu, "SORGULA" adlı bir JButton'e bir ActionListener ekleyerek, kullanıcının belirli kriterlere göre veritabanında sorgulama yapmasını sağlar. Kullanıcı, bir JComboBox (comboBox) ile sorgulanacak alanı seçer ve ardından bir JTextField (txt_adsorgu) aracılığıyla sorgu değerini girer. Seçilen alana ve girilen değere göre uygun SQL sorgusu oluşturulur ve bu sorgu sonucunda elde edilen veriler, tabloyu temsil eden DefaultTableModel (modelim) üzerinden alınan satırlara eklenerek JTable'da görüntülenir.

```

cikis_buttonu.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0); // Sistemden atar
    }
}

```

Şekil 35: kod içerikleri

Bu Java kodu, "cikis_buttonu" adlı bir JButton'e bir ActionListener ekleyerek, kullanıcının uygulamadan çıkmasını sağlayan bir işlevselliği temsil eder. "System.exit(0);" ifadesi, uygulamanın sistemden çıkmasını ifade eder.

```

public class baglanti {
    static Connection myConn; // Veritabanı bağlantısı için kullanılan nesne
    static Statement myStat; // SQL sorgularını yürütmek için kullanılan nesne
}

```

Şekil 36: kod içerikleri

Bu Java sınıfı, veritabanı ile bağlantı kurmak ve SQL sorgularını yürütmek için kullanılan bağlantı ve ifade nesnelerini içerir.

```

// Veritabanından tüm ürünleri çekmek için kullanılan metod
static ResultSet yap() {
    ResultSet myRs = null;
    try {
        myConn = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/stok_takip","root", "admin");
        myStat = (Statement) myConn.createStatement();
        myRs = myStat.executeQuery("select * from urunler");
    } catch (Exception e) {
        e.printStackTrace();
    }
    return myRs;
}

```

Şekil 37: kod içerikleri

Bu Java metodu, MySQL veritabanına bağlanarak "stok_takip" adlı veritabanındaki "urunler" tablosundan tüm ürünleri çekmek için kullanılır. Bağlantı bilgileri (URL, kullanıcı adı, şifre) hardcoded olarak belirlenmiştir. Metot, bir ResultSet nesnesi döndürerek çekilen ürün verilerini temsil eder.

```
// Veritabanına yeni veri eklemek için kullanılan metot
static void ekle(String sql_sorgu) {
    try {
        myStat.executeUpdate(sql_sorgu);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Şekil 38: kod içerikleri

Bu Java metodu, parametre olarak aldığı SQL sorgusunu kullanarak veritabanına yeni veri eklemek için kullanılır. myStat.executeUpdate(sql_sorgu) ifadesi, veritabanında güncelleme (ekleme, silme veya güncelleme) işlemlerini yürüten bir metodu çağırır. Metot, herhangi bir hata durumunda hata bilgilerini yazdırır.

```
//Mevcut bir veriyi güncellemek için kullanılan metot
static void update(String sql_sorgu) {
    try {
        myStat.executeUpdate(sql_sorgu);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

Şekil 39: kod içerikleri

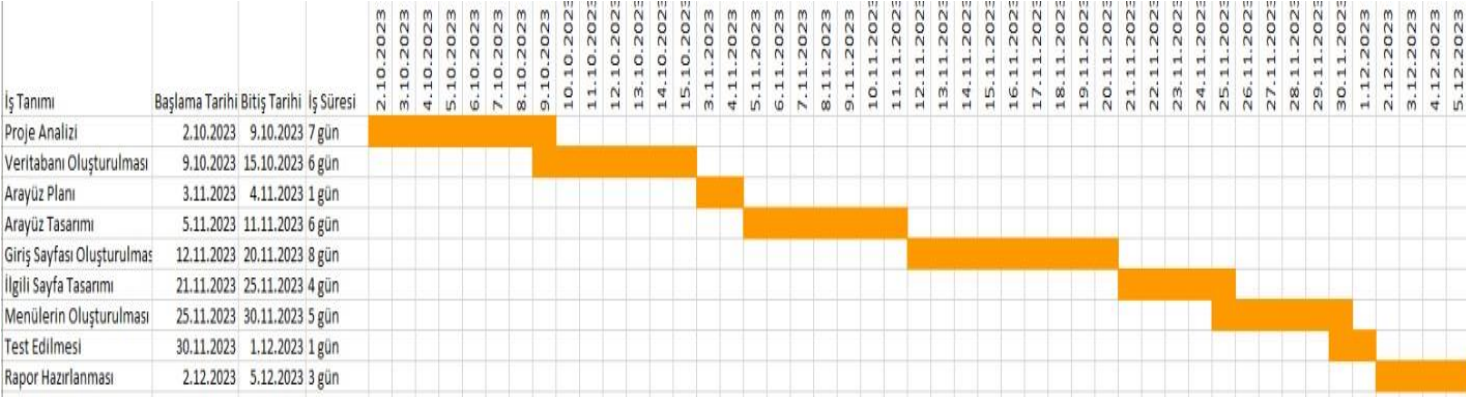
Bu Java metodu, mevcut bir veriyi güncellemek için kullanılır. Parametre olarak aldığı SQL sorgusunu kullanarak myStat.executeUpdate(sql_sorgu) ifadesini çağırır. Bu ifade, veritabanında güncelleme işlemini yürütür. Eğer bir SQL istisnası (SQLException) oluşursa, hata bilgilerini yazdırır.

```
//Belirli bir sorguya göre veri çekmek için kullanılan metot
static ResultSet sorgula(String sql_sorgu) {
    ResultSet myRs = null;
    try {
        myRs = myStat.executeQuery(sql_sorgu);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return myRs;
}
```

Şekil 40: kod içerikleri

Bu Java metodu, veritabanında bir sorgu çalıştırmak için kullanılır. Parametre olarak aldığı SQL sorgusunu kullanarak myStat.executeQuery(sql_sorgu) ifadesini çağırır ve sonuçları temsil eden bir ResultSet nesnesini döndürür. Eğer bir SQL istisnası (SQLException) oluşursa, hata bilgilerini yazdırır. Bu metot, genellikle SELECT sorgularını çalıştırmak ve sonuçları almak için kullanılır.

PROJE ZAMAN ÇİZELGESİ



Şekil 41: proje zaman çizelgesi

PROJE EKİBİ VE YÖNETİMİ

Ekibimiz

Projemiz, beş kişilik bir ekipten oluşmaktadır. Her bir ekip üyesinin belirli bir sorumluluğu bulunmaktadır fakat zorlandığımız yapamadığımız noktalarda herkes birbirine yardım etmiştir.

Proje Yöneticisi: [Şimal BÜLBÜL]

Ana Programcı: [Nisa Usta]

Veritabanı Uzmanı: [Nisa Usta]

Arayüz Tasarımcısı: [Ceydanur ÖZGÜR]

Test ve Kalite Kontrol: [Şimal BÜLBÜL]

PROJE SONUÇLARI

Başarılar

- Temel stok takip işlevselliği başarıyla gerçekleştirildi.
- Kullanıcı dostu arayüz tasarımı oluşturuldu.
- Veritabanı bağlantısı ve güvenliği sağlandı.

Zorluklar

- Proje başlangıcında gereksinimlerin netleştirilmesinde yaşanan belirsizlik.
- Bazı teknik zorluklar nedeniyle gün sürelerinin aşılması.

Gelecek İyileştirmeler

- Raporlama, satış ve analiz özelliklerinin eklenmesi.
- Mobil platformlara uygun bir sürümün geliştirilmesi.

Proje Değerlendirmesi

Proje, belirlenen hedeflere ulaşarak başarıyla tamamlandı. Ancak, gelecekteki versiyonlarda ek özelliklerin ve iyileştirmelerin yapılması planlanmaktadır.

KAYNAKÇA

How do you add an Action Listener on to a JButton in Java. (2008). Erişim Tarihi: 27 Kasım 2023,

<https://stackoverflow.com/questions/284899/how-do-you-add-an-actionlistener-onto-a-jbutton-in-java>

Ayşe Nur Özdemir. (2021). Java Action Listener Sınıfı. Erişim Tarihi: 29 Kasım 2023,

<https://aysenurozdemirr6.medium.com/java-actionlistener-sinifi-6c5b35829758#:~:text=ActionListener%20sınıfı%20java.awt.event%20paketinin%20içinde%20bulunan%20bir%20interfacedir,sınıfı%20içerisinde%20actionPerformed%20metodunu%20bulundurur.>

JTable ve Model Kullanımı. (2013). Erişim Tarihi: 24 Kasım 2023,

<https://yazilimokulu.wordpress.com/2013/08/03/jtable-ve-table-model-kullanimi/>

A. Gökhan Satman. (2013). Java ile Mysql Veritabanı Bağlantısı. Erişim Tarihi: 2 Aralık 2023,

<http://www.phpservisi.com/java-ile-mysql-veritabani-baglantisi-ve-sql-sorgulari/>

Yusuf Sezer. (2021). Sql Tablo Oluşturma. Erişim Tarihi: 3 Aralık 2023,

<https://www.yusufsezer.com.tr/sql-tablo-olusturma/>

Java Veritabanı İşlemleri. (2014). Erişim Tarihi: 23 Kasım 2023,

<https://www.kodlamamerkezi.com/java/java-ile-veritabani-islemleri/>

Onur Gürbüz. (2020). Stok Takip Otomasyonu. Erişim Tarihi: 20 Kasım 2023,

<https://github.com/onurgurbuz/Stok-Takip-Otomasyonu>