

## İçindekiler

Genel Bilgiler .....	1
Sınıflar .....	2
Class Main .....	2
Class AnaEkran .....	3
Class GirişEkranı .....	4
Class KullanıcıArayüz .....	7
Class GörevlilerArayüz .....	12
Class MüdürArayüz .....	17
Class ÜyeOl .....	28
Class BelgeDoğrula .....	29
Class Evrak .....	30
Class Çalışan .....	32
Class Görevli .....	33
Class Müdür .....	35
Class Evrakİşlem .....	35
Class Kullanıcı .....	35
Class SQLBağlantısı .....	37
Class İlkKurulum .....	42

## Genel Bilgiler

Programın geliştirilme süresince programın mümkün olduğu kadar esnek bir şekilde kodlanmasına dikkat edilmiştir. Veri tabanına doğrudan müdahale edilip veriler bozulmadığı sürece programın hata verip çökme ihtimali çok düşüktür. Program, kullanıcının suçsuzluğu kanıtlanana kadar kullanıcı suçludur anlayışı ile kodlanmıştır. Dolayısı ile kullanıcı tarafından yapılacak tüm girişler ve işlemler kontrol edilip, program üzerinde oluşacak olan tüm açıklar engellenmeye çalışılmıştır. Mümkün olduğu kadar try-catch yapısı kullanılmıştır. Proje dokümanında istenilen tüm özellikler fazlası ile yerine getirilmiş olup tüm kontrolleri sağlanmıştır ve ek opsiyonlar eklenmiştir. Optimize algoritmalar oluşturmaya dikkat edilmiştir.

**Programın çalışabilmesi için MySQL veri tabanı kullanılmalıdır.** WampServer veya XAMPP programları bu ihtiyacı karşılamaktadır.

WampServer indirme linki: <https://www.wampserver.com/en/download-wampserver-64bits/>

XAMPP indirme linki: <https://www.apachefriends.org/download.html>

Program içinde kullanılan veri tabanı bilgileri

- Kullanıcı = root
- Şifre =
- Port = 3306

İstenirse RAR dosyasının içindeki SQL yedeği doğrudan import edilebilir veya ilk kurulum özelliği gerçekleştirilebilir. Değerlendirme süresinin daha kısa ve daha verimli olabilmesi için SQL yedeğinin MYSQL'e import edilmesi önerilir. Bu işlemler yapılmaz ise sistemde herhangi bir Müdür kaydı olmayacağı için görevli eklenemeyecektir.

## Sınıflar

### Class Main

#### ArrayListler

“Main” Class’ında program içindeki verilerin kullanılacağı alan olan ArrayList’ler tanımlanmıştır. Bu ArrayList’ler, veri tabanı ve program arasında ara bir katman olarak görev almaktadır. Program başlatıldığında veri tabanının içindeki değerler ArrayList’lere aktarılmaktadır ve herhangi bir işlem yapıldığında ArrayList’lerin içindeki değerler, veri tabanına aktarılmaktadır. Program içinde herhangi bir veriye ihtiyaç duyulduğunda ise bu ArrayList’lere başvurulmaktadır.

```
static ArrayList<evrak> evraklar = new ArrayList<>();
static ArrayList<gorevli> gorevliler = new ArrayList<>();
static ArrayList<kullanici> kullanicilar = new ArrayList<>();
static ArrayList<mudur> mudurler = new ArrayList<>();
```

#### İlk Kurulum

Eğer Program dizini altında ilkKurulum.txt dosyası mevcut ise program SQL veri tabanı kurulumu kodlarını çalıştırarak gerekli veri tabanını ve tabloları oluşturur. Daha sonrasında müdür kayıt ekranını açarak, ilk müdürün kaydının yapılmasını sağlar. Birden fazla müdür kaydı yapabilmek için istendiği kadar ilk kurulum yapılabilir. İlk kurulum sonunda ilkKurulum.txt dosyası otomatik olarak silinmektedir. İlk kurulum özelliği yerine RAR dosyasının içindeki SQL backup dosyası, MySQL’e import edilebilir.

```
URL dizin = null;
File f = null;
try{
    dizin = AnaEkran.class.getResource( name: "/ilkKurulum.txt");
    assert dizin != null;
    f = new File(dizin.toURI());
}catch (Exception ee){

}

if (f!=null&&f.exists()) {
    sqlBaglantisi.sifirKurulum();
    ilkKurulum kurulum = new ilkKurulum();
    kurulum.setVisible(true);
}
```

Eğer herhangi bir ilk kurulum yapılmayacak ise SQL veri tabanına bağlanılarak, ArrayList’lerin içleri veri tabanındaki değerler ile doldurulur.

```
//Kullanıcı Bilgilerini Veri tabanından çekerek ArrayList'e Kaydeder
sqlBaglantisi.kullaniciCek();

//Görevli Bilgilerini Veri tabanından çekerek ArrayList'e Kaydeder
sqlBaglantisi.gorevliCek();

//Müdür Bilgilerini Veri tabanından çekerek ArrayList'e Kaydeder
sqlBaglantisi.mudurCek();

//Evrak Bilgilerini Veri tabanından çekerek ArrayList'e Kaydeder
sqlBaglantisi.evrakCek();
```

## Programın Başlatılması

Veri çekilme işlemi de tamamlandığında programın başlatılması sağlanır.

```
//Eğer ilk kurulum durumu yoksa programı normal olarak çalıştır
if (f==null) {
    AnaEkran.main(args);
}
```

## Class AnaEkran

### Tanımlanma Kısımı

AnaEkran sınıfı ana ekran tasarımı için Swing kodlarını içeren sınıftır. import kısmında Swing kütüphaneleri eklenmiştir. JFrame sınıfından miras almaktadır. “Main” fonksiyonun başlaması ile ana ekranın frame’i görünür olmaktadır ve uygulama ikonu resmi ayarlanmaktadır.

```
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                AnaEkran frame = new AnaEkran();
                frame.setVisible(true);
                frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // Çarpı tuşuna basıldığında kapanması için
                URL resim = AnaEkran.class.getResource("/resimler/ikon.png");
                Image icon = Toolkit.getDefaultToolkit().getImage(resim);
                frame.setIconImage(icon);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

Daha sonrasında yapıcı sınıfın içindeki kodlar çalışmaya başlamaktadır. Frame boyutu 800x600 oranlarına ayarlanarak, büyütme küçültme özelliği kapatılmaktadır.

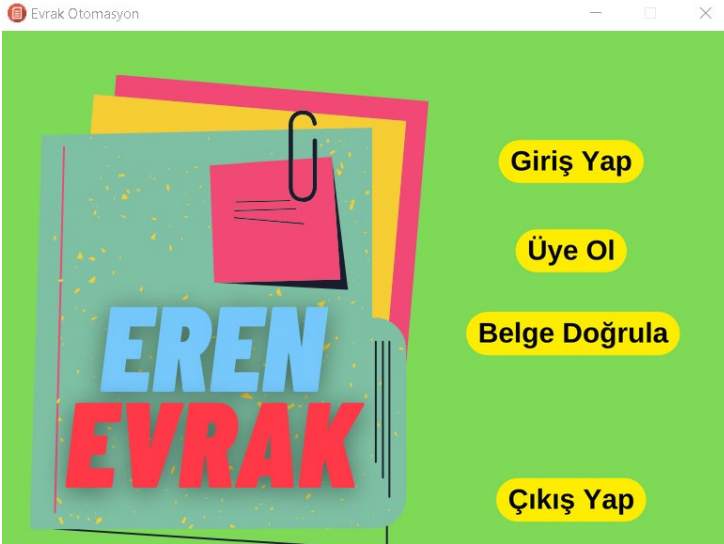
```
setResizable(false);
setPreferredSize(new Dimension(800, 600));
setMinimumSize(new Dimension(800, 600));
setMaximumSize(new Dimension(800, 600));
setTitle("Evrak Otomasyon");
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setSize(800,600);
setBounds(100, 100, 800, 600);
```

Daha sonrasında JPanel eklenerek ana ekranda yer alan butonların ve yazıların tanımlamaları ve gerekli ayarlamaları yapılmaktadır. Butonların resimleri Canva platformu üzerinden tasarlanmıştır. Tanımlanan butonlar JPanel’in içine eklenmektedir.

```
cikisYap.setBorderPainted(false);
cikisYap.setIcon(new ImageIcon(AnaEkran.class.getResource("/resimler/cikisYap.png")));
cikisYap.setBounds(509, 474, 200, 50);
cikisYap.setToolTipText("Çıkış Yap");
```



Ana Ekranda Giriş Yap, Üye Ol, Belge Doğrula ve Çıkış Yap olmak üzere 4 tane buton vardır.



Tüm butonların MouseEvent'leri ayarlanarak tüm butonlara işlev kazandırılmıştır.

```
girisYap.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        dispose();  
        GirisEkrani frm2 = new GirisEkrani();  
        frm2.setVisible(true);  
        frm2.setIconImage(icon);  
    }  
});
```

```
uyeOl.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        UyeOl uyelik = new UyeOl();  
        dispose();  
        uyelik.setVisible(true);  
        uyelik.setIconImage(icon);  
    }  
});
```

```
cikisYap.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        dispose();  
        return;  
    }  
});
```

Arka plan resmi için Frame'in boyutunda bir JLabel tanımlanmıştır ve resim atanmıştır.

```
JLabel arkaPlan = new JLabel("");  
arkaPlan.setIcon(new ImageIcon(AnaEkran.class.getResource("/resimler/Eren Please.png")));  
arkaPlan.setBounds(0, 0, 800, 600);  
contentPane.add(arkaPlan);
```

## Class GirisEkrani

### Tanımlanma Kısım

Ana menüden Giriş Yap butonuna basılırsa GirisEkrani sınıfından bir nesne oluşturulmaktadır ve oluşan bu nesneden frame görünürlüğü true olarak ayarlanmaktadır ve ikon resmi atanmaktadır. Daha sonrasında Ana ekranın frame'i kapatılmaktadır.

```
public void mouseClicked(MouseEvent e) {  
    dispose();  
    GirisEkrani frm2 = new GirisEkrani();  
    frm2.setVisible(true);  
    frm2.setIconImage(icon);  
}
```

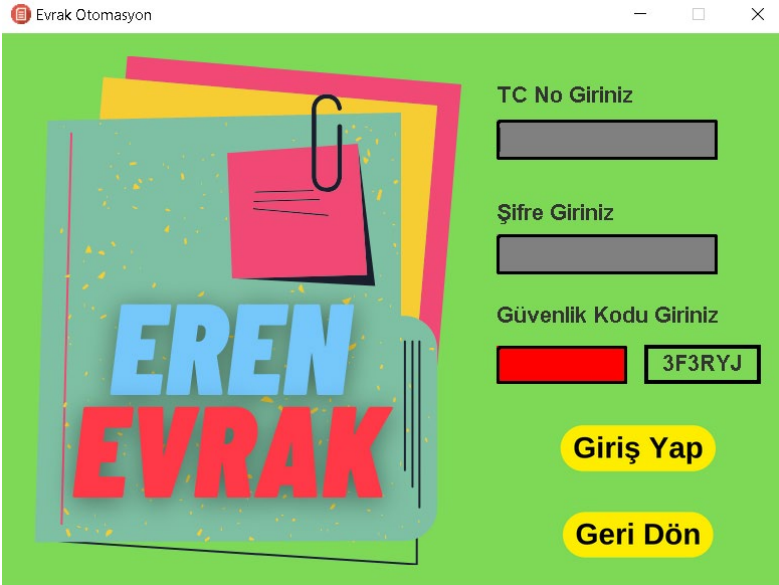
GirisEkranı sınıfından frm2 nesnesinin tanımlanması ile yapıcı sınıfın komutları çalışacaktır. Bu komutlar doğrultusunda frame çözünürlüğü 800x600 boyutunda olacaktır. Yeniden boyutlandırma özelliği kapatılmıştır. Diğer tüm arayüz sınıfları gibi bu sınıf da Swing kütüphanelerini import edilmektedir ve JFrame sınıfından miras almaktadır.

```
public GirisEkranı() {  
    setResizable(false);  
    setPreferredSize(new Dimension(800, 600));  
    setMinimumSize(new Dimension(800, 600));  
    setMaximumSize(new Dimension(800, 600));  
    setTitle("Evrak Otomasyon");  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    setSize(800,600);  
    setBounds(100, 100, 800, 600);  
}
```

Arayüz ekranında 3 adet JLabel, 3 adet JTextField, 1 adet JPasswordField ve 2 adet JButton nesnesi tanımlaması yapılmıştır.

```
JLabel label_TC = new JLabel( text: "TC No Giriniz");  
label_TC.setFont(new Font( name: "Arial", Font.BOLD, size: 22));  
label_TC.setBounds( x: 492, y: 42, width: 158, height: 40);  
contentPane.add(label_TC);  
  
JLabel label_Sifre = new JLabel( text: "Şifre Giriniz");  
label_Sifre.setFont(new Font( name: "Arial", Font.BOLD, size: 22));  
label_Sifre.setBounds( x: 492, y: 159, width: 127, height: 40);  
contentPane.add(label_Sifre);  
  
JLabel arkaplan = new JLabel( text: "");  
arkaplan.setIcon(new ImageIcon(AnaEkran.class.getResource( name: "/resimler/Eren Please.png")));  
arkaplan.setBounds( x: 0, y: -24, width: 800, height: 600);  
contentPane.add(arkaplan);
```

Giriş ekranında kullanıcıdan T.C. No ve Şifre girişi yapması beklenmektedir.



Ayrıca her seferinde rastgele üretilen güvenlik kodu girişinin yapılması da beklenmektedir. Bu kontrol ile kullanıcının insan olup olmadığı tespit edilebilmektedir. Güvenlik kodunun üretiminde aşağıdaki algoritma kullanılmıştır. Giriş ekranı her açıldığında güvenlik kodu değişmektedir.

```

Random rand = new Random();
String kod = "";
String tmp = String.valueOf(rand.nextInt( bound: 999999));
final String Kodd;
for (int i=0;i<tmp.length();i++ ) {
    if (tmp.charAt(i)=='2')
        kod+='F';
    else if (tmp.charAt(i)=='4')
        kod+='Z';
    else if (tmp.charAt(i)=='6')
        kod+='R';
    else if (tmp.charAt(i)=='8')
        kod+='Y';
    else if (tmp.charAt(i)=='0')
        kod+='J';
    else
        kod+=tmp.charAt(i);
}

```

## Giriş Yap

Kullanıcının yaptığı girişler ise Giriş Yap butonunun MouseEvent’inde kontrol edilip gerekli işlemler yapılmaktadır. Bu kontroller;

- T.C. No girişi boş mu?
- T.C. No girişi harf içeriyor mu?
- Şifre Girişi Boş mu?
- T.C. No girişi formatı uygun uzunlukta mı?
- Güvenlik kodu girişi boş mu
- Güvenlik kodu girişi, üretilen güvenlik kodu ile eşleşiyor mu

```

////////////////////////////////////
long tc;
if (input_TC.getText().equals("")) {
    JOptionPane.showMessageDialog( parentComponent: null, message: "TC Kimlik Numarasını Girmediniz !");
    return;
}
else if (input_Sifre.getPassword().length==0) {
    JOptionPane.showMessageDialog( parentComponent: null, message: "Şifre Girmediniz !");
    return;
}
try {
    tc = Long.parseLong(input_TC.getText());
} catch (Exception ee) {
    JOptionPane.showMessageDialog( parentComponent: null, message: "TC Kimlik Numarasını Hatalı Girdiniz !");
    input_TC.setText("");
    return;
}
if (tc <= 9999999999L || tc > 99999999999L) {
    JOptionPane.showMessageDialog( parentComponent: null, message: "TC Kimlik Numarasını Hatalı Girdiniz !");
    input_TC.setText("");
    return;
}
String sifre = String.valueOf(input_Sifre.getPassword());

```

Bu kontroller sağlandıktan sonra kullanıcının girmiş olduğu T.C. No hangi hesap türü (Kullanıcı, Görevli, Müdür) ile eşleşiyor diye bakılmaktadır. Eğer ki kullanıcının girmiş olduğu T.C. No ve şifre, aranan ArrayList’de bulunur ise giriş işlemleri gerçekleştirilmektedir. Eğer şifre yanlış ise, şifrenin yanlış olduğunda dair pop-up mesajı



verilmektedir. Eğer girilen T.C. No herhangi bir ArrayList’de bulunamaz ise böyle bir hesap olmadığı bilgisi pop-up ile gösterilir.

```
//KULLANICI HESAP KONTROL
////////////////////////////////////
for (int i = 0; i < Main.kullaniciilar.size(); i++) {
    if (Objects.equals(sifre, Main.kullaniciilar.get(i).getSifre()) && Main.kullaniciilar.get(i).getTcNo() == tc) {
        JOptionPane.showMessageDialog( parentComponent: null, message: "Kullanıcı Bulundu. Giriş Yapılıyor !");
        int index = i;
        kullaniciArayuz kullanici = new kullaniciArayuz(index);
        kullanici.setVisible(true);
        kullanici.setIconImage(icon);
        dispose();
        return;
    } else if (Main.kullaniciilar.get(i).getTcNo() == tc && !Objects.equals(sifre, Main.kullaniciilar.get(i).getSifre()))
        JOptionPane.showMessageDialog( parentComponent: null, message: "Şifre Yanlış Girildi !");
        input_Sifre.setText("");
        return;
}
```

Aranılan T.C. No ve şifre uyduğunda ise o hesap için oluşturulan arayüze yönlendirme işlemleri başlatılır. Bu işlemler, o hesap türü için oluşturulan arayüz sınıfından bir nesne oluşturulması. Oluşturulan bu nesne ile arayüzün görünürlüğünün true olarak ayarlanması. Başlatılacak olan frame’in ikonunun ayarlanması ve mevcut frame’in kapatılmasıdır.

```
JOptionPane.showMessageDialog( parentComponent: null, message: "Görevli Bulundu. Giriş Yapılıyor")
int index = i;
gorevliArayuz gorevli = new gorevliArayuz(index);
gorevli.setVisible(true);
gorevli.setIconImage(icon);
dispose();
return;
```

## Class KullaniciArayuz

### Tanımlanma Kısım

KullaniciArayuz sınıfı, hesabına giriş yapmış olan kullanıcılar için Swing kodlarını içeren kullanıcı hesabı arayüzü tasarımı yapılmış bir sınıftır. Giriş ekranından başarılı bir giriş gerçekleştirilse yapıcı sınıfı çalışmaktadır. Bu yapıcı sınıfın içinde Frame boyutunun 800x600 olarak ayanması, yeniden boyutlandırmanın kapatılması, JPanel’ in tanımlanması, ekranda gözükecek olan JFrame’lerin JButtonların vs. tanımlanması yapılmıştır.



## Evrak Oluştur

Evrak oluştur bölümünde, oluşturulabilecek evrakların bulunduğu bir JComboBox nesnesi vardır. Combo Box, String türünden veriler tutmaktadır. İçindeki itemler gömülü olarak tanımlanmıştır.

```
JComboBox<String> comboBox_Evraklar = new JComboBox<String>();
comboBox_Evraklar.setModel(new DefaultComboBoxModel<String>(new String[]
    {"İkametgah Evrakı", "Ceza Evrakı",
     "Nüfus Evrakı", "Tapu Evrakı", "Taşıt Evrakı", "Gelir Evrakı"}));
comboBox_Evraklar.setFont(new Font("Arial", Font.PLAIN, 18));
comboBox_Evraklar.setBackground(Color.YELLOW);
comboBox_Evraklar.setBounds(55, 63, 193, 43);
panel.add(comboBox_Evraklar);
```

Evrak oluştur butonunun tanımlanmasını ise aşağıda görebilirsiniz.

```
JButton buton_evrakOlustur = new JButton("");
buton_evrakOlustur.setBounds(23, 137, 250, 50);
buton_evrakOlustur.setBorderPainted(false);
buton_evrakOlustur.setIcon(new ImageIcon(
    kullaniciArayuz.class.getResource("/resimler/Evakk.png")));
buton_evrakOlustur.setToolTipText("Evrak Oluştur");
panel.add(buton_evrakOlustur);
```

Kullanıcı Combo Box içerisinden istediği evrak türünü seçtikten sonra evrak oluştur butonuna basacaktır ve evrak oluştur butonunun MouseEvent'i devreye girecektir. Bu MouseEvent içerisinde, kullanıcının seçmiş olduğu index numarasına göre evrak türünün değeri, "Evrak" sınıfının yapıcı metodu için uygun hale getirilmektedir. Daha sonrasında "Evrak" sınıfından bir nesne üretilerek "Main" sınıfının içindeki evraklar ArrayList'inin içine eklenmektedir. Bu işlemler başarılı bir şekilde yapıldığında, MySQL ile bağlantı kurularak ArrayList'in içindeki bilgiler veri tabanına aktarılmaktadır. Evrakın oluşturulduğuna dair bir pop verilmektedir. Eğer kullanıcı evrak türü seçmeden buton oluşturmaya çalışır ise hata oluşacağından dolayı catch bloğu içerisinde ekrana pop-up mesajı ile uyarı gösterilmektedir.



```

buton_evrakOlustur.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        try {
            String secim="";
            int n= comboBox_Evraklar.getSelectedIndex();
            if (n==0)
                secim = "İkametgah";
            else if (n==1)
                secim = "Ceza";
            else if (n==2)
                secim = "Nüfus";
            else if (n==3)
                secim = "Nüfus";
            else if (n==4) |
                secim = "Taşıt";
            else if (n==5)
                secim = "Gelir";
            Main.evraklar.add(new evrak(secim,Main.kullanıcılar.get(index)));
            sqlBaglantisi.evrakKayit();
            JOptionPane.showMessageDialog(null, "Evrak Oluşturuldu");
        } catch (Exception ee) {
            JOptionPane.showMessageDialog(null, "Evrak Türü Seçmediniz !");
        }
    }
});

```

## Evrakların Hepsini Görüntüle

Kullanıcı istediği kadar evrak oluşturabilmektedir. Oluşturmuş olduğu evrakların hepsini görüntüleyebilmek için tabbed pane sekmeleri içinde yer alan oluşturulan evrakları görüntüle butonuna basarak JTable'ın tanımlanmış olduğu JPanel'e ulaşabilmektedir. Kullanıcının karşısına gelecek olan bu ekran içerisinde yer alan JTable de tüm evrakların görüntülenebilmesi için kullanıcının Tüm Evrakları Görüntüle butonuna basması gerekmektedir. JTable içerisinde, Seri No, Doğrulama Kodu, Tür, Tarih, İmzalayan, Oluşturan, Geçerlilik kolonları yer almaktadır.

Hoşgeldiniz Eren Doğan (Kullanıcı)						
Hesap Ayarları	Oluşturulan Evrakları Görüntüle					
Evrak Oluştur			Belirli Evrakı Görüntüle			
Seri No	Doğrulama Kod	Tür	Tarih	İmzalayan	Oluşturan	Geçerlilik
0	4626	Nüfus	2022-12-25	Mehmet Akif	Eren Doğan	Geçerli
1	9258	Taşıt	2022-12-25	Yok	Eren Doğan	Geçerli
2	5337	Gelir	2022-12-25	Hakan Tunç	Eren Doğan	Geçerli Değil
3	7436	İkametgah	2022-12-25	Mehmet Akif	Eren Doğan	Geçerli Değil
4	6139	Ceza	2022-12-25	Hakan Tunç	Eren Doğan	Geçerli
5	1328	Gelir	2022-12-25	Mehmet Akif	Eren Doğan	Geçerli
9	6614	Ceza	2022-12-26	Yok	Eren Doğan	Geçerli
10	2654	Nüfus	2022-12-29	Yok	Eren Doğan	Geçerli
Oluşturulan Evrakları Görüntüle						

Kullanıcı, "Oluşturulan Evrakları Görüntüle" JButton'una bastığında JButton'un Mouse Event kodları çalışmaya başlamaktadır. Bu kodlar, JTable'ın içerisindeki tüm değerlerin temizlenmesi Bu kodlar JTable içerisinde yer alan tüm satırların silinmesi daha sonrasında imzalanmamış evrakların for döngüsü ile kontrol edilerek kolon bilgilerinin bir String dizi içinde oluşturulup yeni bir satır olarak eklenmesi şeklindedir. Yeni satır eklenmesi işlemi addRow komutu ile yapılmaktadır.

```

public void mouseClicked(MouseEvent e) {
    tblModel.getDataVector().removeAllElements();
    String gecerli;
    //revalidate();
    for (evrak evrak : Main.evraklar) {
        if (evrak.getOlusturan().getTcNo() == Main.kullanici.get(index).getTcNo()) {
            if (evrak.isGecerlilik()) {
                gecerli = "Geçerli";
            }
            else {
                gecerli = "Geçerli Değil";
            }
            String[] tmp = {String.valueOf(evrak.getSeriNo()),
                String.valueOf(evrak.getDogrulamaKod()),
                String.valueOf(evrak.getTur()),
                evrak.getTarih(),
                evrak.ImzalayanKim(),
                evrak.getOlusturan().getAd() + " " + evrak.getOlusturan().getSoyAd(),
                gecerli};
            tblModel.addRow(tmp);
        }
    }
}

```

Kullanıcı tüm evrakları görüntüledikten sonra yalnızca herhangi bir evrakı görüntülemek isterse tabbed pane içerisinde yer alan “Belirli Evrakı Görüntüle” butonuna basarak görüntülemek istediği evrağın seri numarasını girerek “Evrakı Görüntüle” butonuna bastığında görüntülemek isteyeceği evrak bilgileri bir JLabel içerisinde ekranda görüntülenecektir.

Kullanıcı “Evrakı Görüntüle” isimli JButton’a bastığında, JButton’un Mouse Event kodları devreye girmektedir. Bu kodlar, JTextField’a yapılan Seri Numarasının integer tipinde bir değişkene kaydedilmesi ve bu işlemi gerçekleştiren kullanıcının böyle bir evrak oluşturup oluşturmadığı bir döngüsünde kontrol edilerek bir sonuç bulunması halinde evrağın override edilen toString methodu çalıştırılarak çıktısı JLabel içerisine aktarılmaktadır. Eğer kullanıcı hatalı bir giriş yaparsa catch bloğu içerisinde yakalanarak ekrana pop-up mesajı verilmektedir. Aynı şekilde kullanıcının girmiş olduğu seri no ile eşleşen bir evrak bulunamaz ise evrağın bulunmadığına dair pop-up ile uyarı mesajı verilmektedir.

```

public void mouseClicked(MouseEvent e) {
    try {
        int seriNo = Integer.parseInt(input_seriNo.getText());
        for (int i=0; i<Main.evraklar.size(); i++) {
            if (Main.evraklar.get(i).getSeriNo()==seriNo
                && Main.kullanici.get(index).getTcNo()==Main.evraklar.get(i).getOlusturan().getTcNo()) {
                label_belirliEvrak.setText(Main.evraklar.get(i).toString());
                break;
            }
            if (i==Main.evraklar.size()-1) {
                JOptionPane.showMessageDialog(null, "Seri Numara İle Eşleşen Herhangi Bir Evrakınız Bulunamadı!");
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Hatalı Giriş Yaptınız !");
    }
}

```

Kullanıcı; kişisel bilgilerini görüntülemek isterse, hesap şifresini değiştirmek isterse veya hesabından çıkış yapmak isterse tabbed pane içerisinde yer alan “Hesap Ayarları” butonuna tıklayarak hesap ayarlarının olduğu JPanel’e ulaşabilmektedir. Bu JPanel içerisinde “Kişisel Bilgileri Göster”, “Şifre Değiştir”, “Çıkış Yap” JButton nesneleri yer almaktadır. Şifre değiştirme işlemleri için ise 3 adet JPasswordField nesnesi yer almaktadır.

Kullanıcı “Kişisel Bilgileri Göster” JButton’una bastığında “Kullanici” sınıfındaki over ride edilen toString methodu kullanılarak kullanıcının bilgileri ekranın sol bölümünde yer alan boş JLabel içerisine aktarılmaktadır.

```

buton_kisiselBilgileriGoster.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        output_kullaniciBilgileri.setText(Main.kullanici.get(index).toString());
    }
});
panel_3.add(buton_kisiselBilgileriGoster);

```

Kullanıcı şifresini değiştirmek için eski şifresini ve yeni şifresini girmesi gerekmektedir. Olası bir güvenlik açığının önlemek amacı ile kullanıcıdan eski şifresi istenmiştir ve hatalı giriş yapabilme ihtimaline karşın yeni şifresi 2 defa istenmiştir. Kullanıcı bu girişleri JPasswordField alanlarına girecektir. Kullanıcının yaptığı girişlerde çeşitli kontroller sağlanmıştır. Bu kontroller:

- Kullanıcının gireceği şifre güvenlik nedeni ile en az 6 karakter olmalıdır.
- Şifre maksimum 253 karakter olabilir. Bunun nedeni yapılacak girişin veri tabanında sorun çıkarıp programın çökmesini engellemek içindir. Normal şartlarda herhangi bir

şifrenin 254 karakter olabilmesi pek normal değildir fakat veri tabanında ve dahi programda herhangi bir zafiyetin oluşmasını engellemek için bu kontrol sağlanmıştır.

- Kullanıcının giriş yapacağı alanlar boş mu veya boşluk karakteri içeriyor mu?
- Kullanıcının girdiği 2 yeni şifre birbirleri ile uyuyor mu?
- Kullanıcının girdiği eski şifre, gerçek şifresi ile uyuyor mu?

```
public void mouseClicked(MouseEvent e) {
    String eskisifre =String.valueOf(input_eskiSifre.getPassword());
    String yenisifre1 =String.valueOf(input_yeniSifre.getPassword());
    String yenisifre2 =String.valueOf(input_yenisifre2.getPassword());
    if (eskisifre.isBlank()) {
        JOptionPane.showMessageDialog(null,"Eski Şifre Girişi Yapılmamıştır!");
        return;}
    else if (yenisifre1.length()>253) {
        JOptionPane.showMessageDialog(null,"Yeni Şifre Maksimum Uzunluğu Aşmaktadır !");
        return;}
    else if (yenisifre1.length()<6) {
        JOptionPane.showMessageDialog(null,"Yeni Şifre Minimum 6 Karakter olmalıdır !");
        return;}
    else if(Objects.equals(eskisifre,Main.kullaniciilar.get(index).getSifre())){
        if (yenisifre1.isBlank() || yenisifre2.isBlank()) {
            JOptionPane.showMessageDialog(null,"Yeni Şifre İçin Giriş Yapmadınız !");
            return;}
        else if (yenisifre1.equals(yenisifre2)){
            Main.kullaniciilar.get(index).setSifre(yenisifre2);
            sqlBaglantisi.kullaniciKayit();
            JOptionPane.showMessageDialog(null,"Şifre Başarıyla Değiştirilmiştir !");
            return;}
        else {
            JOptionPane.showMessageDialog(null,"Yeni Şifreler Uyuşmamaktadır. Lütfen Kontrol Ediniz !");
            return;}
    }
    else {
        JOptionPane.showMessageDialog(null,"Eski Şifre Yanlış Girilmiştir. Lütfen Kontrol Ediniz !");
        return;}
}
```

Eğer tüm koşullar sağlanırsa kullanıcının şifresi değiştirilecektir, ekranda pop-up mesajı gösterilecektir ve kullanıcılar ArrayList'in içindeki değerler MySQL'e aktarılacaktır.

```
else if (yenisifre1.equals(yenisifre2)){
    Main.kullaniciilar.get(index).setSifre(yenisifre2);
    JOptionPane.showMessageDialog(null,"Şifre Başarıyla Değiştirilmiştir !");
    sqlBaglantisi.kullaniciKayit();
    return;
}
```

Kullanıcı çıkış yapmak istediğinde ise “Çıkış Yap” JButton’una basacaktır. JButton’un Mouse Event kodları devreye girerek “AnaEkran” sınıfından yeni bir nesne oluşturacaktır. “AnaEkran” sınıfının yapıcı metodundan oluşacak JFrame’in görünürlüğü true olarak ayarlanıp mevcut JFrame kapatılacaktır ve kullanıcı hesaptan çıkış yaparak Ana Ekranı geri dönmüş olacaktır.

```
buton_hesaptanCikisYap.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        dispose();
        AnaEkran yeni = new AnaEkran();
        yeni.setVisible(true);
        return;
    }
});
```

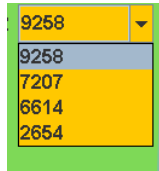
## Class GorevlilerArayuz

“GorevlilerArayuz” sınıfı, hesabına giriş yapmış olan görevliler için Swing kodlarını içeren görevli hesabı arayüzü tasarımı yapılmış bir sınıftır. Giriş ekranından başarılı bir giriş gerçekleştirilse yapıcı sınıfı çalışmaktadır. Bu yapıcı sınıfın içinde Frame boyutunun

800x600 olarak ayanması, yeniden boyutlandırmanın kapatılması, JPanel' in tanımlanması, ekranda gözükecek olan JFrame'lerin JButtonların vs. tanımlanması yapılmıştır.



JFrame görünür olduğunda ekrana bir JPanel gelmektedir. Bu JPanel içerisinde görevlinin evrak imzalama işlemi için gerekli olan nesneler bulunmaktadır. Görevli, imzalayacağı evrağın doğrulama kodunu Integer tipinde veri tutan JComboBox içerisinde görebilmekte ve seçebilmektedir.



JComboBox itemleri yani evrak doğrulama kodları, JFrame başlatılır başlatılmaz otomatik olarak eklenmektedir. Görevli herhangi bir evrağı imzaladığında, Combo Box içerisindeki itemler yenilenmektedir. İmzalanacak hiçbir evrak kalmayınca da bu süreç devam etmektedir. Yalnızca imzalayan bilgisi olmayan evraklar Combo Box içerisinde görüntülenmektedir.

```
JComboBox<Integer> comboBox_evraklar = new JComboBox<Integer>();
comboBox_evraklar.setBackground(Color.ORANGE);
comboBox_evraklar.setFont(new Font("Arial", Font.BOLD, 16));
comboBox_evraklar.setMaximumRowCount(1000);
comboBox_evraklar.setBounds(305, 54, 112, 32);
panel.add(comboBox_evraklar);
for (int i=0;i<Main.evraklar.size();i++) {
    if (Main.evraklar.get(i).getImzalayan()==null)
        comboBox_evraklar.addItem(Main.evraklar.get(i).getDogrulamaKod());
}
```

Görevli imzalayacağı evrağı seçip, "Evrakı İmzala" JButton'una bastığında JButton'un Mouse Event kodları devreye girmektedir. Bu kodların içeriği; görevlinin seçmiş olduğu doğrulama kodu bilgisinin bir değişkene kaydedilmesi, bir for döngüsü ile bu doğrulama kodu ile uyuşan evrağın bulunması ve bu evrağın imzalayan bilgisinin, mevcut görevli olarak değiştirilmesi şeklindedir. Daha sonrasında evrak imzalama ekranındaki JComboBox içerisinde yer alan itemler yenilenerek, imzalanmış olan evrağın o listeden kaldırılması sağlanmaktadır. Bu süreç içerisinde herhangi bir hata oluşur ise catch bloğu içerisinde yakalanarak ekrana pop-up mesajı verilmektedir.

```

public void mouseClicked(MouseEvent e) {
    try {
        int secim = Integer.valueOf(comboBox_evraklar.getSelectedItem().toString());
        for (int i=0;i<Main.evraklar.size();i++) {
            if (secim==Main.evraklar.get(i).getDogrulamaKod()) {
                Main.evraklar.get(i).setImzalayan(Main.gorevliler.get(index));
                sqlBaglantisi.evrakKayit();
                JOptionPane.showMessageDialog(null, "Evrak İmzalandı");
                comboBox_evraklar.removeAllItems();
                for (int j=0;j<Main.evraklar.size();j++) {
                    if (Main.evraklar.get(j).getImzalayan()==null) {
                        comboBox_evraklar.addItem(Main.evraklar.get(j).getDogrulamaKod());
                    }
                }
                return;
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Hata Meydana Geldi !");
    }
}

```

Görevli imzalayacağı evrakların detaylarını görebilmek için tabbed pane içerisinde yer alan “İmza Bekleyen Evrakları Görüntüle” butonuna basarak imza bekleyen evrakların görüntüleneceği JPanel’e ulaşabilmektedir. Bu JPanel içerisinde bir JTable yer almaktadır. Bu JTable içerisinde evrağın özellikleri olan; Seri No, Doğrulama Kodu, Tür, Tarih, İmzalayan, Oluşturan, Geçerlilik kolonları yer almaktadır.

```

table_imzaBekleyenler = new JTable();
table_imzaBekleyenler.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
table_imzaBekleyenler.setFillsViewportHeight(true);
table_imzaBekleyenler.setAutoCreateRowSorter(true);
table_imzaBekleyenler.setInterCellSpacing(new Dimension(5, 5));
table_imzaBekleyenler.setRowMargin(5);
table_imzaBekleyenler.setFont(new Font("Arial", Font.PLAIN, 18));
table_imzaBekleyenler.setBorder(null);
table_imzaBekleyenler.setDefaultEditor(Object.class, null);
table_imzaBekleyenler.setBackground(new Color(126, 217, 87));
scrollPane_1.setViewportView(table_imzaBekleyenler);
DefaultTableModel tblModel1 = (DefaultTableModel)table_imzaBekleyenler.getModel();
String[] cols = {"Seri No", "Dogrulama Kod", "Tür", "Tarih", "İmzalayan", "Oluşturan", "Geçerlilik"};
tblModel1.setColumnIdentifiers(cols);

```

Görevli bu evrakları görüntüleyebilmek için JPanel’in alt kısmında yer alan “İmza Bekleyen Evrakları Görüntüle” JButton’una tıklayarak JButton’un Mouse Event kodlarını devreye sokabilir. Bu kodlar JTable içerisinde yer alan tüm satırların silinmesi daha sonrasında imzalanmamış evrakların for döngüsü ile kontrol edilerek kolon bilgilerinin bir String dizi içinde oluşturulup yeni bir satır olarak eklenmesi şeklindedir. Yeni satır eklenmesi işlemi addRow komutu ile yapılmaktadır.

```

public void mouseClicked(MouseEvent e) {
    tblModel1.getDataVector().removeAllElements();
    String gecerli;
    try {
        for (evrak evrak : Main.evraklar) {
            if (evrak.getImzalayan() == null) {
                if (evrak.isGeçerlilik()) {
                    gecerli = "Geçerli";
                }
                else {
                    gecerli = "Geçerli Değil";
                }
            }
            String[] tmp = {String.valueOf(evrak.getSeriNo()),
                String.valueOf(evrak.getDogrulamaKod()),
                String.valueOf(evrak.getTur()),
                evrak.getTarih(),
                evrak.ImzalayanKim(),
                String.valueOf(evrak.getOluşturan().getAd()+" "+evrak.getOluşturan().getSoyAd()),
                gecerli};
            tblModel1.addRow(tmp);
        }
    }
}

```

Görevli imzaladığı evrakların detaylarını görebilmek için tabbed pane içerisinde yer alan “İmzalanan Evrakları Görüntüle” butonuna basarak imza bekleyen evrakların görüntüleneceği JPanel’e ulaşabilmektedir. Bu JPanel içerisinde bir JTable yer almaktadır. Bu JTable içerisinde evrağın özellikleri olan; Seri No, Doğrulama Kodu, Tür, Tarih, İmzalayan, Oluşturan, Geçerlilik kolonları yer almaktadır.



```

table_imzalananEvraklar = new JTable();
table_imzalananEvraklar.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
table_imzalananEvraklar.setFillViewportHeight(true);
table_imzalananEvraklar.setRowMargin(5);
table_imzalananEvraklar.setRowHeight(30);
table_imzalananEvraklar.setInterCellSpacing(new Dimension(5, 5));
table_imzalananEvraklar.setDefaultEditor(Object.class, null);
table_imzalananEvraklar.setBackground(new Color(126, 217, 87));
table_imzalananEvraklar.setBorder(null);
table_imzalananEvraklar.setFont(new Font("Arial", Font.PLAIN, 18));
scrollPane.setViewportView(table_imzalananEvraklar);
DefaultTableModel tblModel = (DefaultTableModel)table_imzalananEvraklar.getModel();
tblModel.setColumnIdentifiers(cols);

```

Görevli bu evrakları görüntüleyebilmek için JPanel’ın alt kısmında yer alan “İmzalanan Evrakları Görüntüle” JButton’una tıklayarak JButton’un Mouse Event kodlarını devreye sokabilir. Bu kodlar JTable içerisinde yer alan tüm satırların silinmesi daha sonrasında imzalanmamış evrakların for döngüsü ile kontrol edilerek kolon bilgilerinin bir String dizi içinde oluşturulup yeni bir satır olarak eklenmesi şeklindedir. Yeni satır eklenmesi işlemi addRow komutu ile yapılmaktadır.

```

public void mouseClicked(MouseEvent e) {
    tblModel.getDataVector().removeAllElements();
    String gecerli;
    try {
        for (evrak evrak : Main.evraklar) {
            if (evrak.getImzalayan() == Main.gorevliler.get(index)) {
                if (evrak.isGecerlilik()) {
                    gecerli = "Geçerli";
                }
                else {
                    gecerli = "Geçerli Değil";
                }
                String[] tmp = {String.valueOf(evrak.getSeriNo()),
                    String.valueOf(evrak.getDogrulamaKod()),
                    String.valueOf(evrak.getTur()),
                    evrak.getTarih(),
                    evrak.ImzalayanKim(),
                    String.valueOf(evrak.getOlusturan().getAd()+" "+evrak.getOlusturan().getSoyAd()),
                    gecerli};
                tblModel.addRow(tmp);
            }
        }
    }
}

```

Görevli; kişisel bilgilerini görüntülemek isterse, hesap şifresini değiştirmek isterse veya hesabından çıkış yapmak isterse tabbed pane içerisinde yer alan “Hesap Ayarları” butonuna tıklayarak hesap ayarlarının olduğu JPanel’e ulaşabilmektedir. Bu JPanel içerisinde “Kişisel Bilgileri Göster”, “Şifre Değiştir”, “Çıkış Yap” JButton nesneleri yer almaktadır. Şifre değiştirme işlemleri için ise 3 adet JPasswordField nesnesi yer almaktadır.

Evrek Otomasyon

Hoşgeldiniz Hakan Tunç (Görevli)

Hesap Ayarları | İmzalanın Evrakları Görüntüle | İmza Bekleyen Evrakı İmzala | İmza Bekleyen Evrakları Görüntüle

**Kişisel Bilgileri Göster**

Eski Şifrenizi Giriniz:

Yeni Şifrenizi Giriniz:

Yeni Şifreyi Tekrar Giriniz:

**Şifre Değiştir**

TC NO: 12345678911  
Ad: Hakan  
Soyad: Tunç  
Doğum Tarihi: 2002-04-10  
Maaş: 6000.0  
İş Giriş Tarihi: 2022-12-25  
Çalışma Durumu: Aktif

**Çıkış Yap**

Görevli “Kişisel Bilgileri Göster” JButton’una bastığında “Gorevli” sınıfındaki override edilen toString metodu kullanılarak görevlinin bilgileri ekranın sol bölümünde yer alan boş JLabel içerisine aktarılmaktadır.

```
buton_kisiselBilgiGoster.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        output_kisiselBilgi.setText(Main.gorevliiler.get(index).toString());  
    }  
});
```

Görevli şifresini değiştirmek için eski şifresini ve yeni şifresini girmesi gerekmektedir. Olası bir güvenlik açığını önlemek amacı ile kullanıcıdan eski şifresi istenmiştir ve hatalı giriş yapabileme ihtimaline karşın yeni şifresi 2 defa istenmiştir. Görevli bu girişleri JPasswordField alanlarına girecektir. Görevlinin yaptığı girişlerde çeşitli kontroller sağlanmıştır. Bu kontroller:

- Görevlinin gireceği şifre güvenlik nedeni ile en az 6 karakter olmalıdır.
- Şifre maksimum 253 karakter olabilir. Bunun nedeni yapılacak girişin veri tabanında sorun çıkarıp programın çökmesini engellemek içindir. Normal şartlarda herhangi bir şifrenin 254 karakter olabilmesi pek normal değildir fakat veri tabanında ve dahi programda herhangi bir zafiyetin oluşmasını engellemek için bu kontrol sağlanmıştır.
- Görevlinin giriş yapacağı alanlar boş mu veya boşluk karakteri içeriyor mu?
- Görevlinin girdiği 2 yeni şifre birbirleri ile uyuyor mu?
- Görevlinin girdiği eski şifre, gerçek şifresi ile uyuyor mu?

Eğer tüm koşullar sağlanırsa kullanıcının şifresi değiştirilecektir, ekranda pop-up mesajı gösterilecektir ve ArrayList içindeki değerler MySQL’e aktarılacaktır.

```

public void mouseClicked(MouseEvent e) {
    String eskisifre =String.valueOf(input_eskiSifre.getPassword());
    String yenisifre1 =String.valueOf(input_yeniSifre.getPassword());
    String yenisifre2 =String.valueOf(input_yeniSifre2.getPassword());
    if (eskisifre.isBlank()) {
        JOptionPane.showMessageDialog(null,"Eski Şifre Girişi Yapılmamıştır!");
        return;}
    else if (yenisifre1.length()>253) {
        JOptionPane.showMessageDialog(null,"Yeni Şifre Maksimum Uzunluğu Aşmaktadır !");
        return;}
    else if (yenisifre1.length()<6) {
        JOptionPane.showMessageDialog(null,"Yeni Şifre Minimum 6 Karakter olmalıdır !");
        return;}
    else if(Objects.equals(eskisifre,Main.gorevliler.get(index).getSifre())){
        if (yenisifre1.isBlank() || yenisifre2.isBlank() ) {
            JOptionPane.showMessageDialog(null,"Yeni Şifre İçin Giriş Yapmadınız !");
            return;}
        else if (yenisifre1.equals(yenisifre2)){
            Main.gorevliler.get(index).setSifre(yenisifre2);
            sqlBaglantisi.gorevliKayit();
            JOptionPane.showMessageDialog(null,"Şifre Başarıyla Değiştirilmiştir !");
            return;}
        else {
            JOptionPane.showMessageDialog(null,"Yeni Şifreler Uyuşmamaktadır. Lütfen Kontrol Ediniz !");
            return;}
    }
    else {
        JOptionPane.showMessageDialog(null,"Eski Şifre Yanlış Girilmiştir. Lütfen Kontrol Ediniz !");
        return;}
}

```

Eğer tüm koşullar sağlanırsa görevlinin şifresi değiştirilecektir, ekranda pop-up mesajı gösterilecektir ve görevliler ArrayList’inin içindeki değerler MySQL’e aktarılacaktır.

```

else if (yenisifre1.equals(yenisifre2)){
    Main.gorevliler.get(index).setSifre(yenisifre2);
    sqlBaglantisi.gorevliKayit();
    JOptionPane.showMessageDialog(null,"Şifre Başarıyla Değiştirilmiştir !");
    return;}

```

Görevli çıkış yapmak istediğinde ise “Çıkış Yap” JButton’una basacaktır. JButton’un Mouse Event kodları devreye girerek “AnaEkran” sınıfından yeni bir nesne oluşturacaktır.

“AnaEkran” sınıfının yapıcı metodundan oluşacak JFrame’in görünürlüğü true olarak ayarlanıp mevcut JFrame kapatılacaktır ve kullanıcı hesaptan çıkış yaparak Ana Ekrana geri dönmüş olacaktır.

```

buton_hesaptanCikisYap.addMouseListener(new MouseAdapter()
@Override
public void mouseClicked(MouseEvent e) {
    dispose();
    AnaEkran yeni = new AnaEkran();
    yeni.setVisible(true);
    return;}

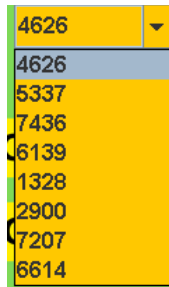
```

## Class MudurArayuz

“MudurArayuz” sınıfı, hesabına giriş yapmış olan müdürler için Swing kodlarını içeren müdür hesabı arayüzü tasarımı yapılmış bir sınıftır. Giriş ekranından başarılı bir giriş gerçekleştirilse yapıcı sınıfı çalışmaktadır. Bu yapıcı sınıfın içinde Frame boyutunun 800x600 olarak ayanması, yeniden boyutlandırmanın kapatılması, JPanel’in tanımlanması, ekranda gözükecek olan JFrame’lerin JButtonların vs. tanımlanması yapılmıştır.



JFrame görünür olduğunda ekrana “Evrak İşlemleri” isimli bir JPanel gelmektedir. Bu JPanel içerisinde müdürün yapacağı evrak işlemleri için gerekli olan nesneler bulunmaktadır. Müdür, işlem yapacağı evrağın doğrulama kodunu Integer tipinde veri tutan JComboBox içerisinde görebilmekte ve seçebilmektedir.



Müdür işlem yapacağı evrağın doğrulama kodunu seçtikten sonra JPanel içerisinde yer alan “Evrakı Pasif Hale Getir” JButton’una basarsa JButton’un Mouse Event kodları devreye girecektir. Bu kodlar; JComboBox içerisinde seçilmiş olan doğrulama kodunun bir değişkene kaydedilmesi ve bir döngü ile evrak ArrayList’inin içinden doğrulama kodu ile uyuşan evrağın index numarasının tespit edilmesi, bu evrağın aktif mi diye kontrol edilmesi, aktif değil ise ekrana pop-up mesajı verilmesi, aktif ise pasif hale getirilmesi ve ArrayList içindeki yeni bilgilerin MySQL’e kaydedilmesidir. Eğer bir hata meydana gelirse catch bloğu içerisinde yakalanıp ekrana pop-up verilecektir.

```

public void mouseClicked(MouseEvent e) {
    try {
        int secim = Integer.valueOf(KomboBox_evraklar.getSelectedItem().toString());
        for (int i=0;i<Main.evraklar.size();i++) {
            if (secim==Main.evraklar.get(i).getDogrulamaKod() && Main.evraklar.get(i).isGecerlilik() ){
                Main.evraklar.get(i).setGecerlilik(false);
                sqlBaglantisi.evrakKayit();
                JOptionPane.showMessageDialog(null, "Evrak Geçerliliği Pasif Hale Getirildi !");
                return;
            }
            else if (secim==Main.evraklar.get(i).getDogrulamaKod() && !Main.evraklar.get(i).isGecerlilik() ){
                JOptionPane.showMessageDialog(null, "Evrak Geçerliliği Zaten Pasif Haldedir !!!");
                return;
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Hata Meydana Geldi !");
    }
}

```

Müdür işlem yapacağı evrağın doğrulama kodunu seçtikten sonra JPanel içerisinde yer alan “Evrakı Aktif Hale Getir” JButton’una basarsa JButton’un Mouse Event kodları devreye girecektir. Bu kodlar; JComboBox içerisinde seçilmiş olan doğrulama kodunun bir değişkene kaydedilmesi ve bir döngü ile evrak ArrayList’inin içinden doğrulama kodu ile uyuşan evrağın index numarasının tespit edilmesi, bu evrağın pasif mi diye kontrol edilmesi, pasif değil ise ekrana pop-up mesajı verilmesi, pasif ise aktif hale getirilmesi ve ArrayList içindeki yeni bilgilerin MySQL’e kaydedilmesidir. Eğer bir hata meydana gelirse catch bloğu içerisinde yakalanıp ekrana pop-up verilecektir.

```

public void mouseClicked(MouseEvent e) {
    try {
        int secim = Integer.valueOf(KomboBox_evraklar.getSelectedItem().toString());
        for (int i=0;i<Main.evraklar.size();i++) {
            if (secim==Main.evraklar.get(i).getDogrulamaKod() && !Main.evraklar.get(i).isGecerlilik() ){
                Main.evraklar.get(i).setGecerlilik(true);
                sqlBaglantisi.evrakKayit();
                JOptionPane.showMessageDialog(null, "Evrak Geçerliliği Aktif Hale Getirildi !");
                return;
            }
            else if (secim==Main.evraklar.get(i).getDogrulamaKod() && Main.evraklar.get(i).isGecerlilik() ){
                JOptionPane.showMessageDialog(null, "Evrak Geçerliliği Zaten Aktif Haldedir !!!");
                return;
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Hata Meydana Geldi !");
    }
}

```

Müdür işlem yapacağı evrağın doğrulama kodunu seçtikten sonra JPanel içerisinde yer alan “Evrakı Sil” JButton’una basarsa JButton’un Mouse Event kodları devreye girecektir. Bu kodlar; JComboBox içerisinde seçilmiş olan doğrulama kodunun bir değişkene kaydedilmesi ve bir döngü ile evrak ArrayList’inin içinden doğrulama kodu ile uyuşan evrağın index numarasının tespit edilmesi ve daha sonrasında ArrayList’in bu indexindeki değerin silinip yeni verilerin MySQL’e kaydedilmesidir. Bu sayede evrak sistemden silinmiş olacaktır ve JComboBox içerisindeki itemler yenilenip, silinen evrağın itemi kaybolacaktır. Eğer bir hata meydana gelirse catch bloğu içerisinde yakalanıp pop-up mesajı verilecektir.

```

buton_evrakSil.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        try {
            int secim = Integer.valueOf(KomboBox_evraklar.getSelectedItem().toString());
            for (int i=0; i<Main.evraklar.size();i++) {
                if (secim==Main.evraklar.get(i).getDogrulamaKod()) {
                    Main.evraklar.remove(i);
                    sqlBaglantisi.evrakKayit();
                    JOptionPane.showMessageDialog(null, "Evrak Sistemden Silindi !");
                    KomboBox_evraklar.removeAllItems();
                    for (int j=0;j<Main.evraklar.size();j++) {
                        KomboBox_evraklar.addItem(Main.evraklar.get(j).getDogrulamaKod());
                    }
                }
            }
        } catch (Exception ee) {
            JOptionPane.showMessageDialog(null, "Hata Meydana Geldi !");
            return;
        }
    }
});

```

Müdür programda kayıtlı olan tüm evrakları görüntülemek isterse tabbed pane içerisinde yer alan “Tüm Evrakları Görüntüle” butonuna basarak hedef JPanel’e ulaşabilmektedir. Bu JPanel içerisinde bir adet JButton ve JTable yer almaktadır. Bu JTable içerisinde Seri No, Doğrulama Kodu, Tür, Tarih, İmzalayan, Oluşturan, Geçerlilik kolonları yer almaktadır.

EvraK Otomasyon

Hoşgeldiniz Emir Doğan (Müdür)						
Tüm Evrakları Görüntüle		Hesap Ayarları	Görevli İşe Alım			
Evrak İşlemleri		Görevli İşlemleri		Tüm Görevlileri Görüntüle		
Seri No	Doğrulama Kod	Tür	Tarih	İmzalayan	Oluşturan	Geçerlilik
0	4626	Nüfus	2022-12-25	Mehmet Akif	Eren Doğan	Geçerli
2	5337	Gelir	2022-12-25	Hakan Tunç	Eren Doğan	Geçerli De
3	7436	İkametgah	2022-12-25	Mehmet Akif	Eren Doğan	Geçerli De
4	6139	Ceza	2022-12-25	Hakan Tunç	Eren Doğan	Geçerli
5	1328	Gelir	2022-12-25	Mehmet Akif	Eren Doğan	Geçerli
7	2900	Taşıt	2022-12-25	Mehmet Akif	Anıl Yılmaz	Geçerli
8	7207	Gelir	2022-12-25	Yok	Anıl Yılmaz	Geçerli
9	6614	Ceza	2022-12-26	Yok	Eren Doğan	Geçerli
Tüm Evrakları Görüntüle						

```

tumGorevliilerTable = new JTable();
tumGorevliilerTable.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
tumGorevliilerTable.setFillViewportHeight(true);
tumGorevliilerTable.setAutoCreateRowSorter(true);
tumGorevliilerTable.setInterCellSpacing(new Dimension(5, 5));
tumGorevliilerTable.setRowMargin(5);
tumGorevliilerTable.setFont(new Font("Arial", Font.PLAIN, 16));
tumGorevliilerTable.setBorder(null);
tumGorevliilerTable.setDefaultEditor(Object.class, null);
tumGorevliilerTable.setBackground(new Color(126, 217, 87));
panel_2.setBackground(new Color(126, 217, 87));
scrollPane_1.setViewportView(tumGorevliilerTable);
DefaultTableModel tblModell = (DefaultTableModel)tumGorevliilerTable.getModel();
String[] cols = {"T.C Kimlik No", "Ad", "Soyad", "Doğum Tarihi", "Maaş",
    "Giriş Tarihi", "Çıkış Tarihi", "Durumu"};
tblModell.setColumnIdentifiers(cols);

```



Müdür “Tüm Evrakları Görüntüle” isimli JButton’a basarak JButton’un Mouse Event kodlarını devreye sokar. Bu kodlar: JTable içerisindeki tüm satırların temizlenmesi ve daha sonrasında bir döngü ile “evraklar” ArrayList’inin içindeki tüm evraklara ulaşıp, bilgilerinin bir String dizisinde tutularak satır olarak eklenmesidir.

```
public void mouseClicked(MouseEvent e) {
    tblModel.getDataVector().removeAllElements();
    String gecerli;
    try {
        for (Evrak evrak : Main.evraklar) {
            if (evrak.isGecerlilik()) {
                gecerli = "Geçerli";
            }
            else {
                gecerli = "Geçerli Değil";
            }
            String[] tmp = {String.valueOf(evrak.getSeriNo()),
                String.valueOf(evrak.getDogrulamaKod()),
                String.valueOf(evrak.getTur()),
                evrak.getTarih(),
                evrak.ImzalanKim(),
                String.valueOf(evrak.getOlusturan().getAd()+" "+evrak.getOlusturan().getSoyAd()),
                gecerli};
            tblModel.addRow(tmp);
            tumEvraklarTable.setAutoCreateRowSorter(true);
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Kayıtlı Evrak Yoktur !");
    }
}
```

Müdür programda kayıtlı olan tüm görevlileri görüntülemek isterse tabbed pane içerisinde yer alan “Tüm Görevlileri Görüntüle” butonuna basarak hedef JPanel’e ulaşabilmektedir. Bu JPanel içerisinde bir adet JButton ve JTable yer almaktadır. Bu JTable içerisinde görevli özellikleri olan: T.C Kimlik No, ad, soy ad, doğum tarihi, maaş, giriş tarihi, çıkış tarihi, çalışma durumu" kolonları yer almaktadır.

Evrak Otomasyon

Hoşgeldiniz Emir Doğan (Müdür)

Tüm Evrakları Görüntüle

Hesap Ayarları

Görevli İşe Alım

Evrak İşlemleri

Görevli İşlemleri

Tüm Görevlileri Görüntüle

T.C Kimlik No	Ad	Soyad	Doğum Tarihi	Maaş	Giriş Tarihi	Çıkış Tarihi	Durumu
120681321...	Mehmet	Akif	2007-05-04	6500.0	2022-12-25		Çalışıyor
123456789...	Hakan	Tunç	2002-04-10	6000.0	2022-12-25		Çalışıyor
123456789...	Yasin	Bilekçi	2002-03-04	6000.0	2022-12-25	2022-12-26	Çalışmıyor

Tüm Görevlileri Görüntüle

```

tumGorevlilerTable = new JTable();
tumGorevlilerTable.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
tumGorevlilerTable.setFillsViewportHeight(true);
tumGorevlilerTable.setAutoCreateRowSorter(true);
tumGorevlilerTable.setInterCellSpacing(new Dimension(5, 5));
tumGorevlilerTable.setRowMargin(5);
tumGorevlilerTable.setFont(new Font("Arial", Font.PLAIN, 16));
tumGorevlilerTable.setBorder(null);
tumGorevlilerTable.setDefaultEditor(Object.class, null);
tumGorevlilerTable.setBackground(new Color(126, 217, 87));
panel_2.setBackground(new Color(126, 217, 87));
scrollPane_1.setViewportView(tumGorevlilerTable);
DefaultTableModel tblModel1 = (DefaultTableModel)tumGorevlilerTable.getModel();
String[] cols = {"T.C Kimlik No", "Ad", "Soyad", "Doğum Tarihi", "Maaş",
    "Giriş Tarihi", "Çıkış Tarihi", "Durumu"};
tblModel1.setColumnIdentifiers(cols);

```

Müdür “Tüm Görevlileri Görüntüle” isimli JButton’a basarak JButton’un Mouse Event kodlarını devreye sokar. Bu kodlar: JTable içerisindeki tüm satırların temizlenmesi ve daha sonrasında bir döngü ile “gorevliler” ArrayList’inin içindeki tüm görevlilere ulaşıp bilgilerinin bir String dizisinde tutularak satır olarak eklenmesidir.

```

buton_tumGorevlileriGoruntule.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        tblModel1.getDataVector().removeAllElements();
        String gecerli;
        try {
            for (gorevli gorevli: Main.gorevliler) {
                if (gorevli.isCalismaDurumu()) {
                    gecerli = "Çalışıyor";
                }
                else {
                    gecerli = "Çalışmıyor";
                }
                String[] tmp = {String.valueOf(gorevli.getTcNo()),
                    gorevli.getAd(),
                    gorevli.getSoyAd(),
                    gorevli.getDogumTarihi(),
                    String.valueOf(gorevli.getMaas()),
                    gorevli.getIsGirisTarihi(),
                    gorevli.getIsCikisTarihi(),
                    gecerli};
                tblModel1.addRow(tmp);
            }
        } catch (Exception ee) {
            JOptionPane.showMessageDialog(null, "Görevli Yoktur !");
            return;
        }
    }
});

```

Müdür programda kayıtlı olan görevlilerin maaşlarını değiştirip, işten çıkarmak, işe tekrar almak isterse tabbed pane içerisinde yer alan “Görevli İşlemleri” isimli butona tıklayarak hedef JPanel’e ulaşabilir. Bu JPanel içerisinde işlem yapacağı görevlinin T.C. kimlik numarasını seçebileceği long tipinde değer tutan bir JComboBox, yeni maaş miktarını girebileceği bir JTextField, 3 adet JButton, 1 adet JCheckBox ve işlem yapılacak olan görevlinin bilgilerinin görüntüleneceği 1 adet JLabel mevcuttur.

Evrak Otomasyon

Hoşgeldiniz Emir Doğan (Müdür)

Tüm Evrakları Görüntüle Hesap Ayarları Görevli İşe Alım

Evrak İşlemleri Görevli İşlemleri Tüm Görevlileri Görüntüle

İşlem Yapacağınız Görevlinin T.C. Kimlik Numarasını Seçiniz 12068132160

Yeni Maaş Miktarını Giriniz

**Yeni Maaş Miktarını Ayarla**

TC NO: 12068132160  
Ad: Mehmet  
Soyad: Akif  
Doğum Tarihi: 2007-05-04  
Maaş: 6500.0  
İş Giriş Tarihi: 2022-12-25  
Çalışma Durumu: Aktif

☐ İşe Alma, Çıkarma Koşullarını Kabul Ediyor, Onaylıyorum.

**İşe Tekrar Al** **İşten Çıkar**

Müdür bir görevlinin maaşını değiştirmek isterse sağ üst kısımdaki Combo Box içerisinde işlem yapmak istediği görevlinin T.C. Numarasını seçmelidir. Müdür işlem yapacağı görevlinin bilgilerini sağ bölümdeki JLabel içerisinde görebilir. Combo Box üzerinden yeni item seçildiği zaman JLabel'ın içeriği de otomatik olarak değişecektir. Müdür, yeni maaş miktarını ekrandaki alana girmeli ve "Yeni Maaş Miktarını Ayarla" butonuna basmalıdır. Butona basıldığında JButton'un Mouse Event kodları devreye girecektir. Bu kodlar: JComboBox üzerinden seçilen görevli T.C. kimlik numarasının long tipinde bir değişkene kaydedilmesi, maaş miktarının uzunluğunun kontrol edilmesi, maaş miktarının girilmemesi veya string içeren bir değer girilmesinin kontrol edilmesi daha sonrasında bir for döngüsü ile "gorevliler" ArrayList'inin içindeki bilgilerden görevlinin bulunması, yeni maaş bilgisinin ArrayList'e kaydedilmesi, daha sonrasında bu ArrayList'in içindeki değerlerin MySQL veri tabanına aktarılması ve ekrana pop-up mesajı ile eski ve yeni maaşın gösterilmesi şeklindedir. Eğer müdür hatalı bir giriş yaparsa, catch bloğu içerisinde yakalanarak ekrana pop-up mesajı verilecektir.

```
public void mouseClicked(MouseEvent e) {
    try {
        long secim = (long) KomboBox_gorevliler.getSelectedIndex();
        if (input_yeniMaasMiktar.getText().length() > 15) {
            JOptionPane.showMessageDialog(null, "Maaş Miktarını Çok Fazla Girdiniz !");
            return;
        }
        else if (input_yeniMaasMiktar.getText().isBlank() || input_yeniMaasMiktar.getText().equals("0")) {
            JOptionPane.showMessageDialog(null, "Maaş Miktarını Girmediniz !");
            return;
        }
        for (int i=0; i<Main.gorevliler.size(); i++) {
            if (secim==Main.gorevliler.get(i).getTcNo()) {
                Double yeniMaas = Double.parseDouble(input_yeniMaasMiktar.getText());
                Main.gorevliler.get(i).setMaas(yeniMaas);
                sqlBaglantisi.gorevliKayit();
                JOptionPane.showMessageDialog(null, "Eski Maaş:" + Main.gorevliler.get(i).getMaas() + " Yeni Maaş: " + yeniMaas);
                return;
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Hatalı Giriş Yaptınız !");
        return;
    }
}
```

Müdür bir görevliyi işten çıkarmak isterse sağ üst kısımdaki Combo Box içerisinde işlem yapmak istediği görevlinin T.C. Numarasını seçmelidir ve daha sonrasında ekranda "İşe

alma, çıkarma koşullarını kabul ediyor, onaylıyorum” isimli JCheckBox’a tik koymalı ve “İşten Çıkar” JButton’una basmalıdır. Daha sonrasında JButton’un Mouse Event kodları devreye girecektir. Bu kodlar: JComboBox üzerinden seçilen görevli T.C. kimlik numarasının long tipinde bir değişkene kaydedilmesi, müdürün JCheckBox’u onaylamasının kontrolünün sağlanması, bir for döngüsü ile “gorevliler” ArrayList’inin içinden görevlinin bilgilerine ulaşılması, görevlinin çalışma durumu false olarak ayarlanması ve çıkış tarihi olarak, işlemin yapıldığı günün tarihi belirlenmesi şeklindedir. Daha sonrasında yapılan değişiklikler ArrayList vasıtası ile MySQL veritabanına iletilmektedir. Ekrana görevlinin işten çıkarıldığına dair pop-up mesajı verilmektedir. Bir hata oluşur ise catch bloğunda yakalanıp ekrana pop-up mesajı verilecektir.

```
public void mouseClicked(MouseEvent e) {
    try {
        long secim = (long) KomboBox_gorevliler.getSelectedIndex();
        if (!checkbox_isKosullar.isSelected()){
            JOptionPane.showMessageDialog(null, "Koşulları Onaylamadınız Kardeş !");
            return;
        }
        for (int i=0;i<Main.gorevliler.size();i++) {
            if (secim==Main.gorevliler.get(i).getTcNo() ){
                if (!Main.gorevliler.get(i).isCalismaDurumu()) {
                    JOptionPane.showMessageDialog(null, "Görevli Zaten İşten Çıkarılmıştır !");
                    return;
                }
                Main.gorevliler.get(i).setCalismaDurumu(false);
                Main.gorevliler.get(i).setIsCikisTarihi(LocalDate.now().toString());
                sqlBaglantisi.gorevliKayit();
                JOptionPane.showMessageDialog(null, "Görevli İşten Çıkarıldı !");
                return;
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Hata Meydana Geldi !");
    }
}
```

Müdür bir görevliyi tekrar işe almak isterse sağ üst kısımdaki Combo Box içerisinden işlem yapmak istediği görevlinin T.C. Numarasını seçmelidir ve daha sonrasında ekranda “İşe alma, çıkarma koşullarını kabul ediyor, onaylıyorum” isimli JCheckBox’a tik koymalı ve “İşe Tekrar Al” JButton’una basmalıdır. Daha sonrasında JButton’un Mouse Event kodları devreye girecektir. Bu kodlar: JComboBox üzerinden seçilen görevli T.C. kimlik numarasının long tipinde bir değişkene kaydedilmesi, müdürün JCheckBox’u onaylamasının kontrolünün sağlanması, bir for döngüsü ile “gorevliler” ArrayList’inin içinden görevlinin bilgilerine ulaşılması, görevlinin çalışma durumunun true olarak ayarlanması, çıkış tarihinin null olarak ayarlanması, giriş tarihinin ise işlemin yapıldığı günün tarihi belirlenmesi şeklindedir. Daha sonrasında yapılan değişiklikler ArrayList vasıtası ile MySQL veritabanına iletilmektedir. Ekrana görevlinin tekrar işe alındığına dair bir pop-up mesajı verilmektedir. Bir hata oluşur ise catch bloğunda yakalanıp ekrana pop-up mesajı verilecektir.

```

buton_iseTekrarAl.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        try {
            long secim = (long) KomboBox_gorevliler.getSelectedIndex();
            if (!checkbox_isKosullar.isSelected()){
                JOptionPane.showMessageDialog(null, "Koşulları Onaylamadınız Kardeş !");
                return;
            }
            for (int i=0;i<Main.gorevliler.size();i++) {
                if (secim==Main.gorevliler.get(i).getTcNo()){
                    if (Main.gorevliler.get(i).isCalismaDurumu()) {
                        JOptionPane.showMessageDialog(null, "Görevli Zaten Çalışmaktadır!");
                        return;
                    }
                    Main.gorevliler.get(i).setCalismaDurumu(true);
                    Main.gorevliler.get(i).setIsCikisTarihi(null);
                    Main.gorevliler.get(i).setIsGirisTarihi(LocalDate.now().toString());
                    sqlBaglantisi.gorevliKayit();
                    JOptionPane.showMessageDialog(null, "Görevli Tekrar İşe Alındı!");
                    return;
                }
            }
        } catch (Exception ee) {
            JOptionPane.showMessageDialog(null, "Hatalı Giriş Yaptınız !");
        }
    }
});

```

Müdür yeni bir görevliyi işe almak isterse tabbed pane’de yer alan “Görevli İşe Alım” butonuna basmalıdır. Karşısına gelecek olan JPanel’de giriş yapılacak olan 6 adet JTextField, Onay verilebilecek 2 adet JCheckBox bulunmaktadır ve görevli işe alındıktan sonra görevli bilgilerinin gösterileceği bir JLabel mevcuttur.

EvraK Otomasyon
— □ ×

Hoşgeldiniz Emir Doğan (Müdür)

Tüm Evrakları Görüntüle

Hesap Ayarları

Görevli İşe Alım

Evrak İşlemleri

Görevli İşlemleri

Tüm Görevlileri Görüntüle

Görevli Kayıt Ekranına Hoş Geldiniz.

Ad:

Soyad:

T.C. No:

Maaş:

Doğum Tarihi:

Güvenlik Kodu:

TC NO: 12345679855

Ad: Akif

Soyad: Avcu

Doğum Tarihi: 2004-07-10

Maaş: 6800.0

İş Giriş Tarihi: 2022-12-30

Çalışma Durumu: Aktif

Şifre: 498843

☒ Çalışma Durumu Aktif Olarak Ayarlınsın.
 ☒ Çalışan İşe Alım Koşullarını Kabul Ediyor, Onaylıyorum.

İşe Al

Müdür gerekli olan bilgileri doldurduktan ve onayları verdikten sonra “İşe Al” JButton’una tıkladığında JButton’un Mouse Event kodları çalışmaktadır ve çeşitli kontroller sağlanmaktadır. Bu kontroller:

- Ad, Soyad, T.C. No, Maaş, Doğum Tarihi, Güvenlik Kodu girişlerine giriş yapılmış mı?
- Ad, Soyad, T.C. No, Doğum Tarihi, Güvenlik Kodu girişlerinin metin uzunluğu 253 karakterden fazla mı?
- Maaş bölümünün uzunluğu 10 karakterden fazla mı?
- T.C. No uygun bir formatta girilmiş mi?
- İşe alım koşulları onaylanmış mı?
- Doğrulama kodu doğru girilmiş mi?
- Girilen T.C. No ile mevcut bir çalışana ait mi?
- Çalışma durumu aktif olarak ayarlı mı?

```
double Maas = Double.valueOf( input_Maas.getText());
if (Ad.isBlank()) {
    JOptionPane.showMessageDialog(null,"Ad Girmediniz !");
    return;
}
else if (Ad.length()>253) {
    JOptionPane.showMessageDialog(null,"Ad Uzunluğu Çok Fazla. "
        + "Lütfen Sistem Yöneticisi İle İletişime Geçin !");
    return;
}
else if (Soyad.isBlank()) {
    JOptionPane.showMessageDialog(null,"Soyad Girmediniz !");
    return;
}
else if (Soyad.length()>253) {
    JOptionPane.showMessageDialog(null,"Soyad Uzunluğu Çok Fazla. "
        + "Lütfen Sistem Yöneticisi İle İletişime Geçin !");
    return;
}
else if (Tc.isBlank()) {
    JOptionPane.showMessageDialog(null,"T.C. No Girmediniz !");
    return;
}
else if (Tc.length()>12||Tc.length()<9) {
    JOptionPane.showMessageDialog(null,"T.C. No Hatalı Girilmiştir!");
    return;
}
else if (Tarih.isBlank()) {
    JOptionPane.showMessageDialog(null,"Doğum Tarihi Girmediniz !");
    return;
}
```

Kontrollerini içermektedir. Bu kontroller sağlandıktan sonra güvenlik amacı ile rastgele bir şifre üretilmektedir. Yeni görevli “Main” sınıfının içindeki “görevliler” ArrayList’inin içine kaydedilmektedir. ArrayList içindeki bilgiler MySQL veri tabanına aktarılmaktadır ve daha sonrasında bir döngü ile kaydedilen bu kullanıcının bilgileri ekrandaki JLabel içerisine yazılmaktadır ve aynı zamanda rastgele üretilen yeni şifre ekranda pop-up mesajı ile verilmektedir. Daha sonrasında görevli T.C numaralarının olduğu JComboBox itemleri yenilenmektedir.

```
String sifre = String.valueOf( rand.nextInt(900000)+100000);

Main.gorevliler.add(new gorevli(TCNo,Ad,Soyad,Tarih,Maas,sifre));
sqlBaglantisi.gorevliKayit();
JOptionPane.showMessageDialog(null,"İşe Alım Tamamlandı !");
JOptionPane.showMessageDialog(null,"Varsayılan Şifre "+sifre+" Olarak Atanmıştır. Lütfen Not Alınız.");
for (int i=0;i<Main.gorevliler.size();i++){
    if (Main.gorevliler.get(i).getTcNo()==TCNo) {
        if (!checkbox_calismaDurum.isSelected()) {
            Main.gorevliler.get(i).setCalismaDurumu(false);
            Main.gorevliler.get(i).setIsGirisTarihi("");
        }
        label_yeniCalisan.setText(Main.gorevliler.get(i).toString()+ "\nŞifre: "+sifre);
        KomboBox_gorevliler.removeAllItems();
        for (gorevli gorevli:Main.gorevliler){
            KomboBox_gorevliler.addItem(gorevli.getTcNo());
        }
    }
}
```



Müdür kişisel bilgilerini görüntülemek isterse, hesap şifresini değiştirmek isterse veya hesabından çıkış yapmak isterse tabbed pane içerisinde yer alan “Hesap Ayarları” butonuna tıklayarak hesap ayarlarının olduğu JPanel’e ulaşabilmektedir. Bu JPanel içerisinde “Kişisel Bilgileri Göster”, “Şifre Değiştir”, “Çıkış Yap” JButton nesneleri yer almaktadır. Şifre değiştirme işlemleri için ise 3 adet JPasswordField nesnesi yer almaktadır.

Müdür “Kişisel Bilgileri Göster” JButton’una bastığında “Mudur” sınıfındaki override edilen toString metodu kullanılarak görevlinin bilgileri ekranın sol bölümünde yer alan boş JLabel içerisine aktarılmaktadır.

```
buton_kisiselBilgiGoster.addMouseListener(new MouseAdapter() {  
    @Override  
    public void mouseClicked(MouseEvent e) {  
        output_kisiselBilgi.setText(Main.gorevliiler.get(index).toString());  
    }  
});
```

Müdür şifresini değiştirmek için eski şifresini ve yeni şifresini girmesi gerekmektedir. Olası bir güvenlik açığını önlemek amacı ile kullanıcıdan eski şifresi istenmiştir ve hatalı giriş yapabilme ihtimaline karşın yeni şifresi 2 defa istenmiştir. Müdür bu girişleri JPasswordField alanlarına girecektir. Müdürün yaptığı girişlerde çeşitli kontroller sağlanmıştır. Bu kontroller:

- Müdürün gireceği şifre güvenlik nedeni ile en az 6 karakter olmalıdır.
- Şifre maksimum 253 karakter olabilir. Bunun nedeni yapılacak girişin veri tabanında sorun çıkarıp programın çökmesini engellemek içindir. Normal şartlarda herhangi bir şifrenin 254 karakter olabilmesi pek normal değildir fakat veri tabanında ve dahi programda herhangi bir zafiyetin oluşmasını engellemek için bu kontrol sağlanmıştır.
- Müdürün giriş yapacağı alanlar boş mu veya boşluk karakteri içeriyor mu?
- Müdürün girdiği 2 yeni şifre birbirleri ile uyuyor mu?
- Müdürün girdiği eski şifre, gerçek şifresi ile uyuyor mu?

Eğer tüm koşullar sağlanırsa kullanıcının şifresi değiştirilecektir, ekranda pop-up mesajı gösterilecektir ve ArrayList içindeki değerler MySQL’e aktarılacaktır.

Müdür çıkış yapmak istediğinde ise “Çıkış Yap” JButton’una basacaktır. JButton’un Mouse Event kodları devreye girerek “AnaEkran” sınıfından yeni bir nesne oluşturacaktır. “AnaEkran” sınıfının yapıcı metodundan oluşacak JFrame’in görünürlüğü true olarak ayarlanıp mevcut JFrame kapatılacaktır ve müdür hesaptan çıkış yaparak Ana Ekranı geri dönmüş olacaktır.

```
JButton buton_cikisYap = new JButton();
buton_cikisYap.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        dispose();
        AnaEkran yeni = new AnaEkran();
        yeni.setVisible(true);
        return;
    }
});
```

## Class UyeOl

Yeni bir kullanıcı üye olmak istediğinde ana ekrandan bu JFrame’e ulaşmalıdır. Karşısına gelecek olan JPanel çeşitli komponentler içermektedir. 6 adet JTextField, Onay verilebilecek 1 adet JCheckBox ve 2 adet JButton bulunmaktadır. Kullanıcı kendi bilgilerini ekrandaki boş alanlara girdikten sonra “Üye Ol” JButton’una basmalıdır. Butona basıldığında JButton’un Mouse Event kodları çalışacaktır. Bu kodlar içerisinde çeşitli kontroller sağlanmaktadır. Bu Kontroller:

- Ad, Soyad, T.C. No, Şifre, Doğum Tarihi, Güvenlik Kodu girişlerine giriş yapılmış mı?
- Ad, Soyad, T.C. No, Şifre, Doğum Tarihi, Güvenlik Kodu girişlerinin metin uzunluğu 253 karakterden fazla mı?
- Şifre 6 karakterden küçük mü
- T.C. No uygun bir formatta girilmiş mi?
- Kayıt koşulları onaylanmış mı?
- Doğrulama kodu doğru girilmiş mi?
- Girilen T.C. No ile mevcut bir kullanıcıya ait mi?

```
if (Ad.isBlank()) {
    JOptionPane.showMessageDialog(null, "Ad Girmediniz !");
    return;
}
else if (Ad.length() > 253) {
    JOptionPane.showMessageDialog(null, "Ad Uzunluğu Çok Fazla. "
        + "Lütfen Sistem Yöneticisi İle İletişime Geçin !");
    return;
}
else if (Soyad.isBlank()) {
    JOptionPane.showMessageDialog(null, "Soyad Girmediniz !");
    return;
}
else if (Soyad.length() > 253) {
    JOptionPane.showMessageDialog(null, "Soyad Uzunluğu Çok Fazla. "
        + "Lütfen Sistem Yöneticisi İle İletişime Geçin !");
    return;
}
else if (Tc.isBlank()) {
    JOptionPane.showMessageDialog(null, "T.C. No Girmediniz !");
    return;
}
else if (Tc.length() != 11) {
    JOptionPane.showMessageDialog(null, "T.C. No Hatalı Girilmiştir!");
    return;
}
else if (Tarih.isBlank()) {
    JOptionPane.showMessageDialog(null, "Doğum Tarihi Girmediniz !");
    return;
}
else if (Tarih.length() > 253) {
    JOptionPane.showMessageDialog(null, "Tarih Uzunluğu Çok Fazla. "
        + "Hatalı Giriş Yaptınız ");
    return;
}
```

Bu kontroller sağlandıktan sonra kullanıcının girmiş olduğu bilgiler overload yapılmış olan yapıcı sınıfa yollanarak ArrayList'in içine eklenecektir. Daha sonrasında ArrayList içindeki veriler MySQL veri tabanına iletilecektir. Bu süreçte herhangi hatalı bir giriş yapılmış ise catch bloğu içerisinde yakalanarak ekrana pop-up mesajı verilecektir.

```
Main.kullanicilar.add(new kullanıcı(TCNo,Ad,Soyad,Tarih,Sifre));
sqlBaglantisi.kullaniciKayit();
JOptionPane.showMessageDialog(null,"Hesabınız Oluşturuldu !");
return;
```

Kullanıcı, hesaptan çıkış yapmak istediğinde ise “Çıkış Yap” JButton’una basacaktır. JButton’un Mouse Event kodları devreye girerek “AnaEkran” sınıfından yeni bir nesne oluşturacaktır. “AnaEkran” sınıfının yapıcı metodundan oluşacak JFrame’in görünürlüğü true olarak ayarlanıp mevcut JFrame kapatılacaktır ve kullanıcı çıkış yaparak Ana Ekrana geri dönmüş olacaktır.

```
button_cikisYap.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        dispose();
        AnaEkran frm = new AnaEkran();
        frm.setVisible(true);
    }
})
```

## Class BelgeDogrula

Sisteme kayıt veya üyelik gerektirmeden herhangi bir şekilde belge doğrulayabilmek için ana ekran JFrame’inden bu class çağırılmalıdır. “BelgeDogrula” JFrame’i başlatıldığında kullanıcının karşısına doğrulama kodunun girileceği 1 adet JTextField ve 2 adet JButton bulunmaktadır. Belge doğrulandığında belge bilgilerinin gösterileceği 1 adet JLabel mevcuttur.



Kullanıcı belgeyi doğrulayabilmek için doğrulama kodunu ekrandaki boş alanına girdikten sonra “Belge Doğrula” JButton’una tıklamalıdır. JButton’un Mouse Event kodları devreye girecektir. Bu kodlar: doğrulama kodu girişi boş mu diye kontrol edilmesi, doğrulama kodunun integer tipinde bir değişkene kaydedilmesi bir for döngüsü ile “Main” sınıfı içindeki “evraklar” ArrayList’inin içindeki evraklardan aranan evrağın bulunması ve “Evrak” sınıfının override edilen toString methodu ile evrak bilgilerinin ekrandaki JLabel’a yazdırılmasıdır. Eğer aranan evrak bulunamaz ise ekrana bir pop-up mesajı verilecektir.

Kullanıcı doğrulama kodunu hatalı bir formatta girmiş ise catch bloğu içerisinde yakalanarak ekrana pop-up mesajı verilecektir.

```
public void mouseClicked(MouseEvent e) {
    if (input_dogrulamaKod.getText().isBlank()) {
        JOptionPane.showMessageDialog(null, "Doğrulama Kodu Girmediniz !");
        return;
    }
    try {
        int tmp = Integer.valueOf(input_dogrulamaKod.getText());
        for (int i = 0; i < Main.evraklar.size(); i++) {
            if (tmp == Main.evraklar.get(i).getDogrulamaKod()) {
                label_evrakBilgiler.setText(Main.evraklar.get(i).toString());
                return;
            }
            if (i == Main.evraklar.size() - 1) {
                JOptionPane.showMessageDialog(null, "Doğrulama Kodu İle Uyuşan Evrak Bulunamamıştır !");
                return;
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, "Doğrulama Kodunu Hatalı Girdiniz !");
        return;
    }
}
```

Kullanıcı çıkış yapmak istediğinde ise “Çıkış Yap” JButton’una basacaktır. JButton’un Mouse Event kodları devreye girerek “AnaEkran” sınıfından yeni bir nesne oluşturacaktır. “AnaEkran” sınıfının yapıcı metodundan oluşacak JFrame’in görünürlüğü true olarak ayarlanıp mevcut JFrame kapatılacaktır ve kullanıcı çıkış yaparak Ana Ekran’a geri dönmüş olacaktır.

```
button_cikisYap.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        dispose();
        AnaEkran frm = new AnaEkran();
        frm.setVisible(true);
    }
});
```

## Class Evrak

Bu sınıf içerisinde evrak özelliklerinin ve metodlarının tanımlandığı sınıftır. Bir Evrak nesnesinin 7 adet özelliği vardır. Bunlar: Seri No, Doğrulama Kodu, Tarih, Tür, İmzalayan, Oluşturan ve Geçerlilik. Tüm değişkenler Encapsulation edilerek private olarak ayarlanmıştır. Programdaki “Evrak” nesneleri bu sınıftan üretilmektedir.

```
public class evrak {
    static private int no = 0;
    private final int seriNo;
    private int dogrulamaKod;
    private final String tarih;
    private String tur;
    private gorevli imzalayan;
    private kullanıcı olusturan;
    private boolean gecerlilik;
```

Seri No 0’den başlayarak 1’er 1’er artan bir değişkendir. Benzersiz bir değişkendir başka hiçbir evrakın seri numarası ile aynı olamamaktadır.

Doğrulama kodu değerini rastgele alan bir değişkendir. Benzersiz bir değişkendir başka hiçbir evrakın doğrulama kodu ile aynı olamamaktadır. Random kütüphanesi kullanılarak rastgele değerini almaktadır.

Tarih değişkeni, evrağın üretildiği tarihi saklamaktadır. Evrak Üretildiğinde LocalDate kütüphanesi kullanılarak, işlemin yapıldığı günün tarihi alınıp tarih değişkeni içerisinde saklanmaktadır

Tür değişkeni, evrak türünün saklandığı String bir değişkendir. Evrak üretilirken kullanıcı tarafından seçilip, yapıcı sınıfa parametre olarak yollanmaktadır.

İmzalayan değişkeni, “Gorevli” sınıfından bir nesneyi tutmaktadır. Evrağı imzalayan kişinin nesnesi yer almaktadır.

Oluşturan değişkeni, “Kullanici” sınıfından bir nesneyi tutmaktadır. Evrağı oluşturan kişinin nesnesi yer almaktadır.

Geçerlilik değişkeni boolean tipinde bir değişkendir. Evrağın geçerlilik durumunu tutmaktadır.

Evrak sınıfının yapıcı sınıfı overload edilmiştir.

Kullanıcı üye olarak kaydolduğunda yapıcı sınıfın içerisine yalnızca 2 parametre gelecektir. Bu parametreler, evrak türü ve evrağı oluşturan kişi parametreleridir. Diğer değişkenler sistem tarafından otomatik olarak atanan veya null olarak kaydedilmektedir.

```
//kullanici constructor
public evrak(String tur, kullanici olusturan) {
    for (int i=0;i<Main.evraklar.size();i++) {
        if (Main.evraklar.get(i).getSeriNo()==no) {
            no++;
            i=0;
        }
    }
    int tmp = rnd.nextInt(10000);
    if (no == 0)
        this.dogrulamaKod=tmp;
    else {
        for (int i =0;i<Main.evraklar.size();i++) {
            if (Main.evraklar.get(i).getDogrulamaKod() == tmp) {
                tmp = rnd.nextInt(10000);
                i=0;
            }
            if (i==Main.evraklar.size()-1) {
                this.dogrulamaKod=tmp;
                break;
            }
        }
    }
    this.tarih = String.valueOf(LocalDate.now());
    this.seriNo =no++;
    this.tur = tur;
    this.olusturan = olusturan;
    this.gecerlilik = true;
    this.imzalayan=null;
}
}
```

Diğer yapıcı sınıf ise veri tabanından gelecek bilgiler doğrultusunda kayıt yapacağı için tüm değişkenler parametre olarak gelmektedir.

```
// veritabanı constructor
public evrak(int seriNo,int dogrulamaKod,String tarih,String tur,boolean gecerlilik) {
    this.seriNo=seriNo;
    this.dogrulamakod=dogrulamaKod;
    this.tarih = tarih;
    this.tur =tur;
    this.gecerlilik = gecerlilik;
    this.olusturan = null;
    this.imzalayan=null;
    no = seriNo+1;
}
}
```

Sınıf içerisinde getter ve setter metotları eklenmiştir. Kullanılmayan getter veya setter metodu yoktur.

```
public int getSeriNo() {  
    return seriNo;  
}  
public void setOlusturan(kullanici olusturan) {  
    this.olusturan = olusturan;  
}  
public int getDogrulamaKod() {  
    return dogrulamaKod;  
}  
public String getTarih() {  
    return tarih;  
}  
  
public String getTur() {  
    return tur;  
}
```

toString methodu override edilmiştir.

```
@Override  
public String toString() {  
    String tmp = "\nSeri Numarası = " + getSeriNo() +  
                "\nDoğrulama Kodu = " + getDogrulamaKod() +  
                "\nTarih = " + getTarih() +  
                "\nTür = " + getTur() +  
                "\nOlusturan TC NO = " + olusturan.getTcNo();  
    if (isGecerlilik())  
        tmp += "\nGeçerlilik = Belge Geçerlidir";  
    else  
        tmp += "\nGeçerlilik = Belge Geçerli Değildir";  
    if (getImzalayan()==null)  
        tmp+="\nİmzalayan = Yok";  
    else  
        tmp+="\nİmzalayan =" + imzalayan.getAd() + " " + imzalayan.getSoyAd();  
    return tmp;  
}
```

## Class Calisan

“Calisan” sınıfı abstract (soyut) bir sınıftır. “Mudur” ve “Gorevli” sınıfları “Calisan” sınıfından miras almıştır. “Calisan” sınıfından herhangi bir nesne oluşturulamamaktadır.

6 özelliği vardır. Bunlar: T.C No, Ad, Soyad, Doğum Tarihi, Maaş ve Şifredir. Tüm özellikler Encapsulation edilerek private olarak ayarlanmıştır. Değişmeyecek olan özellikler final olarak ayarlanmıştır.

```
public abstract class calisan {  
    private final long tcNo;  
    private final String ad;  
    private final String soyAd;  
    private final String dogumTarihi;  
    private double maas;  
    private String sifre;
```

“tcNo” Değişkeni long tipinde sabit bir değişkendir. Çalışanın T.C. kimlik numarası bilgisini tutmaktadır.

“ad” değişkeni String tipinde sabit bir değişkendir. Çalışanın ad bilgisini tutmaktadır.



“sayAd” değişkeni String tipinde sabit bir değişkendir. Çalışanın soy adı bilgisini tutmaktadır.

“dogumTarihi” değişkeni String tipinde sabit bir değişkendir. Çalışanın doğum tarihi bilgisini tutmaktadır.

“maas” değişkeni double tipinde bir değişkendir. Çalışanın maaş bilgisini tutmaktadır.

“sifre” değişkeni String tipinde bir değişkendir. Çalışanın şifre bilgisini tutmaktadır.

“calisan” sınıfının bir adet yapıcı sınıfı vardır ve bu yapıcı sınıf, “calisan” sınıfından miras alan sınıflar tarafından kullanılmaktadır.

```
//constructor
public calisan(long tcNo, String ad, String soyAd, String dogumTarihi, double maas,String sifre) {
    this.tcNo = tcNo;
    this.ad = ad;
    this.soyAd = soyAd;
    this.dogumTarihi = dogumTarihi;
    this.maas = maas;
    this.sifre=sifre;
}
```

Getter ve setter metotları tanımlanmıştır. Kullanılmayan getter veya setter metodu yoktur.

```
public String getSifre() {
    return sifre;
}

public void setSifre(String sifre) {
    this.sifre = sifre;
}

public long getTcNo() {
    return tcNo;
}

public String getAd() {
    return ad;
}

public String getSoyAd() {
    return soyAd;
}
```

toString methodu override edilmiştir.

```
@Override
public String toString() {
    return "\nTC NO: " + getTcNo() + "\nAd: " + getAd() + "\nSoyad: " + getSoyAd() + "\nDoğum Tarihi: " + getDogumTarihi()
        + "\nMaaş: " + getMaas();
}
```

## Class Gorevli

“Gorevli” sınıfı “Calisan” sınıfından miras almıştır. Kendine özgü 3 adet özelliği vardır. Bunlar iş giriş tarihi, iş çıkış tarihi ve çalışma durumudur. Tüm değişkenler Encapsulation edilerek private olarak ayarlanmıştır. Programdaki görevli nesneleri bu sınıftan üretilacaktır.

```
public class gorevli extends calisan {
    private String isGirisTarihi;
    private String isCikisTarihi;
    private boolean calismaDurumu;
```

“isGirisTarihi” değişkeni String tipinde bir değişkendir. Görevlinin işe giriş tarihi bilgisini tutmaktadır.

“isCikisTarihi” değişkeni String tipinde bir değişkendir. Görevlinin işten çıkış tarihi bilgisini tutmaktadır.

“calismaDurumu” değişkeni boolean tipinde bir değişkendir. Görevlinin çalışma durumu bilgisini tutmaktadır.

Yapıcı sınıfı overload edilmiştir. 2 farklı yapıcı sınıfı vardır.

Müdür’ün işe alacağı görevlilerin yapıcı sınıfı aşağıdadır. super komutu ile miras alınan sınıfın yapıcı metodu kullanılacaktır. İş giriş tarihi LocalDate kütüphanesi kullanılarak işlemin yapıldığı tarih olarak ayarlanmaktadır.

```
//Mudur kayıt constructor
public gorevli(long tcNo, String ad, String soyAd, String dogumTarihi, double maas,String sifre) {
    super(tcNo, ad, soyAd, dogumTarihi, maas,sifre);
    this.isGirisTarihi = String.valueOf(LocalDate.now());
    this.calismaDurumu = true;
}
```

Veri tabanı vasıtası ile doğrudan kayıt yapılacağı için tüm bilgiler parametre şeklinde gelecektir. super komutu ile miras alınan sınıfın yapıcı metodu kullanılacaktır.

```
//Veri tabanı kayıt constructor
public gorevli(long tcNo, String ad, String soyAd, String dogumTarihi, double maas,String sifre,
String isGirisTarihi,String isCikisTarihi, boolean calismaDurumu) {
    super(tcNo, ad, soyAd, dogumTarihi, maas,sifre);
    this.isGirisTarihi = isGirisTarihi;
    this.isCikisTarihi = isCikisTarihi;
    this.calismaDurumu = calismaDurumu;
}
```

Getter ve setter metotları tanımlanmıştır. Kullanılmayan getter veya setter metodu yoktur.

```
public String getIsGirisTarihi() {
    return isGirisTarihi;
}

public String getIsCikisTarihi() {
    return isCikisTarihi;
}

public void setIsGirisTarihi(String isGirisTarihi) {
    this.isGirisTarihi = isGirisTarihi;
}

public void setIsCikisTarihi(String isCikisTarihi) {
    this.isCikisTarihi = isCikisTarihi;
}

public boolean isCalismaDurumu() {
    return calismaDurumu;
}
```

toString methodu override edilmiştir. Sınıfa özgü olan özellikler eklendikten sonra super komutu ile calisan sınıfının toString metodundaki bilgiler ile birleşmektedir.

```
@Override
public String toString() {
    String txt = "\nİş Giriş Tarihi: "+getIsGirisTarihi();
    if (isCalismaDurumu()){
        txt+="\nÇalışma Durumu: Aktif";
    }
    else{
        txt+="\nÇalışma Durumu: Pasif";
        txt+="\nİş Çıkış Tarihi: "+getIsCikisTarihi();
    }
    return super.toString()+txt;
}
```

## Class Mudur

“Mudur” sınıfı “Calisan” sınıfından miras almaktadır ve “EvrakIptal” ile “ZamIslem” interface (arayüz) sınıflarını implement etmektedir. “Mudur” sınıfının kendine özgür bir değişkeni yoktur fakat kendine özgü metotları vardır. Bunlar evrak iptal et ve evrak aktif et şeklindedir.

1 adet yapıcı sınıfı vardır. Arayüzden kayıt ve veri tabanından kayıt için aynı yapıcı sınıf kullanılmaktadır.

```
// constructor
public mudur(long tcNo, String ad, String soyAd, String dogumTarihi, double maas, String sifre) {
    super(tcNo, ad, soyAd, dogumTarihi, maas,sifre);
}
```

İmplemente edilen sınıfların metotları override edilmiştir.

```
@Override
public void evrakIptalEt(evrak belge) {
    belge.setGecerlilik(false);
}

@Override
public void evrakAktifEt(evrak belge) {
    belge.setGecerlilik(true);
}
```

toString metodu override edilmiştir.

```
@Override
public String toString() {
    return super.toString();
}
```

## Class EvrakIslem

“EvrakIslem” sınıfı interface (arayüz) olarak tanımlanmıştır. “Mudur” sınıfı tarafından implemente edilmiştir. 2 adet metodu vardır. Bunlar “evrakIptalEt” ve “evrakAktifEt” şeklindedir.

```
public interface evrakIslem{
    void evrakIptalEt(evrak belge);
    void evrakAktifEt(evrak belge);
}
```

## Class Kullanici

“Kullanici” sınıfı herhangi bir sınıftan miras almamıştır. Programdaki kullanıcılar için “Kullanici” sınıfından nesne oluşturulmaktadır.

6 özelliği vardır. Bunlar: T.C No, Ad, Soyad, Doğum Tarihi, Üyelik Tarihi ve Şifredir. Tüm özellikler Encapsulation edilerek private olarak ayarlanmıştır. Değişmeyecek olan özellikler final olarak ayarlanmıştır.

```
public class kullanıcı {
    private final long tcNo;
    private final String ad;
    private final String soyAd;
    private final String dogumTarihi;
    private final String uyelikTarihi;
    private String sifre;
```

“tcNo” değişkeni long tipinde sabit bir değişkendir. Kullanıcının T.C. kimlik numarası bilgisi tutulmaktadır.

“ad” değişkeni String tipinde sabit bir değişkendir. Kullanıcının ad bilgisi tutulmaktadır.

“soyAd” değişkeni String tipinde sabit bir değişkendir. Kullanıcının soyadı bilgisi tutulmaktadır.

“dogumTarihi” değişkeni String tipinde sabit bir değişkendir. Kullanıcının doğum tarihi bilgisi tutulmaktadır.

“uyelikTarihi” değişkeni String tipinde sabit bir değişkendir. Kullanıcının üyelik tarihi bilgisi tutulmaktadır.

“sifre” değişkeni String tipinde sabit bir değişkendir. Kullanıcının şifre bilgisi tutulmaktadır.

Yapıcı sınıfı overload edilmiştir. 2 farklı yapıcı sınıfı vardır.

Kayıt ekranı kullanılarak yapılacak olan kayıta üyelik tarihi parametre olarak alınmayıp, LocalDate kütüphanesi kullanılarak işlemin yapıldığı tarih bilgisi kaydedilmektedir.

```
//Kayıt ekranı constructor
public kullanıcı(long tcNo, String ad, String soyAd, String dogumTarihi, String sifre) {
    this.tcNo = tcNo;
    this.ad = ad;
    this.soyAd = soyAd;
    this.dogumTarihi = dogumTarihi;
    this.uyelikTarihi = String.valueOf(LocalDate.now());
    this.sifre = sifre;
}
```

Veri tabanı kullanılarak yapılacak olan kayıtlarda ise tüm parametreler doğrudan kaydedilmektedir.

```
// veritabanı constructor
public kullanıcı(long tcNo, String ad, String soyAd, String dogumTarihi, String sifre, String uyelikTarihi) {
    this.tcNo = tcNo;
    this.ad = ad;
    this.soyAd = soyAd;
    this.dogumTarihi = dogumTarihi;
    this.uyelikTarihi = uyelikTarihi;
    this.sifre = sifre;
}
```

Getter ve setter metotları tanımlanmıştır. Kullanılmayan getter veya setter metodu yoktur.

```
public String getSifre() {
    return sifre;
}
public void setSifre(String sifre) {
    this.sifre = sifre;
}
public long getTcNo() {
    return tcNo;
}
public String getAd() {
    return ad;
}
```

toString metodu override edilmiştir.

```
@Override
public String toString() {
    return "\nTC NO: " + getTcNo() + "\nAd: " + getAd() + "\nSoyad: " + getSoyAd() + "\nDoğum Tarihi: " +
        getDogumTarihi() + "\nÜyelik Tarihi: " + getUyelikTarihi();
}
```

## Class SQLBaglantisi

Bu sınıf içerisinde MySQL ile bağlantı kurabilmek için gerekli olan bilgilerin ve metotların bulunduğu sınıftır. Veri tabanı kullanıcı adı “root” ve şifresi “123456” olarak ayarlanmıştır.

```
public class SQLBaglantisi {
    static String DB_URL = "jdbc:mysql://localhost:3306/evrakotomasyon";
    static String USER = "root";
    static String PASS = "123456";
    static Connection con = null;
```

Sıfır kurulum özelliği için gerekli SQL sorguları tanımlanmıştır. Eğer program içinde kullanılacak olan veri tabanı veya tabloları mevcut değil ise hepsi oluşturulacaktır.

```
public static void sifirKurulum() {
    try {
        con = DriverManager.getConnection("jdbc:mysql://localhost:3306/", USER, PASS);
        PreparedStatement dbOlustur = con.prepareStatement("CREATE DATABASE IF NOT EXISTS `evrakotomasyon`");
        PreparedStatement dbKullan = con.prepareStatement("USE `evrakotomasyon`");
        PreparedStatement dbTablo1 = con.prepareStatement("CREATE TABLE IF NOT EXISTS `evrak` (\r\n"
            + " `seriNo` int NOT NULL,\r\n"
            + " `dogrulamaKod` int NOT NULL,\r\n"
            + " `tarih` varchar(255) NOT NULL,\r\n"
            + " `tur` varchar(255) NOT NULL,\r\n"
            + " `imzalayan` varchar(255) CHARACTER SET utf8 COLLATE utf8_general_ci DEFAULT NULL,\r\n"
            + " `olusturan` varchar(255) NOT NULL,\r\n"
            + " `gecerlilik` tinyint(1) NOT NULL,\r\n"
            + " UNIQUE KEY `dogrulamaKod` (`dogrulamaKod`),\r\n"
            + " UNIQUE KEY `seriNo` (`seriNo`)\r\n"
            + ") ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;");
        PreparedStatement dbTablo2 = con.prepareStatement("CREATE TABLE IF NOT EXISTS `gorevli` (\r\n"
            + " `tcNo` varchar(255) NOT NULL,\r\n"
            + " `ad` varchar(255) NOT NULL,\r\n"
            + " `soyAd` varchar(255) NOT NULL,\r\n"
            + " `dogumTarihi` varchar(255) NOT NULL,\r\n"
            + " `maas` double NOT NULL,\r\n"
            + " `sifre` varchar(255) NOT NULL,\r\n"
            + " `isGirisTarihi` varchar(255) NOT NULL,\r\n"
            + " `isCikisTarihi` varchar(255) DEFAULT NULL,\r\n"
            + " `calismaDurumu` tinyint(1) DEFAULT NULL,\r\n"
            + " UNIQUE KEY `tcNo` (`tcNo`)\r\n"
            + ") ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;");
```

```
dbOlustur.execute();
dbKullan.execute();
dbTablo1.execute();
dbTablo2.execute();
dbTablo3.execute();
dbTablo4.execute();
```

Program ilk başlatıldığında veri tabanındaki kullanıcı bilgilerinin “Main” sınıfı içerisindeki “kullanıcılar” ArrayList’ine kaydedilebilmesi için gerekli olan kullanıcıCek metodu oluşturulmuştur.

```

public static void kullaniciCek() {
    try {
        con = DriverManager.getConnection(DB_URL,USER,PASS);
        java.sql.Statement st = con.createStatement();
        ResultSet data = st.executeQuery("SELECT * FROM kullanici");
        Long tcNo;
        String ad;
        String soyAd;
        String dogumTarihi;
        String sifre;
        String uyelikTarihi;
        while (data.next()) {
            tcNo=Long.valueOf(data.getString(1));
            ad = data.getString(2);
            soyAd = data.getString(3);
            dogumTarihi= data.getString(4);
            sifre = data.getString(5);
            uyelikTarihi= data.getString(6);
            Main.kullaniciIcar.add(new kullanici(tcNo,ad,soyAd,dogumTarihi,sifre,uyelikTarihi));
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
        System.out.println(e.getMessage());
    }
}

```

Program ilk başlatıldığında veri tabanındaki müdür bilgilerinin “Main” sınıfı içerisindeki “mudurler” ArrayList’ine kaydedilebilmesi için gerekli olan mudurCek metodu oluşturulmuştur.

```

public static void mudurCek() {
    try {
        con = DriverManager.getConnection(DB_URL,USER,PASS);
        java.sql.Statement st = con.createStatement();
        ResultSet data = st.executeQuery("SELECT * FROM mudur");
        Long tcNo;
        String ad;
        String soyAd;
        String dogumTarihi;
        double maas;
        String sifre;
        while (data.next()) {
            tcNo=Long.valueOf(data.getString(1));
            ad = data.getString(2);
            soyAd = data.getString(3);
            dogumTarihi= data.getString(4);
            maas = data.getDouble(5);
            sifre = data.getString(6);
            Main.mudurler.add(new mudur(tcNo,ad,soyAd,dogumTarihi,maas,sifre));
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
        System.out.println(e.getMessage());
    }
}

```

Program ilk başlatıldığında veri tabanındaki görevli bilgilerinin “Main” sınıfı içerisindeki “gorevliIcar” ArrayList’ine kaydedilebilmesi için gerekli olan gorevliCek metodu oluşturulmuştur.

```

public static void gorevliCek() {
    try {
        con = DriverManager.getConnection(DB_URL,USER,PASS);
        java.sql.Statement st = con.createStatement();
        ResultSet data = st.executeQuery("SELECT * FROM gorevli");
        Long tcNo;
        String ad;
        String soyAd;
        String dogumTarihi;
        double maas;
        String sifre;
        String isGirisTarihi;
        String isCikisTarihi;
        boolean calismaDurumu;
        while (data.next()) {
            tcNo=Long.valueOf(data.getString(1));
            ad = data.getString(2);
            soyAd = data.getString(3);
            dogumTarihi= data.getString(4);
            maas = data.getDouble(5);
            sifre = data.getString(6);
            isGirisTarihi = data.getString(7);
            isCikisTarihi = data.getString(8);
            calismaDurumu = data.getBoolean(9);
            Main.gorevliEkle(new gorevli(tcNo,ad,soyAd,dogumTarihi,
                maas,sifre,isGirisTarihi,isCikisTarihi,calismaDurumu));
        }
    }
    catch(SQLException e) {
        e.printStackTrace();
        System.out.println(e.getMessage());
    }
}

```

Program ilk başlatıldığında veri tabanındaki evrak bilgilerinin “Main” sınıfı içerisindeki “evraklar” ArrayList’ine kaydedilebilmesi için gerekli olan evrakCek metodu oluşturulmuştur. Burada özel bir algoritma kullanılmıştır. “Evrak” sınıfındaki “olusturan” ve “imzalayan” özellikleri sınıflara ait nesneler oldukları için ve bu nesneler veri tabanına doğrudan kaydedilemeyeceği için o nesnelerin benzersiz tanımlayıcı özelliği olan T.C. No bilgisi veri tabanına kaydedilmiştir ve veri tabanından tekrar okuma esnasında programdaki ArrayList’lere ilk önce kullanıcılar sonra görevliler sonra müdürler en son ise evraklar kaydedilmiştir. Evraklar kaydedilirken, oluşturan veya imzalayan kişinin T.C. No’su “Main” sınıfının içindeki ArrayList’lerden kontrol edilerek yapıcı sınıfın içine direk o kişinin bilgisi gönderilmektedir.

```

public static void evrakCek() {
    try {
        con = DriverManager.getConnection(DB_URL,USER,PASS);
        java.sql.Statement st = con.createStatement();
        ResultSet data = st.executeQuery("SELECT * FROM `evrak` ORDER BY `seriNo` ASC");
        int seriNo;
        int dogrulamaKod;
        String tarih;
        String tur;
        boolean gecerlilik;
        while (data.next()) {
            long imzalayan = -1;
            long olusturan = -1;
            int index = -1;
            seriNo=data.getInt(1);
            dogrulamaKod=data.getInt(2);
            tarih = data.getString(3);
            tur = data.getString(4);
            if (data.getString(5)!=null) {
                imzalayan = Long.parseLong(data.getString(5));
            }
            olusturan = Long.parseLong( data.getString(6));
            gecerlilik = data.getBoolean(7);
            Main.evraklar.add(new evrak(seriNo,dogrulamaKod,tarih,tur,gecerlilik));

            int n = -1;
            for (int i=0;i<Main.evraklar.size();i++){
                if (seriNo==Main.evraklar.get(i).getSeriNo()){
                    n = i;
                }
            }
        }
    }
}

```



```

if (imzalayan!=-1) {
for (int i=0;i<Main.gorevliiler.size();i++) {
    if (imzalayan== Main.gorevliiler.get(i).getTcNo()) {
        index = i;
        break;
    }
    if (i==Main.gorevliiler.size()-1) {
        index = -1;
        break;
    }
}
if (index!=-1) {
    Main.evraklar.get(n).setImzalayan(Main.gorevliiler.get(index));
}
}

for (int i=0;i<Main.kullaniciilar.size();i++) {
    if (olusturan== Main.kullaniciilar.get(i).getTcNo()) {
        index = i;
        break;
    }
}
Main.evraklar.get(n).setOlusturan(Main.kullaniciilar.get(index));
}

```

Program içinde “Main” sınıfındaki “evraklar” ArrayList’i üzerinde herhangi bir değişiklik yapıldığında evrakKayit() fonksiyonu çalıştırılarak veri tabanındaki evrak bilgileri temizlenip en baştan ArrayList’in içindeki tüm değerler kaydedilmektedir.

```

public static void evrakKayit() {
    String sql = "INSERT INTO `evrak` (`seriNo`, `dogrulamaKod`, `tarih`, "
        + "`tur`, `imzalayan`, `olusturan`, `gecerlilik`) "
        + "VALUES (?, ?, ?, ?, ?, ?, ?)";
    String sql2 = "DELETE FROM evrak";
    try {
        PreparedStatement temizle = con.prepareStatement(sql2);
        temizle.execute();
        for (int i=0;i<Main.evraklar.size();i++) {
            PreparedStatement kayit = con.prepareStatement(sql);
            kayit.setInt(1,Main.evraklar.get(i).getSeriNo());
            kayit.setInt(2,Main.evraklar.get(i).getDogrulamaKod());
            kayit.setString(3,Main.evraklar.get(i).getTarih());
            kayit.setString(4,Main.evraklar.get(i).getTur());
            if (Main.evraklar.get(i).getImzalayan()!=null)
                kayit.setString(5,String.valueOf( Main.evraklar.get(i).getImzalayan().getTcNo()));
            else
                kayit.setNull(5, 0);
            kayit.setString(6,String.valueOf( Main.evraklar.get(i).getOlusturan().getTcNo()));
            kayit.setBoolean(7,Main.evraklar.get(i).isGecerlilik());
            kayit.executeUpdate();
        }
    } catch (SQLException ee) {
        ee.printStackTrace();
    }
}

```

Program içinde “Main” sınıfındaki “gorevliiler” ArrayList’i üzerinde herhangi bir değişiklik yapıldığında gorevliKayit() fonksiyonu çalıştırılarak veri tabanındaki görevli bilgileri temizlenip en baştan ArrayList’in içindeki tüm değerler kaydedilmektedir.

```

public static void gorevliKayit() {
    String sql = "INSERT INTO `gorevli` (`tcNo`, `ad`, `soyAd`, `dogumTarihi`,`"
        + "`maas`, `sifre`, `isGirisTarihi`, "
        + "`isCikisTarihi`, `calismaDurumu`) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
    String sql2= "DELETE FROM gorevli";
    try {
        PreparedStatement temizle = con.prepareStatement(sql2);
        temizle.execute();
        for (int i=0;i<Main.gorevliler.size();i++) {
            PreparedStatement kayit = con.prepareStatement(sql);
            kayit.setString(1,String.valueOf( Main.gorevliler.get(i).getTcNo()));
            kayit.setString(2,Main.gorevliler.get(i).getAd());
            kayit.setString(3,Main.gorevliler.get(i).getSoyAd());
            kayit.setString(4,Main.gorevliler.get(i).getDogumTarihi());
            kayit.setDouble(5,( Main.gorevliler.get(i).getMaas()));
            kayit.setString(6,Main.gorevliler.get(i).getSifre());
            kayit.setString(7,Main.gorevliler.get(i).getIsGirisTarihi());
            kayit.setString(8,Main.gorevliler.get(i).getIsCikisTarihi());
            kayit.setBoolean(9,Main.gorevliler.get(i).isCalismaDurumu());
            kayit.executeUpdate();
        }
    }catch(SQLException ee) {
        ee.printStackTrace();
    }
}

```

Program içinde “Main” sınıfındaki “mudurler” ArrayList’i üzerinde herhangi bir değişiklik yapıldığında mudurKayit() fonksiyonu çalıştırılarak veri tabanındaki müdür bilgileri temizlenip en baştan ArrayList’in içindeki tüm değerler kaydedilmektedir.

```

public static void mudurKayit() {
    String sql = "INSERT INTO `mudur` (`tcNo`, `ad`, `soyAd`, `dogumTarihi`,`"
        + "`maas`, `sifre`) VALUES (?, ?, ?, ?, ?, ?)";
    String sql2= "DELETE FROM mudur";
    try {
        PreparedStatement temizle = con.prepareStatement(sql2);
        temizle.execute();
        for (int i=0;i<Main.mudurler.size();i++) {
            PreparedStatement kayit = con.prepareStatement(sql);
            kayit.setString(1,String.valueOf( Main.mudurler.get(i).getTcNo()));
            kayit.setString(2,Main.mudurler.get(i).getAd());
            kayit.setString(3,Main.mudurler.get(i).getSoyAd());
            kayit.setString(4,Main.mudurler.get(i).getDogumTarihi());
            kayit.setDouble(5,( Main.mudurler.get(i).getMaas()));
            kayit.setString(6,Main.mudurler.get(i).getSifre());
            kayit.executeUpdate();
        }
    }catch(SQLException ee) {
        ee.printStackTrace();
    }
}

```

Program içinde “Main” sınıfındaki “kullanıcılar” ArrayList’i üzerinde herhangi bir değişiklik yapıldığında kullanıcıKayit() fonksiyonu çalıştırılarak veri tabanındaki kullanıcı bilgileri temizlenip en baştan ArrayList’in içindeki tüm değerler kaydedilmektedir.

```

public static void kullanicikayit() {
    String sql = "INSERT INTO `kullanici` (`tcNo`, `ad`, `soyAd`, "
        + "`dogumTarihi`, `sifre`, `uyelikTarihi`) VALUES (?, ?, ?, ?, ?, ?);";
    String sql2 = "DELETE FROM kullanici";
    try {
        PreparedStatement temizle = con.prepareStatement(sql2);
        temizle.execute();
        for (int i=0; i<Main.kullanicilar.size(); i++) {
            PreparedStatement kayit = con.prepareStatement(sql);
            kayit.setString(1, String.valueOf(Main.kullanicilar.get(i).getTcNo()));
            kayit.setString(2, Main.kullanicilar.get(i).getAd());
            kayit.setString(3, Main.kullanicilar.get(i).getSoyAd());
            kayit.setString(4, Main.kullanicilar.get(i).getDogumTarihi());
            kayit.setString(5, Main.kullanicilar.get(i).getSifre());
            kayit.setString(6, Main.kullanicilar.get(i).getUyelikTarihi());
            kayit.executeUpdate();
        }
    } catch (SQLException ee) {
        ee.printStackTrace();
    }
}

```

Veri tabanına ait örnek görüntüler aşağıda görüntülenebilir.

seriNo	dogrulamaKod	tarih	tur	imzalayan	olusturan	gecerlilik
0	4626	2022-12-25	Nüfus	12068132160	10397188502	1
2	5337	2022-12-25	Gelir	12345678911	10397188502	0
3	7436	2022-12-25	İkametgah	12068132160	10397188502	0
4	6139	2022-12-25	Ceza	12345678911	10397188502	1
5	1328	2022-12-25	Gelir	12068132160	10397188502	1
7	2900	2022-12-25	Taşıt	12068132160	12345678916	1
8	7207	2022-12-25	Gelir	NULL	12345678916	1
9	6614	2022-12-26	Ceza	NULL	10397188502	1

ad	soyAd	dogumTarihi	maas	sifre	isGirisTarihi	isCikisTarihi	calismaDurumu
Mehmet	Akif	2007-05-04	6500	789866	2022-12-25	NULL	1
Hakan	Tunç	2002-04-10	6000	123456	2022-12-25	NULL	1
Yasin	Bilekçi	2002-03-04	6000	197059	2022-12-25	2022-12-26	0
Akif	Avcu	2004-07-10	6800	498843	2022-12-30	NULL	1

## Class İlkKurulum

İlk kurulum özelliğinin aktif edilmesi durumunda “Main” içinden oluşturulacak olan “İlkKurulum” nesnesi ile “İlkKurulum” sınıfının yapıcı metotları çalışacaktır ve JFrame’in görünürlüğü true olarak ayarlanacaktır. İlk kurulum ekranından, programa yapılacak olan ilk müdür kaydı gerçekleştirilebilmektedir. Bunun için kullanıcının karşısına gelecek olan JPanel’de bir pop-up mesajı belirecektir. Bu pop-up mesajında tüm bilgilerin doğru girilmesi ve şifrenin en az 6 karakter olması gerektiği belirtilmektedir.

Evrak Otomasyon

İlk Kurulum Ekranına Hoş Geldiniz.  
Lütfen Müdür Kaydını Yapınız

Ad:  Maaş:

Soyad:

T.C. No:

Şifre:

Doğum Tarihi:

Güvenlik Kodu:  3Z5RJ

☐ Üyelik Koşullarını Kabul Ediyor, Onaylıyorum.

**Üye Ol**

**EREN EVRAK**

İlk müdür kaydının yapılabilmesi için ekrandaki giriş kutucuklarının uygun bir şekilde bilgiler ile doldurulması gerekmektedir. Bu bilgiler doldurulduktan sonra “Üye Ol” JButton’una tıklanmalıdır. JButton’a tıklandığında butonun Mouse Event kodları devreye girecektir. Bu kodlar içerisinde çeşitli kontroller sağlanmaktadır. Bu Kontroller:

- Ad, Soyad, T.C. No, Şifre, Maaş, Doğum Tarihi, Güvenlik Kodu girişlerine giriş yapılmış mı?
- Ad, Soyad, T.C. No, Şifre, Doğum Tarihi, Güvenlik Kodu girişlerinin metin uzunluğu 253 karakterden fazla mı?
- Maaş 10 karakterden fazla mı?
- Maaş bilgisi içerisinde String var mı?
- Şifre 6 karakterden küçük mü?
- T.C. No uygun bir formatta girilmiş mi?
- T.C. No bilgisi içerisinde String var mı?
- Kayıt koşulları onaylanmış mı?
- Güvenlik kodu doğru girilmiş mi?
- Girilen T.C. No ile mevcut bir kullanıcıya ait mi?

```
if (Ad.isBlank()) {
    JOptionPane.showMessageDialog(null, "Ad Girmediniz !");
    return;
}
else if (Ad.length() > 253) {
    JOptionPane.showMessageDialog(null, "Ad Uzunluğu Çok Fazla. "
        + "Lütfen Sistem Yöneticisi İle İletişime Geçin !");
    return;
}
else if (Soyad.isBlank()) {
    JOptionPane.showMessageDialog(null, "Soyad Girmediniz !");
    return;
}
else if (Soyad.length() > 253) {
    JOptionPane.showMessageDialog(null, "Soyad Uzunluğu Çok Fazla. "
        + "Lütfen Sistem Yöneticisi İle İletişime Geçin !");
    return;
}
else if (Tc.isBlank()) {
    JOptionPane.showMessageDialog(null, "T.C. No Girmediniz !");
    return;
}
else if (Tc.length() != 11) {
    JOptionPane.showMessageDialog(null, "T.C. No Hatalı Girilmiştir!");
}
```

Bu kontroller sağlandıktan sonra “Mudur” sınıfından yeni bir nesne oluşturularak “Main” sınıfının içindeki “mudurler” ArrayList’ine eklenmektedir ve daha sonrasında ArrayList içindeki bilgiler MySQL veri tabanına aktarılmaktadır ve ilk kurulum tamamlandığı için dizin içerisindeki ilkKurulum.txt dosyası silinip “AnaEkran” sınıfının “Main” metodu çalıştırılarak ana ekrana ulaşılmaktadır ve mevcut JFrame Kapatılmaktadır.

