VEDAT ARSLAN
B181210030
2.ÖGRETİM-A GRUBU
vedat.arslan@ogr.sakarya.edu.tr

SEFA AKTÜRK
G201210046
2.ÖĞRETİM-A GRUBU
sefa.akturk1@ogr.sakarya.edu.tr

ÖGRETİM GÖREVLİSİ: CELAL CEKEN

NESNE YÖNELİMLİ ANALİZ VE TASARIM
1.PROJE
AKILLI SOGUTUCU CİHAZ UYGULAMASI

OPEN/CLOSED İLKESİ VE UYGULANMASI:

Bu ilke projemizdeki nesnelerin geliştirmeye açık ama değişime kapalı olmaları anlamına gelmekte-dir. Oluşturduğunuz nesneler zaman içerisinde ek özellikler kazanabilir genişlemeye açık olurlar bu normal bir yazılım projesinde kaçınılmaz bir durumdur. Ama temel nesne değişime kapalı tutul-malıdır. Bu projede sıcaklığı getir fonksiyonu yazılırken open/closed ilkesine göre yazılmıştır.İlerleyen dönemlerde projeye sıcaklık ile ilgili yeni eklemeler yapılabilmektedir. Ancak varolan fonksiyonlar yeni eklemelerden etkilenmemektedir.

OBSERVER TASARIM DESENİ VE UYGULANMASI:

Nesneler arasında one-to-many ilişki sağlar. Bir nesne durumunu değiştirdiğinde, ona bağlı diğer tüm nesneler uyarılır ve otomatik olarak güncellenir. Projemizde soğutucu aç/kapa işlemlerinde bu tasarım ilkesi kullanılmıştır. Soğutucu açma işlemi sırasında sıcaklığa bağlı olan tüm fonksiyonlarda sıcaklık değişir.

KULLANIM DURUMLARI:

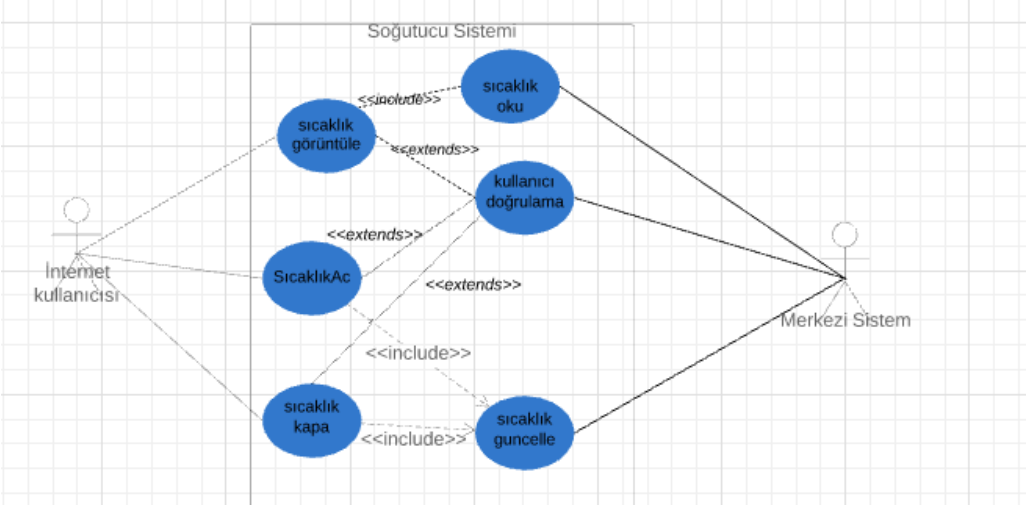
Kullanım Adı: Sıcaklık Göster
Aktör: İnternet Kullanıcısı
Giriş Koşulu: Müşteri internet sitesine girer. Kullanıcı doğrulanır.
Çıkış koşulu : Müşteri sıcaklık değerini okur. Başka herhangi bir işlem yapmaz.
Ana Olay Akışı:
1. Kullanıcı sisteme giriş yapar.
2. Arayüz merkezi sisteme doğrulama gönderir.
3. Merkezi sistem kullanıcıyı doğrular.
4. Arayüz kullanıcıya işlem seçeneklerini sunar ve yapmak istediği işlemi ister.
5. Kullanıcı sıcaklık göster işlemini girer.
6. Arayüz merkezi sisteme işlem doğrulaması gönderir.
7. Merkezi sistem işlemi doğrular ve sıcaklık algılayıcıya işlemi gönderir.
8. Sıcaklık algılayıcı sıcaklığı okur ve merkezi sisteme gönderir.
9. Merkezi sistem sıcaklığı alır, arayüze iletir.
10. Arayüz sıcaklığı gösterir. Kullanıcıya başka işlem yapıp yapılmayacağı sorulur.
11. Kullanıcı başka işlem yapmaz ise çıkış sağlanır.
Alternatif Olay Akışı:
A1. Kullanıcı doğrulanamaz. Üç kez yanlış girişte çıkılır.
A2. İşlem doğrulanamaz. Tekrar işlem alınır.
Özel Gereksinimler: UI gereksinimleri, İşlem gecikmesi en fazla 1 dk. olmalı, 24 saat çalışmalı

Kullanım Adı: Soğutucu Aç
Aktör: İnternet Kullanıcısı
Giriş Koşulu: Müşteri internet sitesine girer. Kullanıcı doğrulanır.
Çıkış koşulu : Müşteri soğutucuyu açar. Başka herhangi bir işlem yapmaz.

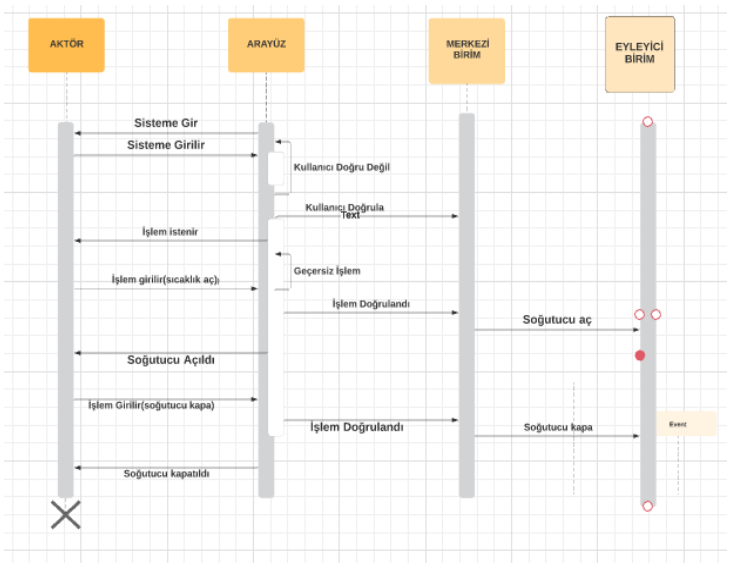
Ana Olay Akışı:
1. Kullanıcı sisteme giriş yapar.
2. Arayüz merkezi sisteme doğrulama gönderir.
3. Merkezi sistem kullanıcıyı doğrular.
4. Arayüz kullanıcıya işlem seçeneklerini sunar ve yapmak istediği işlemi ister.
5. Kullanıcı soğutucu aç işlemini girer.
6. Arayüz merkezi sisteme işlem doğrulaması gönderir.
7. Merkezi sistem işlemi doğrular ve Eyleyici birime işlemi gönderir.
8. Eyleyici soğutucu açar ve işlemin tamamlandığını merkezi sisteme gönderir.
9. Merkezi sistem soğutucu açıldı bildirisini arayüze iletir.
10. Arayüz soğutucu açıldı bildirisini gösterir. Kullanıcıya başka işlem yapıp yapılmayacağı sorulur.
11. Kullanıcı başka işlem yapmaz ise çıkış sağlanır.
Alternatif Olay Akışı:
A1. Kullanıcı doğrulanamaz. Üç kez yanlış girişte çıkılır.
A2. İşlem doğrulanamaz. Tekrar işlem alınır.
Özel Gereksinimler: UI gereksinimleri, İşlem gecikmesi en fazla 1 dk. olmalı, 24 saat çalışmalı

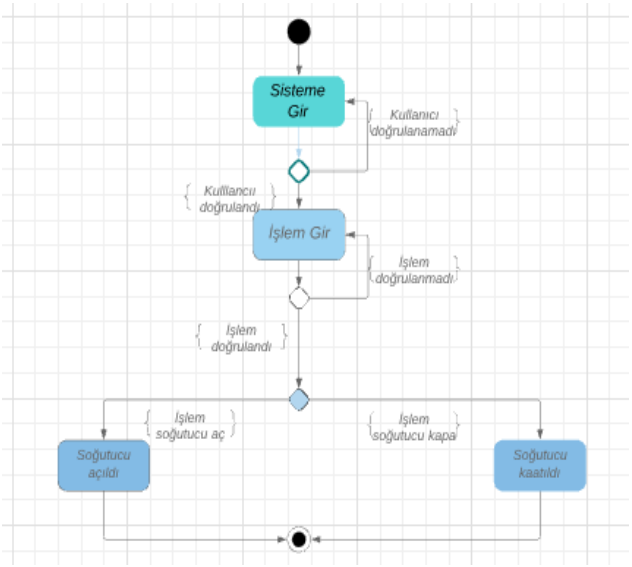
Kullanım Adı: Soğutucu Kapatmak
Aktör: İnternet Kullanıcısı
Giriş Koşulu: Müşteri internet sitesine girer. Kullanıcı doğrulanır.
Çıkış koşulu : Müşteri soğutucuyu kapar. Başka herhangi bir işlem yapmaz.
Ana Olay Akışı:
1. Kullanıcı sisteme giriş yapar.
2. Arayüz merkezi sisteme doğrulama gönderir.
3. Merkezi sistem kullanıcıyı doğrular.
4. Arayüz kullanıcıya işlem seçeneklerini sunar ve yapmak istediği işlemi ister.
5. Kullanıcı soğutucu kapa işlemini girer.
6. Arayüz merkezi sisteme işlem doğrulaması gönderir.
7. Merkezi sistem işlemi doğrular ve Eyleyici birime işlemi gönderir.
8. Eyleyici soğutucu kapar ve işlemin tamamlandığını merkezi sisteme gönderir.
9. Merkezi sistem soğutucu kapandı bildirisini arayüze iletir.
10. Arayüz soğutucu kapandı bildirisini gösterir. Kullanıcıya başka işlem yapıp yapılmayacağı sorulur.
11. Kullanıcı başka işlem yapmaz ise çıkış sağlanır.
Alternatif Olay Akışı:
A1. Kullanıcı doğrulanamaz. Üç kez yanlış girişte çıkılır.
A2. İşlem doğrulanamaz. Tekrar işlem alınır.
Özel Gereksinimler: UI gereksinimleri, İşlem gecikmesi en fazla 1 dk. olmalı, 24 saat çalışmalı

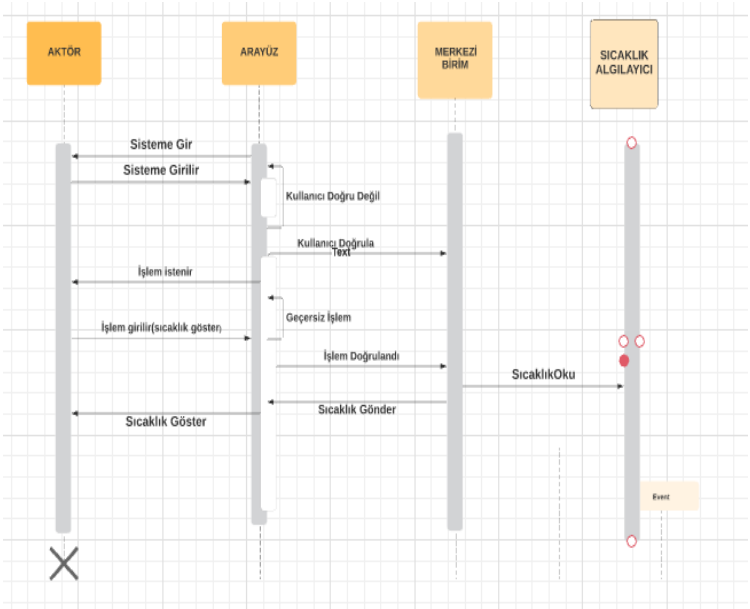
KULLANICI DURUMU USE CASE

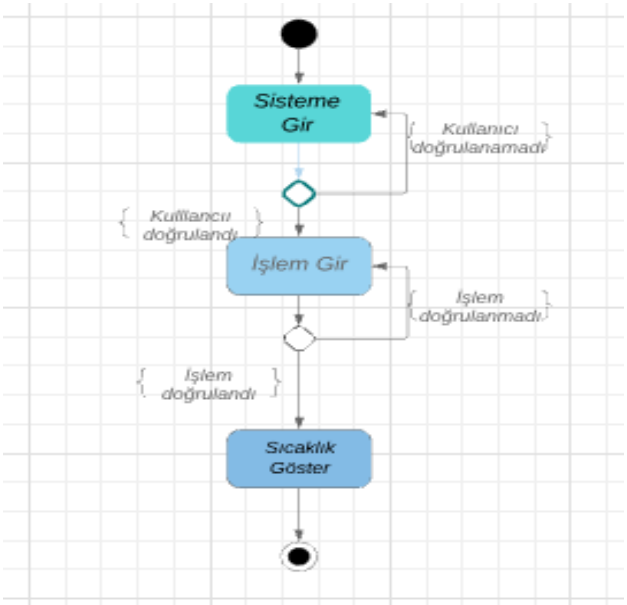


Sıcaklık Oku ve Soğutucu Aç/Kapa durumlarına ait sequence ve activity şemaları

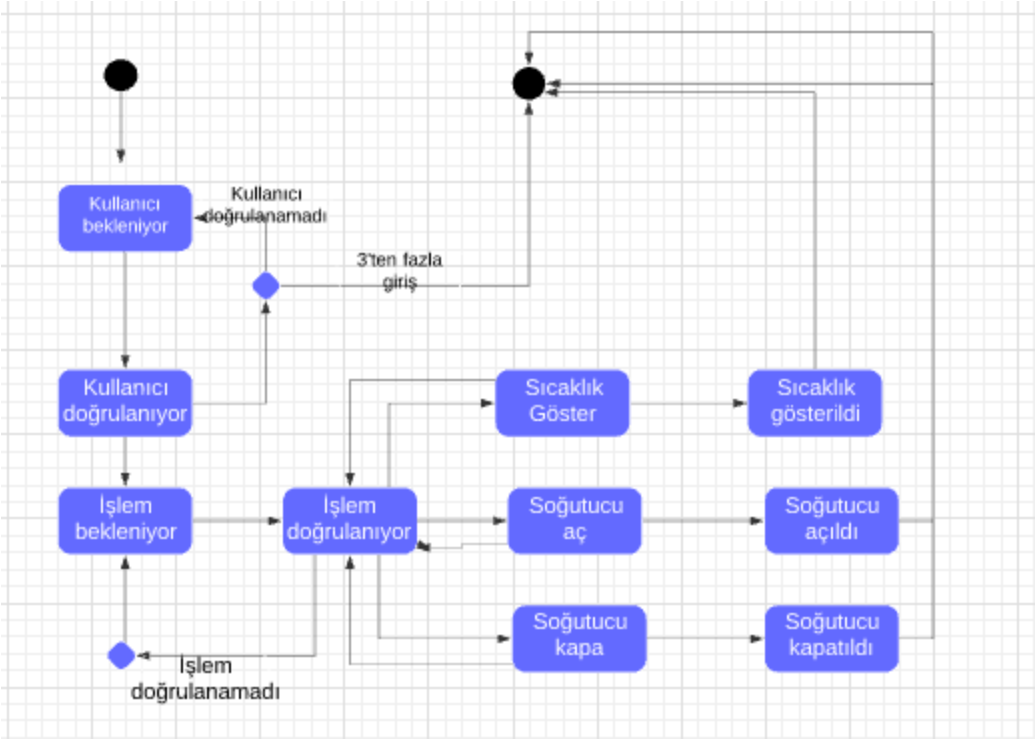




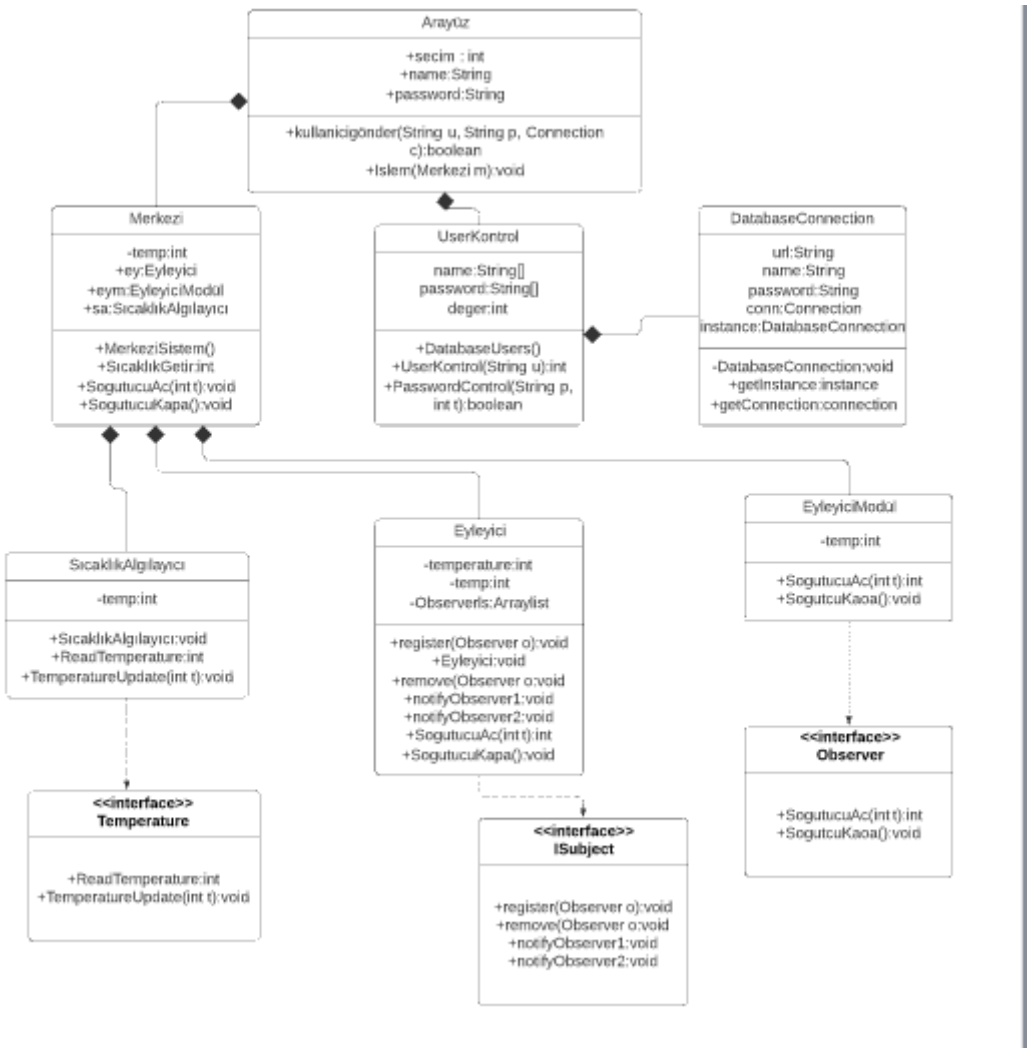




Sistem Durum Diyagramı



Sistem Sınıf Şeması



A)Kullanıcı Doğrulama Ekranı ve Açıklaması:

Program çalıştırıldığında ilk olarak bizden “Ekran” sınıfından kontrol edilen kullanıcı adı (username) ve şifre (password) bilgileri isteniyor ve veri tabanındaki bilgiler ile kıyaslanıyor. Bu bilgiler kullanıcıdan alınırken eğer doğru alındıysa seçenekler kullanıcıya gösteriliyor. Kayda alınan değerler değişkenlere atanıyor. Eğer bilgiler yanlış ise kullanıcı doğrulama ekranına tekrar yönlendiriliyor.

1)Başarılı – Başarısız Giriş Denemsi:

```
Kullanıcı Adınızı Giriniz:
vedat
Şifrenizi Giriniz:
1234
Veritabanına bağlanıldı.
[HATA] Kullanıcı adınız veya şifreniz hatalı, tekrar deneyiniz...
Kullanıcı Adınızı Giriniz:
sefa
Şifrenizi Giriniz:
123
Veritabanına bağlanıldı.
Giriş yapıldı! Menüye yönlendiriliyor...


********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
```

Eğer kullanıcı adımı veya parolamı doğru yazarsam program üstteki gibi kullanıcının kullanacağı seçenekleri ekrana getiriyor.

2)Giriş Yaparken Kullandığım Veritabanı Kodumun Ekran Çıktısı:

```java
public class VeritabaniBaglanti implements IVeritabani {
    Connection conn;
    @Override
    public boolean baglan() {
        try {
            conn = DriverManager.getConnection( url: "jdbc:postgresql://localhost:5432/users", user: "postgres", password: "151048");
            if (conn != null) {
                System.out.println("Veritabanına bağlanıldı.");
                return true;
            }
            else {
                System.out.println("Sistem veritabanına zaten bağlı durumda!");
                return false;
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
        return false;
    }
```

```java
    @Override
    public boolean kullaniciDogrula(String kullaniciAdi, String sifre) {
        try {
            baglan();

            String sql= "SELECT *  FROM \"users\" WHERE \"username\"='"+ kullaniciAdi + "' AND \"password\"='"+ sifre + "';";

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);

            String kAdi;
            String sf;

            while(rs.next()) {
                kAdi = rs.getString("username");
                sf = rs.getString("password");

                if(kAdi.equals(kullaniciAdi) && sf.equals(sifre)) {
                    return true;
                }
            }

            rs.close();
            stmt.close();

            baglantiSonlandir();

            return false;
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
        return false;
    }
```

Bağlantıyı yapabilmek için ilk önce postgreSQL driverını kurup projemin içine oluşturduğum postgreJar klasörü içerisinden dahil ettim. Daha sonra gerekli olan kodlarla kendi dataBase'imden(users) verileri çekmek için şifrem (123) ile bağlandım.

B) Sıcaklığın Görüntülenmesi ve Soğutucunun Açılıp Kapatılmasıyla ilgili Ekran Görüntüleri ve Açıklaması:

Ekran sınıfımın içerisinde bulunan menuYazdır metodunu kullanarak kullanıcıya gerekli olan ekran çıktılarını gösteriyorum. Ve kullanıcıdan 1 ile 4 arasında bir seçim yapmasını istiyorum. 1)Sıcaklık görüntüle 2)Soğutucuyu aç 3)Soğutucuyu kapat 4)Sistemden çıkış yap.

Eğer soğutucu açıkken veya kapalıyken kullanıcı aynı işlemleri tekrarlamak isterse program kullanıcıyı soğutucu zaten açık veya kapalı şeklinde uyarıyor. Bunu yaparken Eyleyici sınıfından metot çağırıyorum. Extra olarak programıma eğer sıcaklık belli bir değerin üzerindeyse otomatik soğutucu açılacak şekilde bir update ekledim.

1)Kullanıcıya Gösterilen Seçenekler:

```
********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
```

2)Ekran Sınıfının Kodu:

```
13      public void menuYazdir() {
14          System.out.println("\n********** ANA MENÜ **********");
15          System.out.println("1- Sıcaklık Görüntüle");
16          System.out.println("2- Soğutucuyu Aç");
17          System.out.println("3- Soğutucuyu Kapat");
18          System.out.println("4- Sistemden Çıkış Yap");
19          System.out.println("******************************");
20      }
21  }
```

3)Soğutucu Zaten Açık mı? Kapalı mı? :

```
********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
2
Soğutucu açılıyor...
Soğutucu açıldı!
********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
2
Soğutucu zaten aktif durumda!
********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
3
Soğutucu kapatılıyor...
Soğutucu kapatıldı!
********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
3
Soğutucu zaten kapalı durumda!
```

4)Sıcaklık Görüntüleme:

```
********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
1
Anlık sıcaklık: 19,4 °C

********** ANA MENÜ **********
1- Sıcaklık Görüntüle
2- Soğutucuyu Aç
3- Soğutucuyu Kapat
4- Sistemden Çıkış Yap
******************************
```

SıcaklıkAlgilayici sınıfımın içerisinde bulunan javanın random kütüphanesi ile random bir sıcaklık değeri getiriyorum.

```java
import java.text.DecimalFormat;
import java.util.Random;

public class SicaklikAlgilayici implements ISicaklikAlgilayici {

    @Override
    public String sicaklikOku() {
        Random r = new Random();
        DecimalFormat formatter = new DecimalFormat("#0.0");

        double sicaklik = 18 + 18 * r.nextDouble();

        return formatter.format(sicaklik) + " °C";
    }
}
```

C)Veritabanında Kullanıcı Verilerinin Saklandığı Tablonun Görüntüsü ve Onun SQL Kodları:

"Users" adlı bir veritabanı oluşturuldu. Daha sonra SQL kodları ile tablo oluşturuldu ve 2 tane doğru giriş örneği verildi.

1)Tablonun Görüntüsü:

	kullaniciNo [PK] integer	username text	password text
1	1	vedat	123
2	2	sefa	123

2)SQL Kodları:

```
Create table "users"(
"kullaniciNo" Serial,
"username" Varchar(80),
"password" Varchar(80),
    Constraint "usersPK" PRIMARY KEY("kullaniciNo")
);

insert into "users"
("kullaniciNo","username","password")
Values
('0001','vedat',123);
insert into "users"
("kullaniciNo","username","password")
Values
('0002','sefa',123);
```

D)”Dependency Inversion” ilkesinin ne olduğu ve uygulamada nasıl kullanıldığı:

1)Dependency Inversion ilkesi nedir ?:

Üst sınıfların alt sınıflara olan bağımlılığını araya bir soyutlama katman ekleyerek kaldırmaktır. Bu soyutlama işlemi ise, üst sınıfların alt sınıfların kendilerini değil de ara yüzlerini (Interface) kullanarak gerçekleşir. Bu sayede alt sınıfta yapılan herhangi bir değişiklik üst sınıfları etkilemez. Böylece üst sınıfların alt sınıflara olan bağımlılığı ortadan kalkar. Direk olarak alt sınıfı kullanmazlar.

2)Dependency Inversion ilkesini uygulamada kullanımı:

```java
package smartDevice;

import java.text.DecimalFormat;
import java.util.Random;

public class SicaklikAlgilayici implements ISicaklikAlgilayici {

    @Override
    public String sicaklikOku() {
        Random r = new Random();
        DecimalFormat formatter = new DecimalFormat("#0.0");

        double sicaklik = 18 + 18 * r.nextDouble();

        return formatter.format(sicaklik) + " °C";
    }
}
```

```java
package smartDevice;

public interface ISicaklikAlgilayici {
    public String sicaklikOku();
}
```

```java
package smartDevice;

public interface IEyleyici {
    public void sogutucuAc();
    public void sogutucuKapat();
}

```

```java
package smartDevice;

public class Eyleyici implements IEyleyici, IObserver {
    public class IEyleyici {

    }

    private boolean sogutucuDurumu = false;

    @Override
    public void sogutucuAc() {
        if (!sogutucuDurumu) {
            System.out.println("Soğutucu açılıyor...");

            sogutucuDurumu = true;

            System.out.printf("Soğutucu açıldı!");
        } else {
            System.out.printf("Soğutucu zaten aktif durumda!");
        }
    }

    @Override
    public void sogutucuKapat() {
        if (sogutucuDurumu) {
            System.out.println("Soğutucu kapatılıyor...");

            sogutucuDurumu = false;

            System.out.printf("Soğutucu kapatıldı!");
        } else {
            System.out.println("Soğutucu zaten kapalı durumda!");
        }
    }
```

2)Observer nesnesi nedir ?:

Bu desenin amacı, çok sayıda nesneye gözlemledikleri nesnede meydana gelen olayları bildirmektedir. Bir nesnenin değişikliğinden farklı nesneler etkilenecekse kullanılır.

```
1  package smartDevice;
2
3  public class AgArayuzu implements IAgArayuzu, IObserver {
4      @Override
5      public void sogucutuAc(IEyleyici eyleyici) {
6          eyleyici.sogutucuAc();
7      }
8
9      @Override
10     public void sogutucuKapat(IEyleyici eyleyici) {
11         eyleyici.sogutucuKapat();
12     }
13
14     @Override
15     public String sicaklikGonder(ISicaklikAlgilayici sicaklikAlgilayici) {
16         return sicaklikAlgilayici.sicaklikOku();
17     }
18
19     @Override
20     public void update(String mesaj, String derece) {
21         VeritabaniIslem veritabaniIslem = new VeritabaniIslem(new VeritabaniBaglanti());
22         veritabaniIslem.yuksekSicaklikLogla(derece);
23     }
24 }
```

F)Uygulamanın Kaynak Kodları:

1)AgArayuzu.java

```
package smartDevice;

public class AgArayuzu implements IAgArayuzu, IObserver {
    @Override
    public void sogucutuAc(IEyleyici eyleyici) {
        eyleyici.sogutucuAc();
    }

    @Override
    public void sogutucuKapat(IEyleyici eyleyici) {
        eyleyici.sogutucuKapat();
    }

    @Override
    public String sicaklikGonder(ISicaklikAlgilayici
sicaklikAlgilayici) {
        return sicaklikAlgilayici.sicaklikOku();
    }

    @Override
    public void update(String mesaj, String derece) {
        VeritabaniIslem veritabaniIslem = new
VeritabaniIslem(new VeritabaniBaglanti());
        veritabaniIslem.yuksekSicaklikLogla(derece);
    }
     }
```

2)AkilliCihaz.java

```
package smartDevice;

import java.time.format.DateTimeFormatter;

public class AkilliCihaz {
```

```
    private String cihazIsmi;
    private String cihazSahibininAdı;
    private String cihazZamanDilimi;
    private int cihazOtoUykuModuSaat;

    private IEkran ekran;
    private ITusTakimi tusTakimi;

    private IEyleyici eyleyici;
    private ISicaklikAlgilayici sicaklikAlgilayici;
    private IAgArayuzu agArayuzu;

    private VeritabaniIslem veritabaniIslem;

    private Publisher publisher;

    private static final int SICAKLIK_GORUNTULE = 1;
    private static final int SOGUTUCUYU_AC = 2;
    private static final int SOGUTUCUYU_KAPAT = 3;
    private static final int CIKIS = 4;

    private AkilliCihaz(AkilliCihazBuilder builder) {
        ekran = new Ekran();
        tusTakimi = new TusTakimi();

        eyleyici = new Eyleyici();
        sicaklikAlgilayici = new SicaklikAlgilayici();
        agArayuzu = new AgArayuzu();

        veritabaniIslem = new VeritabaniIslem(new
VeritabaniBaglanti());

        publisher = new Publisher();

        this.cihazIsmi = builder.cihazIsmi;
        this.cihazSahibininAdı =
builder.cihazSahibininAdı;
        this.cihazZamanDilimi = builder.cihazZamanDilimi;
        this.cihazOtoUykuModuSaat =
builder.cihazOtoUykuModuSaat;
    }
```

```
    public void baslat() {

        // Bildirim gidecek modüller ayarlanıyor...
        publisher.attach((IObserver) agArayuzu);
        publisher.attach((IObserver) eyleyici);

        boolean girisYapilmadiMi = true;
        do {
            ekran.mesajGoruntule("Kullanıcı Adınızı
Giriniz: ");
            String kullaniciAdi =
tusTakimi.stringVeriAl();

            ekran.mesajGoruntule("Şifrenizi Giriniz: ");
            String sifre = tusTakimi.stringVeriAl();


if(veritabaniIslem.kullaniciDogrula(kullaniciAdi, sifre))
{
                ekran.mesajGoruntule("Giriş yapıldı!
Menüye yönlendiriliyor... \n");
                girisYapilmadiMi = false;
            } else {
                ekran.hataMesajiGoruntule("Kullanıcı
adınız veya şifreniz hatalı, tekrar deneyiniz...");
            }
        } while(girisYapilmadiMi);

        menuSecimleri();
    }

    public void menuSecimleri() {
        int secim;
        do {
            ekran.menuYazdir();
            secim = tusTakimi.intVeriAl();
            switch (secim) {
                case SICAKLIK_GORUNTULE:
                    String gelenSicaklik =
agArayuzu.sicaklikGonder(sicaklikAlgilayici);
                    ekran.mesajGoruntule("Anlık sıcaklık:
" + gelenSicaklik);
```

```
if(Integer.parseInt(gelenSicaklik.substring(0, 2)) >= 30)
                        publisher.notify("Fazla sıcaklık
uyarısı", gelenSicaklik);
                    break;
                case SOGUTUCUYU_AC:
                    agArayuzu.sogucutuAc(eyleyici);
                    break;
                case SOGUTUCUYU_KAPAT:
                    agArayuzu.sogutucuKapat(eyleyici);
                    break;
                case CIKIS:
                    ekran.mesajGoruntule("Çıkış
yapılıyor...");
                    break;
                default:
                    ekran.hataMesajiGoruntule("1-4
arasında bir seçenek girmelisiniz");
                    break;
            }
        } while(secim != CIKIS);
    }

    public static class AkilliCihazBuilder {
        private String cihazIsmi;
        private String cihazSahibininAdı;
        private String cihazZamanDilimi;
        private int cihazOtoUykuModuSaat;

        public AkilliCihazBuilder(String cihazIsmi) {
            this.cihazIsmi = cihazIsmi;
        }

        public AkilliCihazBuilder sahibininAdı(String
cihazSahibininAdı) {
            this.cihazSahibininAdı = cihazSahibininAdı;
            return this;
        }

        public AkilliCihazBuilder ZamanDilimi(String
cihazZamanDilimi) {
```

```
            this.cihazZamanDilimi =                 ;
            return this;
        }

        public AkilliCihazBuilder OtoUykuModuSaat(int
cihazOtoUykuModuSaat) {
            this.cihazOtoUykuModuSaat =
cihazOtoUykuModuSaat;
            return this;
        }

        public AkilliCihaz build() {
            return new AkilliCihaz(this);
        }
    }
     }
```

3)Ekran.java

```
package smartDevice;

public class Ekran implements IEkran {
    @Override
    public void mesajGoruntule(String mesaj) {
        System.out.println(mesaj);
    }

    public void hataMesajiGoruntule(String mesaj) {
        System.out.println("[HATA] " + mesaj);
    }

    public void menuYazdir() {
        System.out.println("\n********** ANA MENÜ
**********");
        System.out.println("1- Sıcaklık Görüntüle");
        System.out.println("2- Soğutucuyu Aç");
        System.out.println("3- Soğutucuyu Kapat");
        System.out.println("4- Sistemden Çıkış Yap");

System.out.println("*****************************");
    }
     }
```

4)Eyleyici.java

```
package smartDevice;

public class Eyleyici implements IEyleyici, IObserver {
    public class IEyleyici {

     }

     private boolean sogutucuDurumu = false;

    @Override
    public void sogutucuAc() {
        if (!sogutucuDurumu) {
            System.out.println("Soğutucu açılıyor...");

            sogutucuDurumu = true;

            System.out.printf("Soğutucu açıldı!");
        } else {
            System.out.printf("Soğutucu zaten aktif
durumda!");
        }
    }

    @Override
    public void sogutucuKapat() {
        if (sogutucuDurumu) {
            System.out.println("Soğutucu kapatılıyor...");

            sogutucuDurumu = false;

            System.out.printf("Soğutucu kapatıldı!");
        } else {
            System.out.println("Soğutucu zaten kapalı
durumda!");
        }
    }

    @Override
    public void update(String mesaj, String derece) {
        if (!sogutucuDurumu) {
```

```
            System.out.println(mesaj + " , (" + derece +
") soğutucu otomatik olarak açık duruma getiriliyor");
            sogutucuAc();
        }

    }
}
```

5)IAgArayuzu.java

```
package smartDevice;

public interface IAgArayuzu {
    public void sogucutuAc(IEyleyici eyleyici);
    public void sogutucuKapat(IEyleyici eyleyici);

    public String sicaklikGonder(ISicaklikAlgilayici
sicaklikAlgilayici);
}
```

6)IEkran.java

```
package smartDevice;

public interface IEkran {
    public void mesajGoruntule(String mesaj);
    public void hataMesajiGoruntule(String mesaj);
    public void menuYazdir();
}
```

7)IEyleyici.java

```
package smartDevice;

public interface IEyleyici {
    public void sogutucuAc();
    public void sogutucuKapat();
}
```

8)IObserver.java

```java
package smartDevice;

public interface IObserver {
    public void update(String m, String d);
}
```

9)ISicaklikAlgilayici.java

```java
package smartDevice;

public interface ISicaklikAlgilayici {
    public String sicaklikOku();
}
```

10)ISubject.java

```java
package smartDevice;

public interface ISubject {
    public void attach(IObserver o);
    public void detach(IObserver o);
    public void notify(String mesaj, String derece);
}
```

11)ITusTakimi.java

```java
package smartDevice;

import java.util.Scanner;

public interface ITusTakimi {
    public String stringVeriAl();
    public int intVeriAl();
}
```

12)IVeritabani.java

```java
package smartDevice;

import java.sql.ResultSet;
```

```
public interface IVeritabani {
    public boolean baglan();
    public boolean kullaniciDogrula(String kullaniciAdi,
String sifre);
    public void yuksekSicaklikLogla(String sicaklik);
    public void baglantiSonlandir();
}
```

13)kullanici.java

```
package smartDevice;

public class kullanici {

    public class KullaniciBuilder {

     }

     private int id;
    private String isim;
    private String sifre;
    private int yas;

    public kullanici(Builder builder){
        this.id = builder.id;
        this.isim = builder.isim;
        this.sifre = builder.sifre;
        this.yas = builder.yas;
    }

    public int getId() {
        return id;
    }

    public String getIsim() {
        return isim;
    }

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

```
    public int getYas() {
        return yas;
    }

    @Override
    public String toString() {
        return "Kullanici {" +
                " isim = " + isim +
                " , yaş = " + yas +
                '}';
    }

    public static class Builder  {
        private int id;
        private String isim;
        private String sifre;
        private int yas;

        public Builder(String isim, String sifre){
            this.isim = isim;
            this.sifre = sifre;
        }

        public Builder setId(int id){
            this.id = id;
            return this;
        }

        public Builder setYas(int yas) {
            this.yas = yas;
            return this;
        }

        public kullanici build(){
            return new kullanici(this);
        }
    }
}
```

14)Main.java

```
package smartDevice;
```

```
public class Main {

    public static void main(String[] args) {

        AkilliCihaz akilliCihaz = new
AkilliCihaz.AkilliCihazBuilder("Akıllı Cihazım")
                .sahibininAdı("Burak")
                .ZamanDilimi("GMT+3")
                .OtoUykuModuSaat(12)
                .build();

        akilliCihaz.baslat();
    }
}
```

15)Publisher.java

```
package smartDevice;

import java.util.ArrayList;
import java.util.List;

public class Publisher implements ISubject {

    private List<IObserver> subscribers = new
ArrayList<>();

    @Override
    public void attach(IObserver subscriber) {
        subscribers.add(subscriber);
    }

    @Override
    public void detach(IObserver subscriber) {
        subscribers.remove(subscriber);
    }

    @Override
    public void notify(String mesaj, String derece) {
        for(IObserver subscriber: subscribers) {
            subscriber.update(mesaj, derece);
```

```
        }
    }
}
```

16)SicaklikAlgilayici.java

```
package smartDevice;

import java.text.DecimalFormat;
import java.util.Random;

public class SicaklikAlgilayici implements
ISicaklikAlgilayici {

    @Override
    public String sicaklikOku() {
        Random r = new Random();
        DecimalFormat formatter = new
DecimalFormat("#0.0");

        double sicaklik = 18 + 18 * r.nextDouble();

        return formatter.format(sicaklik) + " °C";
    }
}
```

17)TusTakimi.java

```
package smartDevice;

import java.util.Scanner;

public class TusTakimi implements ITusTakimi {

    @Override
    public String stringVeriAl() {
        Scanner in = new Scanner(System.in);
        return in.nextLine();
    }

    @Override
    public int intVeriAl() {
```

```
        Scanner in = new Scanner(System.in);
        return in.nextInt();
    }
}
```

18)VeritabaniBaglantisi.java

```
package smartDevice;


import java.sql.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;


public class VeritabaniBaglanti implements IVeritabani {


   Connection conn;


   @Override
   public boolean baglan() {
      try {
         conn =
DriverManager.getConnection("jdbc:postgresql://localhost:5432/burakxx",
"postgres", "123");
         if (conn != null) {
            System.out.println("Veritabanına bağlanıldı.");
            return true;
         }
         else {
            System.out.println("Sistem veritabanına zaten bağlı
durumda!");
```

```
                return false;
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
        return false;
    }

    @Override
    public boolean kullaniciDogrula(String kullaniciAdi, String sifre) {
        try {
            baglan();

            String sql= "SELECT *  FROM \"users\" WHERE \"username\"='"+ kullaniciAdi + "' AND \"password\"='"+ sifre + "';";

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery(sql);

            String kAdi;
            String sf;

            while(rs.next()) {
                kAdi = rs.getString("username");
                sf = rs.getString("password");

                if(kAdi.equals(kullaniciAdi) && sf.equals(sifre)) {
```

```
                return true;
            }
        }

        rs.close();
        stmt.close();

        baglantiSonlandir();

        return false;
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
    return false;
}

@Override
public void yuksekSicaklikLogla(String sicaklik) {
    try {
        baglan();

        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-
MM-dd HH:mm:ss");
        LocalDateTime tarih = LocalDateTime.now();

        String sql= "INSERT INTO LOGLAR (sicaklik, tarih) VALUES (\'" +
sicaklik + "\',\'" + dtf.format(tarih) + "\');";
```

```
            Statement stmt = conn.createStatement();
            stmt.executeUpdate(sql);

            conn.close();

            stmt.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }

    @Override
    public void baglantiSonlandir() {
        try {
            if (conn != null) {
                conn.close();
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

19)VeritaniIslem.java

```
package smartDevice;

public class VeritabaniIslem {
```

```
    private IVeritabani veritabani;

    public VeritabaniIslem(IVeritabani veritabani) {
        this.veritabani = veritabani;
    }

    public boolean kullaniciDogrula(String kullaniciAdi,
String sifre) {
        return  veritabani.kullaniciDogrula(kullaniciAdi,
sifre);
    }

    public void yuksekSicaklikLogla(String sicaklik) {
        veritabani.yuksekSicaklikLogla(sicaklik);
    }
}
```

: