T.C. SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

NESNE YÖNELİMLİ ANALİZ VE TASARIM DERSİ

PROJE ÇALIŞMASI

NESNELERİN İNTERNETİ SİSTEMLERİ İÇİN AKILLI CİHAZ TASARIMI

Hazırlayan

ADEM YILMAZ

G191210305

2.ÖĞRETİM B GRUBU

adem.yilmaz10@ogr.sakarya.edu.tr

Kaynak Kod Link: https://github.com/ademylmz34/Projem

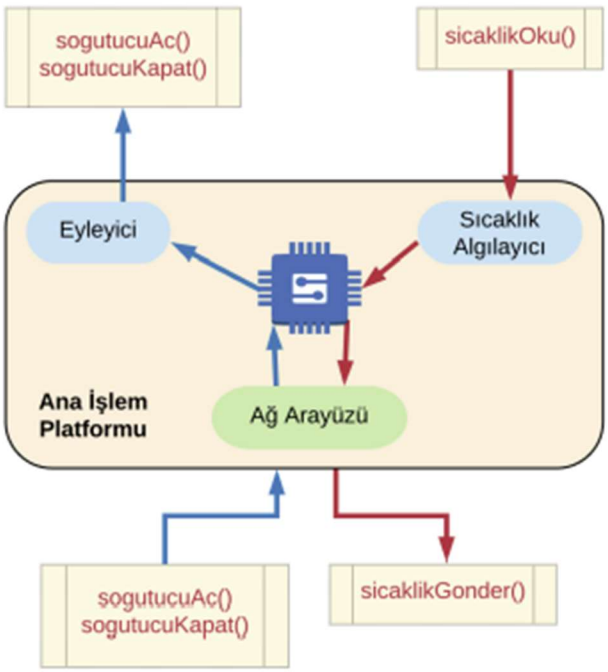
PROJE TANITIMI

Açıklama

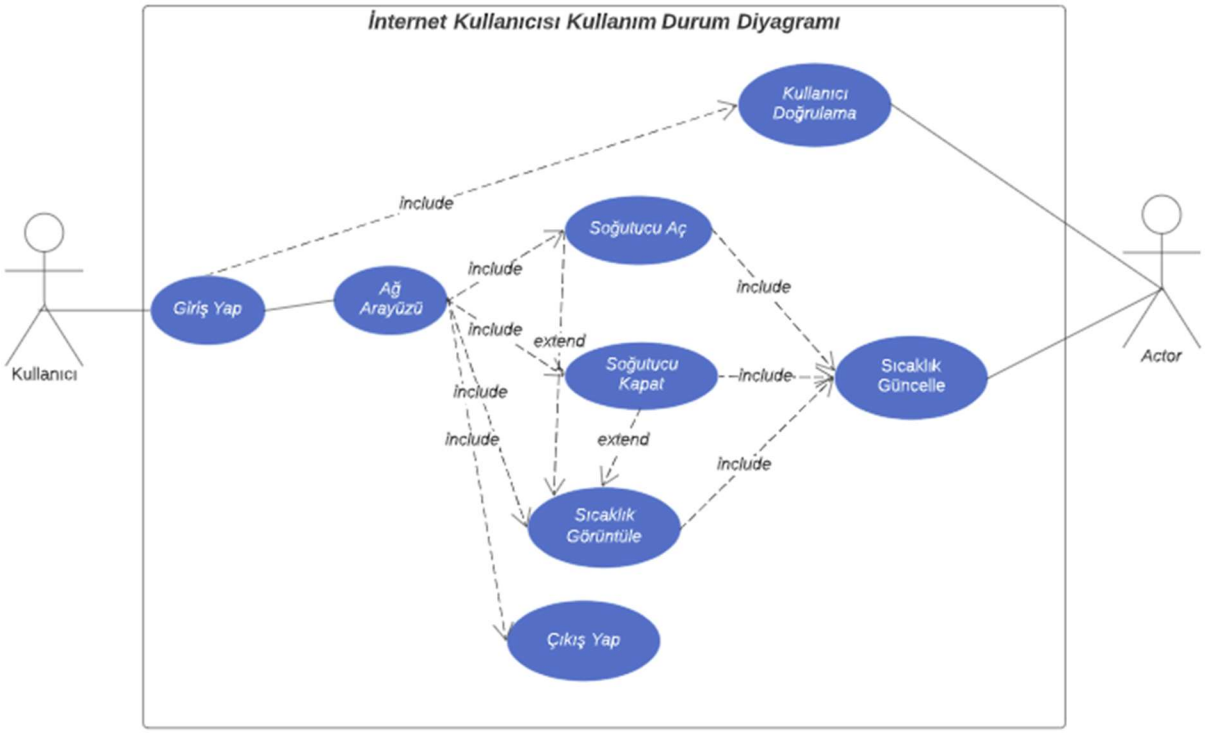
Elimizde, elektromekanik sistemi kullanılabilir durumda olan, fakat denetleyicisi çalışmayan bir soğutucu bulunmaktadır. Bu soğutucuyu internet üzerinden kontrol etmek üzere, aşağıda ana hatları verilen bir akıllı cihaz üretilmiştir. Çalışma kapsamında, bu cihaz için geliştirilmesi beklenen yazılımın; analizini, tasarımını ve gerçeklemesini yapmanız beklenmektedir.

Akıllı Cihazın Tanıtımı

- **Akıllı cihaz**, şekilde görüldüğü gibi, ana işlem platformu ile çevresel birimlerden (eyleyici, sıcaklık algılayıcısı ve ağara yüzü) oluşmaktadır.
- **Ana işlem platformu** geliştirilecek yazılımın çalıştırılacağı birimdir. Çevre birimleri ile şekilde verilen ara yüzler üzerinden haberleşmektedir.
- **Sıcaklık algılayıcı** modül ortam sıcaklığını ölçer.
- **Eyleyici modül** soğutucunun açılması ve kapatılması işlemlerini yerine getirir.
- İnternet kullanıcıları **ağ arayüzünü** kullanarak; sıcaklık görüntüleme, soğutucuyu açma ve soğutucuyu kapatma işlemlerini yerine getirebilirler.
- **Akıllı cihaz**; kapalı, açılış testi yapılıyor, bekleme, algılama, servis dışı ve işlem yapılıyor gibi durumlara sahiptir.



İnternet Kullanıcısı için Kullanım(Use-Case) Diyagramı



Kullanım Durumu: Sıcaklığın Görüntülenmesi

Hazırlayan: Adem Yılmaz

Sürüm: v1.0

Tarih: 19/04/2022

İlgili Aktörler: İnternet Kullanıcısı, Soğutucu Bilgi Sistemi

Giriş Koşulu:

- Kullanıcı sisteme giriş yapmış olmalıdır.
- Cihaz açık ve kullanılır durumda olmalıdır.

Çıkış Koşulu: Kullanıcı işlemi tamamlanmıştır.

Ana Olay Akışı(Ana Senaryo Başarılı)

1. Soğutucu sistemi, ekrana kullanıcıdan kullanıcı adını girmesini isteyen bir mesaj yazdırır.
2. Kullanıcı, tuş takımını kullanarak kullanıcı adını girer.
3. Kullanıcı adı doğrulaması için Soğutucu Bilgi Sistemine istek gönderilir.
4. Sistem isteği kabul eder ve ekrana kullanıcıdan şifre girilmesini isteyen mesaj gönderilir.
5. Kullanıcı, tuş takımını kullanarak şifreyi girer.
6. Kullanıcı doğrulaması için soğutucu sistemine istek gönderilir.
7. Soğutucu sistemi erişim isteğini kabul eder ve arayüze erişim sağlanır.
8. Kullanıcı, arayüzden tuş takımı yardımıyla sıcaklık görünteleme seçeneğini seçer.
9. Soğutucu Sistemi, sıcaklık algılayıcısından sıcaklık değerini alır ve ekrana sıcaklık değeri yazdırılır.

Alternatif Olay Akışı:

A. Kullanıcı Adı doğrulanmadı.
   - 3 den az kez yanlış ise yeniden gir.
   - 3 kez yanlış girilmiş ise işlemi sonlandır ve çıkış yap.

B. Yanlış şifre girildi.
   - 3 den az kez yanlış ise yeniden gir.
   - 3 kez yanlış şifre girilmiş ise işlemi sonlandır ve çıkış yap.

Özel Gereksinimler:

- Sıcaklık Algılayıcı çalışır durumda olmalıdır.

Kullanım Durumu: Soğutucunun Çalıştırılması

Hazırlayan: Adem Yılmaz

Sürüm: v1.0

Tarih: 19/04/2022

İlgili Aktörler: İnternet Kullanıcısı, Soğutucu Bilgi Sistemi

Giriş Koşulu:

- Kullanıcı sisteme giriş yapmış olmalıdır.
- Cihaz açık ve kullanılır durumda olmalıdır.

Çıkış Koşulu: Kullanıcı işlemi tamamlanmıştır.

Ana Olay Akışı(Ana Senaryo Başarılı)

1. Soğutucu sistemi, ekrana kullanıcıdan kullanıcı adını girmesini isteyen bir mesaj yazdırır.
2. Kullanıcı, tuş takımını kullanarak kullanıcı adını girer.
3. Kullanıcı adı doğrulaması için Soğutucu Bilgi Sistemine istek gönderilir.
4. Sistem isteği kabul eder ve ekrana kullanıcıdan şifre girilmesini isteyen mesaj gönderilir.
5. Kullanıcı, tuş takımını kullanarak şifreyi girer.
6. Kullanıcı doğrulaması için soğutucu sistemine istek gönderilir.
7. Soğutucu sistemi erişim isteğini kabul eder ve arayüze erişim sağlanır.
8. Kullanıcı, arayüzden tuş takımı yardımıyla soğukluk çalıştırma seçeneklerinden (açma veya kapama) birini seçer.
9. Soğutucu Sistemi, eyleyici çağırı ve gerekli olan işlemi(soğutucuyu açma veya kapama) yapmasını ister.
10. Soğutucu sistemi seçilen işleme göre soğutucuyu devreye sokar veya devre dışı bırakır.
11. Soğutucu sistemi, sıcaklık algılayıcısını çağırır.Sıcaklık algılayıcısı, soğutucu devrede ise sıcaklığı düşürür veya soğutucu devre dışı ise sıcaklığı oda sıcaklık düzeyine getirir.
12. Soğutucu açma işlemi seçilmiş ise, ekrana soğutucu devrede yazdırılır.
13. Soğutucu kapama işlemi seçilmiş ise, ekrana soğutucu devredışı yazdırılır.
14. Ortalama sıcaklık değeri ekrana yazdırılır.

Alternatif Olay Akışı:

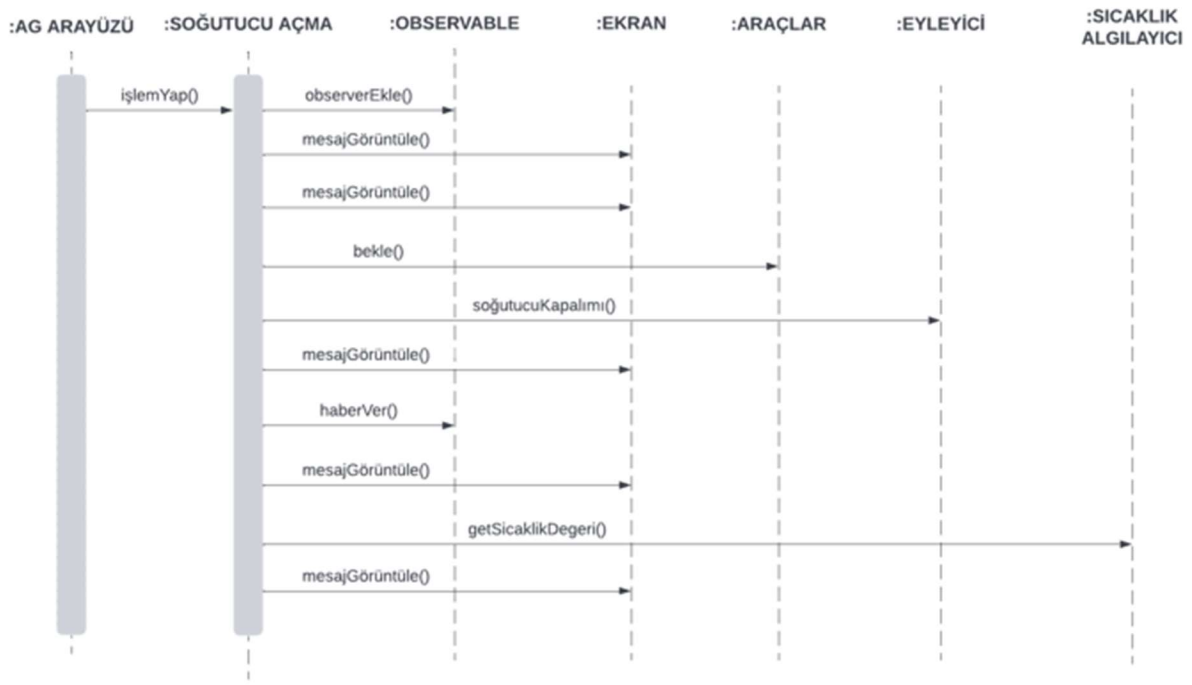
- **A.** Kullanıcı Adı doğrulanmadı.
  - 3 den az kez yanlış ise yeniden gir.
  - 3 kez yanlış girilmiş ise işlemi sonlandır ve çıkış yap.
- **B.** Yanlış şifre girildi.
  - 3 den az kez yanlış ise yeniden gir.
  - 3 kez yanlış şifre girilmiş ise işlemi sonlandır ve çıkış yap.
- **C.** Soğutucu Açık(Soğutucu Açma İşlemi)
  - Ekrana soğutucunun açık durumda olduğu ve sıcaklık değeri yazdırılır.
  - İşlem sonlandırılır.
- **D.** Soğutucu Kapalı(Soğutucu Kapama İşlemi)
  - Ekrana soğutucunun kapalı durumda olduğu ve sıcaklık değeri yazdırılır.
  - İşlem sonlandırılır.

Özel Gereksinimler:

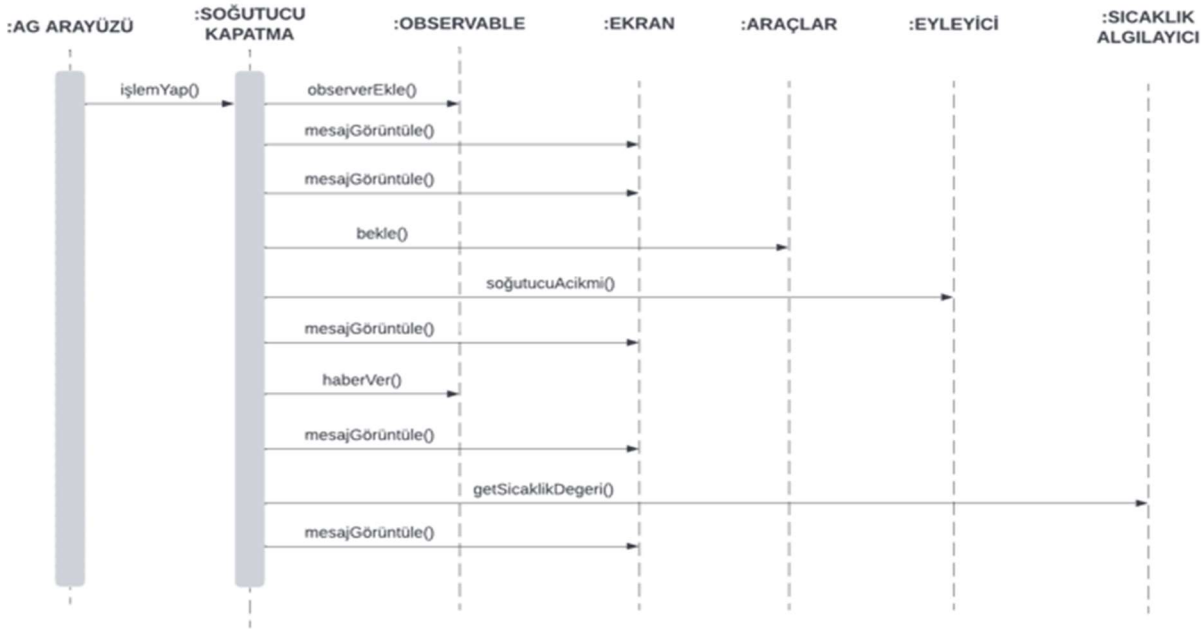
- Sıcaklık Algılayıcı çalışır durumda olmalıdır.

SIRALAMA ŞEMALARI

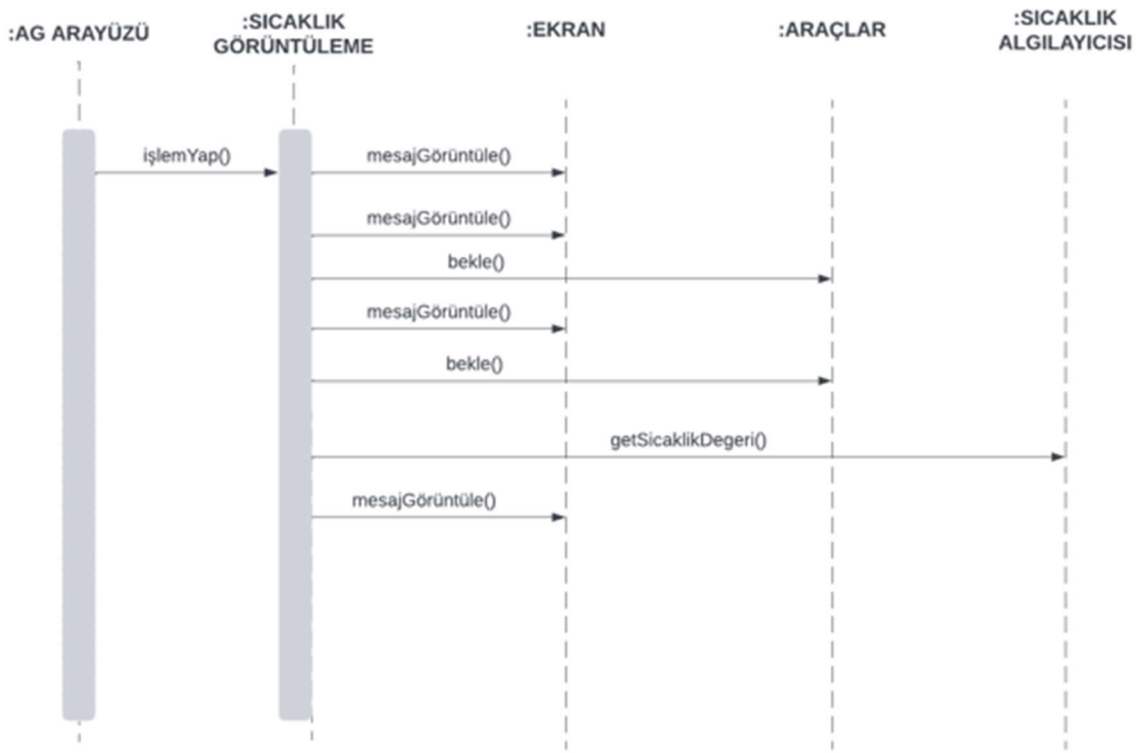
Soğutucunun Çalıştırılması



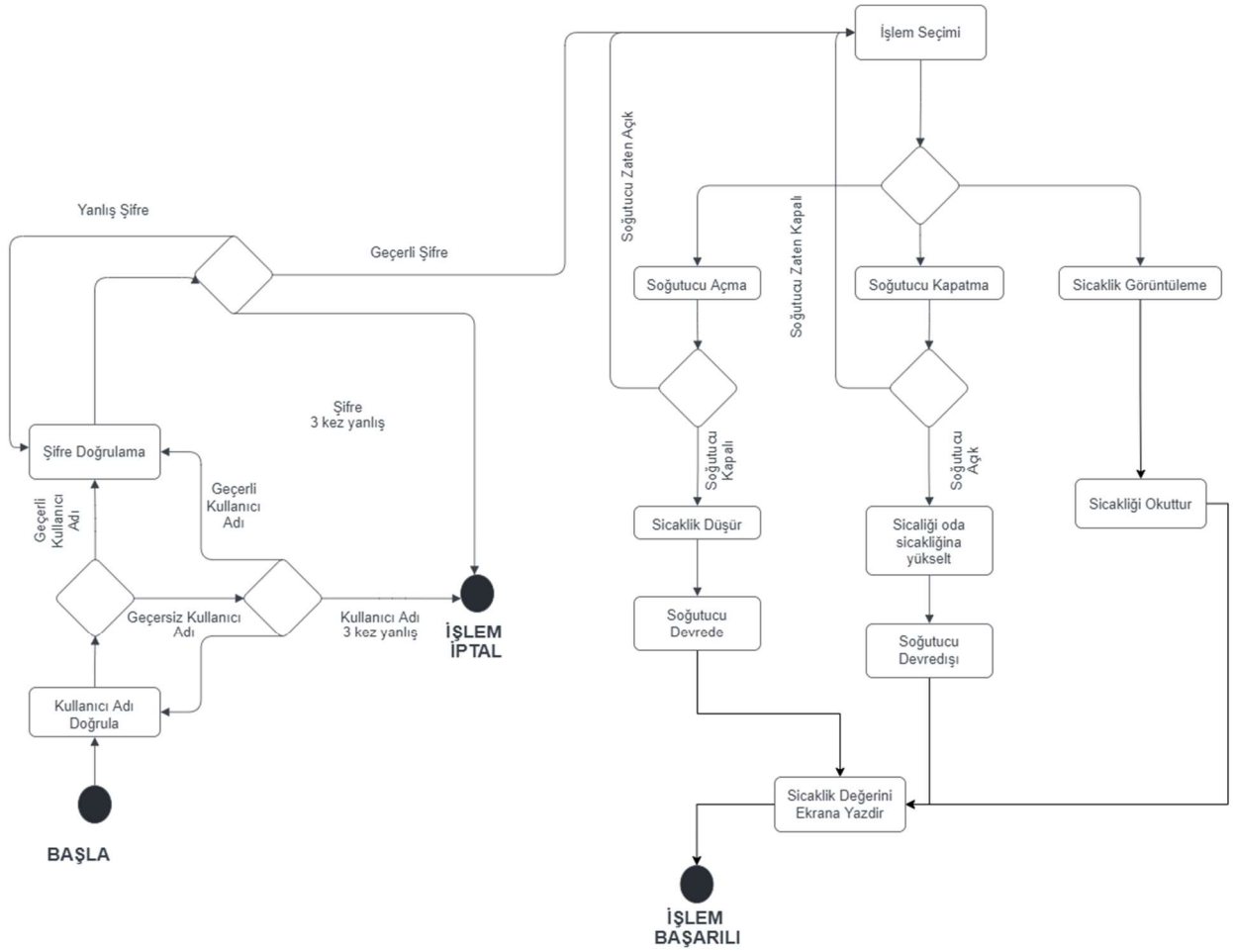
Soğutucunun Kapatılması



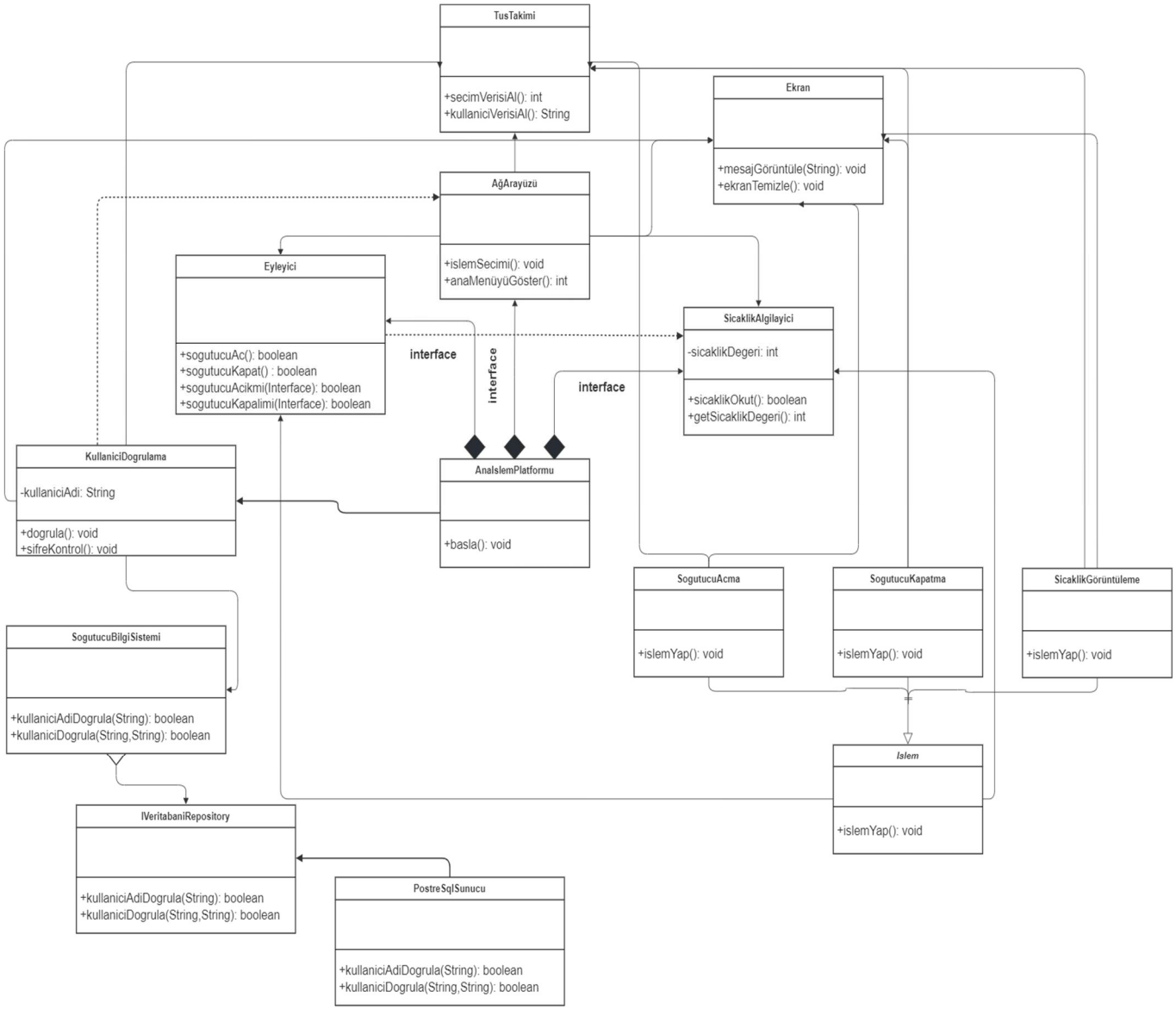
Sıcaklığın Görüntülenmesi



ETKİNLİK ŞEMASI



SINIF ŞEMASI

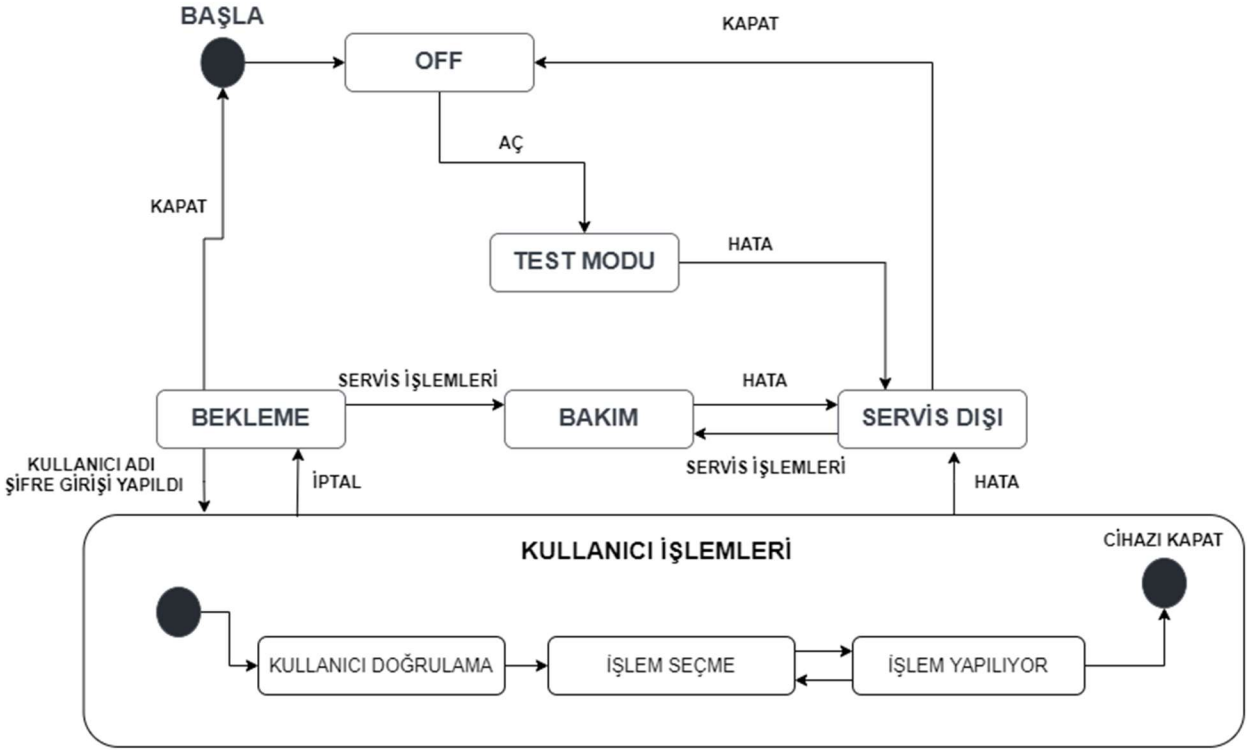


CRC KARTLARI

Soğutucu Açma(Soğutucu Açma işlemini gerçekleştirir)	
SORUMLULUK	**İŞBİRLİĞİ YAPILAN SINIF**
Mesaj yazdırma	**Ekran**
Soğutucu Kontrolü(Açık mı, Kapalı mı?)	**Eyleyici**
Soğutucu açma	**kendisi**
Sıcaklık alma	**Sicaklik Algilayici**

Kullanici Dogrulama(Kullanici doğrulama ve anamenüye yönlendirme işlemini gerçekleştirir)	
SORUMLULUK	**İŞBİRLİĞİ YAPILAN SINIF**
Mesaj yazdırma	**Ekran**
Kullanici adi ve şifre alma	**Tuş takımı**
Kullanici adi ve şifre doğrulama	**SoğutucuBilgiSistemi(IVeritabanıRepository, PostresqlSunucu)**
Anamenüye yönlendirme	**kendisi**

DURUM MAKİNESİ DİYAGRAMI



UYGULAMA EKRAN GÖRÜNTÜLERİ

Hatalı Kullanıcı

```
Hoşgeldiniz...
Kullanici Adinizi giriniz:
adem
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz kullanıcı adi hatali..
Tekrar Kullanici Adi Giriniz:
adksaljdklsajlda
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz kullanıcı adi hatali..
Tekrar Kullanici Adi Giriniz:
adslşakşkdlsşa
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
3 den fazla hatali kullanici adi girdiniz.Cikis yapiliyor...
```

Hatalı Şifre

```
Hoşgeldiniz...
Kullanici Adinizi giriniz:
adem28
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Sifre giriniz:
dsadasddd
Sifre Dogrulanıyor...
Veritabanına bağlandı!
Girdiginiz Sifre Hatali..
Tekrar Sifre Giriniz:
12345
Sifre Dogrulanıyor...
Veritabanına bağlandı!
Girdiginiz Sifre Hatali..
Tekrar Sifre Giriniz:
456445
Sifre Dogrulanıyor...
Veritabanına bağlandı!
3 den fazla hatali sifre girdiniz.Cikis yapiliyor...
```

Başarılı Doğrulama

```
Hoşgeldiniz...
Kullanici Adinizi giriniz:
adem
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Girdiginiz kullanıcı adi hatali..
Tekrar Kullanici Adi Giriniz:
adem28
Kullanici Adi Dogrulaniyor...
Veritabanına bağlandı!
Sifre giriniz:
1245
Sifre Dogrulanıyor...
Veritabanına bağlandı!
Girdiginiz Sifre Hatali..
Tekrar Sifre Giriniz:
123
Sifre Dogrulanıyor...
Veritabanına bağlandı!
Giris Basarılı...adem28
**********************************************
Ana Menu
1-Sogutucu Ac
2-Sogutucu Kapat
3-Sicaklik Görüntüle
4-Cikis Yap
Seciminiz:
**********************************************
```

Soğutucunun Çalıştırılması

```
**********************************************
Ana Menu
1-Sogutucu Ac
2-Sogutucu Kapat
3-Sicaklik Görüntüle
4-Cikis Yap
Seciminiz:
**********************************************
1
Sogutucu acma islemini sectiniz.
Sogutucu acma islemi baslatiliyor...
Sicaklik düsürülüyor...
Sogutucu Devrede !!!
Ortalama Sicaklik Degeri: -7
Ana Menüye Dönmek için '5' i tuşlayınız.
```

Soğutucunun Kapatılması

```
Ana Menüye Dönmek için '5' i tuşlayınız.
5
**********************************************
Ana Menu
1-Sogutucu Ac
2-Sogutucu Kapat
3-Sicaklik Görüntüle
4-Cikis Yap
Seciminiz:
**********************************************
2
Sogutucu kapama islemini sectiniz.
Sogutucu kapama islemi baslatiliyor...
Sicaklik oda sicakligina yükseltiliyor...
Sogutucu Devrededışı !!!
Ortalama Sicaklik Degeri: 13
Ana Menüye Dönmek için '5' i tuşlayınız.
```

Sıcaklığın Görüntülenmesi

```
2
Sogutucu kapama islemini sectiniz.
Sogutucu kapama islemi baslatiliyor...
Sicaklik oda sicakligina yükseltiliyor...
Sogutucu Devrededışı !!!
Ortalama Sicaklik Degeri: 13
Ana Menüye Dönmek için '5' i tuşlayınız.
5
**********************************************
Ana Menu
1-Sogutucu Ac
2-Sogutucu Kapat
3-Sicaklik Görüntüle
4-Cikis Yap
Seciminiz:
**********************************************
3
Sicaklik görüntüleme islemini sectiniz.
Sicaklik görüntüleme islemi baslatiliyor...
Algilayici'dan sicaklik degeri aliniyor...
Ortalama Sicaklik Degeri: 13
Ana Menüye Dönmek için '5' i tuşlayınız.
```

VERİTABANI EKRAN GÖRÜNTÜLERİ

Kullanicilar Tablosu ve Verileri

Data Output | Explain | Messages | Notifications

	id [PK] integer	kullaniciadi character varying (50)	sifre character varying (50)
1	1	adem28	123
2	2	kullanici	789
3	3	kullanici2	123456

SQL Kodları

Tablo Oluşturma ve İlgili tabloya veri girilmesi

```
CREATE TABLE "Users" (
    "UserId" SERIAL,
    "UserName" VARCHAR(50) NOT NULL,
    "Password" VARCHAR(50) NOT NULL,
    CONSTRAINT "userIdPK" PRIMARY KEY("UserId"),
    CONSTRAINT "userNameUnique" UNIQUE("UserName")

);

INSERT INTO "Users"("UserName","Password") VALUES ('adem28','123')
INSERT INTO "Users"("UserName","Password") VALUES ('kullanici','789')
INSERT INTO "Users"("UserName","Password") VALUES ('kullanici2','123456')
```

Kullanıcı Adı ve Kullanıcı Doğrulama Fonksiyonlarının Oluşturulması

```
15  CREATE FUNCTION kullaniciadidogrula(_userName character varying) returns integer
16  AS
17  $$
18  BEGIN
19      IF (select count(*) from "Users" where kullaniciadi=_userName)>0 THEN
20          return 1;
21      ELSE
22          return 0;
23      END IF;
24  END;
25  $$
26  LANGUAGE "plpgsql";
27
28  CREATE FUNCTION kullanicidogrula(_userName character varying,_sifre character varying) returns integer
29  AS
30  $$
31  BEGIN
32      IF (select count(*) from "Users" where kullaniciadi=_userName and sifre=_sifre)>0 THEN
33          return 1;
34      ELSE
35          return 0;
36      END IF;
37  END;
38  $$
39  LANGUAGE "plpgsql";
40
41  select * from kullaniciadidogrula('adem28');
42  select * from kullanicidogrula('adem28','123');
```

Data Output | Explain | Messages | Notifications

	kullanicidogrula integer
1	1

DEPENDENCY INVERSİON İLKESİ

Bir sınıfın, metodun ya da özelliğin, onu kullanan diğer sınıflara karşı olan bağımlılığı en aza indirgenmelidir. Bir alt sınıfta yapılan değişiklikler üst sınıfları etkilememelidir.

Dependency Inversion ilkesinde ki temel amaç, bir sınıf içerisinde başka bir sınıfı kullanıyorsanız kısacası kullanılan sınıf direkt olarak somut haliyle değil, onu soyut haliyle kullanmaktır.

Burada Soğutucu Bilgi Sistemi, bir veritabanı kullanarak kullanıcı doğrulaması yapacaktır.

Sistem de birden fazla veritabanı olduğunu varsayalım. Dolayısıyla ben sistemdeki herhangi bir veritabanını kullanarak doğrulama yapabilirim. Her bir veritabanının ise kendine özgü ayrı kodları komutları vardır. Bundan dolayı benim bu veritabanlarını kolay olarak kullanabilmek adına bir interface tanımlayarak bunları bir interface altında toplayabilirim ve istediğim veritabanını bu interface yardımıyla kolayca kullanabilirim. İnterface kullanmasaydım ben bu veritabanı sınıfının somutunu kullanmış olacaktım ve bu istenilen bir durum değildir. Dolayısıyla ben bu sınıfların somutunu değilde soyutunu kullanmak istersem veritabanı sınıfları ile Soğutucu Bilgi Sistemi sınıfı arasına bir interface koymam yeterlidir. Bu interface ise aşağıda görüldüğü üzere IVeritabanıRepository dir. Dolayısıyla artık ben IVeritabanıRepository interface ini kullanarak ilgili veritabanı sınıfın soyutunu kullanabilirim ve diğer veritabanı sınıflarına olan bağımlılığımı azaltabilirim.

```
public class SogutucuBilgiSistemi implements ISogutucuBilgiSistemi {
    private IVeritabaniRepository veritabaniRepository;
    public SogutucuBilgiSistemi(IVeritabaniRepository veritabaniRepository){
        this.veritabaniRepository=veritabaniRepository;
    }

    @Override
    public boolean kullaniciAdiDogrula(String kullaniciAdi) {
        if(veritabaniRepository.kullaniciAdiDogrula(kullaniciAdi)) { return true; }
        return false;
    }

    @Override
    public boolean kullaniciDogrula(String kullaniciAdi, String sifre) {
        if(veritabaniRepository.kullaniciDogrula(kullaniciAdi,sifre)){ return true; }
        return false;

    }
```

```java
public interface IVeritabaniRepository {
    public boolean kullaniciAdiDogrula(String kullaniciAdi);
    public boolean kullaniciDogrula(String kullaniciAdi,String sifre);
}
```

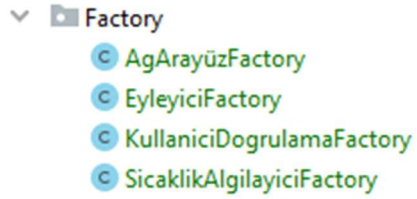
```java
public class PostreSqlSunucu implements IVeritabaniRepository {

    private Connection baglan(){
        Connection conn=null;
        try {
            conn = DriverManager.getConnection( url: "jdbc:postgresql://localhost:5432/veritabani",
                    user: "postgres",  password: "123");

            System.out.println("Veritabanına bağlandı!");

        } catch (Exception e) {
            System.out.println("Bağlantı girişimi başarısız!");
            e.printStackTrace();
        }
        return conn;
    }
    @Override
    public boolean kullaniciAdiDogrula(String kullaniciAdi) {
        Connection conn=this.baglan();
        try {
            CallableStatement stmt = conn.prepareCall( sql: "{?= call kullaniciadidogrula(?)}");
            stmt.setString( parameterIndex: 2,kullaniciAdi);
            stmt.registerOutParameter( parameterIndex: 1, Types.INTEGER);
            stmt.execute();
            conn.close();
            if(stmt.getInt( parameterIndex: 1)==1)
                return true;
```

FACTORY METHOD DESENİ

Factory Method deseni, yazılım ortamında birden çok aynı özelliği gösterebilecek sınıflar için gerekli nesne üretiminin kalıtım yolu ile kalıtım yapılan nesne tarafından yapılmasını sağlamaktır.

Factory method kullanmamızdaki temel amaç, sınıflarımızın içinde yeni bir nesne oluşturmamaktır. Çünkü sınıflarımızda ilgili nesneyi oluşturduğumuz da onunla ilgili değişiklik yapılabilir ve bu istenen durum değildir. Dolayısıyla Factory Method desenini kullanarak, ilgili sınıfların nesnelerini onlarla bağlantılı olan factory sınıflarında oluşturup bu durumun önüne geçebiliriz.

Factory Sınıflarımı ayrı bir paketin içine yerleştirdim.



Sicaklik Algilayicisina ait nesne aşağıdaki gibi oluşturulmuştur.

```
public class SicaklikAlgilayiciFactory {

    public ISicaklikAlgilayici factoryMethod() {
        Islem islem = new SicaklikGörüntüleme();
        ISicaklikAlgilayici sicaklikAlgilayici = new SicaklikAlgilayici(islem);
        return sicaklikAlgilayici;
    }
```

Eyleyiciye ait nesne aşağıdaki gibi oluşturulmuştur.

```
public class EyleyiciFactory {

    public IEyleyici factoryMethod() {
        SicaklikUretici sicaklikUretici = new SicaklikUretici();
        Islem sogutucuAcma = new SogutucuAcma();
        Islem sogutucuKapatma=new SogutucuKapatma();
        IEyleyici eyleyici=new Eyleyici(sogutucuAcma,sogutucuKapatma);
        eyleyici.observerEkle(sicaklikUretici);
        return eyleyici;
    }
}
```

```
public class AgArayüzFactory {

    public IAgArayüzü factoryMethod() { return new AgArayüzü(); }
}
```

Kullanici Dogrulamaya ait nesne aşağıdaki gibi oluşturulmuştur.

```
public class KullaniciDogrulamaFactory {

    public IKullaniciDogrulama factoryMethod() {
        SogutucuBilgiSistemi sogutucuBilgiSistemi=new SogutucuBilgiSistemi(new PostreSqlSunucu());
        IKullaniciDogrulama kullaniciDogrulama = new KullaniciDogrulama(sogutucuBilgiSistemi);
        return kullaniciDogrulama;
    }
}
```

Uygulamanın main kısmından Anaİslem platformu adında nesne oluşturulup ilgili Factory lerin oluşturulması aşağıdaki gibidir.

```
public class SogutucuUygulamasi {
    public static void main(String[] args) throws IOException {
        AnaislemPlatformu anaislemPlatformu =new AnaislemPlatformu(new SicaklikAlgilayiciFactory(),new EyleyiciFactory(),
                new AgArayüzFactory(),new KullaniciDogrulamaFactory());
        anaislemPlatformu.basla();
    }
}
```

```
public class AnaislemPlatformu {
    private ISicaklikAlgilayici sicaklikAlgilayici;
    private IEyleyici eyleyici;
    private IAgArayüzü agArayüzü;
    private IKullaniciDogrulama kullaniciDogrulama;

    private SicaklikAlgilayiciFactory sicaklikAlgilayiciFactory;
    private EyleyiciFactory eyleyiciFactory;
    private AgArayüzFactory agArayüzFactory;
    private KullaniciDogrulamaFactory kullaniciDogrulamaFactory;

    private static final int sogutucuAcma = 1;
    private static final int sogutucuKapama = 2;
    private static final int sicaklikGörüntüleme = 3;
    private static final int cikis = 4;

    public AnaislemPlatformu(SicaklikAlgilayiciFactory sicaklikAlgilayiciFactory, EyleyiciFactory eyleyiciFactory,
                             AgArayüzFactory agArayüzFactory, KullaniciDogrulamaFactory kullaniciDogrulamaFactory) {

        this.sicaklikAlgilayiciFactory=sicaklikAlgilayiciFactory;
        this.eyleyiciFactory=eyleyiciFactory;
        this.agArayüzFactory=agArayüzFactory;
        this.kullaniciDogrulamaFactory=kullaniciDogrulamaFactory;
    }

    public void basla() throws IOException {
        sicaklikAlgilayici = sicaklikAlgilayiciFactory.factoryMethod();
        eyleyici = eyleyiciFactory.factoryMethod();
        agArayüzü = agArayüzFactory.factoryMethod();
        kullaniciDogrulama = kullaniciDogrulamaFactory.factoryMethod();
```

OBSERVER TASARIM DESENİ

Observer tasarım deseni, birden fazla nesneyi takip ettikleri başka bir nesnede gerçekleşen olaylarla ilgili bilgilendirmeyi sağlayan bir abonelik mekanizması oluşturmayı amaçlar.

Gerçekleştirdiğim uygulamada observer tasarım desenini aşağıdaki ekran görüntülerinde görüldüğü üzere Soğutucu açma, soğutucu kapatma gibi durumlarda sıcaklik seviyesinin tekrardan güncellenmesi (sicakliğin düşürülmesi veya oda sıcaklığına yükseltilmesi) için kullandım.

Burada observer'ım sıcaklık üreticisi , observer im ise soğutucu açma ve soğutucu kapama işlemlerini yapan eyleyici sınıfıdır.

```
public interface Observer {
    void update(String deger);
}
```

```
public class Eyleyici extends Observable implements IEyleyici {
```

```
public class SicaklikUretici implements Observer {
```

Birden fazla sıcaklık üreticim olabilir ve bunların her birine güncelleme yapmak gereklidir. Dolayısıyla bu durumu dikkate alarak bir observer listesi tanımladım.

```
public abstract class Observable {
    private List<Observer> observerList;

    public Observable() { observerList = new ArrayList<>(); }

    public void ekle(Observer observer) { observerList.add(observer); }

    public void cikar(Observer observer) { observerList.remove(observer); }

    protected void haberVer(String deger){
        for (Observer observer : observerList) {
            observer.update(deger);
        }
    }
}
```

Observer ın update durumunda yapacağı işlemler aşağıda görülmek üzere sıcaklık düşürme ve oda sıcakliği yöntemlerinde belirtilmiştir.

```
@Override
public void update(String deger) {
    if (deger.equalsIgnoreCase( anotherString: "sicaklikdüsür"))
        sicaklikDusur();
    else if (deger.equalsIgnoreCase( anotherString: "odasicakligi"))
        odaSicakliginaAyarla();
}
```

Soğutucu Açılırken sıcaklığın düşürülmesi durumudur.

```
public void SogutucuAc() {
    if(SogutucuKapalimi()){
        sogutucuAcma.islemYap();
        haberVer( deger: "sicaklikdüsür");
    }
    else{
        Ekran.mesajGoruntule("Soğutucu zaten açık..");
    }
    Ekran.mesajGoruntule("Ortalama sicaklik degeri: "+SicaklikUretici.getSicaklikDegeri());
}
```

Soğutucu kapanırken sıcaklığın oda sıcaklığına getirilmesi durumudur.

```
public void SogutucuKapat() {
    if(SogutucuAcikmi()){
        sogutucuKapatma.islemYap();
        haberVer( deger: "odasicakligi");
    }
    else{
        Ekran.mesajGoruntule("Soğtucu zaten kapalı..");
    }
    Ekran.mesajGoruntule("Ortalama sicaklik degeri: "+SicaklikUretici.getSicaklikDegeri());
}
```