



SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
NESNE YÖNELİMLİ ANALİZ VE TASARIM DERSİ

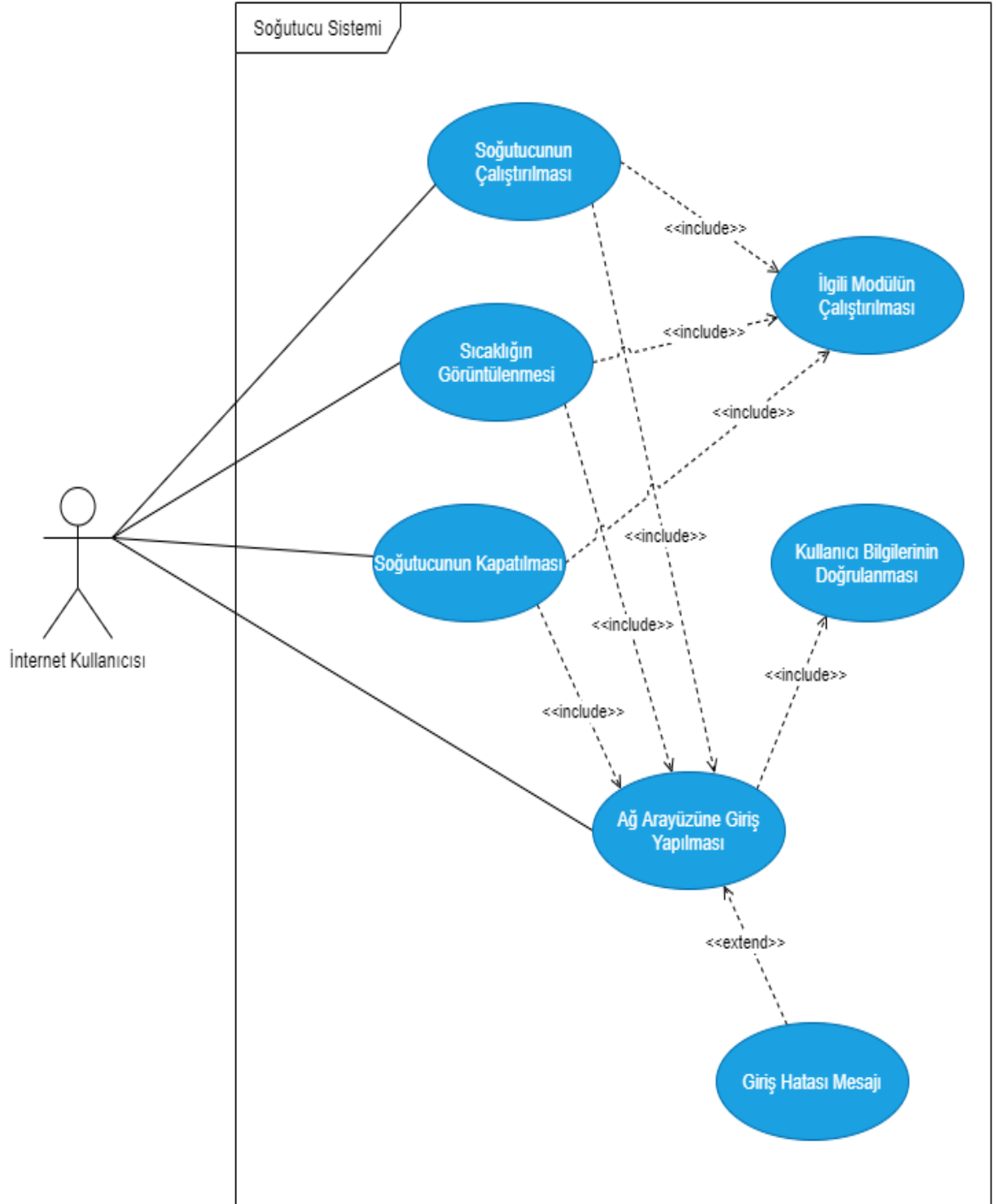
PROJE RAPORU

AD SOYAD.....: SEDANUR SARIKOÇ
ÖĞRENCİ NUMARASI.: B181210110
SINIF.....: 2
ÖĞRETİM TÜRÜ-ŞUBE: 2A
E-POSTA.....: sedanursrkoc@gmail.com

SAKARYA

Nisan, 2020

KULLANIM DURUMU (USE CASE) DİYAGRAMI



KULLANIM DURUMLARI

USE CASE - SOĞUTUCUNUN ÇALIŞTIRILMASI:

- "Soğutucunun Çalıştırılması" Kullanım Durumu
 - Eşsiz bir ad: Soğutucunun Çalıştırılması
 - 04.05.2020,08.05.2020 v1.1.1 sedanursarıkoç
 - ilgili aktörler: İnternet kullanıcısı
 - Giriş koşulu: Kullanıcı internet ile soğutucunun ağ arayüzüne bağlanarak soğutucuyu aç seçeneğini seçer
 - Çıkış koşulu: Müşteri işlemini (transation) tamamlar
 - Özel gereksinim: Elektrik gücü, Soğutucu gazlar içermesi, Ağ arayüzüne sahip olması, İnternet bağlantısı
 - Olay akışı:

Ana Olay Akışı:

1. Kullanıcı ağ arayüzünü açar
2. Ağ arayüzü kullanıcıya giriş ekranını gösterir
3. Kullanıcı bilgilerini girer
4. Kullanıcı bilgileri soğutucu veri tabanı yardımıyla doğrulanır
5. Ağ arayüzü kullanıcıya menüyü gösterir
6. Kullanıcı menüden soğutucuyu açmayı seçer
7. Soğutucu zaten açık değilse merkezi işlem birimi kullanıcının isteğini eyleyiciğe iletir
8. Eyleyici soğutucuyu açma işlemini gerçekleştirir

Alternatif Olay Akışı:

A1. Kullanıcının bilgileri doğrulanamadı(4)

5. Kullanıcının bilgilerinin doğrulanamadığı mesajı ekrana yazdırılır
6. İşlem sonlandırılır

A2. Soğutucu zaten açık(7)

8. Merkezi işlem birimi isteği eyleyiciye iletmez
9. Kullanıcıya mesaj iletilir
10. İşlem sonlandırılır

USE CASE – SICAKLIĞIN GÖRÜNTÜLENMESİ:

- "Sıcaklığın Görüntülemesi " Kullanım Durumu
 - Eşsiz bir ad: Ağ Arayüzü Kullanarak Sıcaklık Görüntüleme
 - 04.05.2020,08.05.2020 v1.1.1 sedanursarıkoç
 - ilgili aktörler: İnternet kullanıcısı
 - Giriş koşulu: İnternet yardımıyla soğutucunun ağ arayüzüne bağlanarak sıcaklık görüntüle seçeneğini seçer
 - Çıkış koşulu: Müşteri işlemini (transation) tamamlar
 - Özel gereksinim: Elektrik gücü, Ağ arayüzüne sahip olması, İnternet bağlantısı
 - Olay akışı:

Ana Olay Akışı:

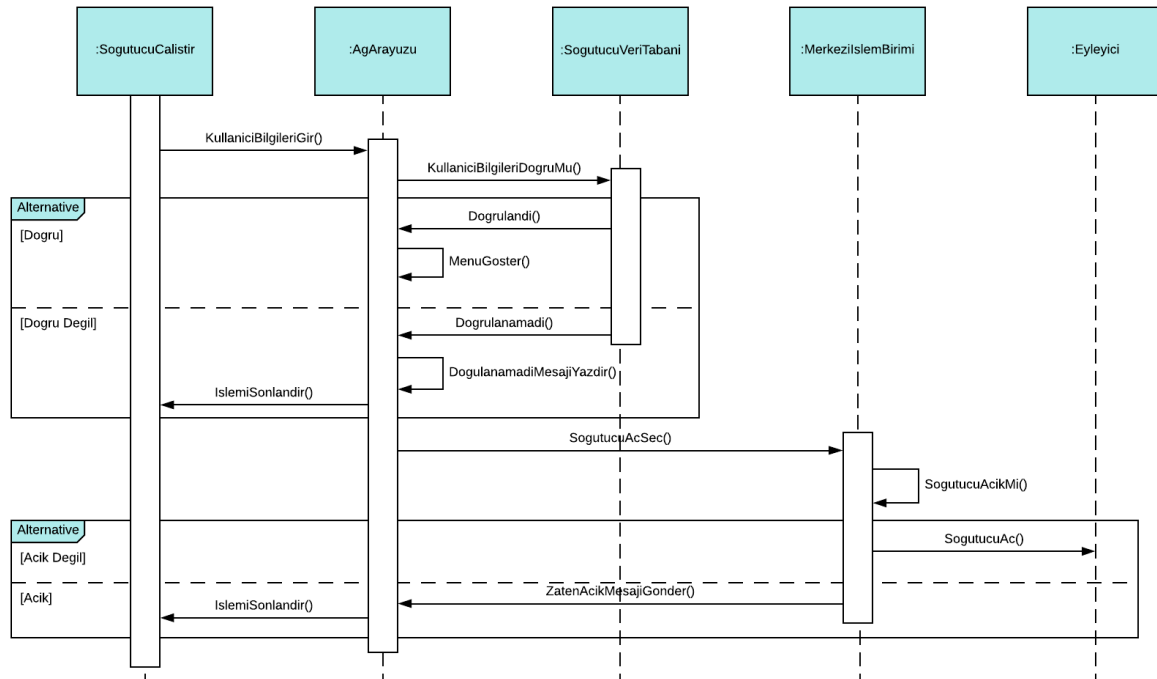
1. Kullanıcı ağ arayüzünü açar
2. Ağ arayüzü kullanıcıya giriş ekranını gösterir
3. Kullanıcı bilgilerini girer
4. Kullanıcı bilgileri soğutucu veri tabanı yardımıyla doğrulanır
5. Ağ arayüzü kullanıcıya menüyü gösterir
6. Kullanıcı menüden sıcaklığı görüntülemeyi seçer
7. Merkezi İşlem birimi kullanıcının isteğini gerçekleştirmek için sıcaklık algılayıcıyı çalıştırır
8. Sıcaklık görüntülenir

Alternatif Olay Akışı:

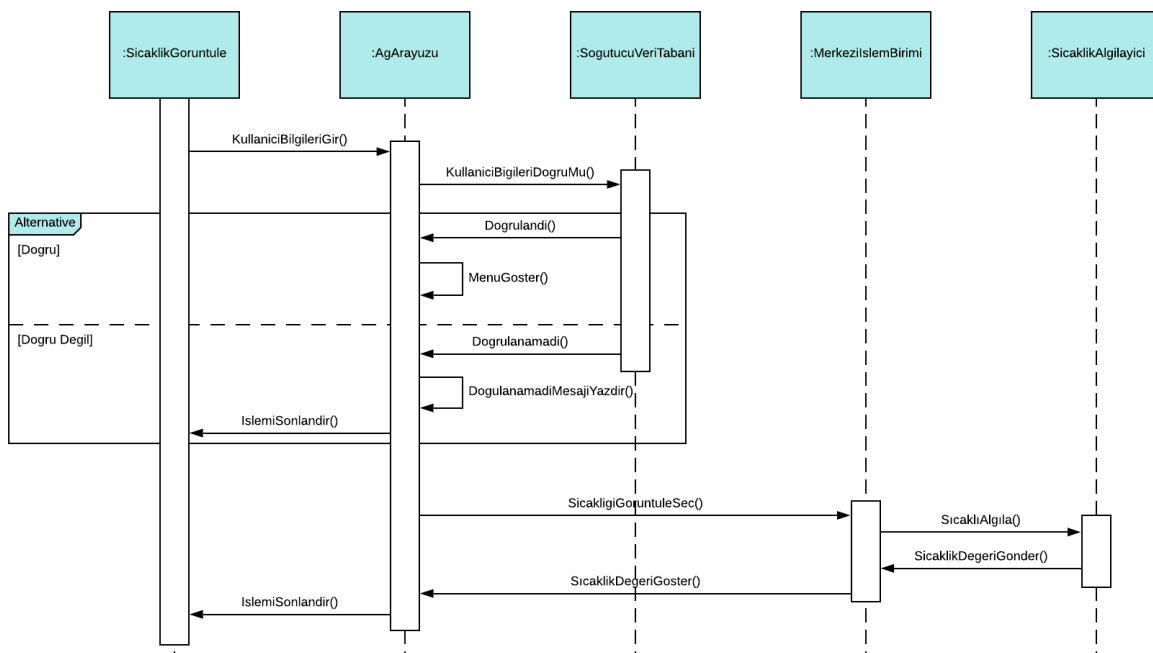
- A1. Kullanıcının bilgileri doğrulanamadı(4)
5. Kullanıcının bilgilerinin doğrulanamadığı mesajı ekrana yazdırılır
6. İşlem sonlandırılır

SIRALAMA ŞEMALARI (SEQUENCE DIAGRAM)

SOĞUTUCUNUN ÇALIŞTIRILMASI - SIRALAMA ŞEMASI:

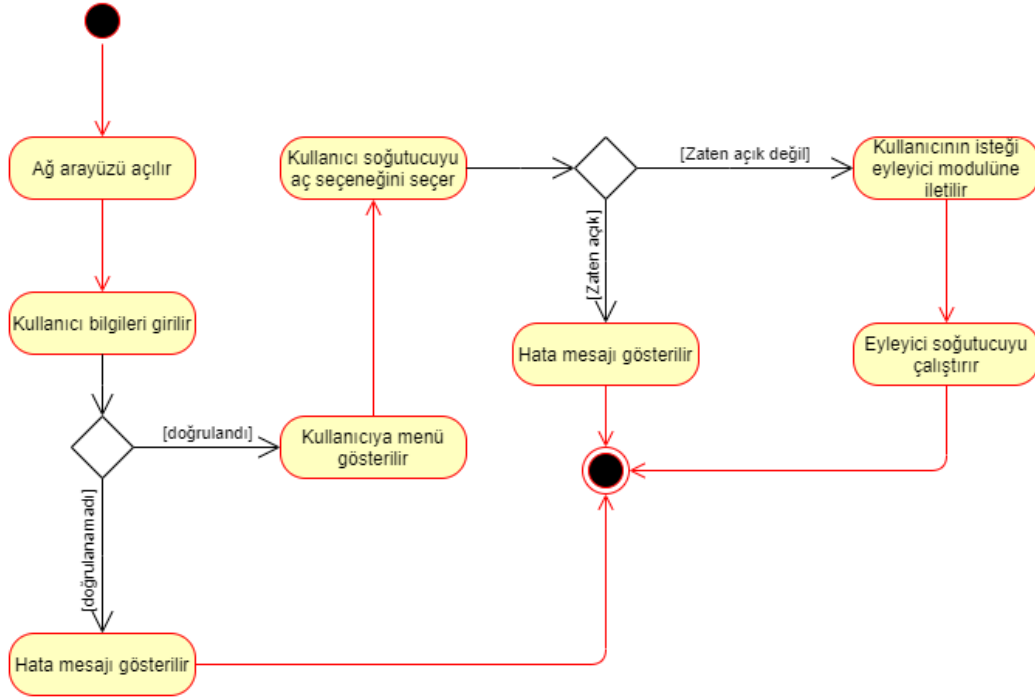


SICAKLIĞIN GÖRÜNTÜLENMESİ - SIRALAMA ŞEMASI

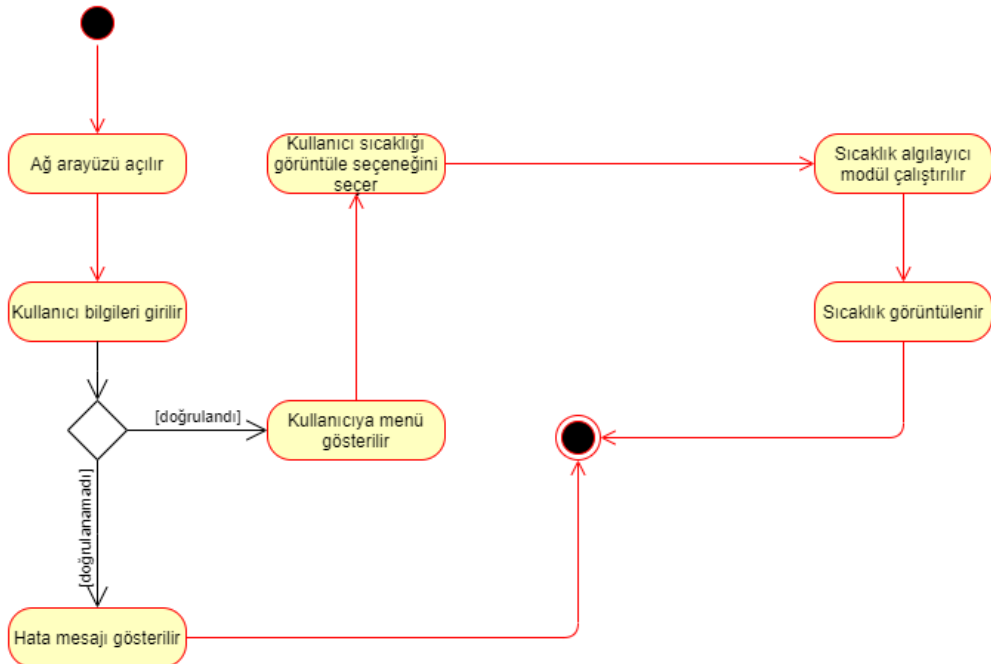


ETKİNLİK ŞEMALARI (ACTIVITY DIAGRAM)

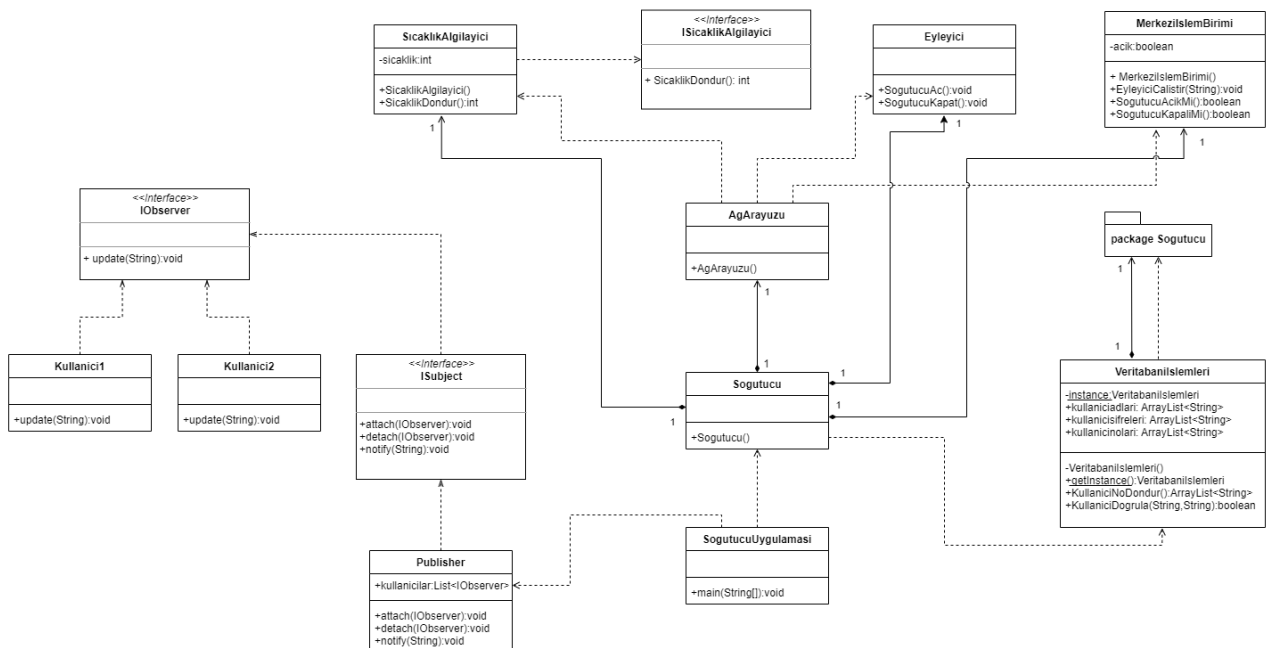
SOĞUTUCUNUN ÇALIŞTIRILMASI - ETKİNLİK ŞEMASI:



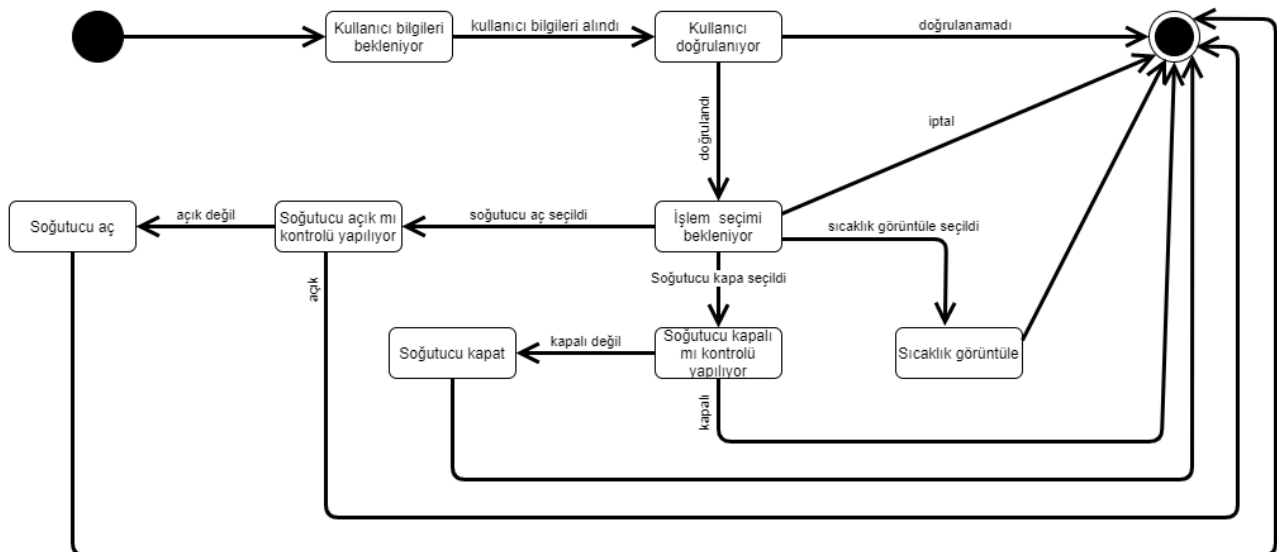
SICAKLIĞIN GÖRÜNTÜLENMESİ - ETKİNLİK ŞEMASI:



SINIF ŞEMASI



SİSTEMİN DURUM MAKİNE (STATE MACHINE) DİYAGRAMI



Kullanıcı doğrulama ekranı ve açıklaması:

```
Output - SogutucuUygulamasi (run)
run:
Kullanici Adi: Devrim
Sifre: 1234
Veritabanına bağlandı!
Kullanıcı doğrulanıyor...
Veritabanına bağlandı ve kullanıcı hesap bilgilerini aldı...
Kullanici dogrulanamadi...

Kullanici Adi: |
```

Kullanıcı bilgilerini girdiğinde veritabanından kullanıcı bilgileri kontrol edilir ve bilgiler veritabanındaki kullanıcı bilgileriyle eşleşmiyorsa tekrar kullanıcı bilgileri istenir. Bu işlem kullanıcı bilgileri doğrulanana kadar devam eder.

```
Output - SogutucuUygulamasi (run)
Kullanıcı doğrulanıyor...
Veritabanına bağlandı ve kullanıcı hesap bilgilerini aldı...
Kullanici dogrulanamadi...

Kullanici Adi: Seda
Sifre: 123456
Kullanıcı doğrulanıyor...
Veritabanına bağlandı ve kullanıcı hesap bilgilerini aldı...
Kullanici dogrulandi...

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklik goruntule
[4] Cikis yap
Secim:
```

Kullanıcı bilgilerini girdiğinde veritabanından kullanıcı bilgileri kontrol edilir ve bilgiler veritabanındaki kullanıcı bilgileriyle eşleşiyorsa ağ arayüzü kullanıcıya menüyü gösterir.

SOĞUTUCUNUN AÇILMASI:

```
Output - SogutucuUygulaması (run)

Kullanici dogrulandi...

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:1
İstek eyleyiciye bildiriliyor...
Sogutucu acildi.

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:|
```

Kullanıcı soğutucu aç seçeneğini seçtiğinde soğutucu zaten açık değilse merkezi işlem birimi eyleyiciye isteği bildirir. Eyleyici soğutucuyu açar. Ağ arayüzü kullanıcıya ilgili mesajı iletir.

```
Output - SogutucuUygulaması (run)

İstek eyleyiciye bildiriliyor...
Sogutucu acildi.

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:1
Sogutucu zaten acik...

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:
```

Kullanıcı soğutucu aç seçeneğini seçtiğinde soğutucu zaten açık ise merkezi işlem birimi eyleyiciye isteği bildirmez ve ağ arayüzü kullanıcıya hata mesajını iletir.

SOĞUTUCUNUN KAPATILMASI:

```
Output - SogutucuUygulaması (run)

Sogutucu zaten acik...

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:2
İstek eyleyiciye bildiriliyor...
Sogutucu kapatıldı.

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:|
```

Kullanıcı soğutucu kapat seçeneğini seçtiğinde soğutucu zaten kapalı değilse merkezi işlem birimi eyleyiciye isteği bildirir. Eyleyici soğutucuyu kapatır. Ağ arayüzü kullanıcıya ilgili mesajı iletir.

```
Output - SogutucuUygulaması (run)

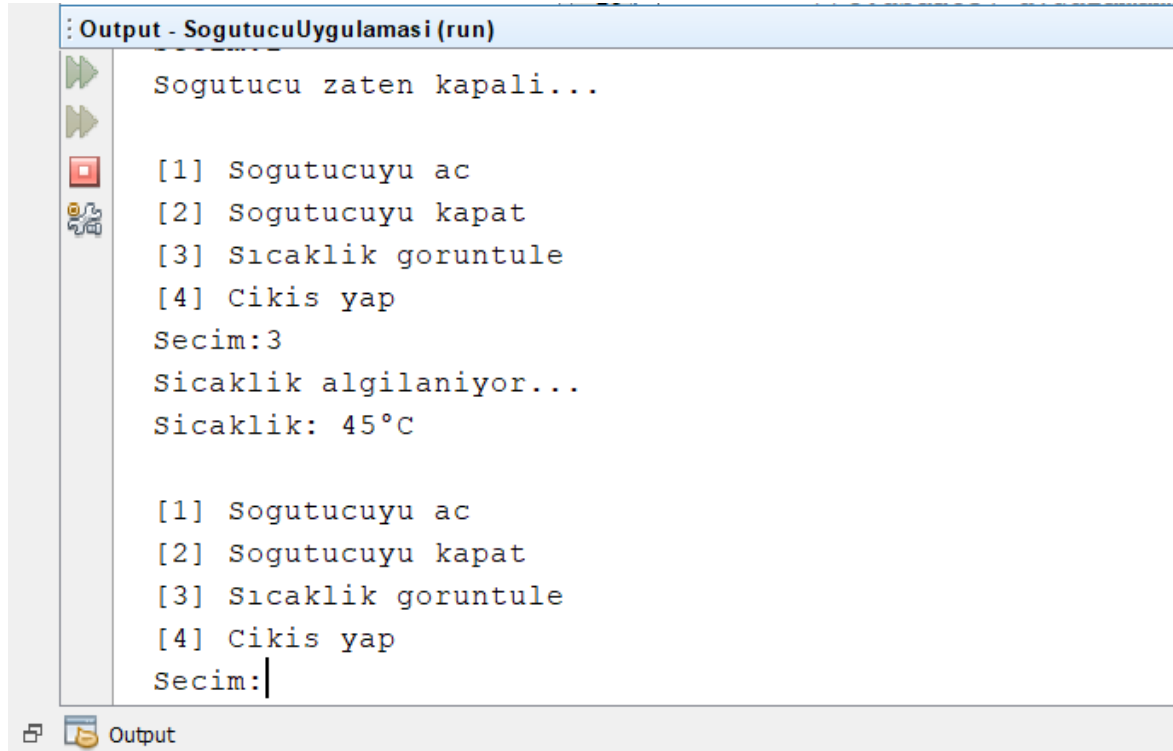
İstek eyleyiciye bildiriliyor...
Sogutucu kapatıldı.

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:2
Sogutucu zaten kapali...

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkış yap
Secim:|
```

Kullanıcı soğutucu aç seçeneğini seçtiğinde soğutucu zaten kapalı ise merkezi işlem birimi eyleyiciye isteği bildirmez ve ağ arayüzü kullanıcıya hata mesajını iletir.

SICAKLIĞIN GÖRÜNTÜLENMESİ:



```
Output - SogutucuUygulaması (run)

Sogutucu zaten kapali...

[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkis yap
Secim:3
Sıcaklık algılanıyor...
Sıcaklık: 45°C

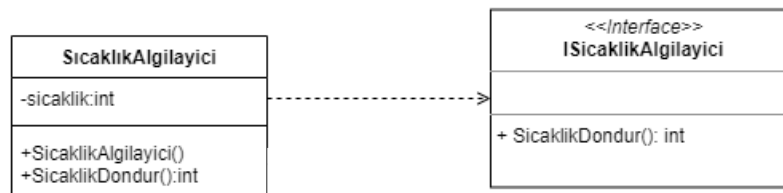
[1] Sogutucuyu ac
[2] Sogutucuyu kapat
[3] Sıcaklık görüntüle
[4] Çıkis yap
Secim:|
```

Kullanıcı sıcaklık görüntüle seçeneğini seçtiğinde sıcaklık algılayıcı sıcaklığı algılayarak merkezi işlem birimine iletir. Ağ arayüzü sıcaklık değerini kullanıcıya gösterir.

OPEN/CLOSED İLKESİ:

Bu ilke herhangi bir modülü geliştirmek istediğimizde modülde değişiklik yapmadan yalnızca modülü genişleterek ilgili modülü geliştirmemizi amaçlar. Bu ilkeyi kullandığımız modül geliştirmelere açık(open) değiştirmelere ise kapalıdır(closed).

Projemde bu ilkeyi “Sıcaklık Algılayıcı” modülünü gerçeklemek için kullandım. Bu modül için Interface kullanarak soyutlama yaptım. Sıcaklık algılayıcının farklı sıcaklık ölçü birimlerini de döndürmesi gerektiği zaman modülü değiştirmeden genişleterek modülün farklı sıcaklık ölçü birimleri için de çalışmasını sağlayabileceğim.



SINGLETON ve OBSERVER DESENLERİ

Singleton:

Bir sınıftan sadece tek bir nesne oluşturulabilmesini amaçlar. Yapıcı gibi çalışan bir metod kullanarak sınıftan nesne yoksa nesne oluşturulur ve bu nesne saklanır, nesne önceden oluşturulduysa o nesneyi döndürür.

Projemde bu deseni “VeritabaniIslemleri” isimli sınıfta kullandım. Kurucu fonksiyonu private yaparak dışarıdan erişimi engelledim.

```
public class VeritabaniIslemleri {  
  
    private static VeritabaniIslemleri instance;  
    ArrayList<String> kullaniciadlari = new ArrayList<String>();  
    ArrayList<String> kullanicisifreleri = new ArrayList<String>();  
    ArrayList<String> kullanicinolari = new ArrayList<String>();  
  
    private VeritabaniIslemleri()  
    {  
    }
```

Sonrasında nesneye global erişebilmek için nesne yoksa oluşturan, nesne varsa onu döndüren bir fonksiyon yazdım.

```
public static synchronized VeritabaniIslemleri getInstance() {  
  
    if(instance==null) instance = new VeritabaniIslemleri();  
  
    return instance;  
  
}
```

Observer: Çok sayıda nesneye, gözlemlenen nesnede gerçekleşen bir olayı bildirmek için kullanılan desendir.

Projemde bu deseni IObserver, ISubject interfaceleri ve Publisher sınıfı yardımı ile gerçekleştirdim. IObserver interface’i kullanıcılara bağladım böylelikle gözlemlenen nesnede bir olay geldiğinde bağlı olan kullanıcılara Publisher sınıfından nesne oluşturularak ve bu kullanıcı nesnelerini publisher’a ekledim. Publisher’ın notify metodunu çağırarak kullanıcılara mesaj iletilmesini sağladım.

