

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

Ders : Nesne Yönelimli Analiz ve Tasarım

Hazırlayan : Yusuf ÖZASLAN (B201210033)

Şube : 1. Öğretim B Grubu

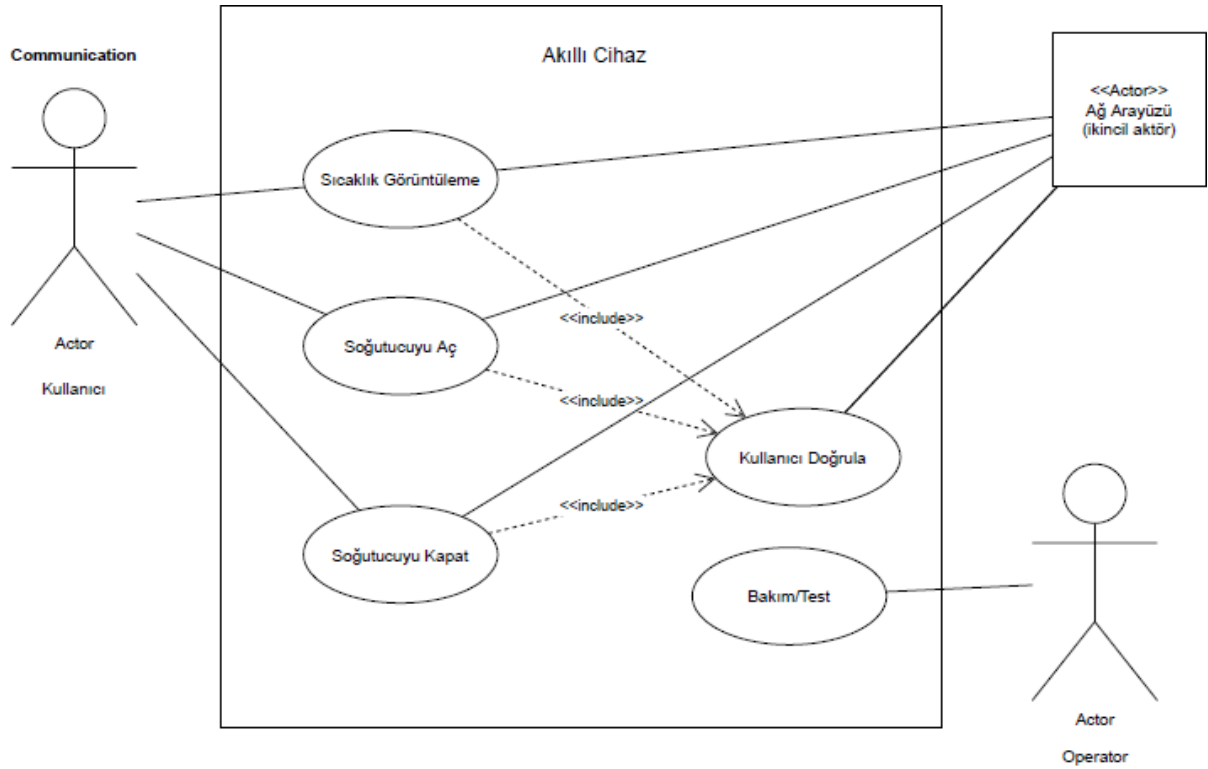
E-Posta : yusuf.ozaslan@ogr.sakarya.edu.tr

GitHub : <https://github.com/YusufOzaslan/NYA-Proje>

Nisan – 2022



Use Case Diagram



Kullanım Durumları

Sıcaklığın Görüntülenmesi :

04.05.2022 V1.1.1 Yusuf Özasan

İlgili Aktörler : Kullanıcı, Ağ Arayüzü

Giriş Koşulu : Kullanıcının kullanıcı adı ve şifresi doğrulanır.

Çıkış Koşulu : Kullanıcı işlemlerini tamamlar.

Ana Olay Akışı:

- Kullanıcının şifresi onaylanır.
- Sıcaklık görüntüleme isteği gönderilir.
- Sıcaklık değeri okunur.
- Okunan değer döndürülür.
- Değer kullanıcıya gösterilir.

Alternatif Olay Akışı:

- Kullanıcının şifresi onaylanmaz.
- Veri tabanı bağlantısı kesilir.
- İşlem sonlandırılır.

Özel Gereksinimler:

- Akıllı cihaz ağa bağlı olmalı.

Soğutucunun Çalıştırılması:

04.05.2022 V1.1.1 Yusuf Özaslan

İlgili Aktörler : Kullanıcı, Ağ Arayüzü

Giriş Koşulu : Kullanıcının kullanıcı adı ve şifresi doğrulanır.

Çıkış Koşulu : Kullanıcı işlemlerini tamamlar.

Ana Olay Akışı:

- Kullanıcının şifresi onaylanır.
- Soğutucu kapatma talebi gönderilir.
- Soğutucu kapatılır.
- Kullanıcı bilgilendirilir.

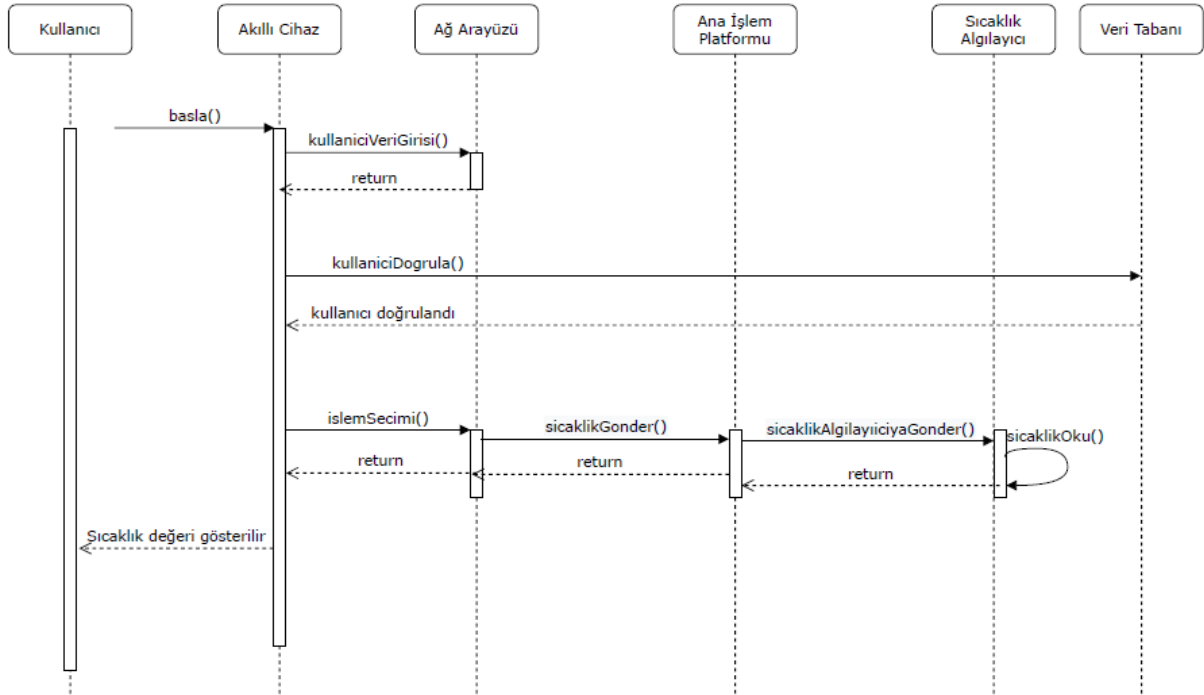
Alternatif Olay Akışı:

- Kullanıcının şifresi onaylanmaz.
- Veri tabanı bağlantısı kesilir.
- İşlem sonlandırılır.

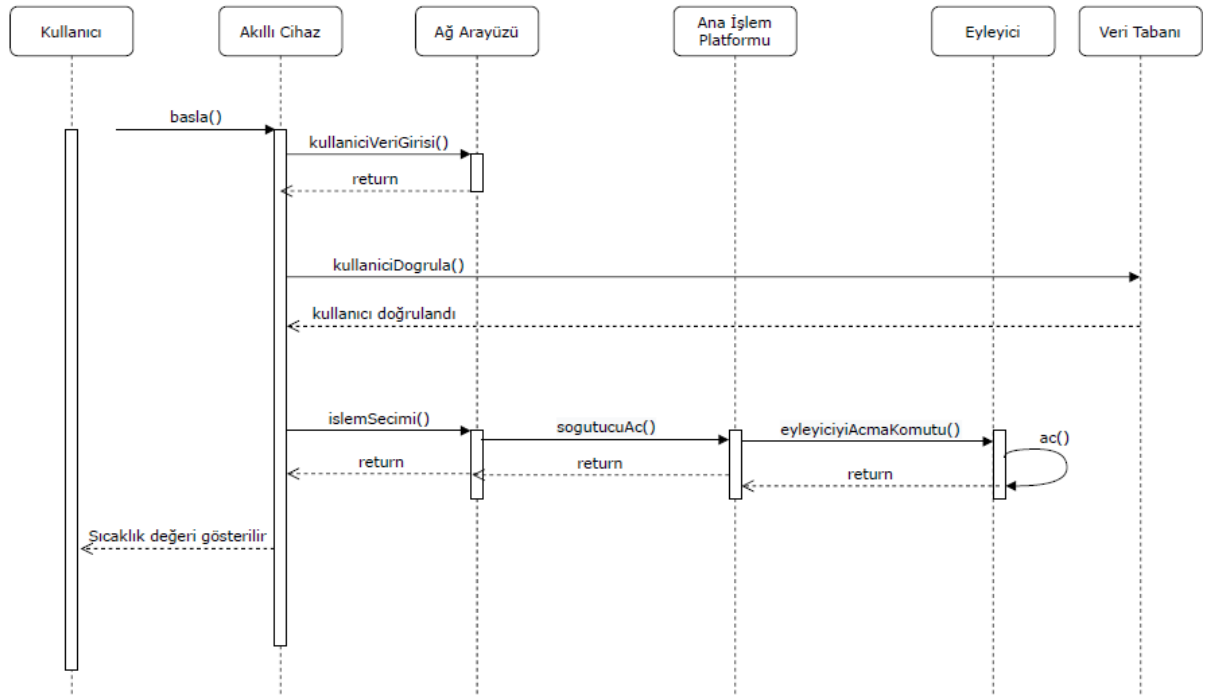
Özel Gereksinimler:

- Akıllı cihaz ağa bağlı olmalı.

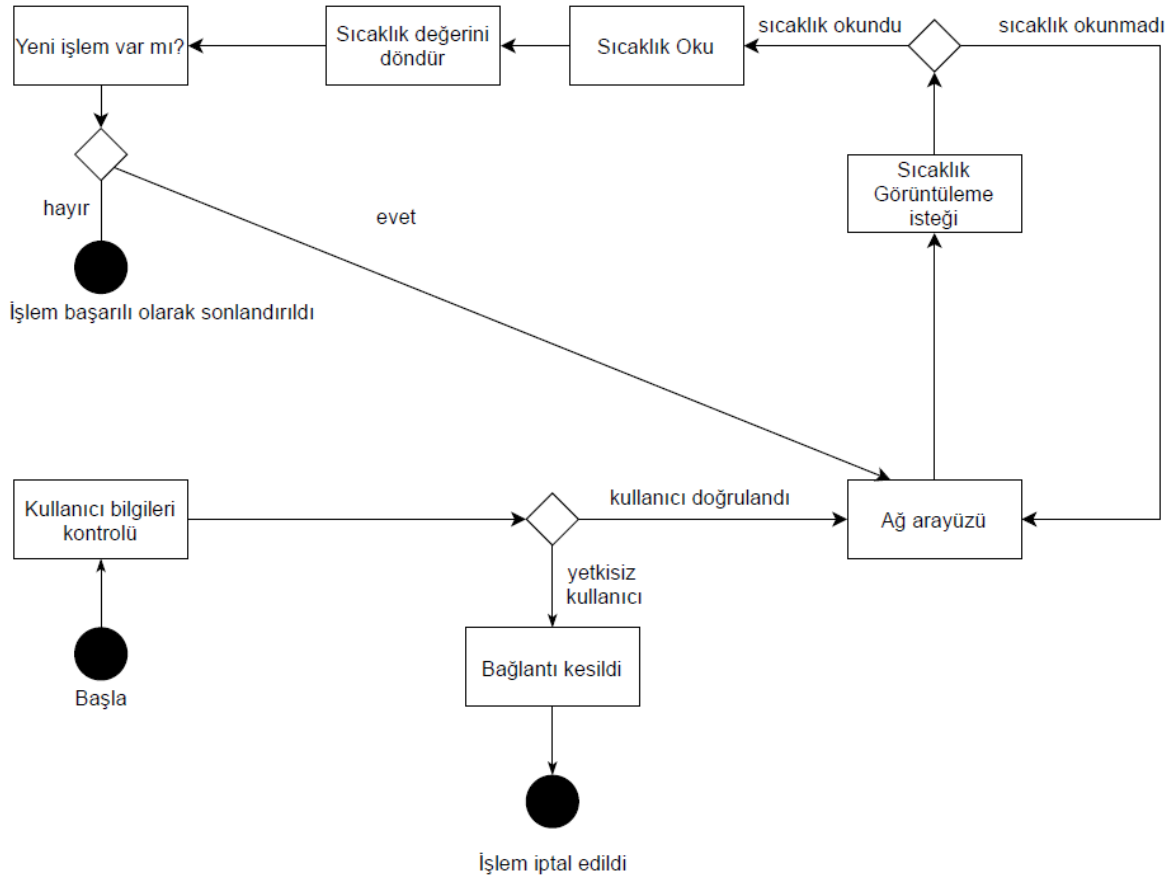
Sıcaklığın Görüntülenmesi (Sequence Diagram)



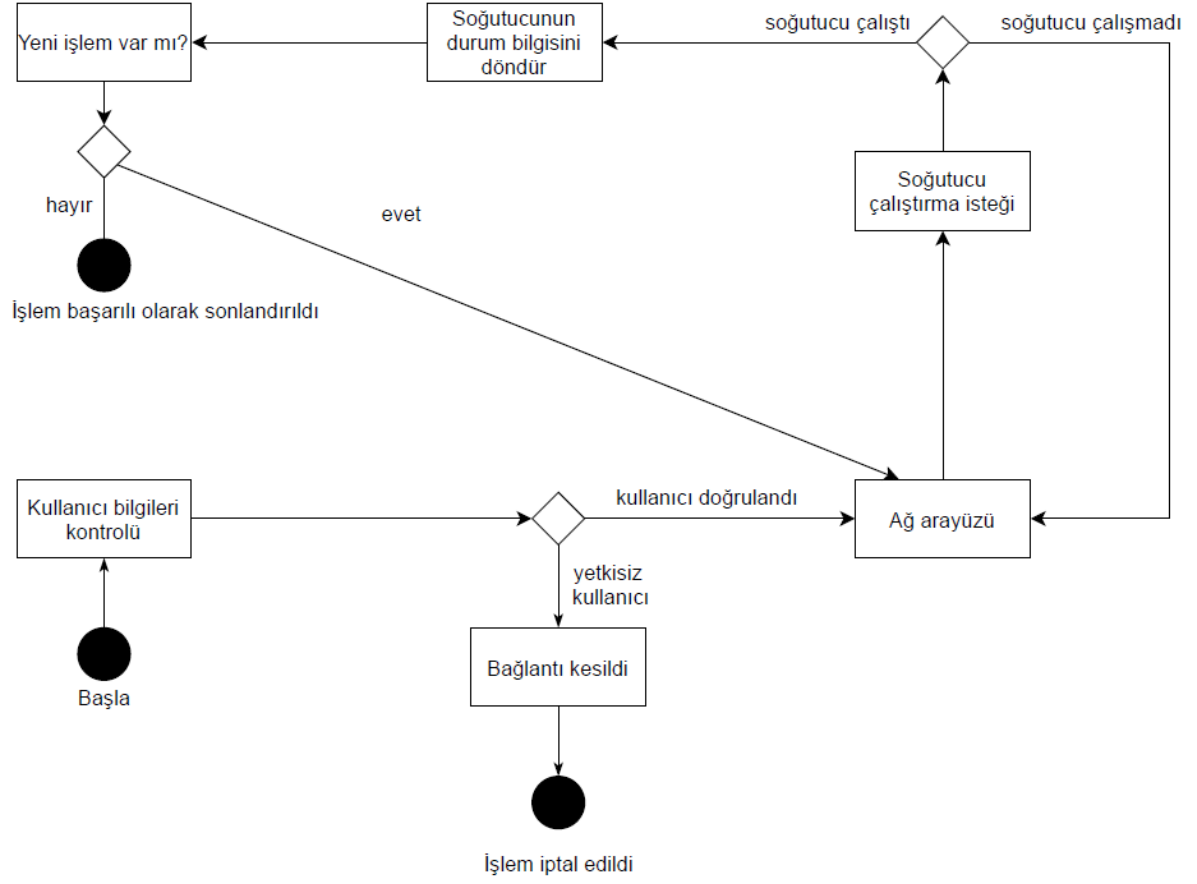
Soğutucunun Çalıştırılması (Sequence Diagram)



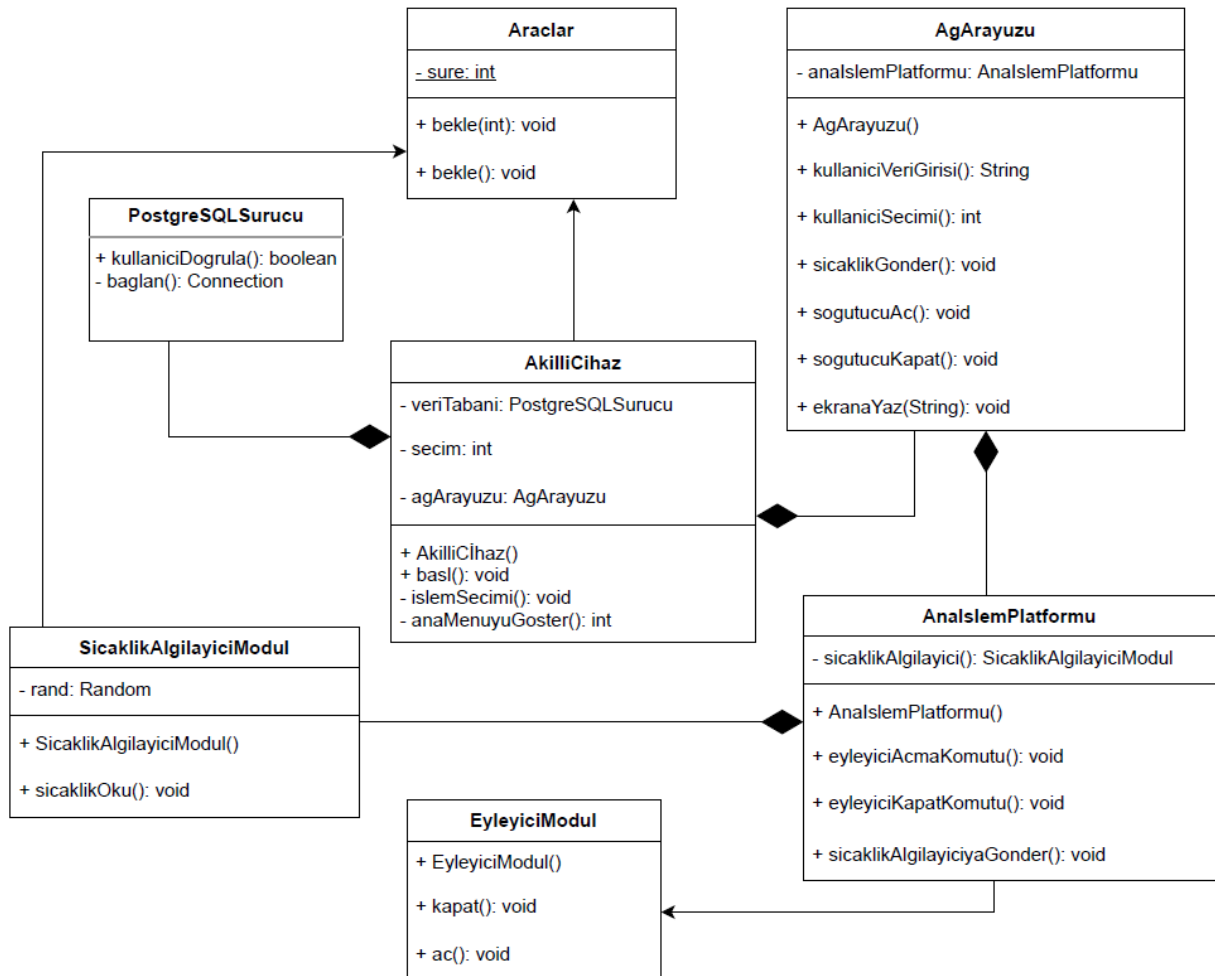
Sıcaklığın Görüntülenmesi(Activity Diagram)



Soğutucunun Çalıştırılması(Activity Diagram)



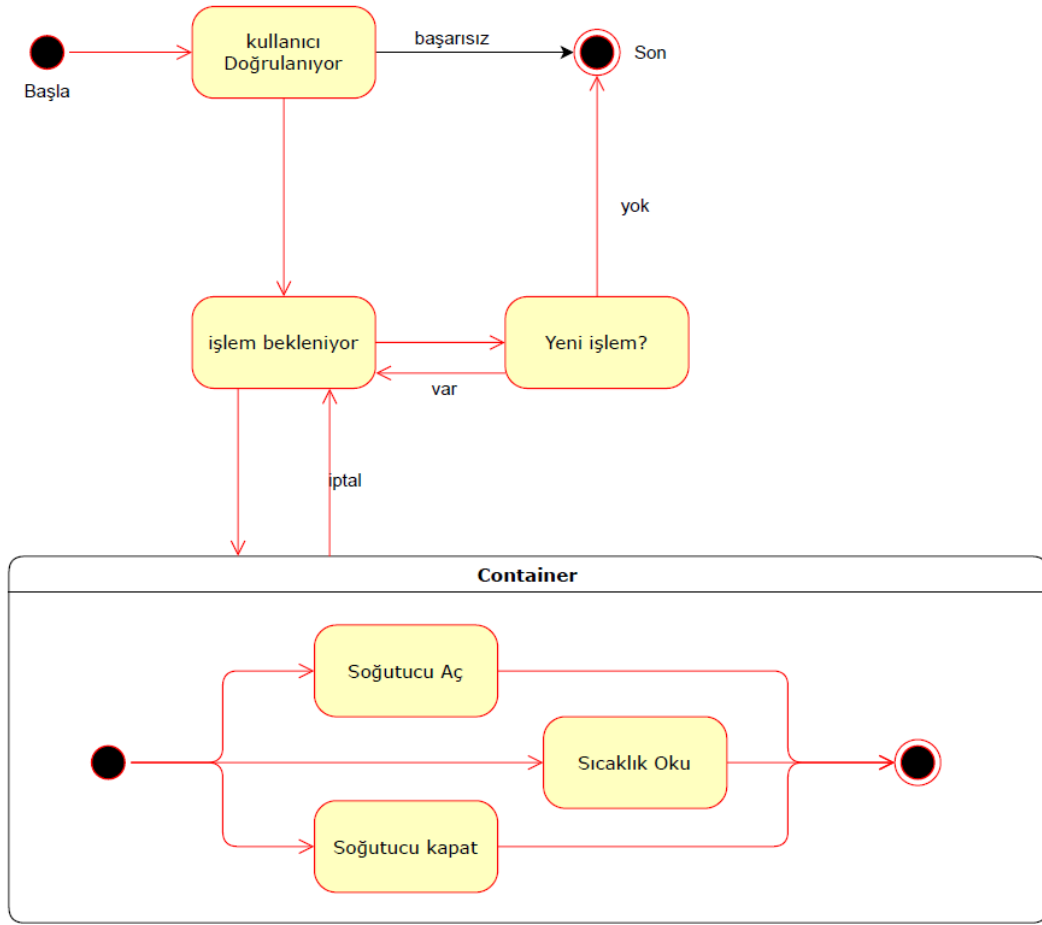
Class Diagram



| AgArayuzu CRC kartı | |
|------------------------------------|-------------------------|
| Sorumluluk | İşbirliği Yapılan Sınıf |
| Kullanıcı veri girişı | Kendisi |
| Mesaj yaz | AnalslemPlatformu |
| Soğutucu açma isteęi gönderilir | AkilliCihaz |
| Soğutucu kapatma isteęi gönderilir | |
| Sıcaklık bilgisi istenir | |

| AnalslemPlatformu CRC kartı | |
|--|-------------------------|
| Sorumluluk | İşbirliği Yapılan Sınıf |
| eyleyiciyi açma komutu çalıştırılır | Kendisi |
| eyleyiciyi kapatma komutu çalıştırılır | AgArayuzu |
| Sıcaklık algılayıcıyı çalıştır | SicaklikAlgılayıcıModul |
| | EyleyiciModul |

State Machine Diagram



Kullanıcı Doğrulama

```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" -ja
Kullanici Bilgileri
Kullanici Adi:yusuf
Sifre:123
Veritabanına bağlandı!
Kullanıcı doğrulandı!
*****
Ana Menu
1-Sıcaklık Goruntule
2-Sogutucu Ac
```


Sıcaklık Görüntüleme

```
1-Sıcaklık Görüntüle
2-Sogutucu Aç
3-Sogutucu Kapat
4-Cıkıs
Seciminiz:
*****
1
Sıcaklık Ölçülüyor
Sıcaklık: 70
Abone1 e gelen mesaj→Sıcaklık 50 derecenin üzerinde, sogutucuyu acin
Abone2 ye gelen mesaj→Sıcaklık 50 derecenin üzerinde, sogutucuyu acin
*****
```

Soğutucunun Açılıp Kapatılması

```
Ana Menu
1-Sıcaklık Görüntüle
2-Sogutucu Aç
3-Sogutucu Kapat
4-Cıkıs
Seciminiz:
*****
2
Cihaz Aciliyor!
Cihaz Durumu: ACIK
```

```
Ana Menu
1-Sıcaklık Görüntüle
2-Sogutucu Aç
3-Sogutucu Kapat
4-Cıkıs
Seciminiz:
*****
3
Cihaz Kapatiliyor!
Cihaz Durumu: KAPALI
```

Veri Tabanı Görüntüsü

| Database: | AkıllıCihaz | Schema: | public | Table: | KullaniciHesabi |
|-----------|-------------|--------------|--------|--------|-----------------|
| | id | kullaniciAdi | sifre | | |
| 1 | 1 | yusuf | 123 | | |
| 2 | 2 | ozaslan | 456 | | |

```
--
-- PostgreSQL database dump
--

-- Dumped from database version 13.4
-- Dumped by pg_dump version 14.0

SET statement_timeout = 0;
SET lock_timeout = 0;
SET idle_in_transaction_session_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = on;
SELECT pg_catalog.set_config('search_path', '', false);
SET check_function_bodies = false;
SET xmloption = content;
SET client_min_messages = warning;
SET row_security = off;

SET default_tablespace = '';

SET default_table_access_method = heap;

--
-- Name: KullaniciHesabi; Type: TABLE; Schema: public; Owner: postgres
--

CREATE TABLE public."KullaniciHesabi" (
    id integer NOT NULL,
    "kullaniciAdi" character varying(40) NOT NULL,
    sifre character varying
);

ALTER TABLE public."KullaniciHesabi" OWNER TO postgres;

--
-- Name: KullaniciHesabi_id_seq; Type: SEQUENCE; Schema: public; Owner: postgres
--

CREATE SEQUENCE public."KullaniciHesabi_id_seq"
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;
```

```

ALTER TABLE public."KullaniciHesabi_id_seq" OWNER TO postgres;

--
-- Name: KullaniciHesabi_id_seq; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE public."KullaniciHesabi_id_seq" OWNED BY
public."KullaniciHesabi".id;

--
-- Name: KullaniciHesabi_id_seq1; Type: SEQUENCE; Schema: public;
Owner: postgres
--

CREATE SEQUENCE public."KullaniciHesabi_id_seq1"
    AS integer
    START WITH 1
    INCREMENT BY 1
    NO MINVALUE
    NO MAXVALUE
    CACHE 1;

ALTER TABLE public."KullaniciHesabi_id_seq1" OWNER TO postgres;

--
-- Name: KullaniciHesabi_id_seq1; Type: SEQUENCE OWNED BY; Schema:
public; Owner: postgres
--

ALTER SEQUENCE public."KullaniciHesabi_id_seq1" OWNED BY
public."KullaniciHesabi".id;

--
-- Name: KullaniciHesabi id; Type: DEFAULT; Schema: public; Owner:
postgres
--

ALTER TABLE ONLY public."KullaniciHesabi" ALTER COLUMN id SET
DEFAULT nextval('public."KullaniciHesabi_id_seq1"'::regclass);

--
-- Data for Name: KullaniciHesabi; Type: TABLE DATA; Schema: public;
Owner: postgres
--

INSERT INTO public."KullaniciHesabi" VALUES
    (1, 'yusuf', '123'),
    (2, 'ozaslan', '456');

```

```
--
-- Name: KullaniciHesabi_id_seq; Type: SEQUENCE SET; Schema: public;
Owner: postgres
--

SELECT pg_catalog.setval('public."KullaniciHesabi_id_seq"', 1,
false);

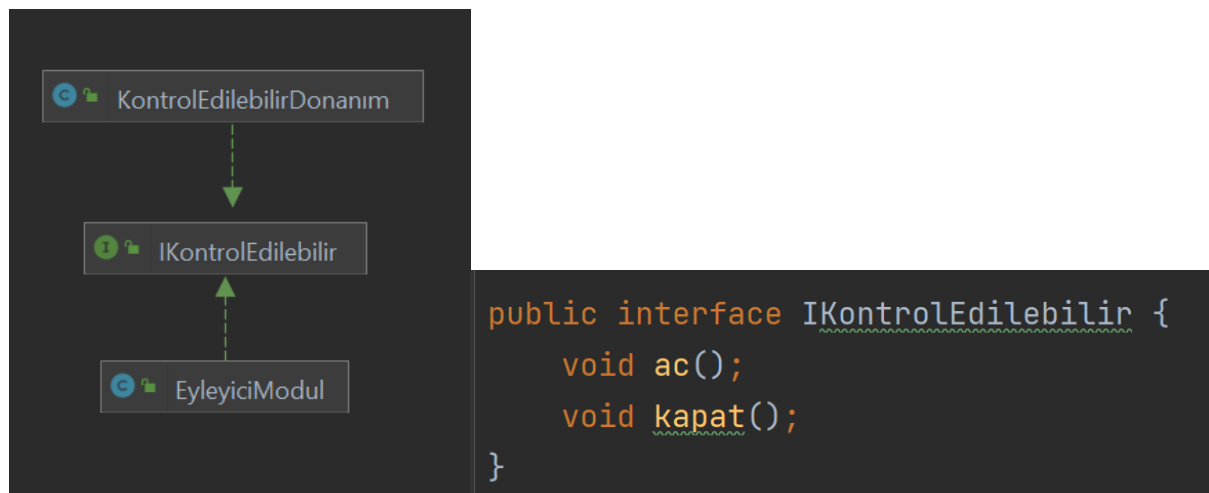
--
-- Name: KullaniciHesabi_id_seq1; Type: SEQUENCE SET; Schema:
public; Owner: postgres
--

SELECT pg_catalog.setval('public."KullaniciHesabi_id_seq1"', 2,
true);

--
-- PostgreSQL database dump complete
--
```

Dependency Inversion Principle

Nesneler arasındaki bir bağlantıda, yüksek seviyeli modül ile düşük seviyeli modül (her ikisi birden) soyutlamaya bağlı olmalı. Her ikisi birbirine doğrudan değil arayüz üzerinden bağlanmalı.

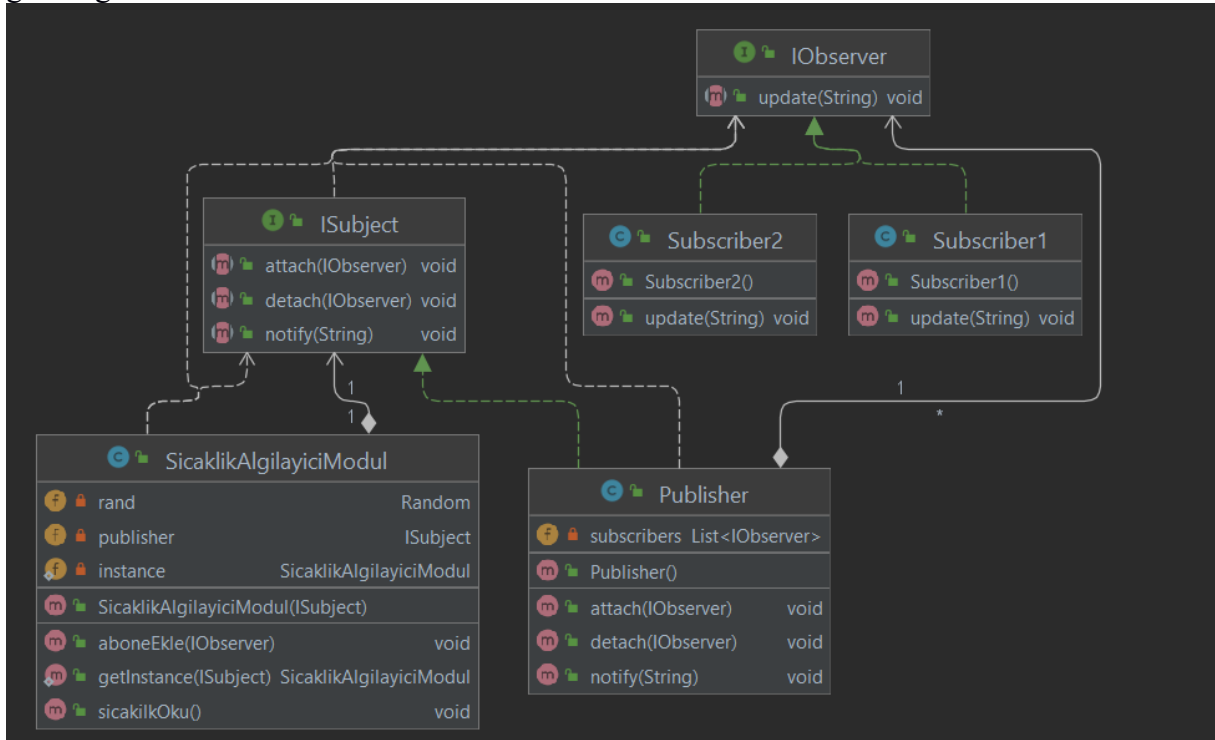


Dependency Inversion ilkesi kullanılarak EyleyiciModul nesnesinin soyutlamaya (IKontrolEdilebilir) bağlı olması sağlanıyor.

Dolayısıyla aralarındaki bağlantı zayıflıyor. Birindeki değişikliğin diğerini etkilemesinin önüne geçilmiş oluyor. KontrolEdilebilirDonanim nesnesinin başka nesnelerle (örneğin; EyleyiciModul, led, su ısıtıcısı vb.) de kullanılabilmesi sağlanıyor (kod tekrar kullanımı artırılıyor).

Observer Design

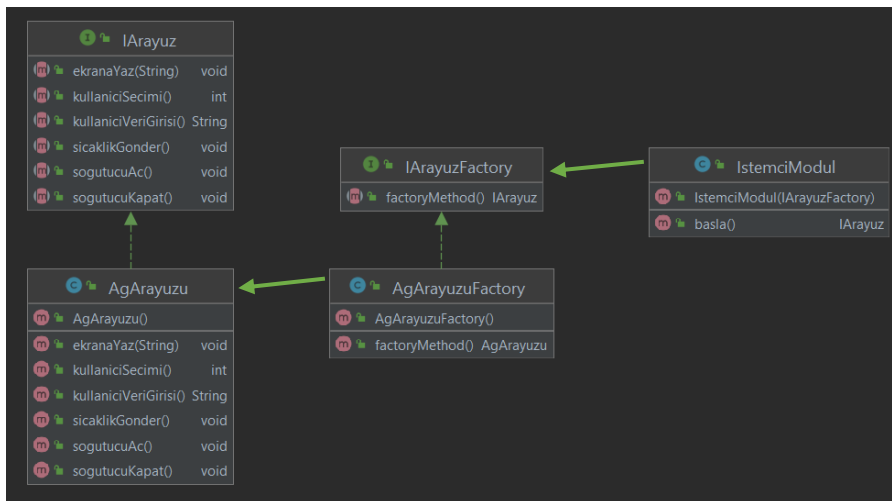
Çok sayıda nesneye, gözlemledikleri nesnede meydana gelen olayı bildirmek gerektiğinde kullanılabilir.



İki kullanıcı sınıfı bulunur(Subscriber1, Subscriber2). SicaklikAlgilyiciModul nesnesinde Publisher nesnesi oluşturulur. Kullanıcılar bu Publisher nesnesinde kaydedilir. Okunan sıcaklık değerine göre kullanıcılara mesaj gönderilir.

Factory Method

Nesne oluşturmayla (creational) ilgili desenlerden biridir. Bir sınıftan nesne oluşturmak gerektiğinde, bu sorumluluğu istemci koddan ayırmak (kapsülleme/SRP) için kullanılır.



AgArayuzu nesnesi oluşturulurken new methodu yerine factory method çağrılarak nesne oluşturulur. Böylece istemci kod, nesne oluşturmayla ilgili herhangi bir değişiklikten etkilenmez

