

# SAKARYA ÜNİVERSİTESİ

## BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

### BİLGİSAYAR MÜHENDİSLİĞİ

İsim:Furkan

Soyisim:Tektaş

Numara:B181210049

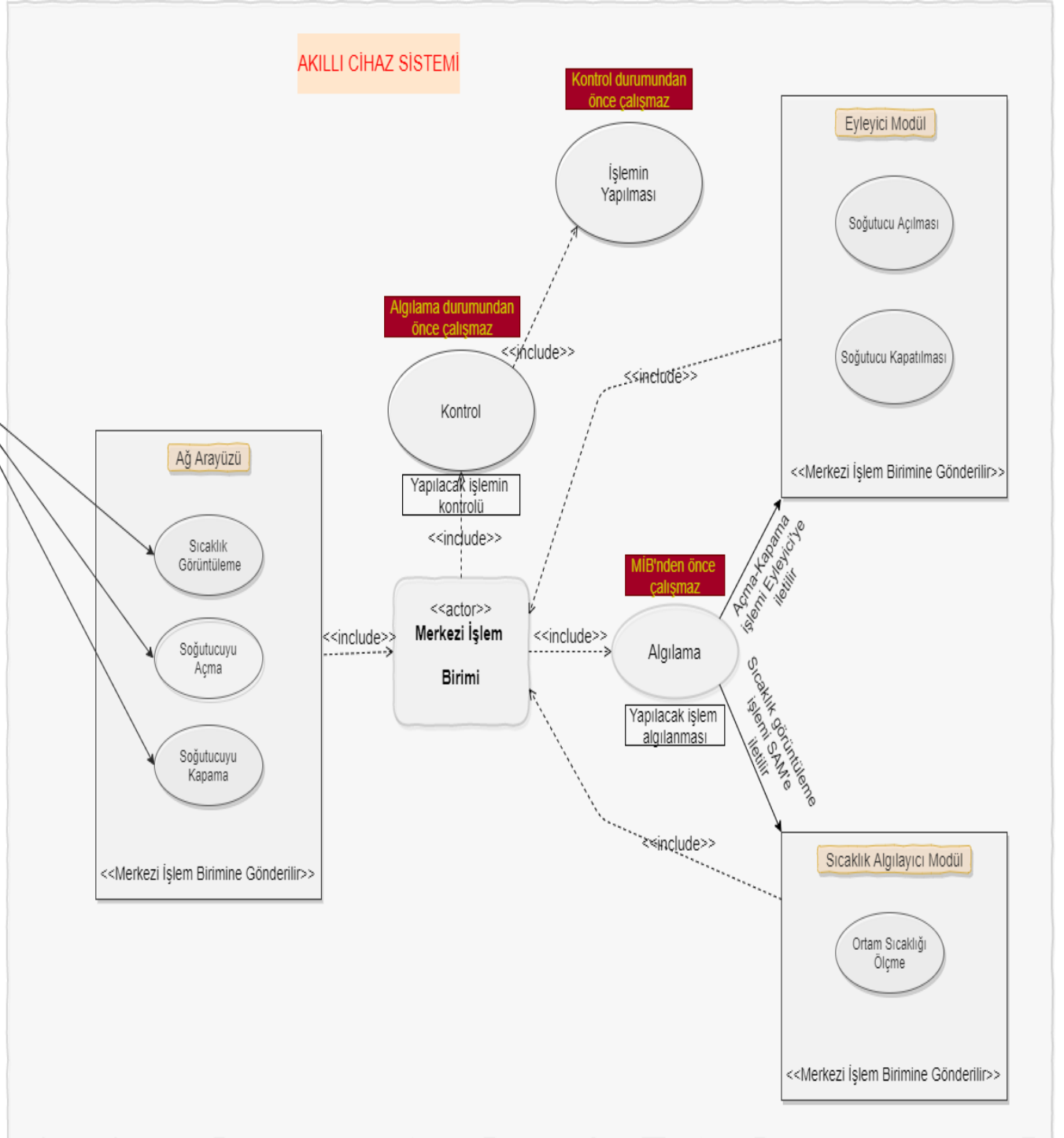
Sınıf:2

Öğretim Türü:1. Öğretim

Şube:21B(2. Sınıf,1. Öğretim,B grubu)

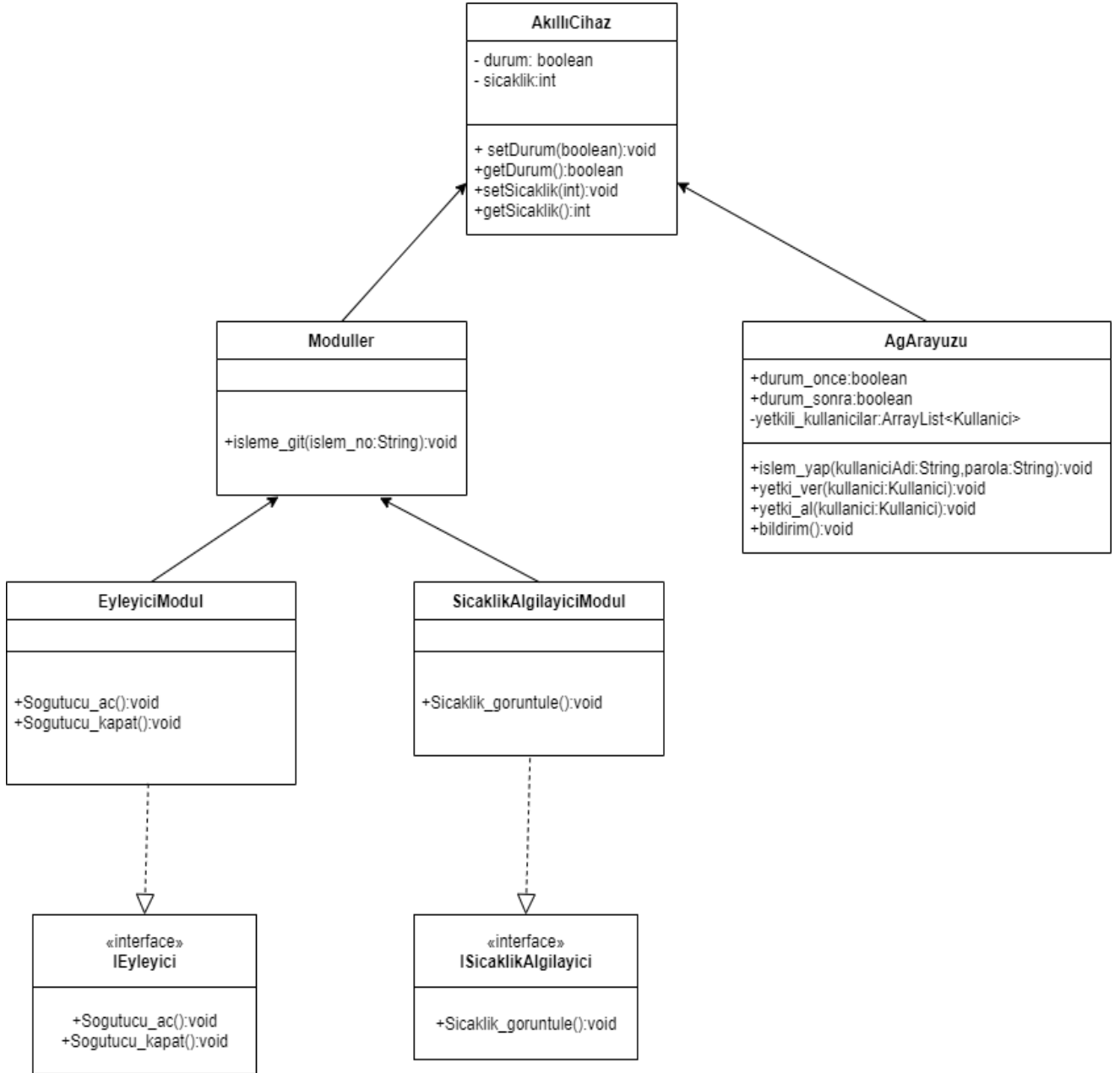
E-Posta:furkan.tektas@ogr.sakarya.edu.tr

# USE-CASE DIAGRAM

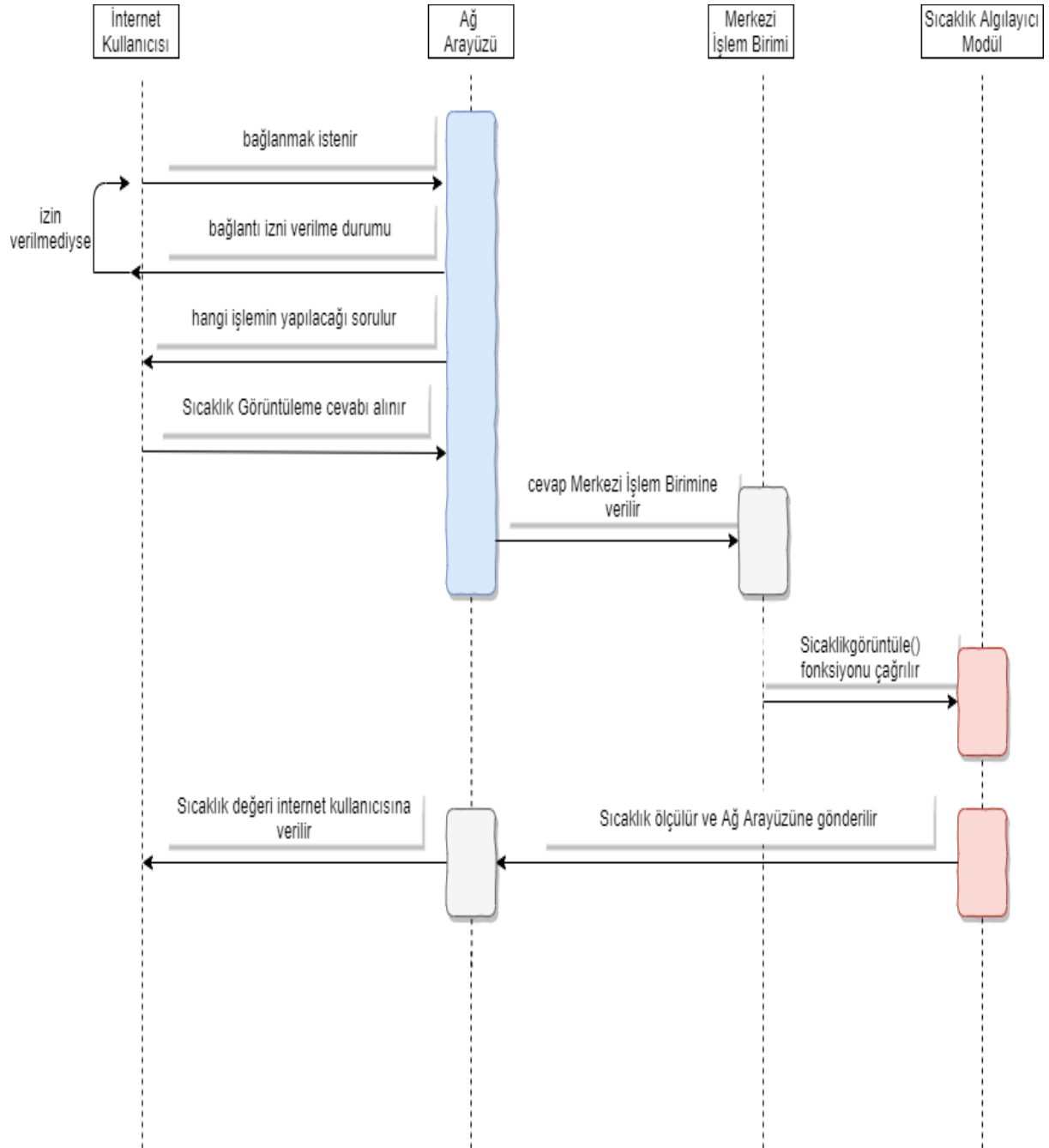


Sıcaklık görüntüleme işlemi yalnızca soğutucu açık ise gerçekleştirilebilir. Soğutucu sıcaklığı -10 ile 20 SANTIĞRAT derece arasında değişebilmektedir. Soğutucu açıldığında ya kapanabilir ya da sıcaklığı görüntülenebilir.

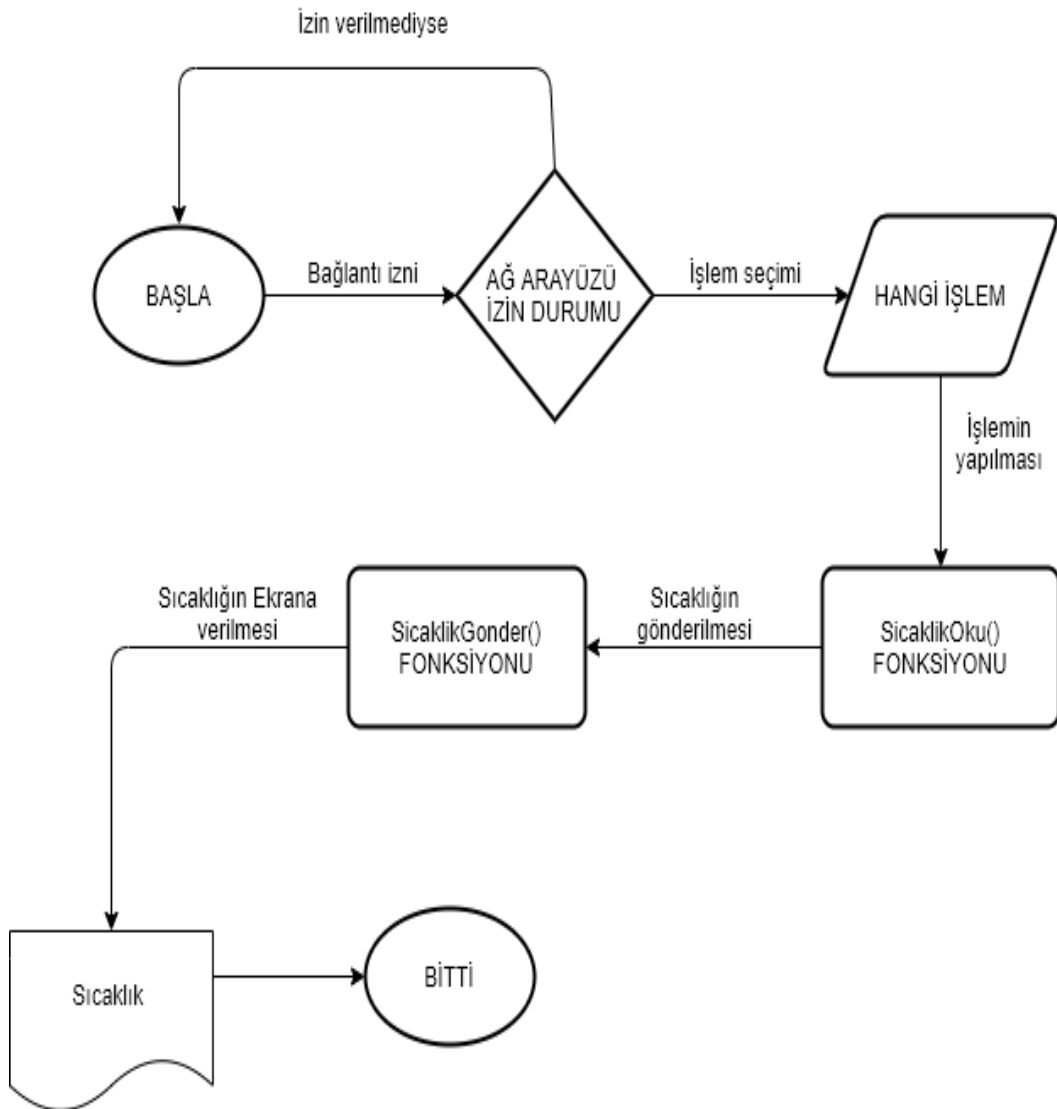
## SINIF ŞEMASI



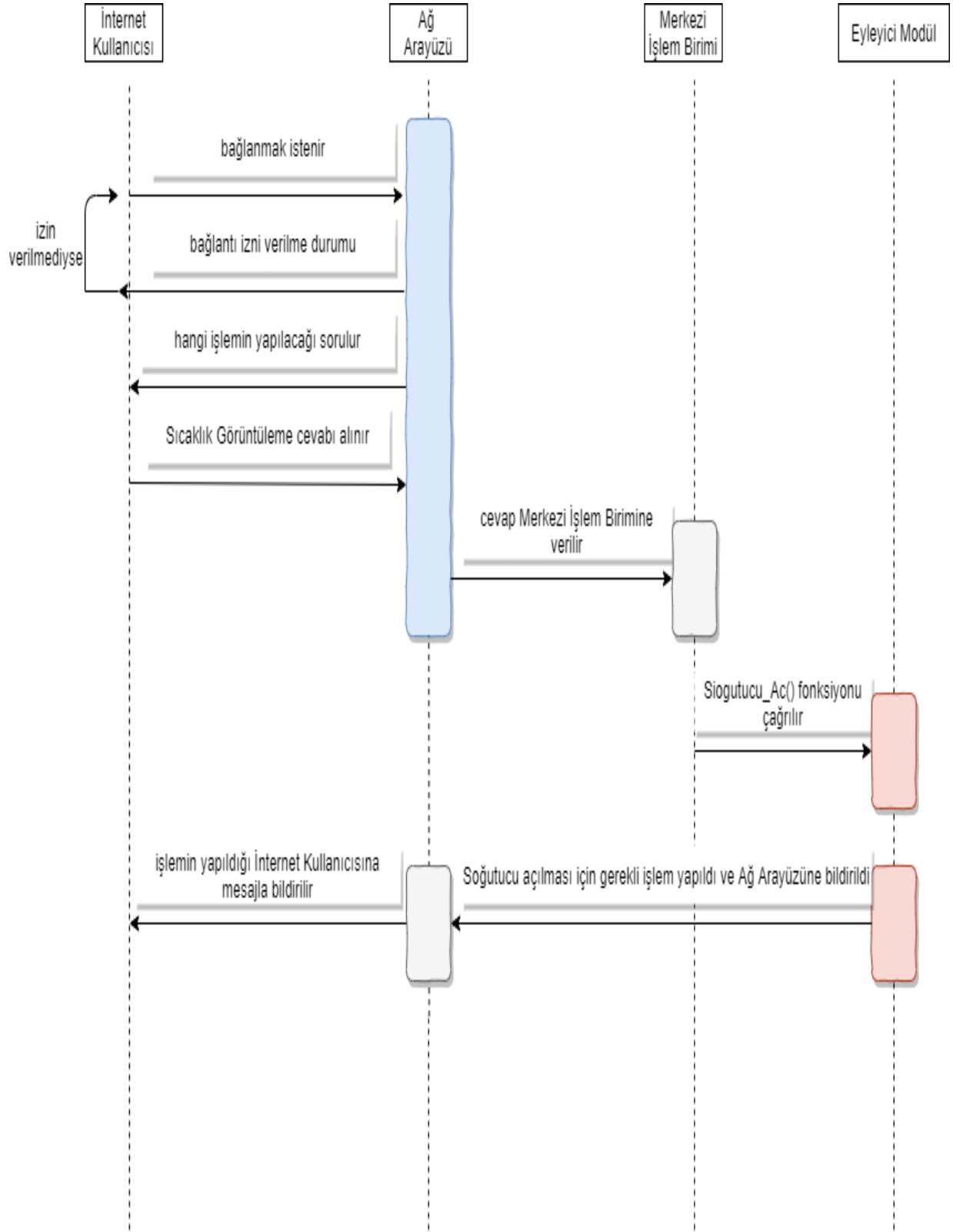
## SICAKLIK GÖRÜNTÜLEMESİ-SUQUENCE DIAGRAM



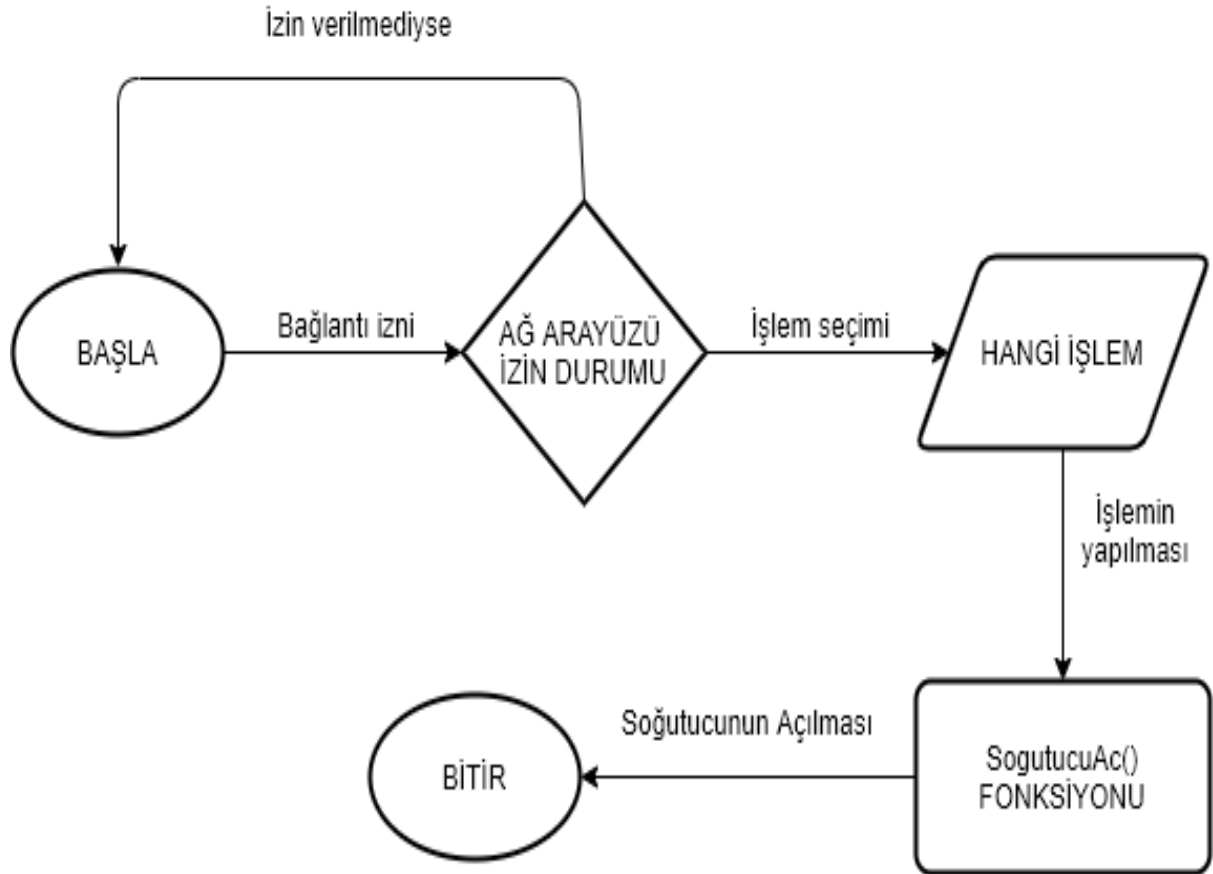
## SICAKLIK GÖRÜNTÜLEMESİ-ACTİVİTY DİAGRAM



## SOĞUTUCU ÇALIŞTIRILMASI-SEQUENCE DIAGRAM

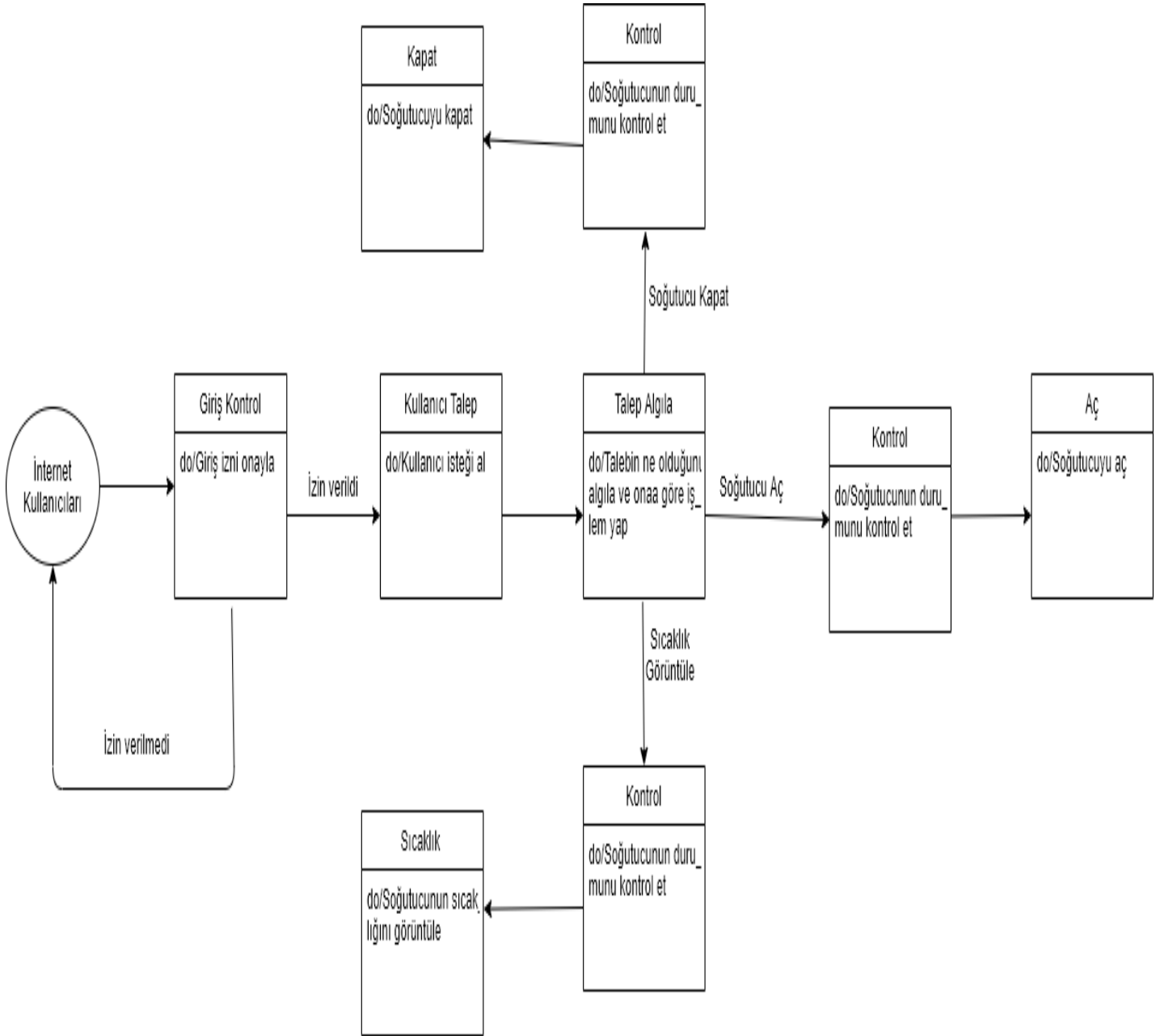


## SOĞUTUCU ÇALIŞTIRILMASI-ACTİVİTY DİAGRAM

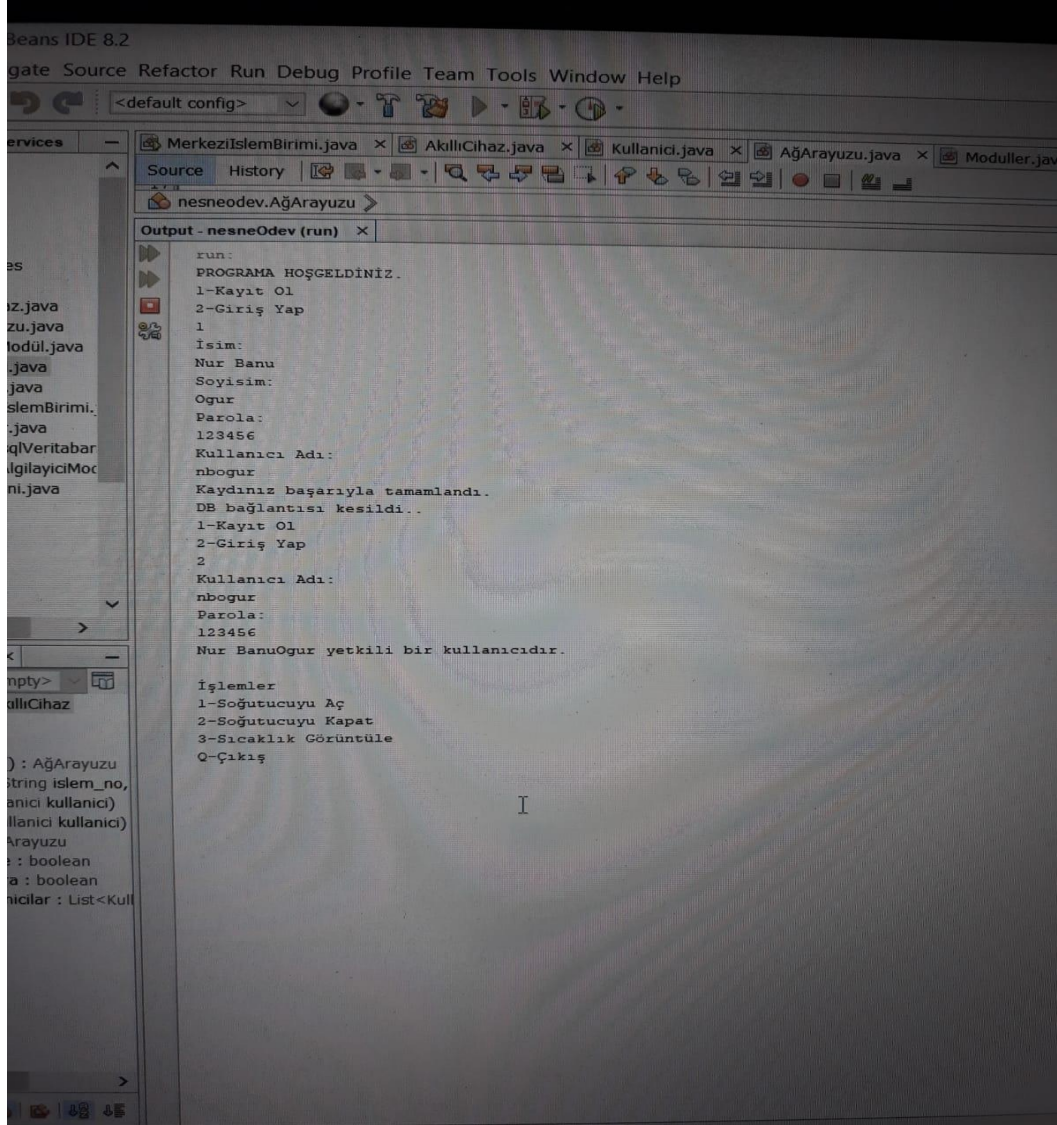




## SİSTEMİN DURUM DİAGRAMI



# KULLANICI DOĞRULAMA EKRANI VE AÇIKLAMASI



```
Run:
PROGRAMA HOŞGELDİNİZ.
1-Kayıt Ol
2-Giriş Yap
1
İsim:
Nur Banu
Soyisim:
Ogur
Parola:
123456
Kullanıcı Adı:
nbogur
Kaydınız başarıyla tamamlandı.
DB bağlantısı kesildi..
1-Kayıt Ol
2-Giriş Yap
2
Kullanıcı Adı:
nbogur
Parola:
123456
Nur BanuOgur yetkili bir kullanıcıdır.

İşlemler
1-Soğutucuyu Aç
2-Soğutucuyu Kapat
3-Sıcaklık Görüntüle
Q-Çıkış
```

BU KISIMDA KULLANICININ SİSTEME GİRİŞ YAPABİLMESİ İÇİN ÖNCELİKLE SİSTEME KAYDOLMASI GEREKMEKTEDİR.GEREKLİ BİLGİLERİ GİRDİĞİ ZAMAN, POSTGRESQL VERİTABANININ YETKİLİ KULLANICI TABLOSUNA BİLGİLERİYLE BİRLİKTE KAYIT OLMUŞ OLUR.VE SİSTEME GİREBİLME YETENEĞİ KAZANMIŞ OLUR.GİRİŞ YAPTIĞI TAKDİRDE İSE ÖNÜNE YAPABİLECEĞİ BİRKAÇ SEÇENEK SUNULUR.BU SEÇENEKLERDEN İSTEDİĞİNİ YAPABİLME YETKİSİNE SAHİPTİR.

# SICAKLIĞIN GÖRÜNTÜLENMESİ VE SOĞUTUCUNUN AÇILIP KAPATILMASIYLA İLGİLİ EKRAN GÖRÜNTÜLERİ VE AÇIKLAMASI

```
Output - nesneOdev (run) x
123456
Nur BanuOgur yetkili bir kullanıcıdır.

İşlemler
1-Soğutucuyu Aç
2-Soğutucuyu Kapat
3-Sıcaklık Görüntüle
0-Çıkış
1
Soğutucuyu açıldı.

Hey Nur Banu soğutucu açıldı
Hey Celal soğutucu açıldı
Hey Furkan soğutucu açıldı
Hey ali soğutucu açıldı
İşlemler
1-Soğutucuyu Aç
2-Soğutucuyu Kapat
3-Sıcaklık Görüntüle
0-Çıkış
3
Sıcaklık: 6
İşlemler
1-Soğutucuyu Aç
2-Soğutucuyu Kapat
3-Sıcaklık Görüntüle
0-Çıkış
2
Soğutucuyu kapandı.

Hey Nur Banu soğutucu kapandı
Hey Celal soğutucu kapandı
Hey Furkan soğutucu kapandı
Hey ali soğutucu kapandı
İşlemler
1-Soğutucuyu Aç
2-Soğutucuyu Kapat
3-Sıcaklık Görüntüle
0-Çıkış
3
Soğutucu kapalı olduğu için bu işlem gerçekleştirilemiyor.

İşlemler
1-Soğutucuyu Aç
2-Soğutucuyu Kapat
3-Sıcaklık Görüntüle
0-Çıkış
2
Soğutucu zaten kapalı.

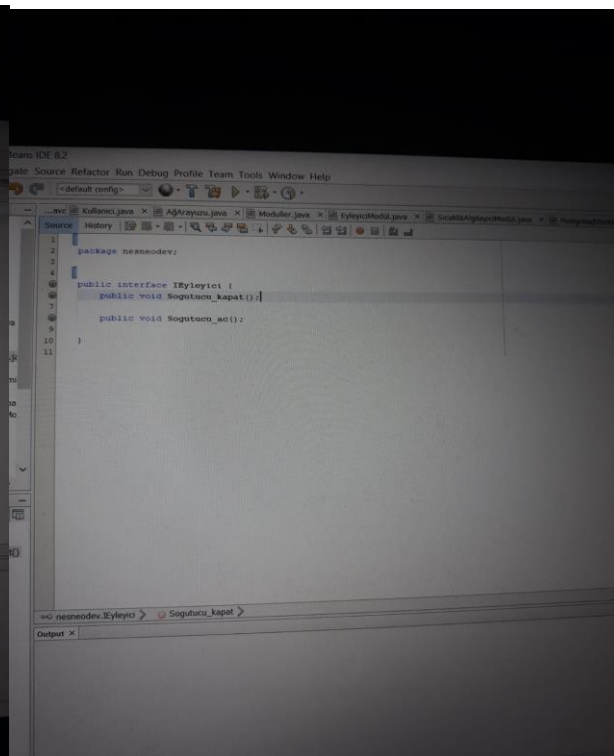
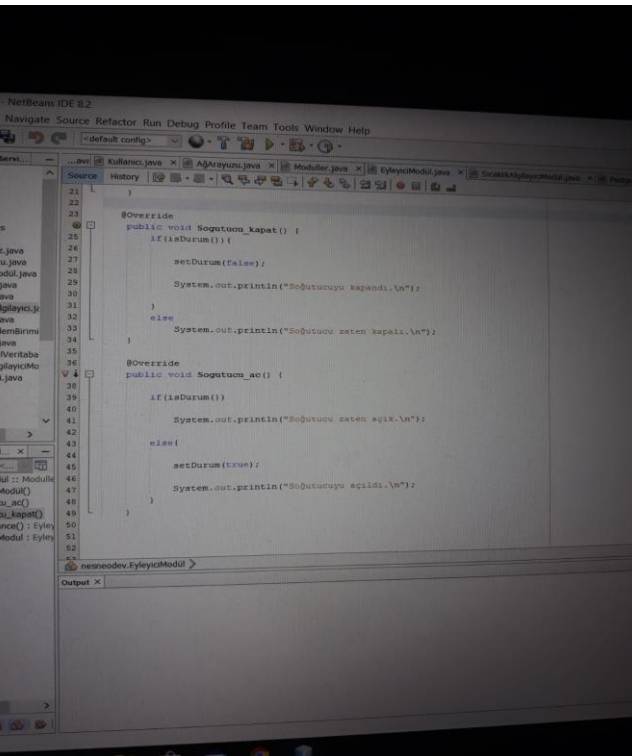
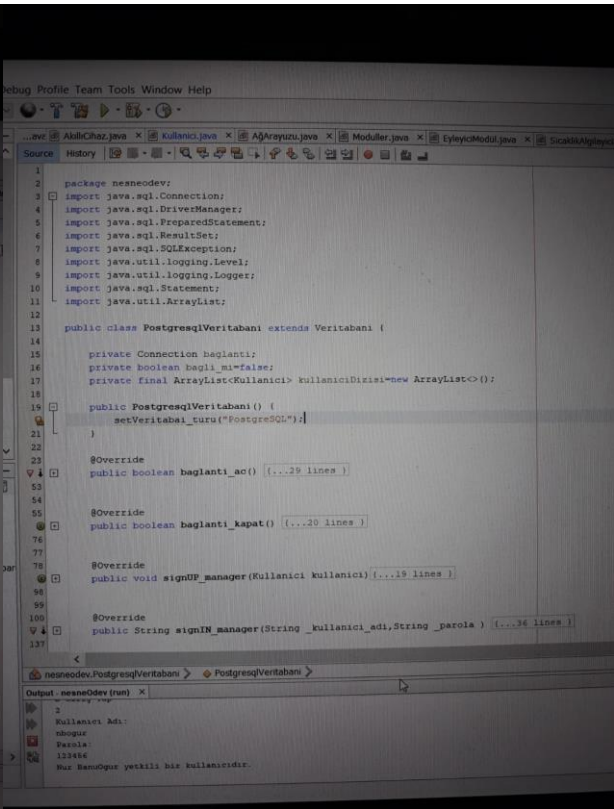
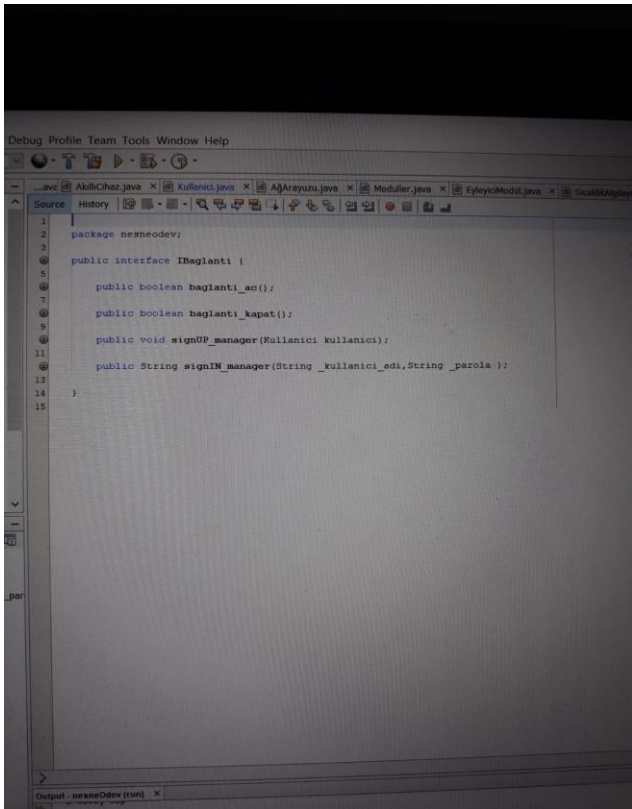
İşlemler
```

BU KISIMDA İSE YETKİLİ KULLANICI SİSTEMİ AÇABİLME,KAPAYABİLME VE SİSTEMİN SICAKLIĞINI GÖRÜNTÜLEYEBİLME YETKİLERİNE SAHİPTİR.ANCAK SİSTEM KAPALI DURUMDA OLDUĞUNDA SICAKLIĞI GÖRÜNTÜLEYEMEZ.AYRICA BURADA BİR KULLANICI BİR İŞİ YAPTIĞINDA BUNUN BİLGİSİ DİĞER KULLANICILARA DA GİTMEKTEDİR.BU OBSERVER TASARIM DESENİYLE SAĞLANMIŞTIR.ÖRNEKTE GÖRÜLDÜĞÜ GİBİ NUR BANU İSİMLİ KULLANICININ YAPMIŞ OLDUĞU

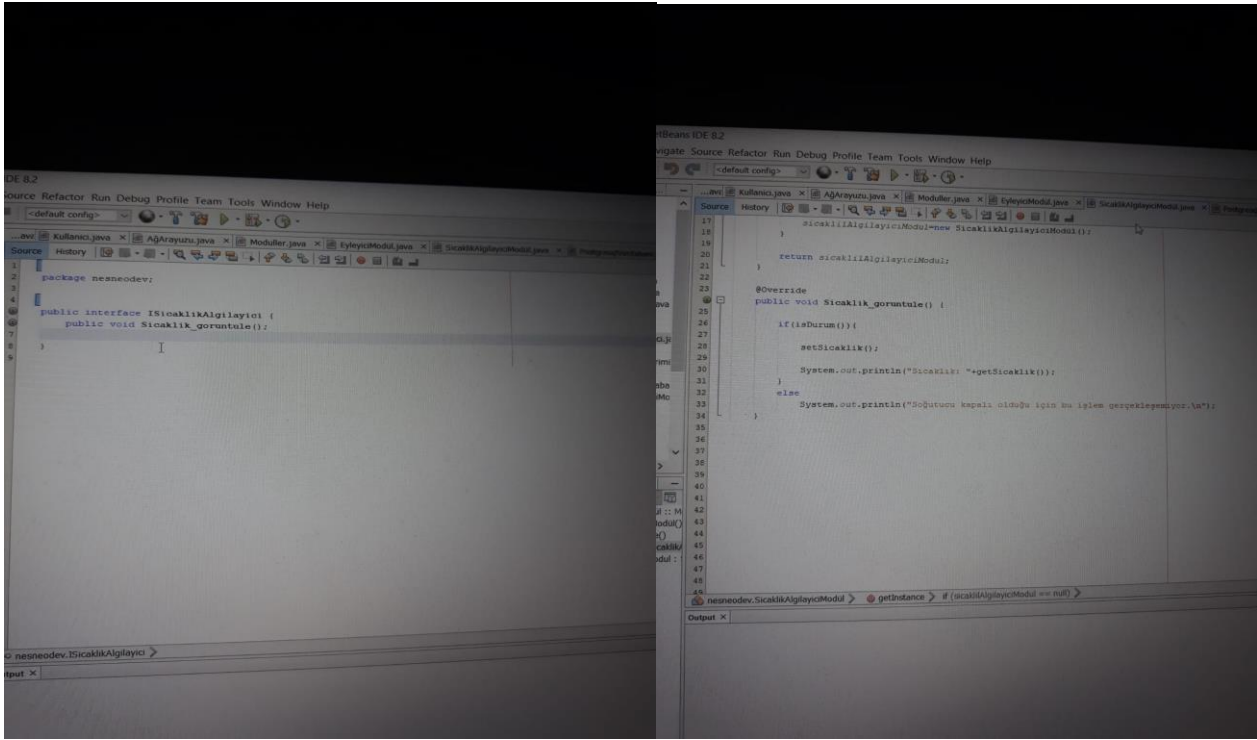
İŞLERDEN DİĞER KULLANICILAR DA HABER ALMAKTADIRLAR.VE SİSTEM EĞER AÇIK VEYA KAPALI DURUMDA İSE YETKİLİ KULLANICI BU SİSTEMİ TEKRARDAN AÇAMAZ VEYA KAPATAMAZ.BUNLAR KONTROLLER İLE SAĞLANMAKTADIR VE UYARI MESAJLARI ALMAKTADIRLAR.

## “Open/Closed” İLKESİNİN NE OLDUĞU VE UYGULAMA İÇERİSİNDE NASIL GERÇEKLENDİĞİ

Bir sınıf ya da fonksiyon var olan özellikleri korumalı yani davranışını değiştirmiyor olmalı ve yeni özellikler kazanabilmelidir. Gereksinimler değiştiğinde, yeni gereksinimlerin karşılanabilmesi için bir sınıfa yeni veya farklı davranışlar eklenebilir olmasıdır.Uygulama içerisinde ise bu interface ile sağlanmıştır.Yani bir programcı bu sistemi Postgresql veritabanıyla kullanmak yerine başka bir veritabanı kullanmak isteseydi yapması gereken şey interface sınıfını implement etmesi ve Veritabanı sınıfını extends etmesi olacaktı.Bu ilke bir yazılımın sürdürülebilir olmasını sağlar.Yani programcı bu veritabanına yeni bir özellik eklemek istediğinde mevcut sisteme müdahale etmeden interface ve kalıtım yoluyla rahatlıkla bunu yapabilmektedir.Ayrıca programcı Eyleyici modül'e veya Sıcaklık Algılayıcı Modül'e yeni bir özellik kazandırmak istediğinde de bunu metodlar yardımıyla yapabilmektedir.Veyahut bir özelliği kaldırmak istediğinde de bu özelliğe ait fonksiyonları silerek bunu yapabilir.Yani bir değişiklik yapılacaksa eğer bunu tüm kodları değiştirmek yerine sadece o değişiklik ile alakalı olan yerler değiştirilmelidir.Aşağıdaki görsellerde bu ilkenin uygulamada nasıl kullanıldığı gösterilmektedir.







## “Singleton” ve “Observer” DESENLERİNİN NE OLDUĞU VE UYGULAMA İÇERİSİNDE NASIL GERÇEKLENDİĞİ

**Singleton Tasarım Deseni;** yazılım mühendisliğinde bir sınıf tipinden sadece tek bir nesnenin yaratılmasına izin veren bir yazılım tasarım örüntüsüdür. Bu kavram bazen daha da geliştirilerek sınıfın belirli sayıda nesnesinin yaratılması şeklinde genelleştirilebilir. Örneğin, oluşturulacak nesne sayısı beş ile sınırlanabilir.

**Observer Tasarım Deseni;**özne olarak adlandırılan bir nesnenin, bağımlıları olan, gözlemci adı verilen bir listesini tuttuğu ve genellikle yöntemlerinden birini çağırarak durum değişikliklerini otomatik olarak bildirdiği bir yazılım tasarım modelidir.

**Signleton Tasarım Deseni;**programımızda Akıllı cihaz ve alt alanlarını(Ağ Arayüzü ve Moduller) bir kez oluşturma amacıyla kullanılmıştır.Yani her bir işlem(Soğutucu açma-kapama vs.) yapılırken yeni bir akıllı cihaz ve alt alan oluşturmamak amacıyla bu tasarım deseni önemli bir görev almıştır.Ayrıca sınıfların constructor'ı da protected yapılmıştır.Aslolan private olmasıdır ancak sınıflardan kalıtım alındığı için private buna engel olmaktadır.Bunu önlemenin yolu da protected yapmaktır.Aşağıda bu kalıbın program içinde nasıl kullanıldığına dair görseller vardır...

```

public class AğArayuzu extends AkıllıCihaz {
    private final List<Kullanici> yetkili_kullanicilar;
    private boolean durum_once, durum_sonra;

    protected AğArayuzu() {
        super();
    }

    private static AğArayuzu arayuz=null;

    public static AğArayuzu getInstance() {
        if(arayuz==null) {
            arayuz=new AğArayuzu();
            Moduller.getInstance();
        }
        return arayuz;
    }

    public void islemi_yap(String islem_no,String
        durum_sonra):
}

package nesneodev;

public class Moduller extends AkıllıCihaz {
    protected Moduller() {
        super();
    }

    private static Moduller moduller=null;

    public static Moduller getInstance() {
        if(moduller==null) {
            moduller=new Moduller();
            SicaklikAlgılayıcıModül.getInstance();
            EyleyiciModül.getInstance();
        }
        return moduller;
    }

    public void isleme_git(String islem_no) {
        if(null!=islem_no)
            switch (islem_no) {
                case "1":

```

```

Profile Team Tools Window Help
package nesneodev;

public class EyleyiciModül extends Moduller {
    private EyleyiciModül() {
        super();
    }

    private static EyleyiciModül eyleyiciModul=null;

    public static EyleyiciModül getInstance() {
        if(eyleyiciModul==null) {
            eyleyiciModul=new EyleyiciModül();
        }
        return eyleyiciModul;
    }

    public void Soğutucu_kapat() {
        if(isDurum()) {
            setDurum(false);
            System.out.println("Soğutucuyu kapandı.\n");
        }
        else
            System.out.println("Soğutucu zaten kapalı.\n");
    }
}

Profile Team Tools Window Help
package nesneodev;

public class SicaklikAlgılayıcıModül extends Moduller {
    private SicaklikAlgılayıcıModül() {
        super();
    }

    private static SicaklikAlgılayıcıModül sicaklikAlgılayıcıModul=null;

    public static SicaklikAlgılayıcıModül getInstance() {
        if(sicaklikAlgılayıcıModul==null) {
            sicaklikAlgılayıcıModul=new SicaklikAlgılayıcıModül();
        }
        return sicaklikAlgılayıcıModul;
    }

    public void Sicaklik_goruntule() {
        if(isDurum()) {
            setSicaklik();
            System.out.println("Sicaklık: "+getSicaklik());
        }
        else
            System.out.println("Soğutucu kapalı olduğu için bu işlem geçerli

```

**Observer Tasarım Deseni** ;programımızda yetkili herhangi bir kullanıcının yapmış olduğu değişikliği diğer yetkili kullanıcılara haber vermesi amacıyla kullanılmıştır. Aşağıda bu kalıbın program içinde

nasıl kullanıldığına dair görseller vardır.Sistemde bir değişiklik yapıldığında yani sistem kapalıyken açık duruma getirildiğinde herkese bunun bilgisi gitmektedir.Veya sistem açık durumdayken kapalı duruma getirildiğinde de bunun bilgisi gitmektedir.Ancak sistem kapalı durumdayken kapatılmaya çalışıldığında bu bir değişiklik olmadığı için kimseye böyle bir bilgi gitmeyecektir.

```
public void islemi_yap(AğArayuzu ağarayuzu,String islem_no){
    ağarayuzu.islemi_yap(islem_no,getName());
}

public void Update(boolean durum){
    if(durum==true){
        System.out.println("Hey "+getName()+" soğutucu açıldı");
    }
    else{
        System.out.println("Hey "+getName()+" soğutucu kapandı");
    }
}

neodev.Kullanici > Update > if (durum == true) >
nesneOdev (run) X
Kullanici Adı:
soğutucu:
Pazara:
123456
Bur BanuOgur yetkili bir kullanıcıdır.
```

```
public void islemi_yap(String islem_no,String yapan_kisi){
    durum_once=isdurum();
    Moduller.getInstance().isleme_git(islem_no);
    durum_sonra=isdurum();
    if(durum_once!=durum_sonra){
        bildirim();
    }
}

public void yetki_ver(Kullanici kullanici){
    yetkili_kullanicilar.add(kullanici);
}

public void yetki_al(Kullanici kullanici){
    yetkili_kullanicilar.remove(kullanici);
}

public void bildirim(){
    yetkili_kullanicilar.forEach((k) -> {
        k.Update(durum_sonra);
    });
}

neodev.AğArayuzu >
nesneOdev (run) X
Kullanici Adı:
soğutucu:
Pazara:
123456
Bur BanuOgur yetkili bir kullanıcıdır.
```

Github proje linki:<https://github.com/furkan177/Nesne-Yonelimli-Analiz>

Youtube video

linki:<https://www.youtube.com/watch?v=sEHDCjf3bZ>

Q



Configurations

Queries

Tables

Views

Columns

Constraints

Indexes

Triggers

Actions

id

[PK] integer

name

character varying (10)

surname

character varying (10)

password

character varying (10)

kullanici\_adi

character varying (30)

1	529171	Nur Banu	Ogur	123456	nbogur
2	905174	Celal	Ceken	123	cceken
3	-91030	Furkan	Tekta?	12	ftektas
4	-256648	ali	gulbag	145	agulbag

Windows Taskbar

Icons: File Explorer, Mail, Google Chrome, VS Code