

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

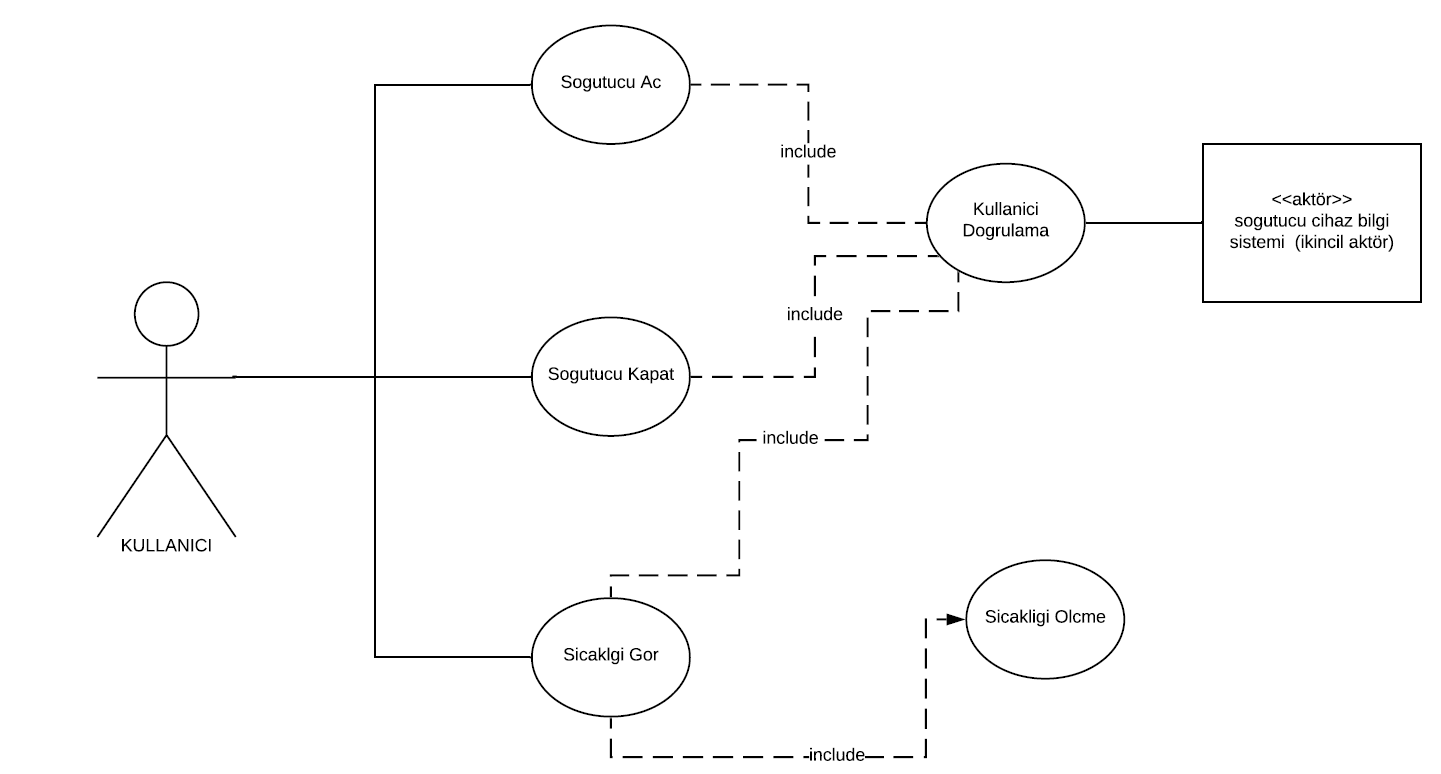
Nesne Yönelimli Analiz ve Tasarım

**PROJE/TASARIM**

**Hazırlayan:**

**B171210082 - Emre Akcan 2. Sınıf 2. Öğretim A grubu**

**USE CASE DIAGRAM**



**“SOĞUTUCUNUN ÇALIŞTIRILMASI DURUMU”**

Eşsiz bir ad “Soğutucunun çalıştırılması”

* Soğutucunun başlatılması sürecini tanımlar
* 20.04.2020 v1.1 Emre Akcan

**İlgili Aktörler:** Kullanıcı, Cihaz aygıtları

**Giriş Koşulu:** Soğutucu başlatılma isteği

**Çıkış Koşulu:** Soğutucunun başlatılması

**Olay Akışı:**

1. Kullanıcı soğutucu cihazı çalıştırır.
2. Ekrana kullanıcı adı ve şifre girilmesini istenen bir mesaj gönderilir.
3. Kullanıcı, kullanıcı adı ve şifresini girer.
4. Cihaz klavyeden girilen değerleri alır.
5. Kullanıcı doğrulama yapmak için cihaza istekte bulunur.
6. Soğutucu isteği kabul eder ve gerekli işlemleri başlatır.
7. Kullanıcı soğutucuyu başlatma isteği gönderir.
8. Soğutucu başlatılır.

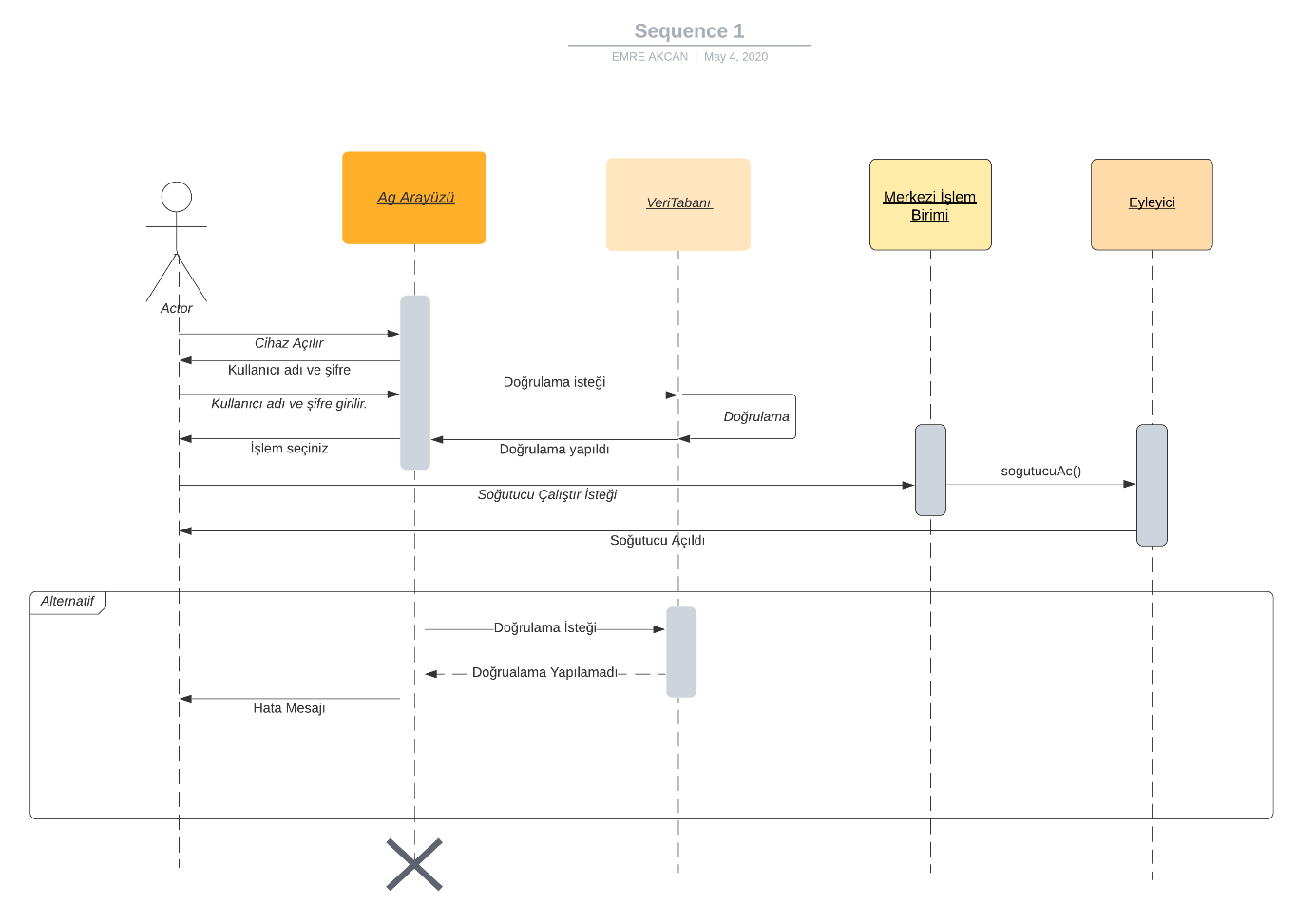
**Alternatif Olay Akışı:**

A1.Kullanıcı adı ve şifre yanlış

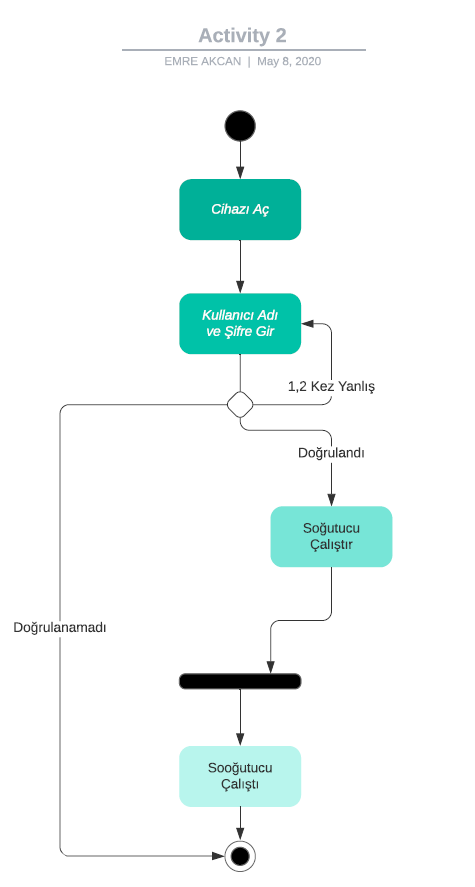
7. 3 kez yanlış girildi.

8. Çıkış.

**SOĞUTUCU ÇALIŞTIRILMASI SEQUENCE DIAGRAM**



**SOĞUTUCUNUN ÇALIŞTIRILMASI AKTİVİTE DİYAGRAMI**

****

**“SICAKLIGIN GÖRÜNTÜLENMESİ DURUMU”**

Eşsiz bir ad “Sıcaklığın görüntülenmesi”

* Sıcaklığın görüntülenme sürecini tanımlar
* 20.04.2020 v1.0 Emre Akcan

**İlgili Aktörler:** Kullanıcı, Fiziksel Ortam(Hava), Cihaz aygıtları

**Giriş Koşulu:** Sıcaklık görüntüleme isteği

**Çıkış Koşulu:** Sıcak görüntülenmesi

**Olay Akışı:**

1. Kullanıcı soğutucu cihazı çalıştırır.
2. Ekrana kullanıcı adı ve şifre girilmesini istenen bir mesaj gönderilir.
3. Kullanıcı, kullanıcı adı ve şifresini girer.
4. Cihaz klavyeden girilen değerleri alır.
5. Kullanıcı doğrulama yapmak için cihaza istekte bulunur.
6. Soğutucu doğrulama yapar ve gerekli işlemleri başlatır.
7. Kullanıcı sıcaklık görüntülemek ister.
8. Cihaz, algılama ve kontrol durumuna geçer.
9. Sıcaklık ölçülür.
10. Sıcaklık görüntülenir.

**Alternatif Olay Akışı:**

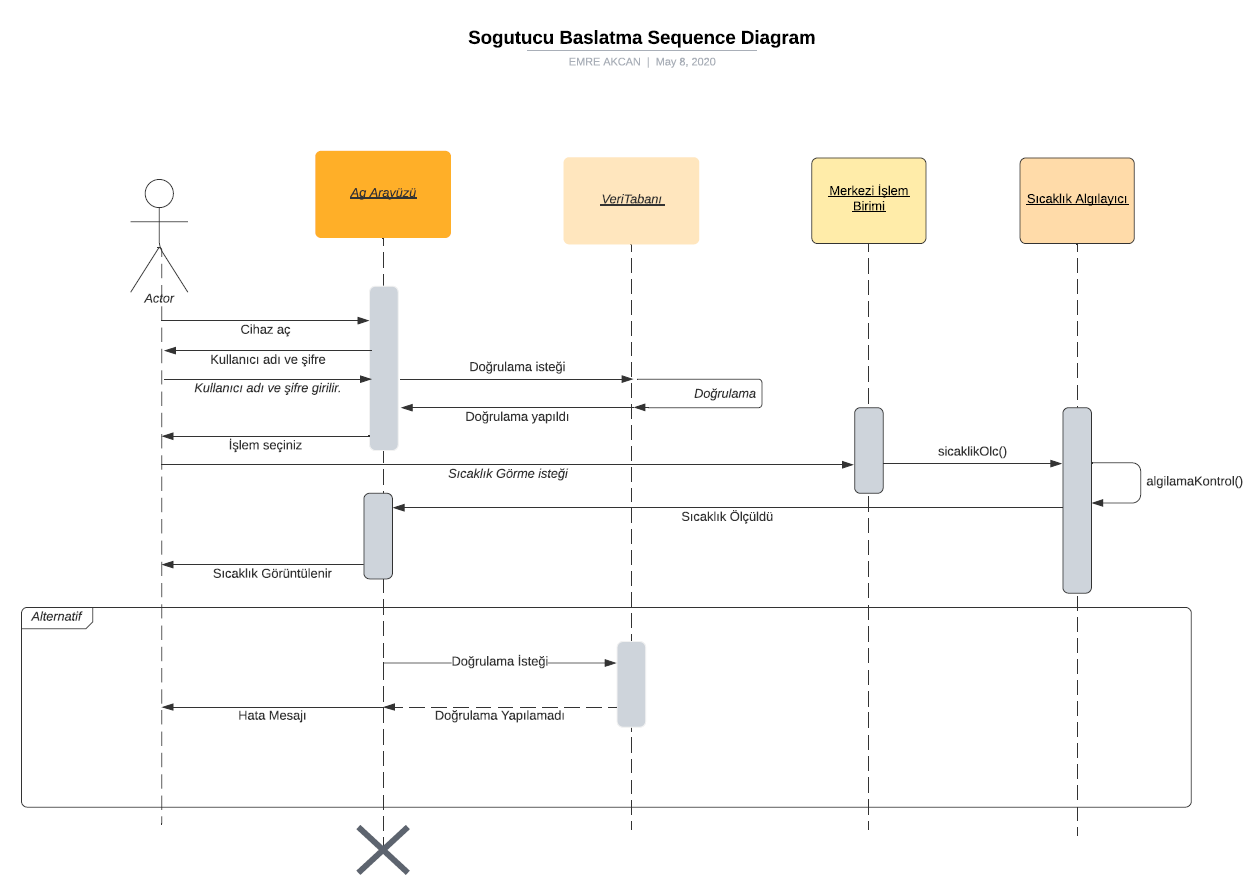
1. Kullanıcı adı ve şifre yanlış

6. 3 kez yanlış girildi.

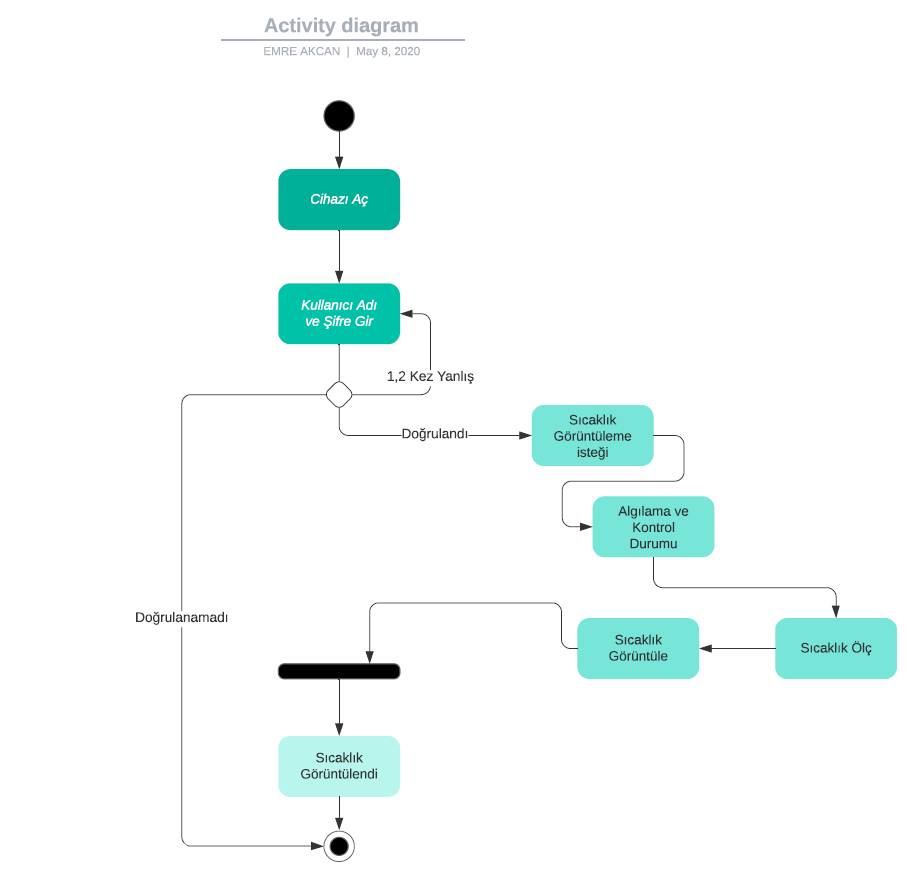
7. Çıkış.

**Özel Gereksinimler:** Cihaz 24 saat çalışır. İstenilen sıcaklığa ulaştığında bekleme durumuna geçer. Soğutucu açılmadan herhangi bir işlem yapılamaz.

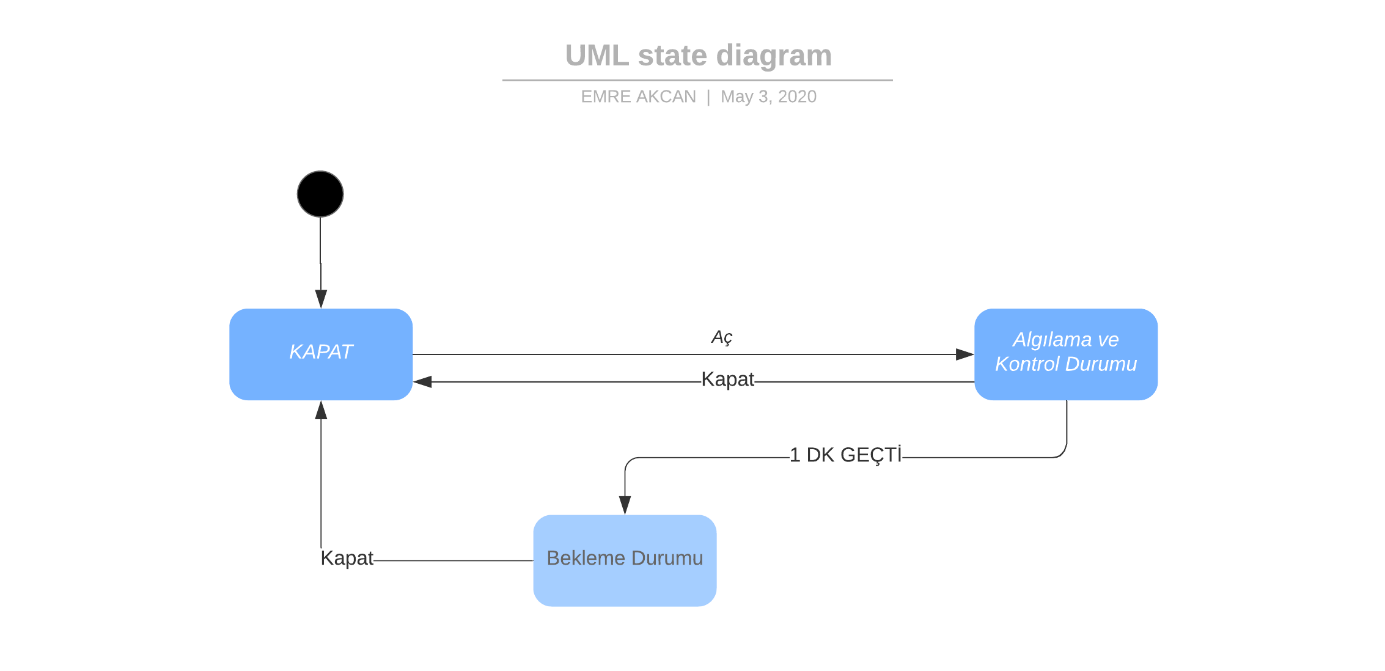
**SICAKLIK GÖRÜNTÜLEME**



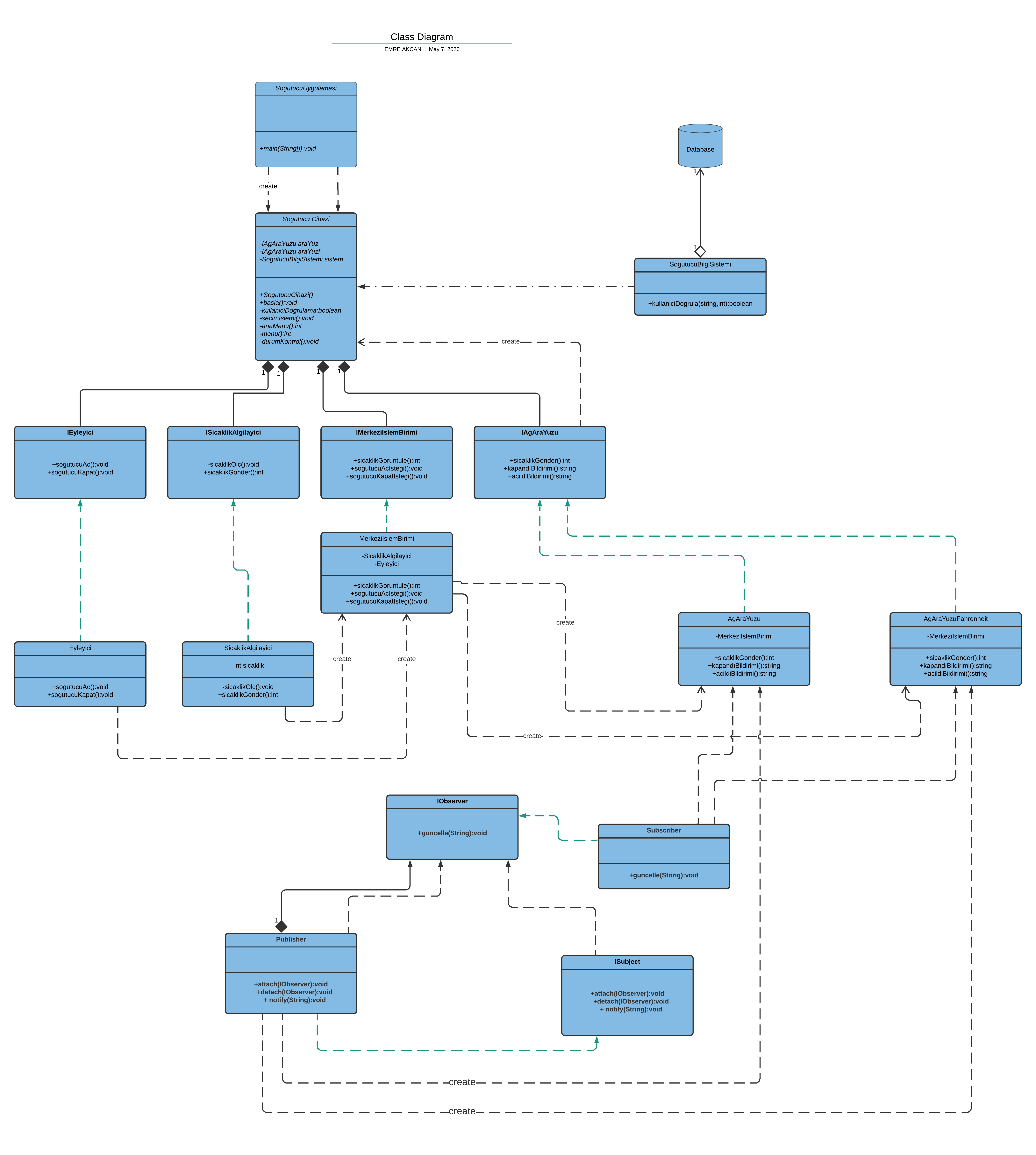
**SICAKLIK GÖRÜNTÜLEME AKTİVİTE DİYAGRAMI**



**DURUM DİYAGRAMI**



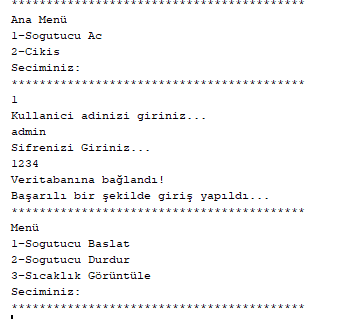
**SINIF DİYAGRAMI**

****

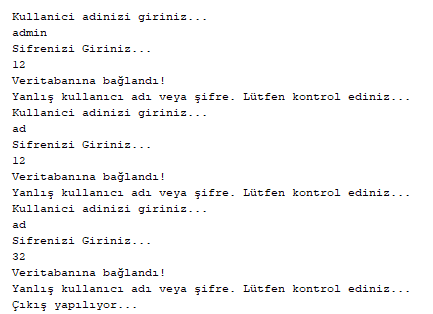
**KULLANICI DOĞRULAMA EKRANI**

Programımız çalıştığında, ilk olarak kullanıcı cihazı açması gerekiyor. Daha sonrasında kullanıcı adı ve şifresini girmesi gerekiyor. Girilen kullanıcı adı ve şifre değerleri veri tabanında kontrol edilip sisteme giriş yapılmasını sağlıyor. Eğer minimum 3 kere yanlış girilirse sistem kendi otomatikman kapanıyor.

*Başarılı Giriş:*



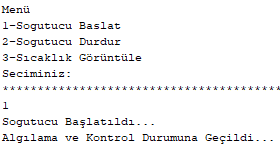
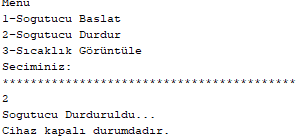
*Başarısız Giriş:*



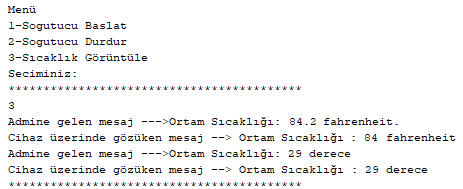
**SOĞUTUCUNUN BAŞLATILMASI VE SICAKLIĞIN GÖRÜNTÜLENMESİ**

Soğutucu cihaza erişim gerçekleştirdikten sonra kendimizi menüde buluyoruz. Menü ekranında kullanıcıyı soğutucu başlat, soğutucu durdur ve sıcaklık görüntüle seçenekleri karşılıyor. Soğutucu başlat seçeneği ortamdaki havanın soğutulmaya başlanmasını sağlıyor. Soğutucu durdur seçeneği soğutma işleminin durdurulmasını sağlıyor. Eğer kullanıcı sıcaklık görüntülemek isterse 3. seçenekten sıcaklık fahrenheit ve celcius cinsinden görüntüleniyor.

*Soğutucu Başlatılması: Soğutucu Durdurulması:*

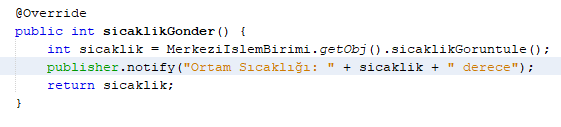
*Sıcaklığın Görüntülenmesi:*

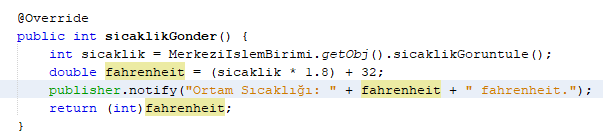
**

**OPEN/CLOSED İLKESİ**

Open/Closed ilkesinde bir sınıfın farklı davranışlara açık olması gerekir ve temel özelliklerinin sabit kalması zorunludur. Bu ilkede sınıfa yeni davranışların eklenebilmesi kolaylıkla sağlanır. Bu ilkeyi kodumdan örnek verecek olursam:

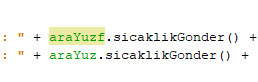
Aynı ara yüzden kalıtım alan farklı iki sınıfımız var. Biri sıcaklığın fahrenheit cinsinden gözükmesini diğeri ise celcius cinsinden gözükmesini sağlayan metodlara sahip. Fakat bu metodların ikisi de aynı metod sadece işlevleri farklı.





Ara yüzden nesneler tanımlayıp bu nesneler farklı sınıflardan oluşturuyorum.

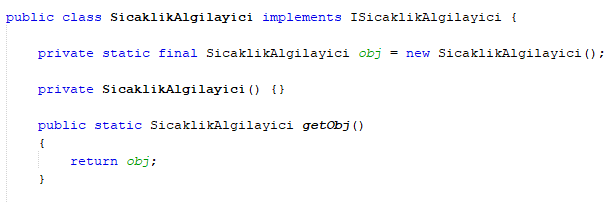




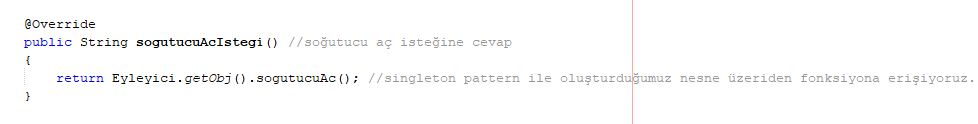
Burada nesnelerden aynı metodları çağırdığımda biri fahrenheit cinsinde değer döndürürken diğeri ise celcius cinsinde değer döndürüyor. Böylelikle open/closed ilkesini kodda uygulamış oluyoruz.

**SINGLETON ve OBSERVER DESENLERİ**

**Singleton** deseni tanımı şudur ki; yalnızca bir örneği olan ve buna global erişim noktası sağlayan bir sınıf tanımlamasıdır. Başka bir deyişle, bir sınıf yalnızca tek bir örneğin oluşturulmasını ve tek bir nesnenin diğer tüm sınıflar tarafından kullanılabilmesini sağlamalıdır.



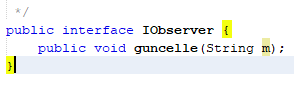
Sıcaklık algılayıcı sınıfı içerisinde ona ait bir örnek oluşturup, merkezi işlem birimi sınıfı içerisinde ondan nesne üretmeden içerisindeki metoda ulaşıyoruz.



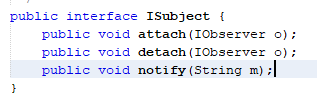
**Observer** deseninden, bire bir bağımlılığı durumunda bir nesnenin durumu değiştiğinde buna bağlı olan nesnelere bu değişim bildirilir ve nesneler güncellenir.

***Observer*** *deseninin koddaki uygulanması:*

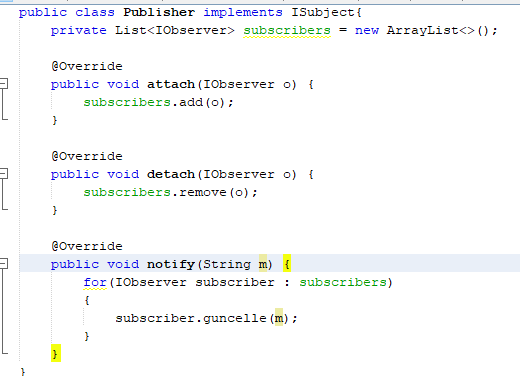
Observer, ara yüz sınıfı bir değişiklik olduğunda nesneyi uyaracaktır.



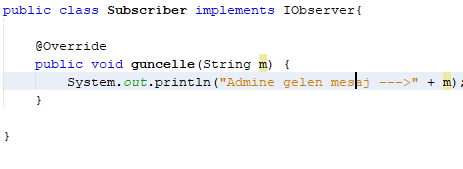
Subject ara yüz sınıfı ise durumu değişecek nesneyi temsil eder.



Publisher(haber verici) sınıfına subject ara yüzünü implemente ederek metod gövderlini tanımladık.



Subscriber(abone) sınıfına observer ara yüzünü implemente ederek metod gövderlini tanımladık.



SıcaklıkAlgılayıcı sınıfı içerisinde sicaklikGonder() fonksiyonundaki sıcaklık değişimini buna bağlı olan nesnelere duyuruyoruz.

