****

**T.C.**

**SAKARYA ÜNİVERSİTESİ**

**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

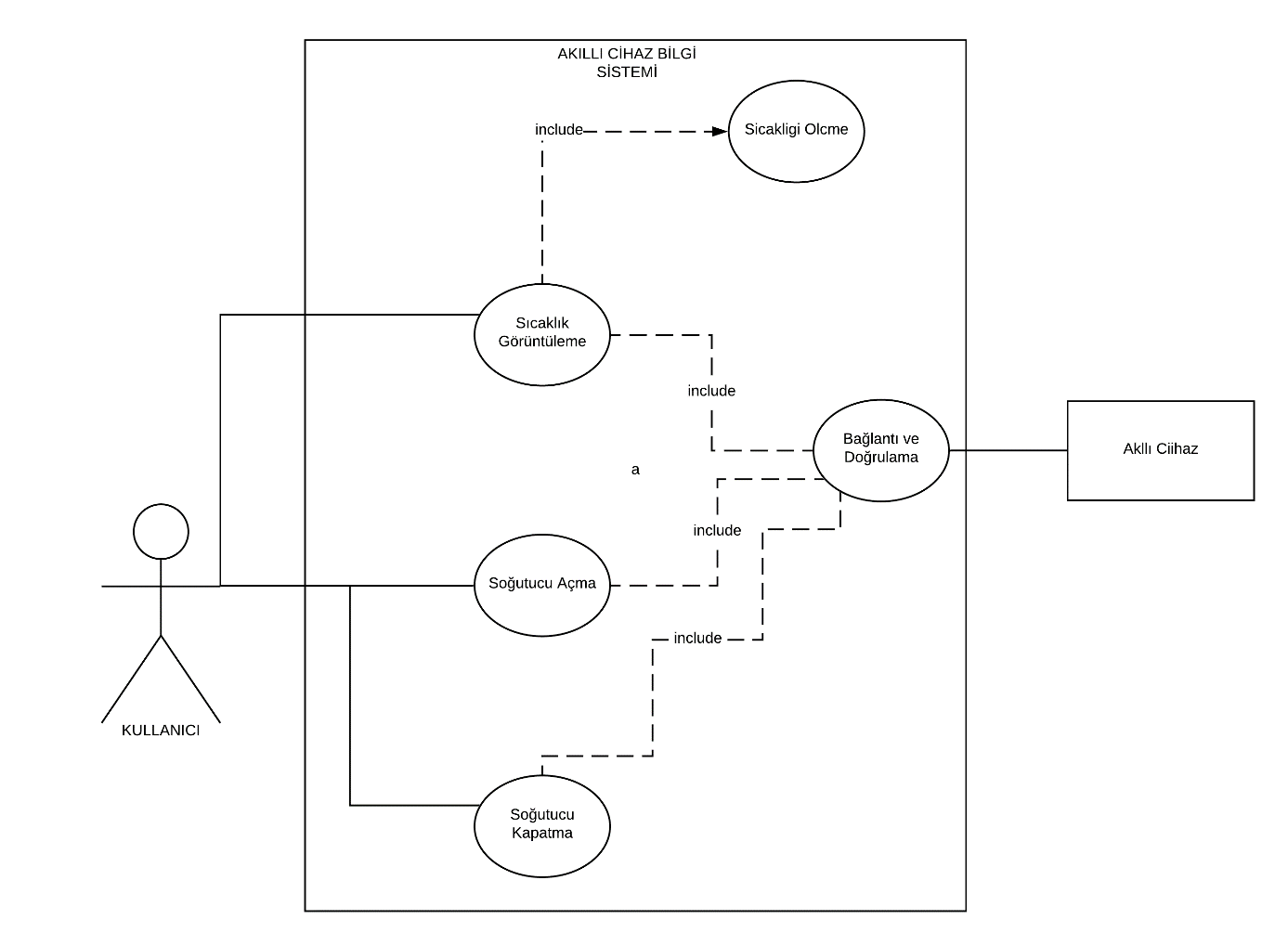
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Nesne Yönelimli Analiz ve Tasarım**

**PROJE/TASARIM**

**Hazırlayan:**

**B171210084 – Serhat Erenel 2. Sınıf 1. Öğretim A Grubu**



**SICAKLIĞIN GÖRÜNTÜLENMESİ KULLANIM DURUMU**

* Eşsiz bir ad: Sıcaklığın Görüntülenmesi,
* Akıllı cihazın sıcaklığının görüntülenmesi sürecini tanımlar.
* 8.05.2020 v1.1 Serhat Erenel.

**İlgili Aktörler:** İnternet Kullanıcıları

**Giriş Koşulu:** Sıcaklığın görüntülenmesi isteği.

**Çıkış Koşulu:** Sıcaklığın görüntülenmesi.

**Ana Olay Akışı;**

**1.**Kullanıcı ağ arayüzü giriş ekranına gelir.

**2.**Ekrana kullanıcı adı ve şifre erkanı gelir.

**3.**Kullanıcı şifreyi ve kullanıcı adını girer.

**4.**Ağ arayüzü erişimi onaylar ve seçim menüsünü gösterir.

**5.**Kullanıcı sıcaklığın görüntülenmesi işlemini seçer .

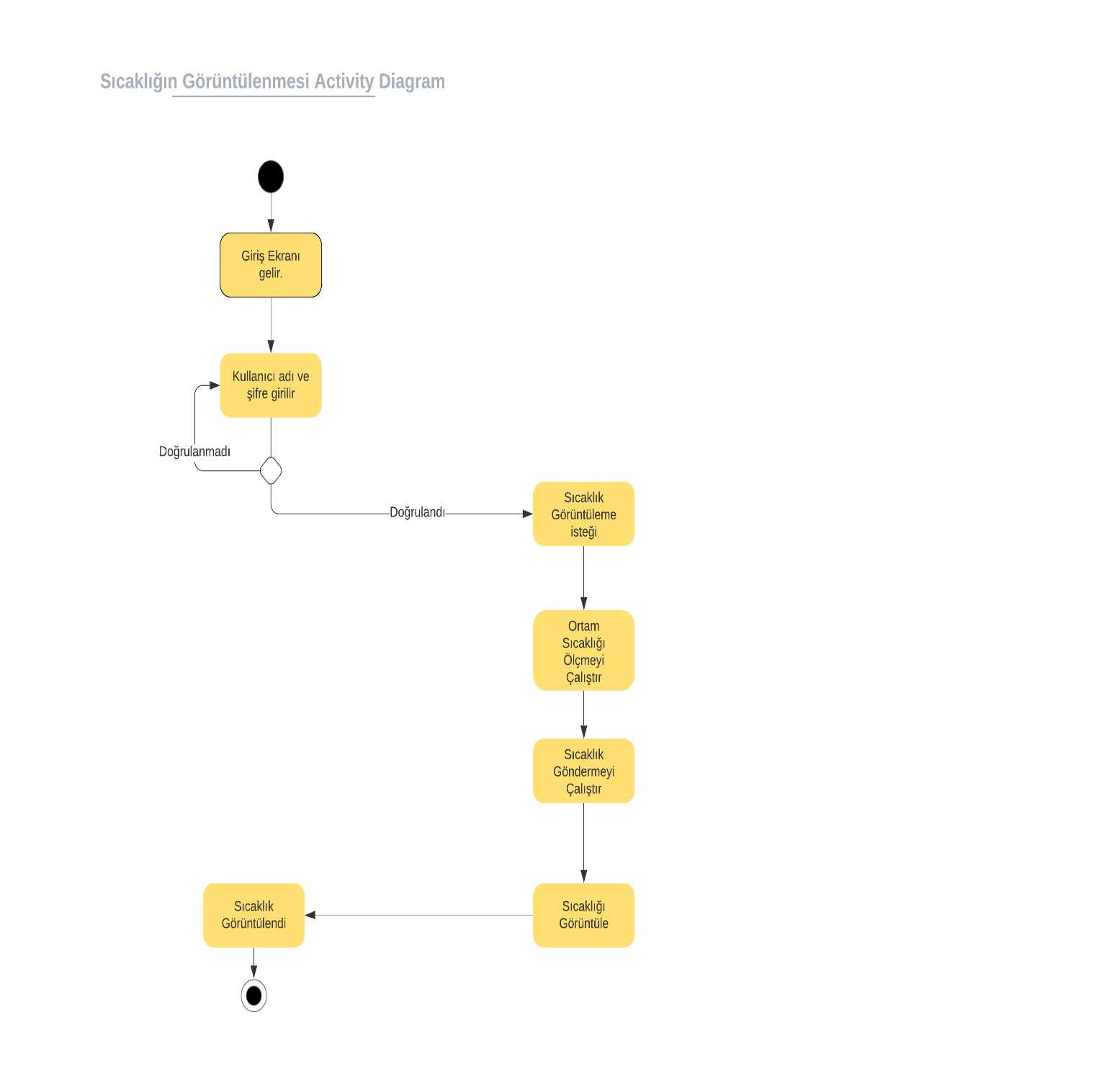
**6.**Sistem seçimi algılar ve seçimi çalıştırır.

**Alternatif Olay Akışı;**

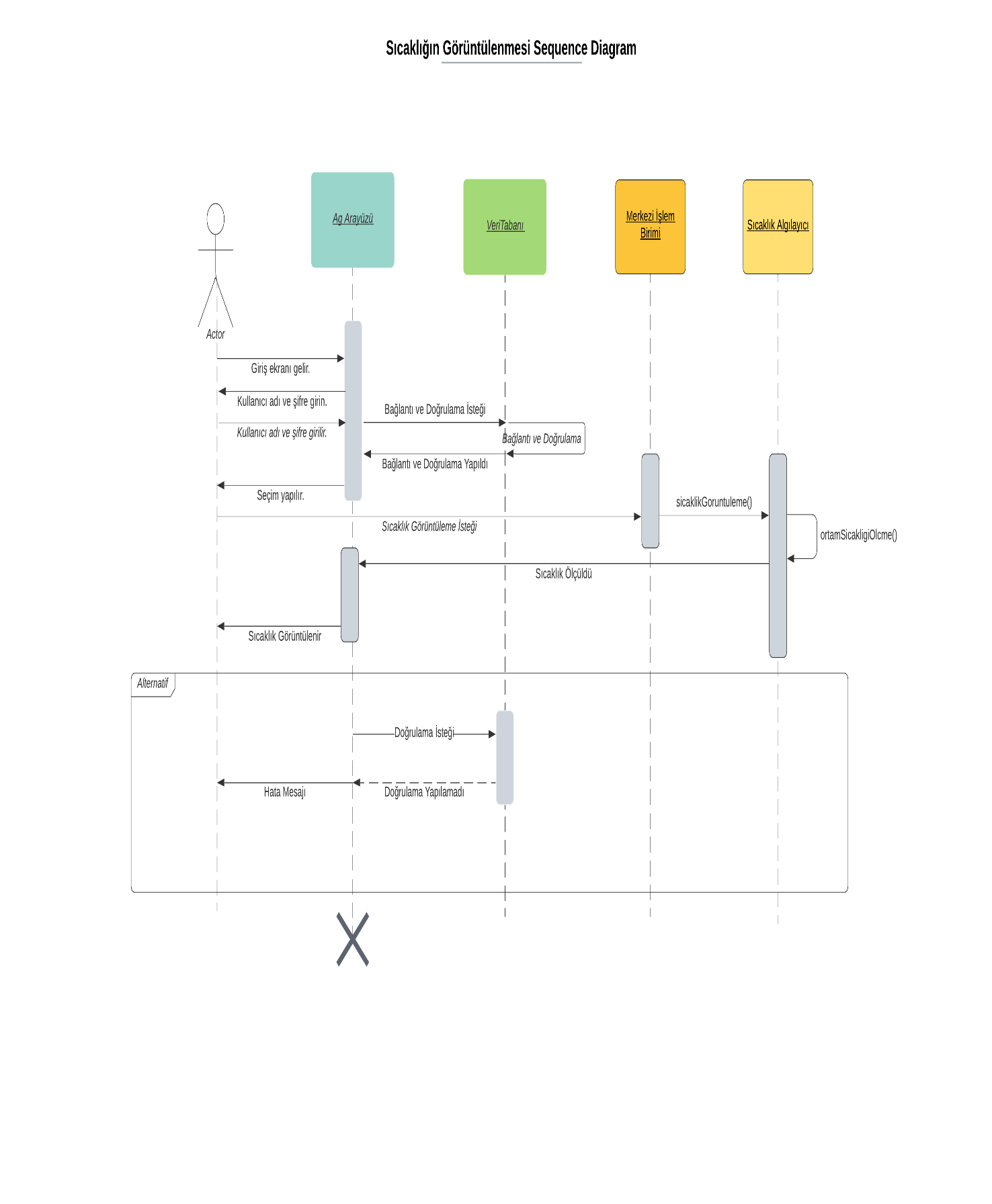
**A1.** Kullanıcı adı ve şifre yanlış.(3)

**4.** Giriş ekranına dönülür.

**SICAKLIĞIN GÖRÜNTÜLENMESİ ACTIVITY DIAGRAM**



**SICAKLIĞIN GÖRÜNTÜLENMESİ SEQUENCE DIAGRAM**



**SOĞUTUCUNUN ÇALIŞTIRILMASI KULLANIM DURUMU**

* Eşsiz bir ad: Soğutucunun Çalıştırılması
* Akıllı cihazın soğutucusunun başlatılması sürecini tanımlar.
* 8.05.2020 v1.1 Serhat Erenel.

**İlgili Aktörler:** İnternet Kullanıcıları

**Giriş Koşulu:** Soğutcunun çalıştırılması isteği.

**Çıkış Koşulu:** Soğutucunun çalıştırılması.

**Ana Olay Akışı;**

**1.**Kullanıcı ağ arayüzü giriş ekranına gelir.

**2.**Ekrana kullanıcı adı ve şifre erkanı gelir.

**3.**Kullanıcı şifreyi ve kullanıcı adını girer.

**4.**Ağ arayüzü erişimi onaylar ve seçim menüsünü gösterir.

**5.**Kullanıcı soğutucu açma işlemini seçer .

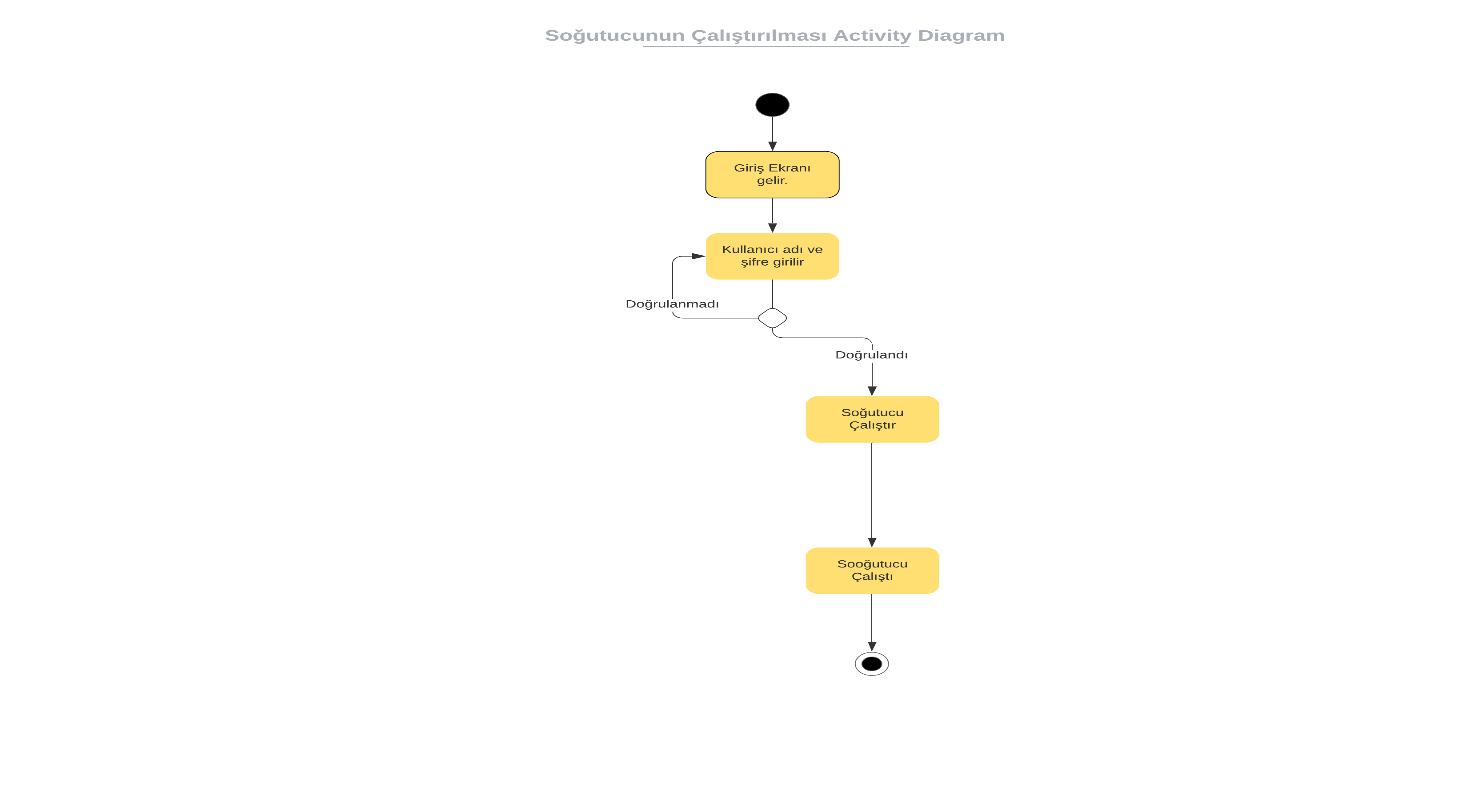
**6.**Sistem seçimi algılar ve seçimi çalıştırır.

**Alternatif Olay Akışı;**

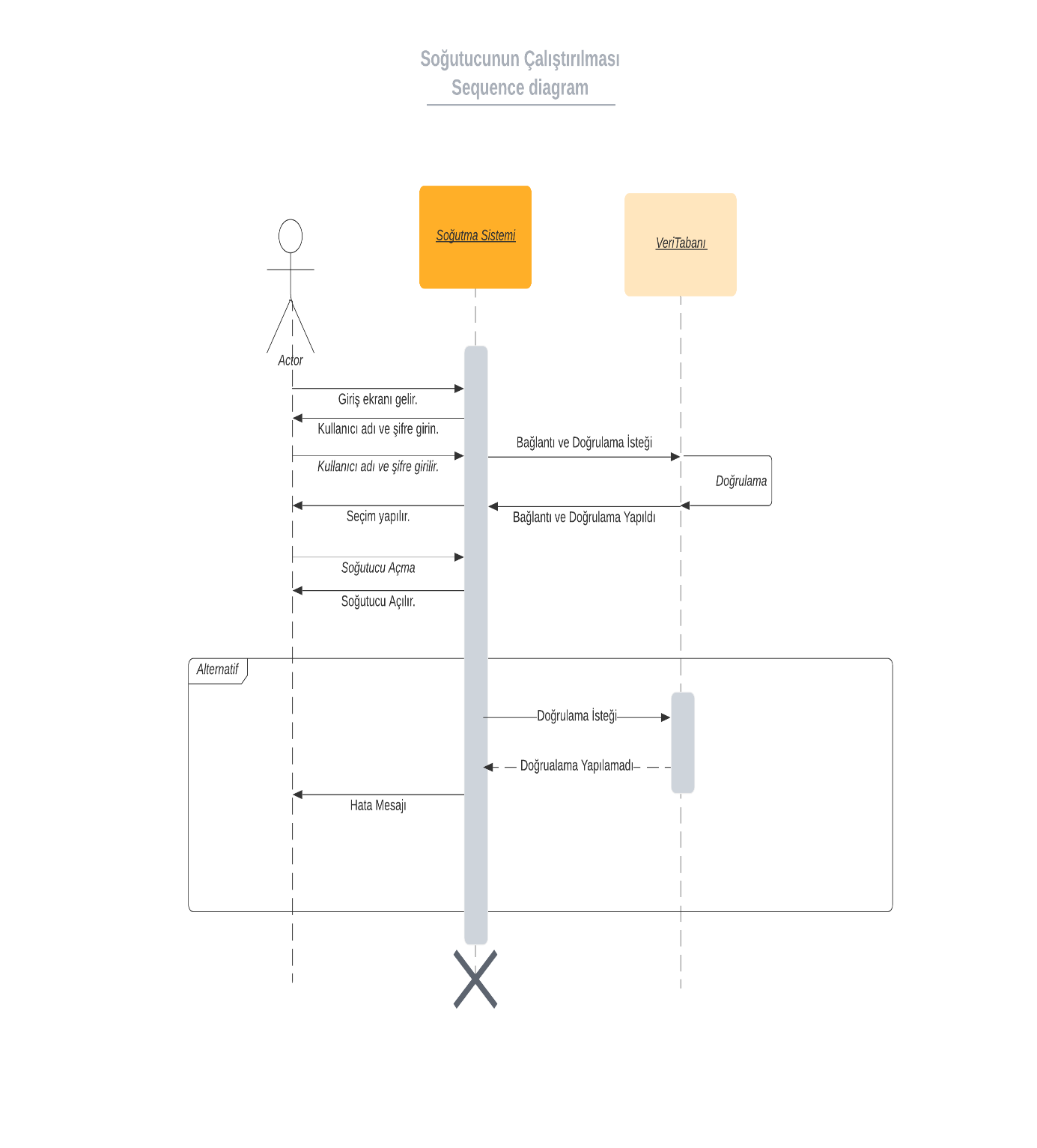
**A1.** Kullanıcı adı ve şifre yanlış.(3)

**4.** Giriş ekranına dönülür.

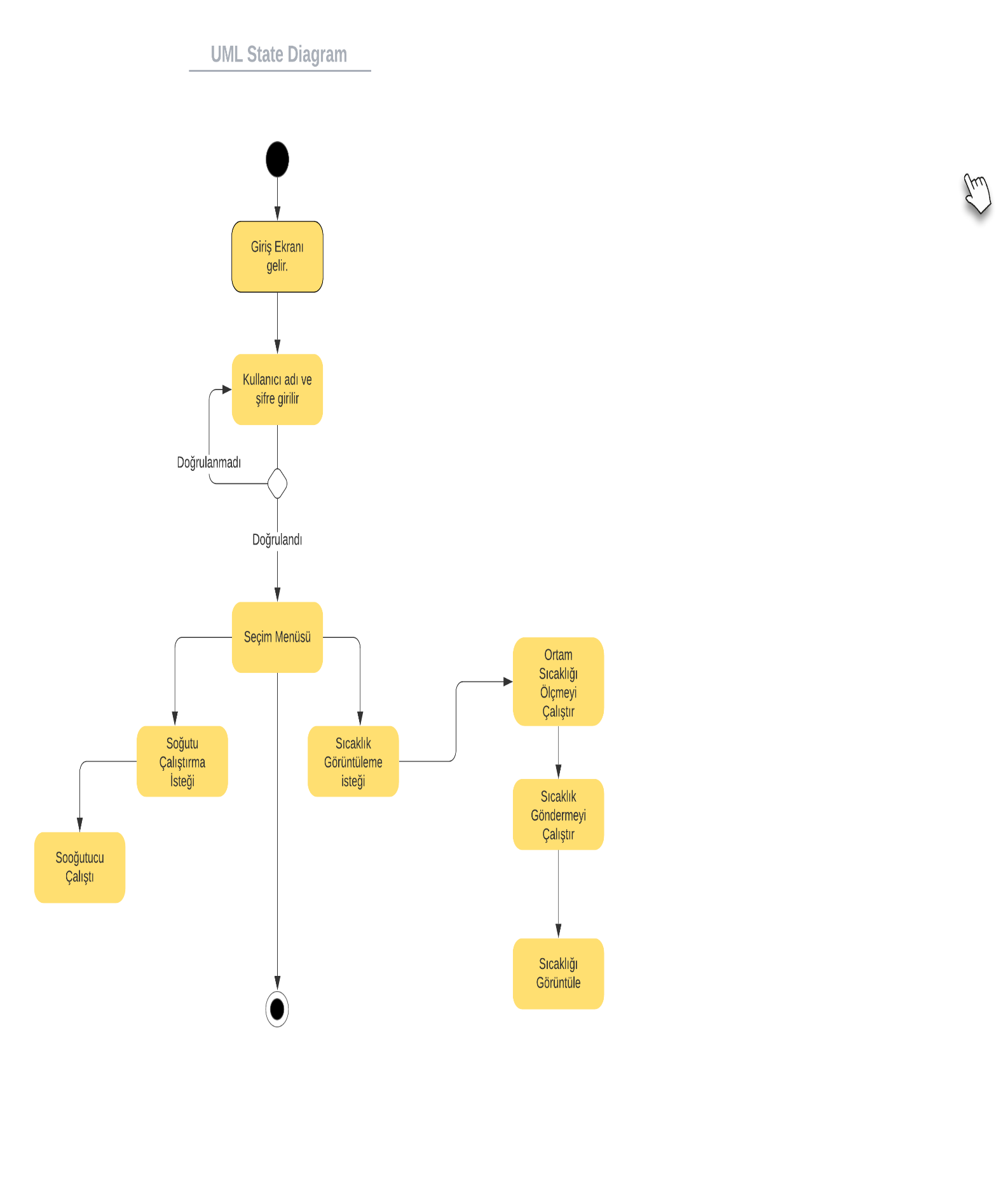
**SOĞUTUCUNUN ÇALIŞTIRILMASI ACTIVITY DIAGRAM**



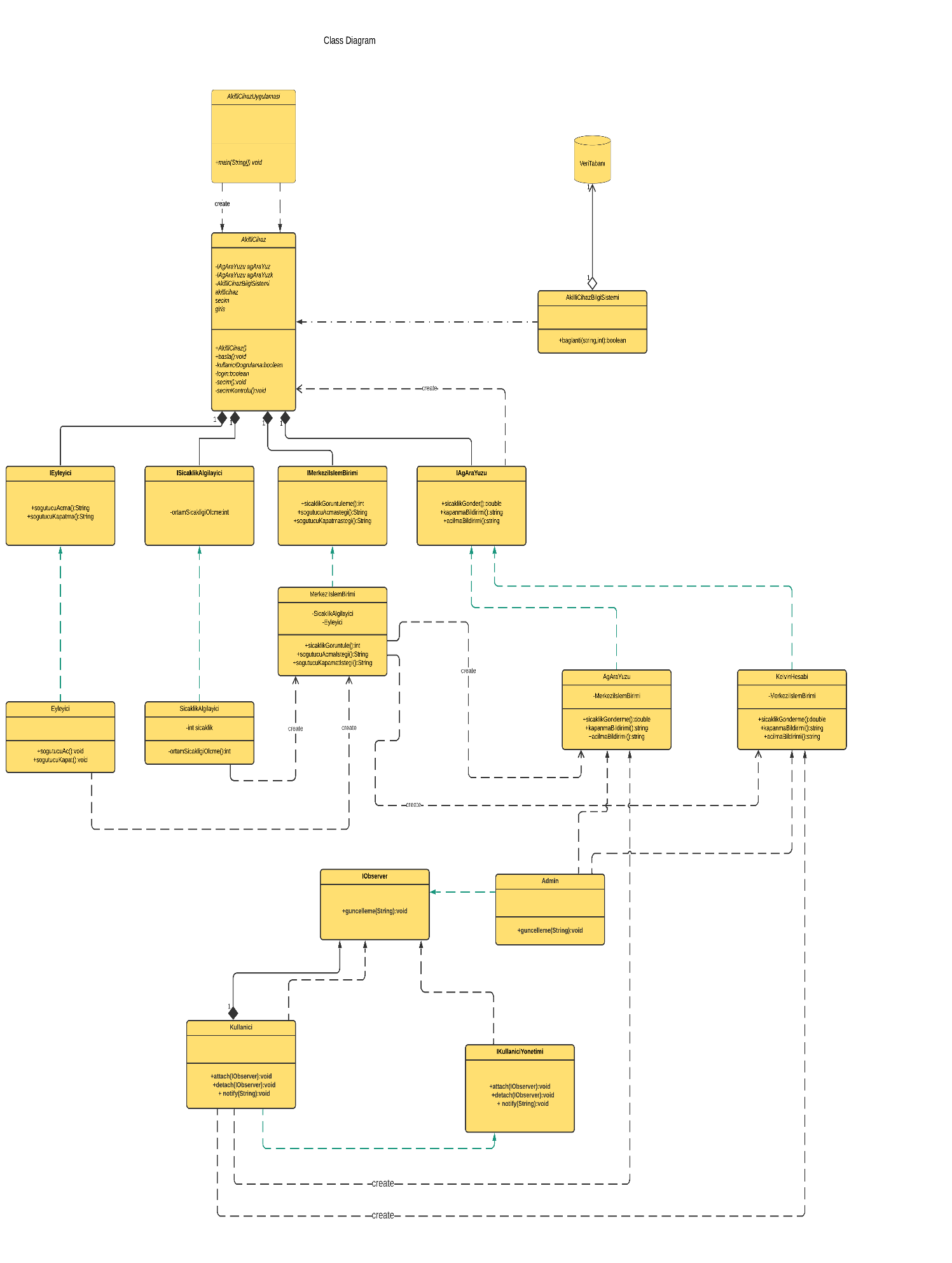
**SOĞUTUCUNUN ÇALIŞTIRILMASI SEQUENCE DIAGRAM**



**DURUM STATE MACHINE DIAGRAM**



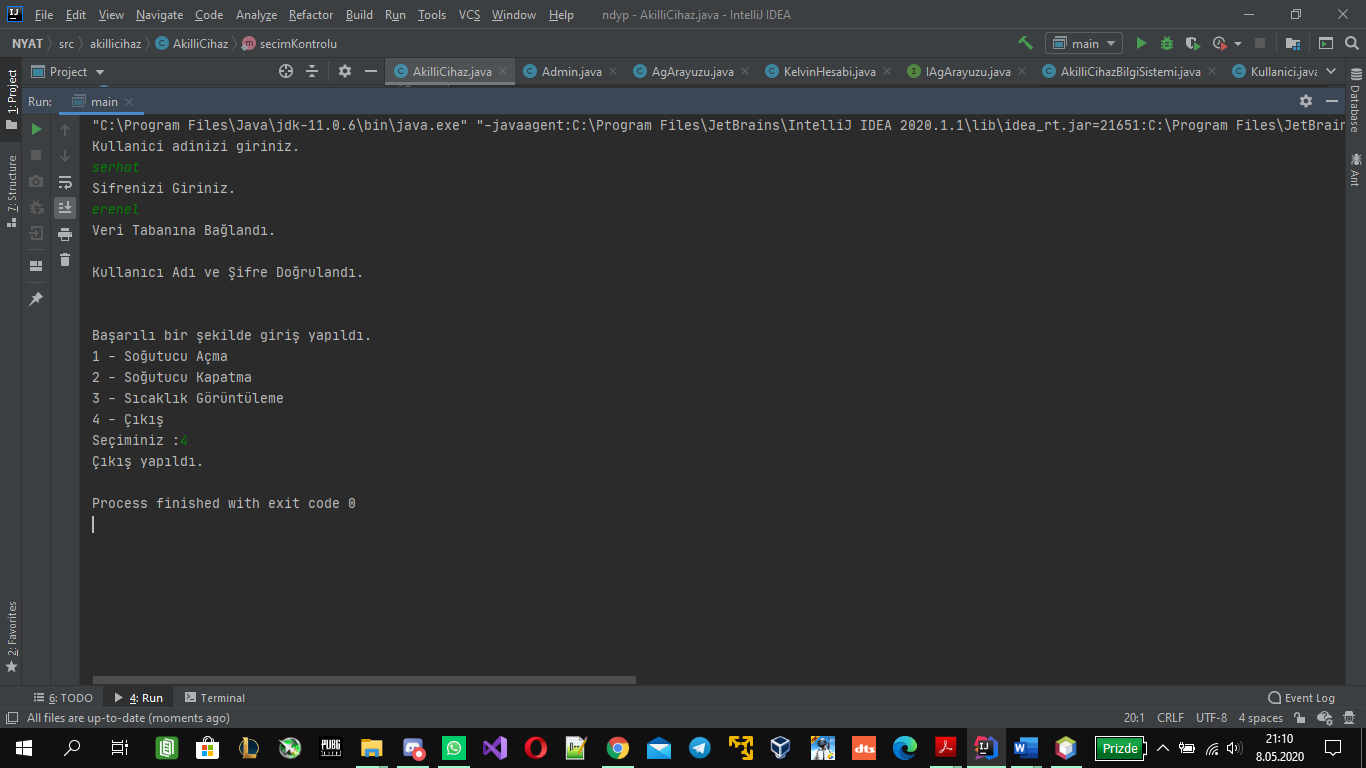
**CLAS DIAGRAM**

****

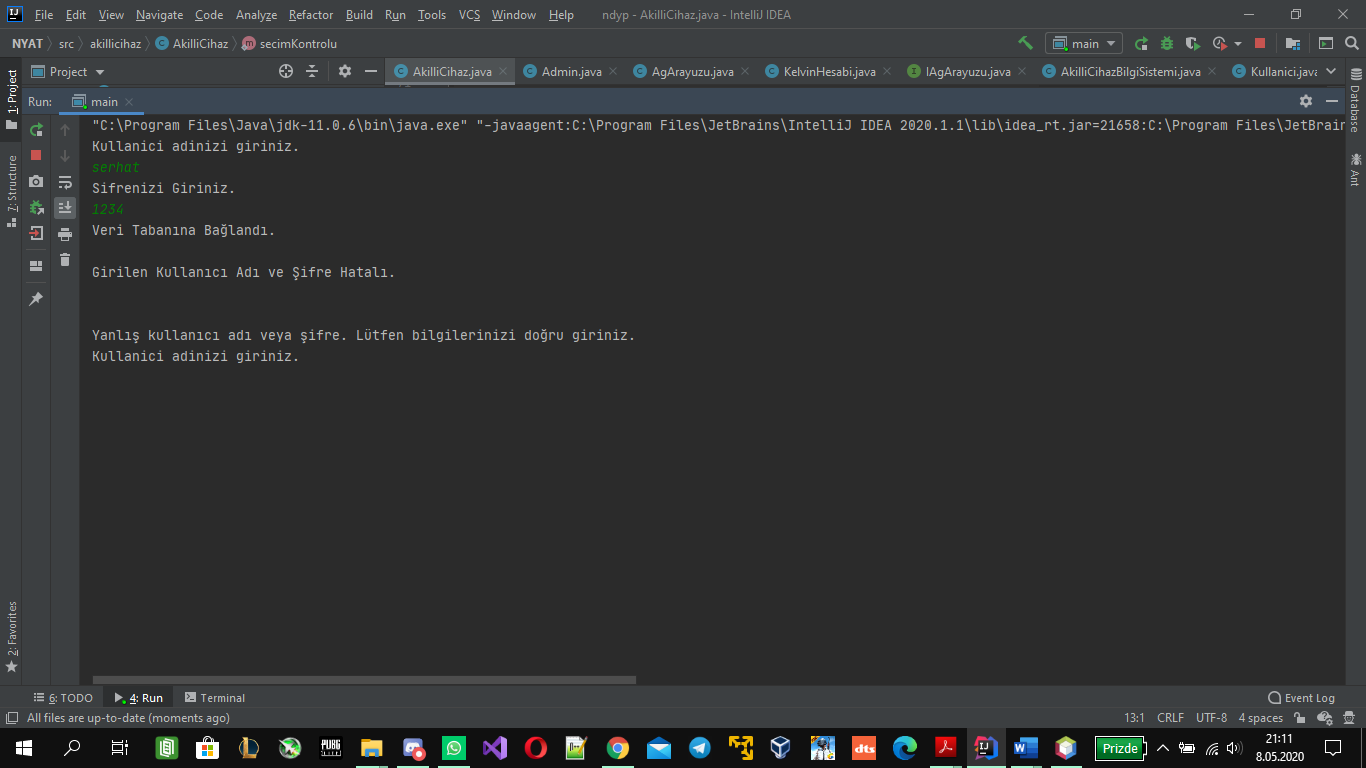
**KULLANICI DOĞRULAMA EKRANI**

Kullanıcının sisteme giriş yapabilmesi için ilk önce kullanıcı adı ve şifre ekranından veritabanında bulunan bilgileri doğru bir şekilde girerek giriş yapması gerekiyor. Aksi takdirde giriş yapamaz ve o ekranda kalır. Giriş yaptıktan çıkmak isterse çıkış seçeneği ile çıkış yapabilir.

**BAŞARILI GİRİŞ**

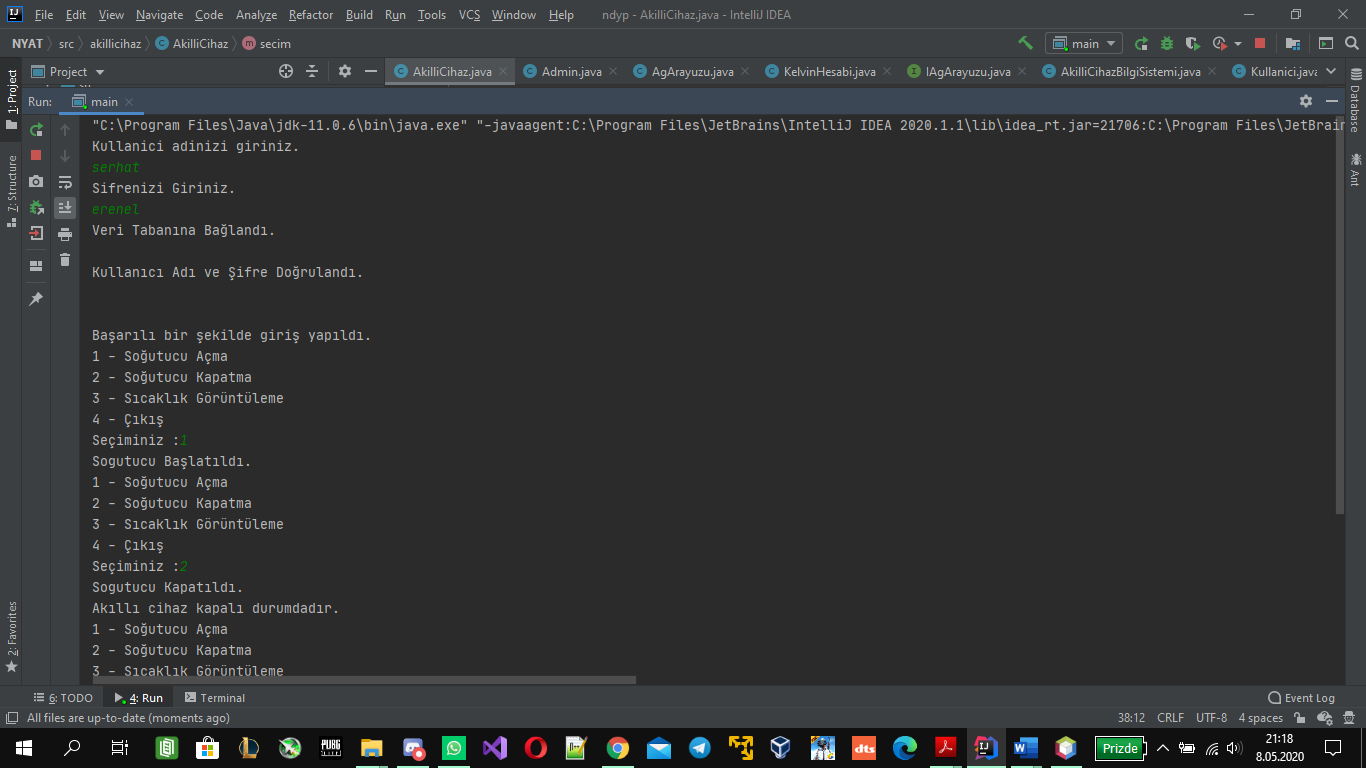


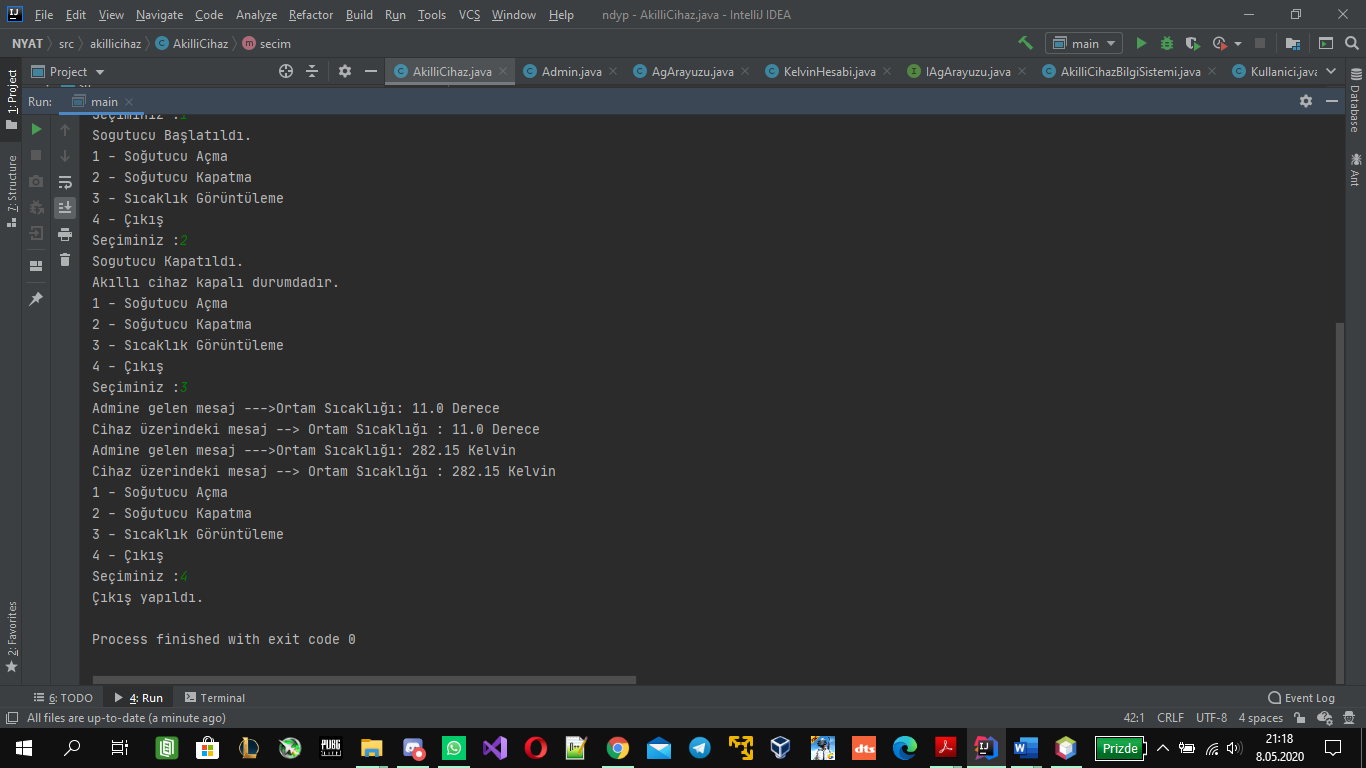
**BAŞARISIZ GİRİŞ**



**SICAKLIĞIN GÖRÜNTÜLENMESİ VE SOĞUTUCUNUN AÇILIP KAPANMASI**

**Sıcaklığın görüntülenmesi ve soğutucunun açılması işlemleri için ilk önce sisteme başarılı bir şekilde giriş yapıyoruz. Daha sonra giriş yaptıktan sonra çıkan menüde gerekli numaralara basarak yapmak istediğimiz işlemi seçiyoruz. 1 numarali işlemi seçersek soğutucu açılıyor. 2 numaralı işlemi seçersek soğutucu kapanıyor. 3 numaralı işlemi seçersek sıcaklık görüntüleniyor derece ve kelvin cinsinden. 4 numaralı işlemi seçersek sistemden çıkış yapıyoruz.**

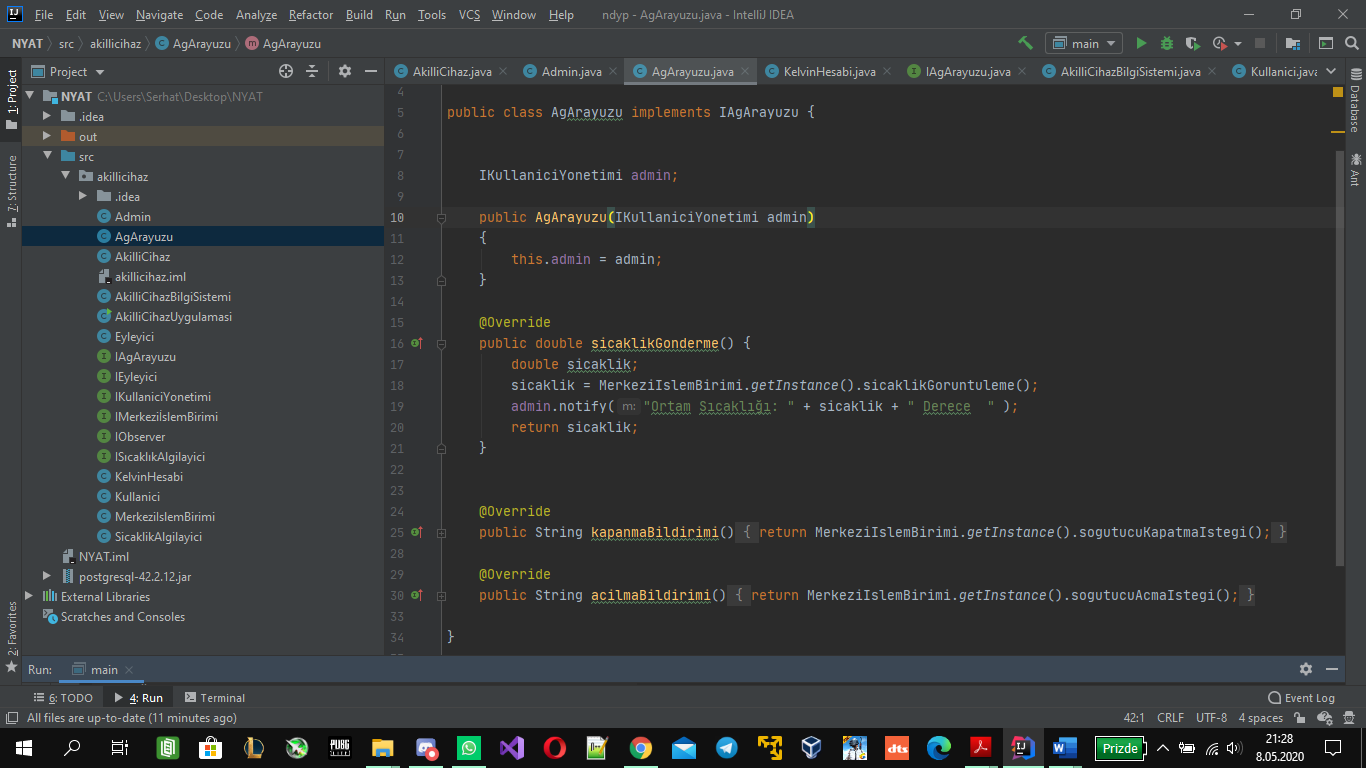


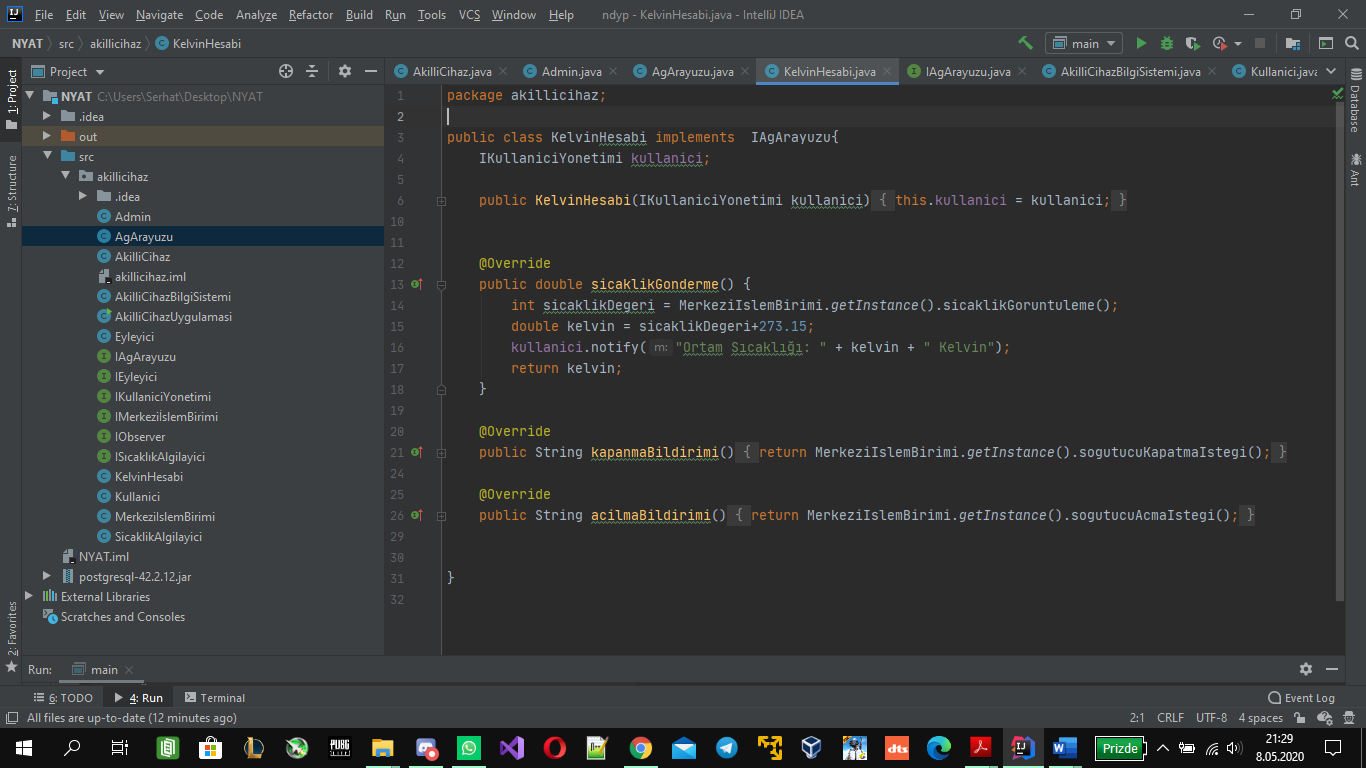


**OPEN/CLOSED İLKESİ**

Müşteriden gelen isteklerin veya sistemimizde değişiklik yapmak istediğimizde, sistemimizin rahatlıkla geliştirilebiliyor olması ve başka geliştirmelere de uzun kodlar yazmadan imkanı sağlanması adına, uygulamamızı Open/Closed prensibine göre geliştirmek, bizler için hayat kurtaran bir özelliktir.

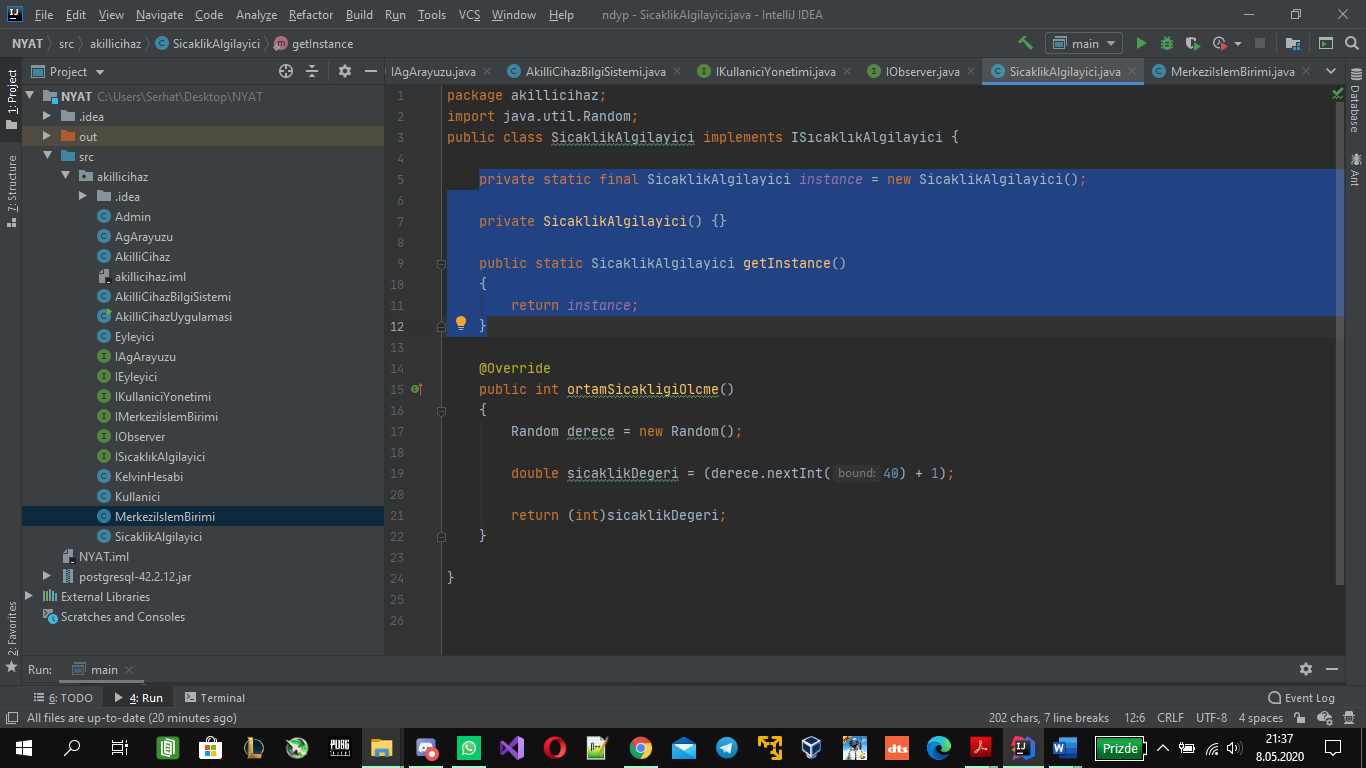
Bende programımda hem sınıflar için arayüz yapıp hem de aynı arayuzden kalıtım alan iki sınıf oluşturdum. Bu sınıflar AgArayuzu ve KelvinHesabi sınıflarıdır. Kalıtım aldıkları arayuz ise IAgArayuzu interface i dir. İki sınıfta sıcaklık görüntülenmesi işlemlerini yapan methodlara sahip bu methodlar aynı ama işlevleri farklı.

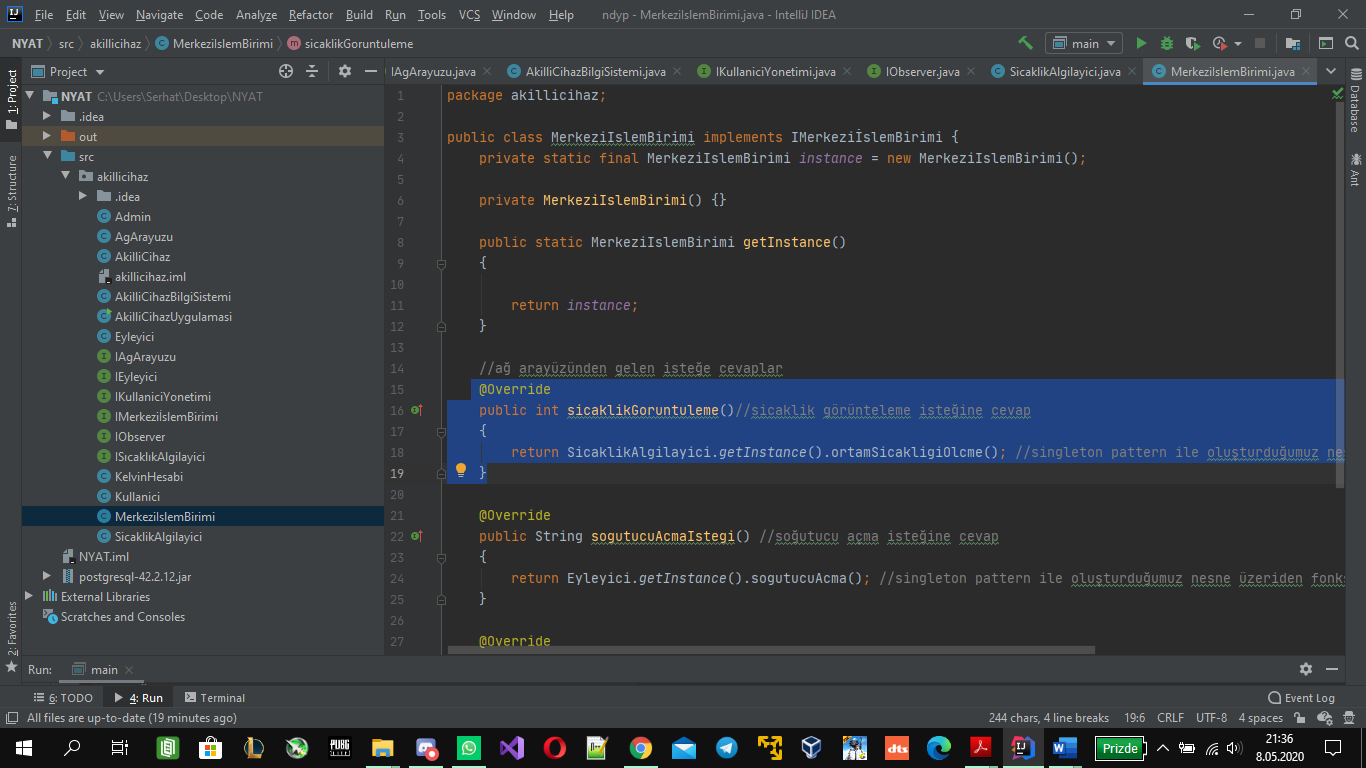




**SINGLETON VE OBSERVER DESENLERİ**

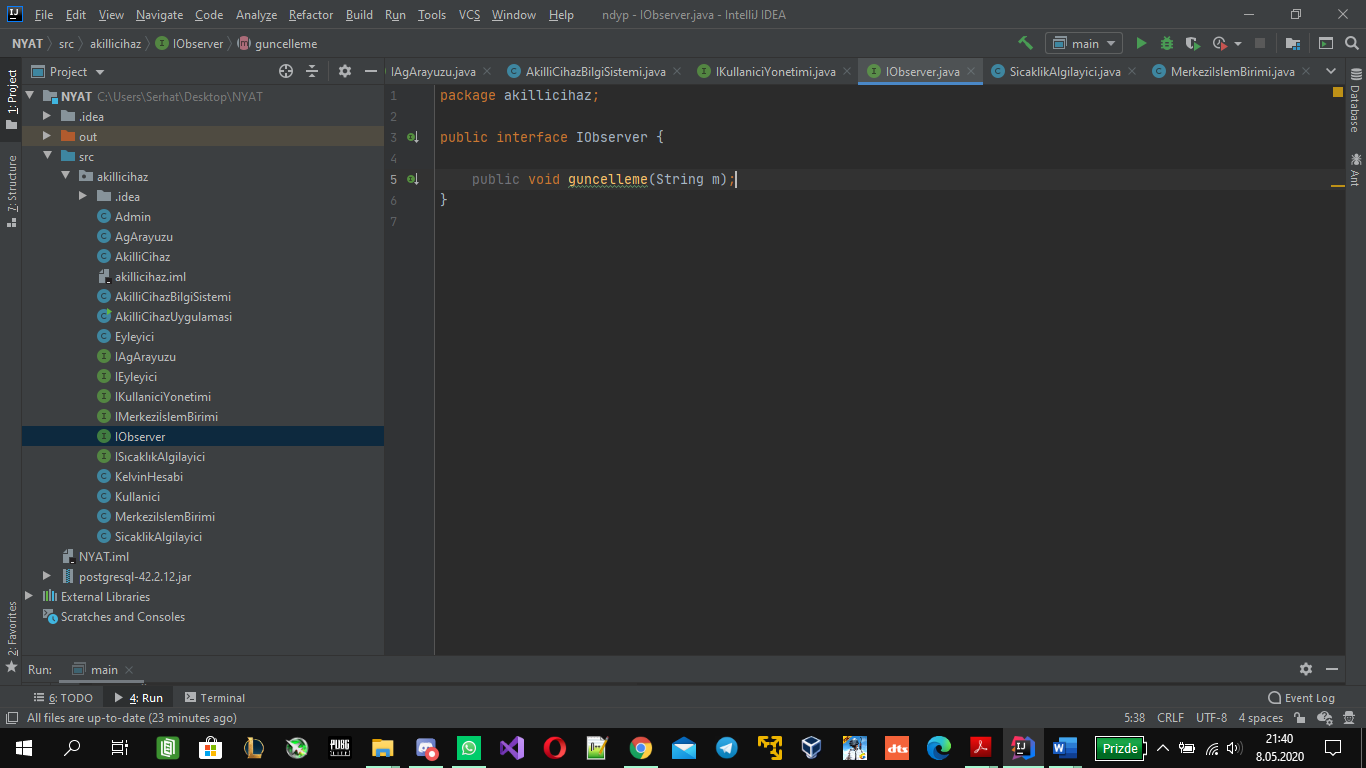
**Singleton** deseni, hazırlayacağımız sınıftan sadece bir örneğinin oluşturulmasını sağlar. Bu sayede nesnenin kopyalanmasını ya da yeni bir tane oluşturmasını engeller ve nesneye ihtiyaç duyulduğunda o nesnenin daha önceden oluşturulan örneğini çağırır.

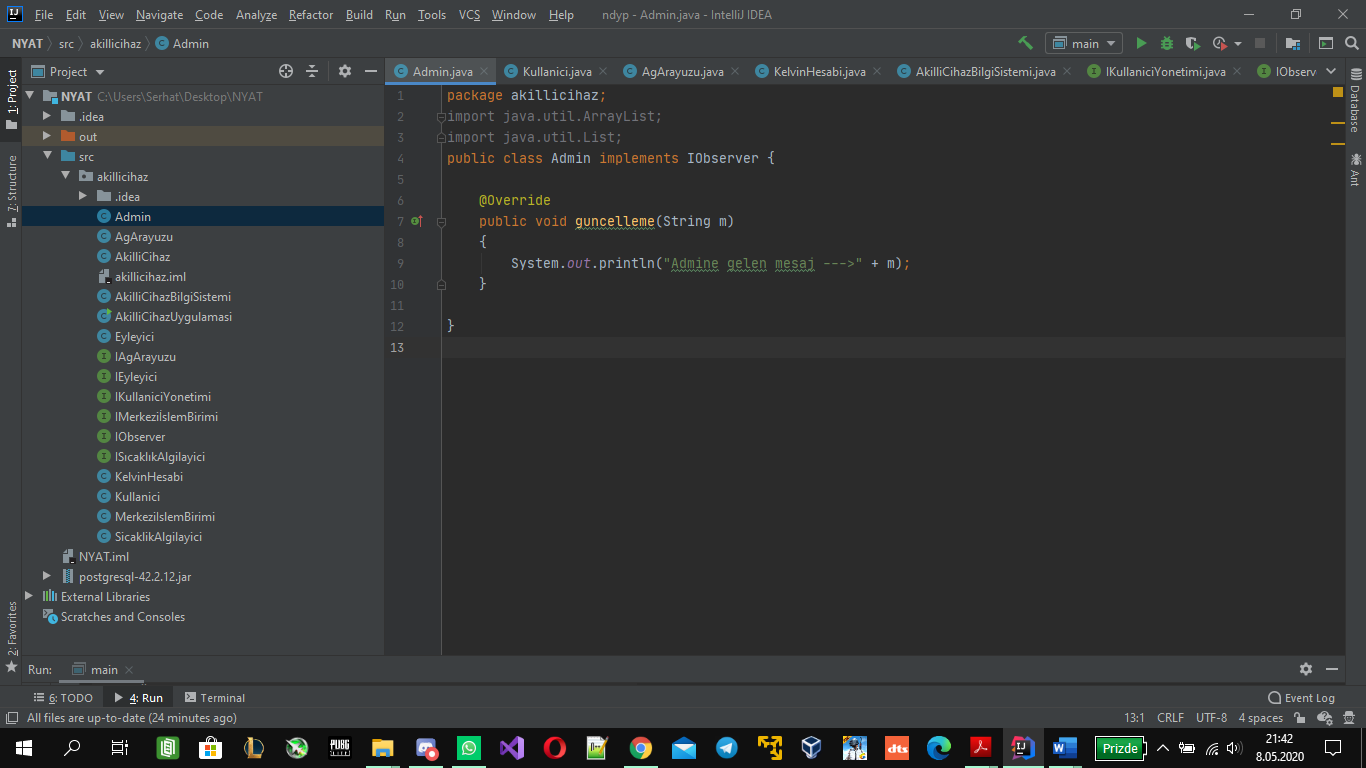




Yukardaki fotoğraflarda mavili olan kısımlar bu desenin kullanımını gösteriyor. İlk resimde nesnemizi üretiyoruz ikinci resim yani alttaki resimde nesnemizi başka sınıfta kullanıyoruz.

**Observer** deseni, behavioral tasarım desenlerinden biridir. Nesneler arasında one-to-many ilişki sağlar. Bir nesne durumunu değiştirdiğinde, ona bağlı diğer tüm nesneler uyarılır ve otomatik olarak güncellenir.

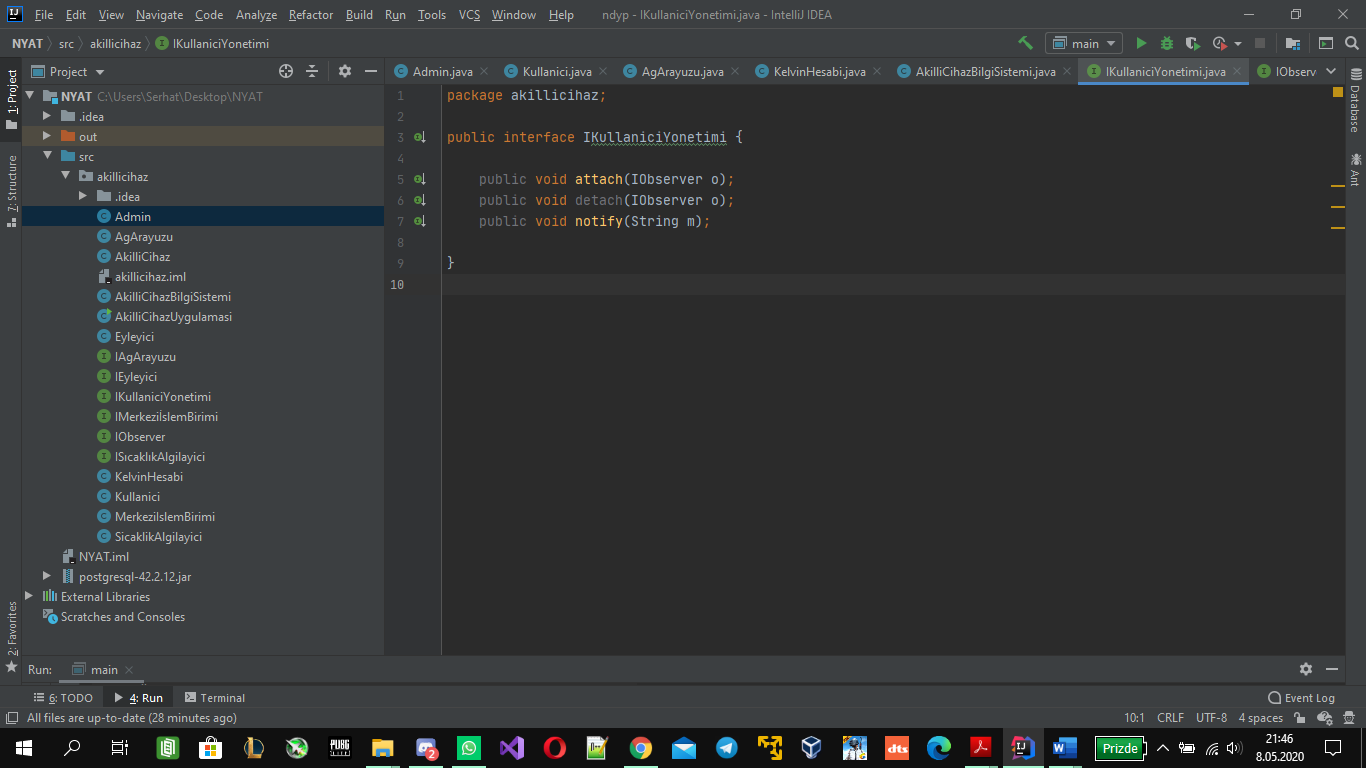


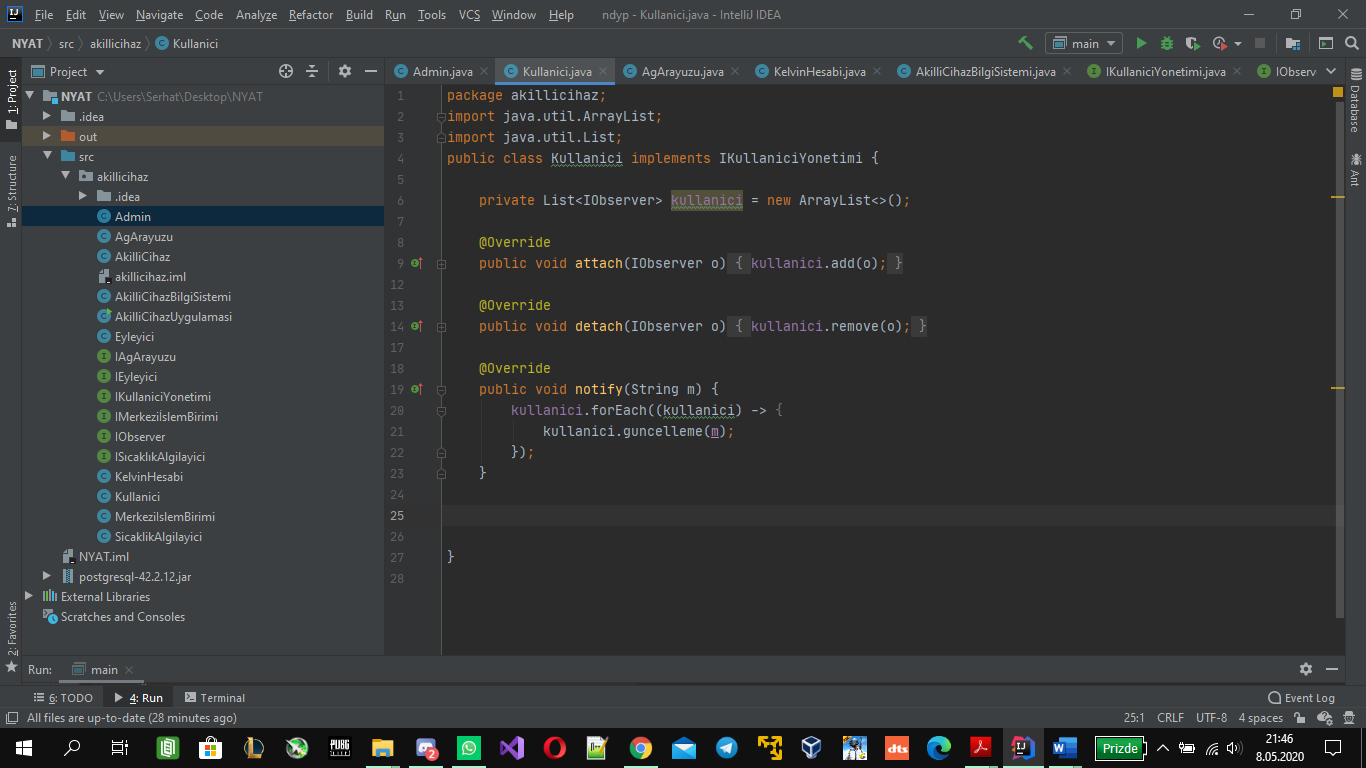


İlk fotoğrafta IObserver interface i değişiklik olduğunda nesneyi uyarakcatır.

İkinci fotoğrafta ise Admin sınıfı IObserver interface inden kalıtım alarak değişimi yazdırıyor.

Değişim işlemleri ise aşağıdaki fotoğrafladaki sınıflarda gerçekleşiyor.





**VİDEO LİNKİ:** **https://drive.google.com/file/d/1KADPaRRU0ywgEfij29QEXJKNd\_eVEjK9/view?usp=sharing**