



T.C.

SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

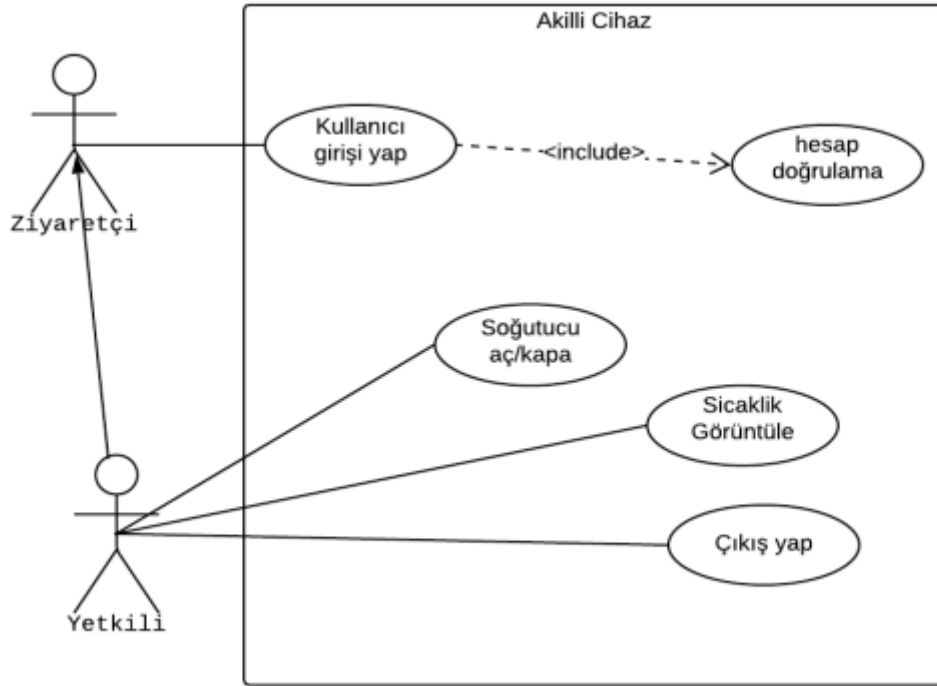
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Nesne Yönelimli Analiz Ve Tasarım Proje Ödevi

MAYIS 2020

a.

Use-Case “Akıllı Cihaz Kontrol”



b.

Akıllı cihaz kontrol Kullanım durumu

- Akıllı cihazın uzaktan kontrol edilmesi durumunu tanımlar
- 06.05.2020 v1 Omar Garibov

İlgili Aktörler: İnternet Kullanıcısı

Giriş koşulu: Kullanıcı şifre ve adını girer.

Çıkış koşulu: Kullanıcı doğrulaması yanlışa ve ya menüden “kapat” seçeneği seçilirse.

Olay Akışı:

1. Kullanıcıdan cihaz adını ve şifresini girmesi istenir.
2. Kullanıcı cihaz adını ve şifresini girer.
3. Sistem kullanıcıyı doğrular.
4. Kullanıcıya sıcaklık durumu gösterilir.
5. Kullanıcı Soğutucu durumunu seçer.
6. Bekleme durumunda olan soğutucu kullanıcı seçimine göre güncellenir
7. Sistem kullanıcıya işlem seçimlerini tekrar sunar
8. Sistem seçilen seçimi yerine getirir ve kullanıcıya bildirir

Alternatif Akışlar:

- A1

3. Kullanıcı doğrulanmaz

4. Kullanıcıya doğrulamanın yapılamadığı ekrana yazılan mesajla bildirilir ve programı tekrar çalıştırması istenir

5. program sonlanır

- A2

5. Kullanıcı Sıcaklık ölçme işlemini seçer

6. Akıllı cihaz ölçülen sıcaklık değerini ekranda gösterir.

7. Kullanıcıya aynı seçenekleri tekrardan sunar.

- A3

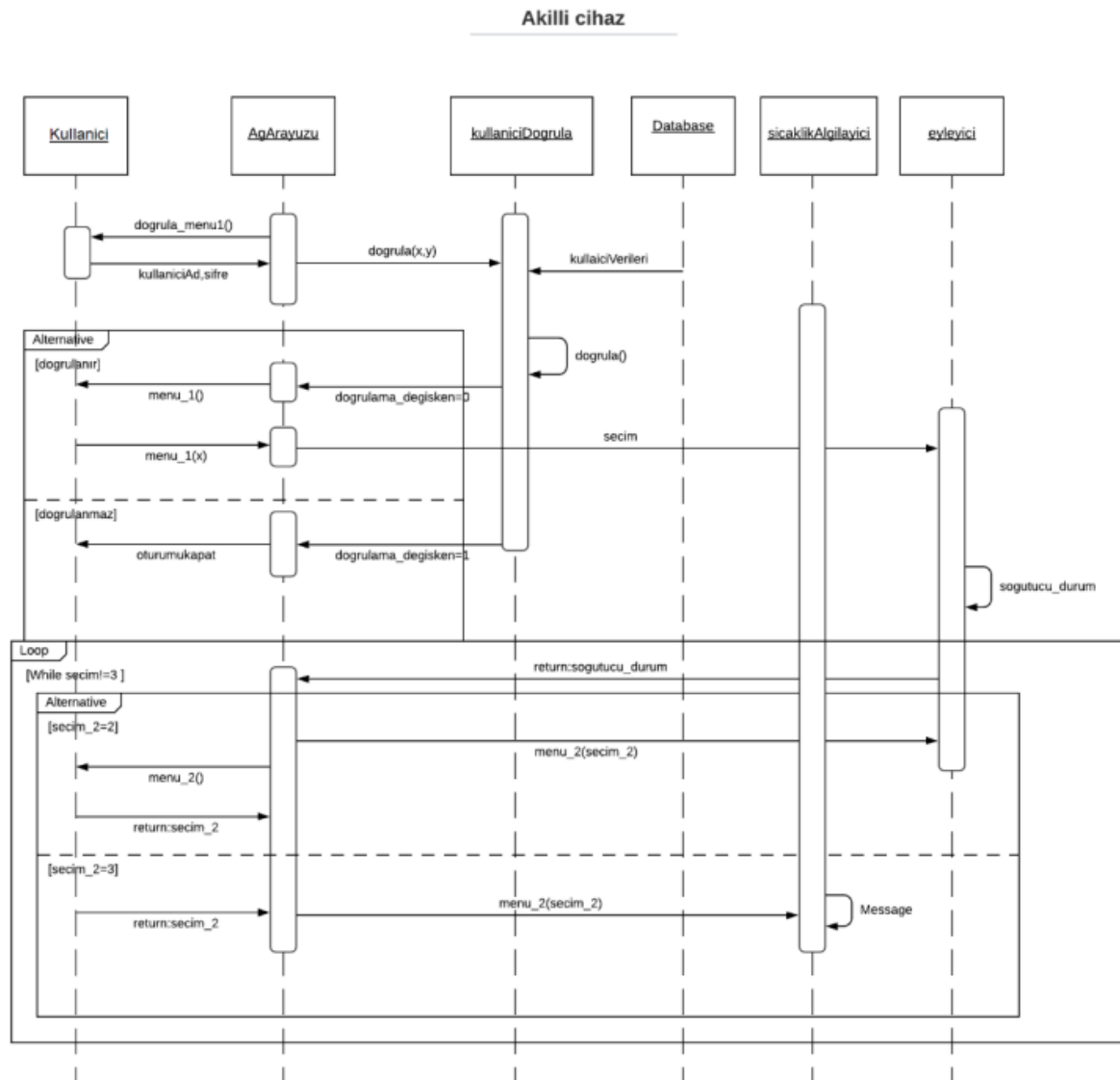
5. Kullanıcı Oturumu kapat seçeneğini seçer

6. Kullanıcının oturumu kapatılır.

Özel Gereksinimler: Kullanıcının geçersiz seçim numarası girmemesi, 24 saat çalışması

c.

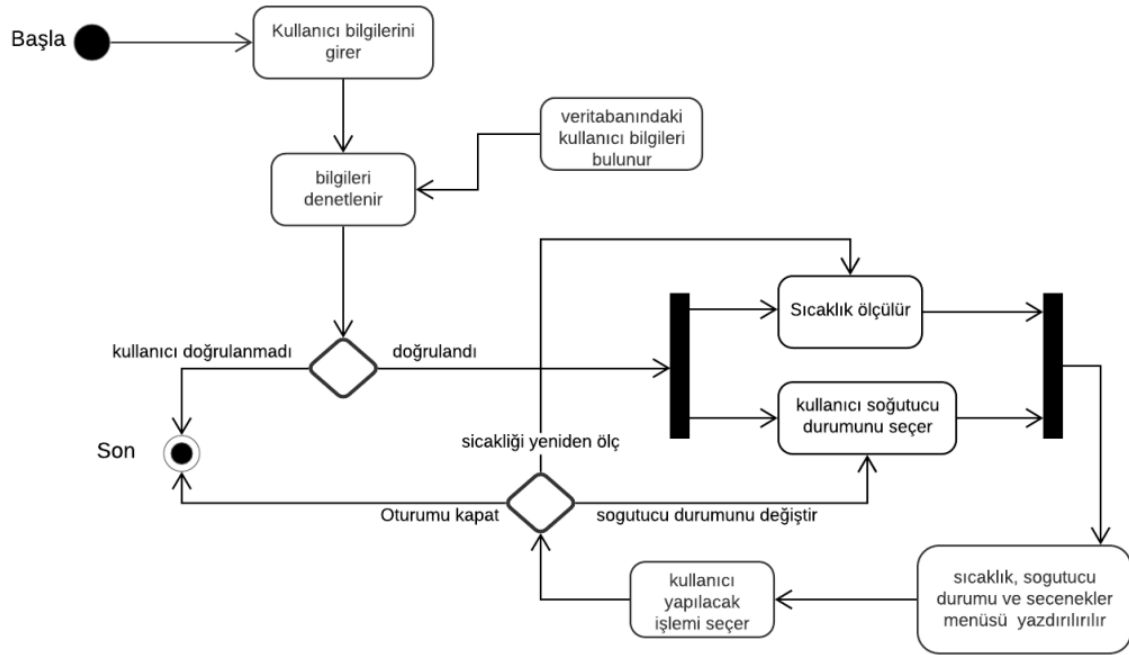
Sequence diagram



d.

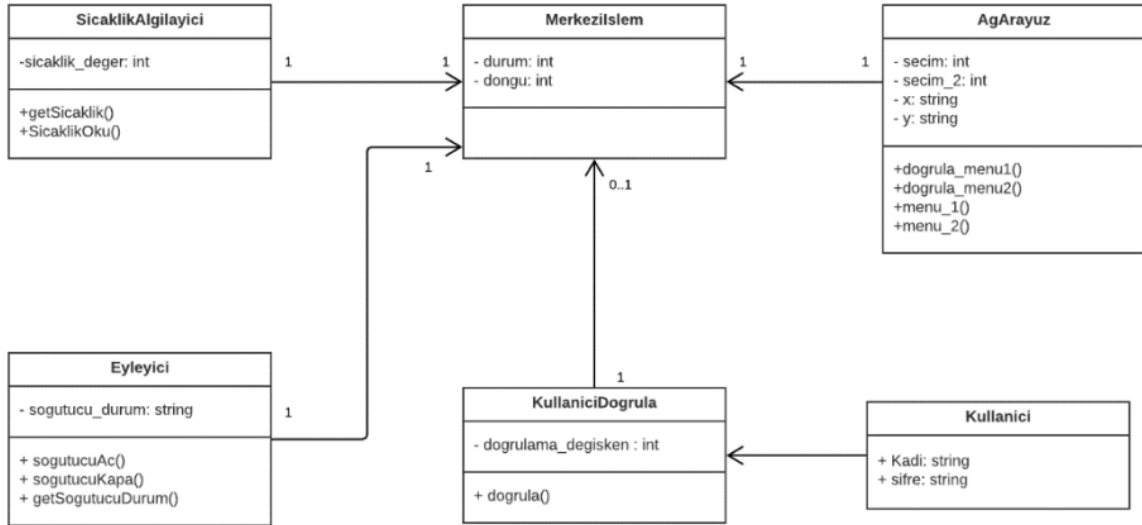
Activity diagram

Activity Diagram



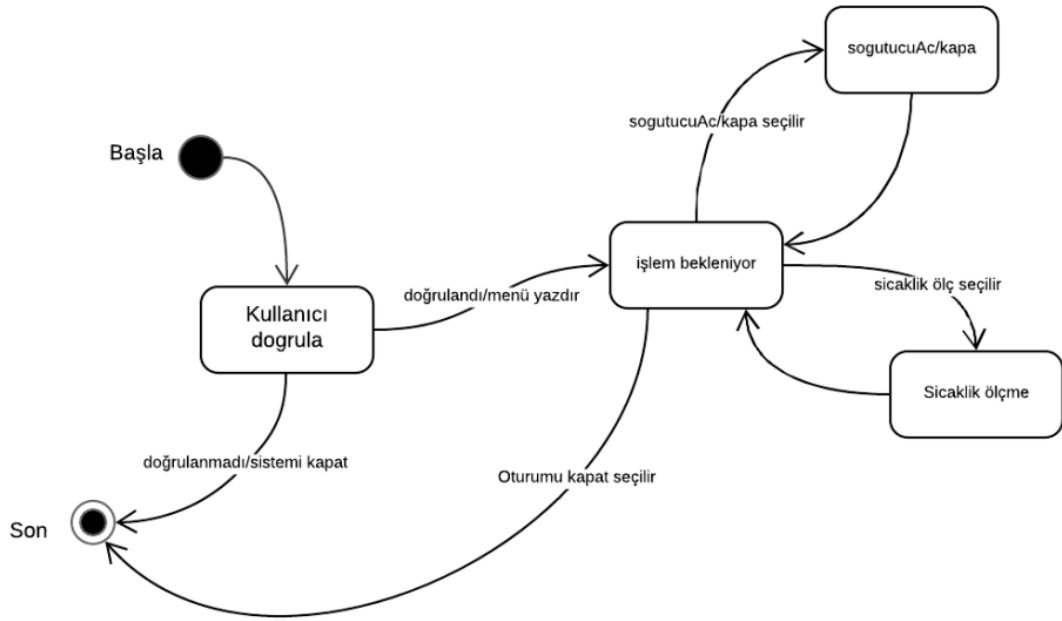
e.

Sınıf Şeması



f.

Sistemin durum diyagramı



g.

Kullanıcı Doğrulama İşlemi

```
*****
Kullanıcı Adını girin.....:cihaz4
Kullanıcı Şifresini girin.....:4444

*****

*****
Veritabanına bağlandı!
*****

*****
Kullanıcı doğrulama başarılı
*****
```

Program çalıştırıldığında ilk olarak AgArayuzu sınıfında tanımlanmış dogrula_Menu() fonksiyonu ile kullanıcıdan Adını ve kullanıcı şifresini girmesi istenir. Kullanıcının girdiği veriler AgArayuzu sınıfında tutulur. Daha sonra bu bilgiler MerkeziIslem sınıfında çağırılıp dogrula() fonksiyonu ile KullanıcıDogrula sınıfına gönderilir. KullanıcıDogrula sınıfında veri tabanına bağlanılır. Bağlanma gerçekleştiğinde fotoğraftaki mesaj yazdırılır, devamında ise veri tabanındaki kullanıcı bilgileri while döngüsü yardımıyla tek tek karşılaştırılır. Girilen

```
-----!!!!-----!!!!-----
Hatalı giriş lütfen programı tekrar baslatın
-----!!!!-----!!!!-----
```

Kullanıcı adı ve şifre kayıtlı ise “Kullanıcı Doğrulama Başarılı” yazdırılır ve dogrulama_degisken=1 yapılır. dogrulama_degisken MerkeziIslem sınıfında denetlenir. Sıfıra eşit ise hatalı giriş yapılmış demektir ve ekran çıktısı bu şekilde olur ve program sonlandırılır:

h. Sıcaklığın görüntülenmesi ve soğutucunun açılıp kapatılması

```
Sicaklik Degeri.....: 1
Sogutucu Durumu.....: Beklemede

[1] Sogutucuyu Ac
[2] Sogutucuyu Kapat
Sogutucu Durumunu secin.(1/2)
*****
```

Kullanıcı doğrulaması yapıldıktan sonra SicaklikAlgilyici sınıfındaki sicaklikOku() metoduyla rastgele 0-50 arası üretilen sicaklik değeri AgArayuzu sınıfındaki menu_1() fonksiyonu ile yazdırılır ve kullanıcıdan bekleme durumundaki soğutucunun ne yapacağını girmesi istenir (dokümanda soğutucunun başlangıçtaki durumu belirtilmediği için kullanıcı tarafından seçilmesine karar verdim). Kullanıcı seçimine göre Eyleyici sınıfındaki sogutucuAc() ve ya sogutucuKapa() fonksiyonu çalışır. Bu fonksiyonlar Eyleyici classındaki String durumuna “açık” ya da “kapalı” değerini atar. Değer atandıktan sonra Merkezilslem sınıfında AgArayuzundeki menu_2() fonksiyonu çağrılır. Menü_2 fonksiyonu sıcaklık değeri ve soğutucu durumunu tekrardan yazdırır. Son durum(kullanıcının seçimine bağlı olarak değişebilir) bu şekilde olur:

```
*****

Sicaklik Degeri.....: 1
Sogutucu Durumu.....: KAPALI

[1]Oturumu Kapat
[2]Tekrar Ölç
[3]Sogutucu Durumu Degistir
Yapilacak islemi secin.(1/2/3)
*****
```

Tekrar ölç seçildiğinde sicaklik tekrar hesaplanıp yazılır:

```
*****

Sicaklik Degeri.....: 13
Sogutucu Durumu.....: KAPALI

[1]Oturumu Kapat
[2]Tekrar Ölç
[3]Sogutucu Durumu Degistir
Yapilacak islemi secin.(1/2/3)
*****
```

Sogutucu durumu değiştir seçildiğinde soğutucu_durum değişkeninin son değeri kontrol edilir ve değerin terini üretecek eyleyici fonksiyonu çalıştırılır.

```
*****

Sicaklik Degeri.....: 13
Sogutucu Durumu.....: ACIK

[1]Oturumu Kapat
[2]Tekrar Ölç
[3]Sogutucu Durumu Degistir
Yapilacak islemi secin.(1/2/3)
*****
```


j.

Singleton ve Observer

singleton temelinde oluşturulan sınıftan sadece bir adet nesne üretilmesini garantileme durumudur. Ama bunu yaparken sınıfın constructor metodu private olmalıdır. Böylece constructor sadece sınıf içerisinde bir kez çalıştırılır. Instance sınıfın içerisinde private static olarak tutulmalıdır aynı zamanda tutulan instanceı döndüren public static metod sınıf içerisinde yazılmalıdır.

Uygulama içerisinde singleton yapısını Eyleyici, SicaklikAlgilayici ve AgArayuzu classlarında kullandım. Örnek olarak eyleyici classı:

```
public class Eyleyici {  
    private static Eyleyici eyleyici = new Eyleyici(); //singleton yapısı  
    private Eyleyici()  
    {  
    }  
    public static Eyleyici getSingleton()  
    {  
        return eyleyici;  
    }  
}
```

Observer pattern veriyi update metodu ile devamlı olarak almaz. Sunucudaki verinin güncellenmesini denetler ve güncelleme olursa veriyi alır