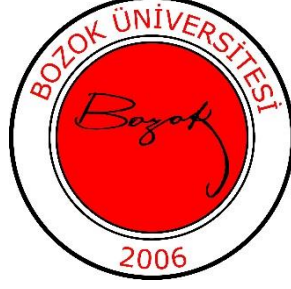


Mart 2020
BOZOK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



EV OTOMASYON SİSTEMİ

12/04/2020

YAZILIM MÜHENDİSLİĞİ ÖDEVİ

9. GRUP

Esra YÜCE

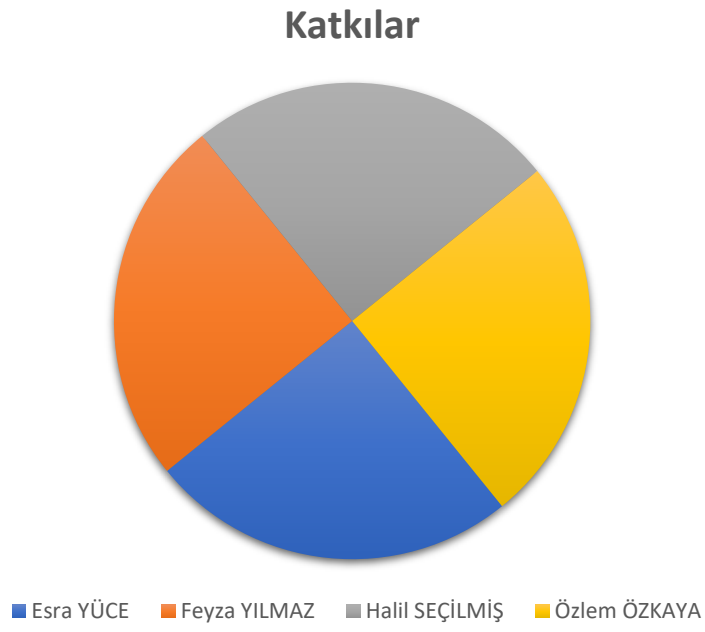
Feyza YILMAZ

Halil SEÇİLMİŞ

Özlem ÖZKAYA

BİREYSEL KATKI DAĞILIMI

Tüm üyeler bu rapora eşit katkıda bulundu.



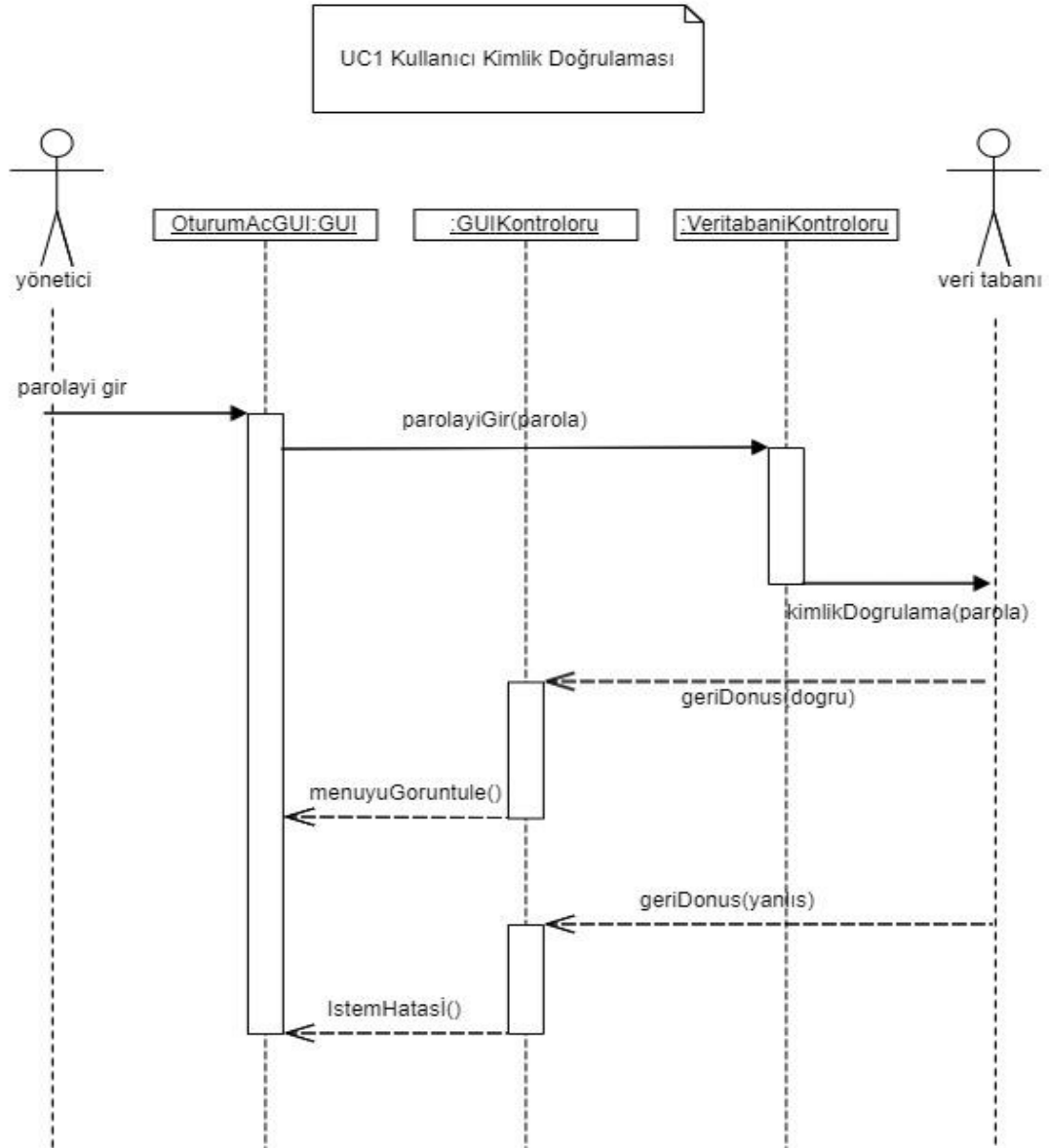
İÇİNDEKİLER

Bölüm 1.....	4
1.1. Etkileşim Diyagramları.....	4
Bölüm 2.....	19
2.1. Sınıf Diyagramı ve Arayüz Özellikleri.....	19
a. Class Diyagramları.....	19
b. Veri Türleri ve İşlem İşaretleri	21
c. İzlenebilirlik Matrisi	28
2.2. Sistem Mimarisi ve Sistem Tasarımı.....	29
a. Mimari Tarzlar.....	29
b. Alt Sistemlerin Tanımlanması	29
c. Alt Sistemleri Donanıma Eşleme	30
d. Kalıcı Veri Depolama	30
e. Ağ Protokolü	35
f. Global Kontrol Akışı	35
g. Donanım Gereksinimleri.....	36
Bölüm 3.....	37
3.1. Kullanıcı Arayüzü Tasarımı ve Uygulaması	37
3.2. Test Tasarımı	42
3.3. Proje Yönetimi ve Çalışma Planı	47
a. Ekip Üyelerinin Bireysel Katkılarını Birleştirme	47
b. Proje Koordinasyon ve İlerleme Raporu	47
c. Çalışma Planı	47
d. Sorumlulukların Dağılımı	48
3.4. Kaynakça	49

Bölüm 1

1.1. Etkileşim Diyagramları

Use Case-1 Kullanıcı Kimlik Doğrulaması



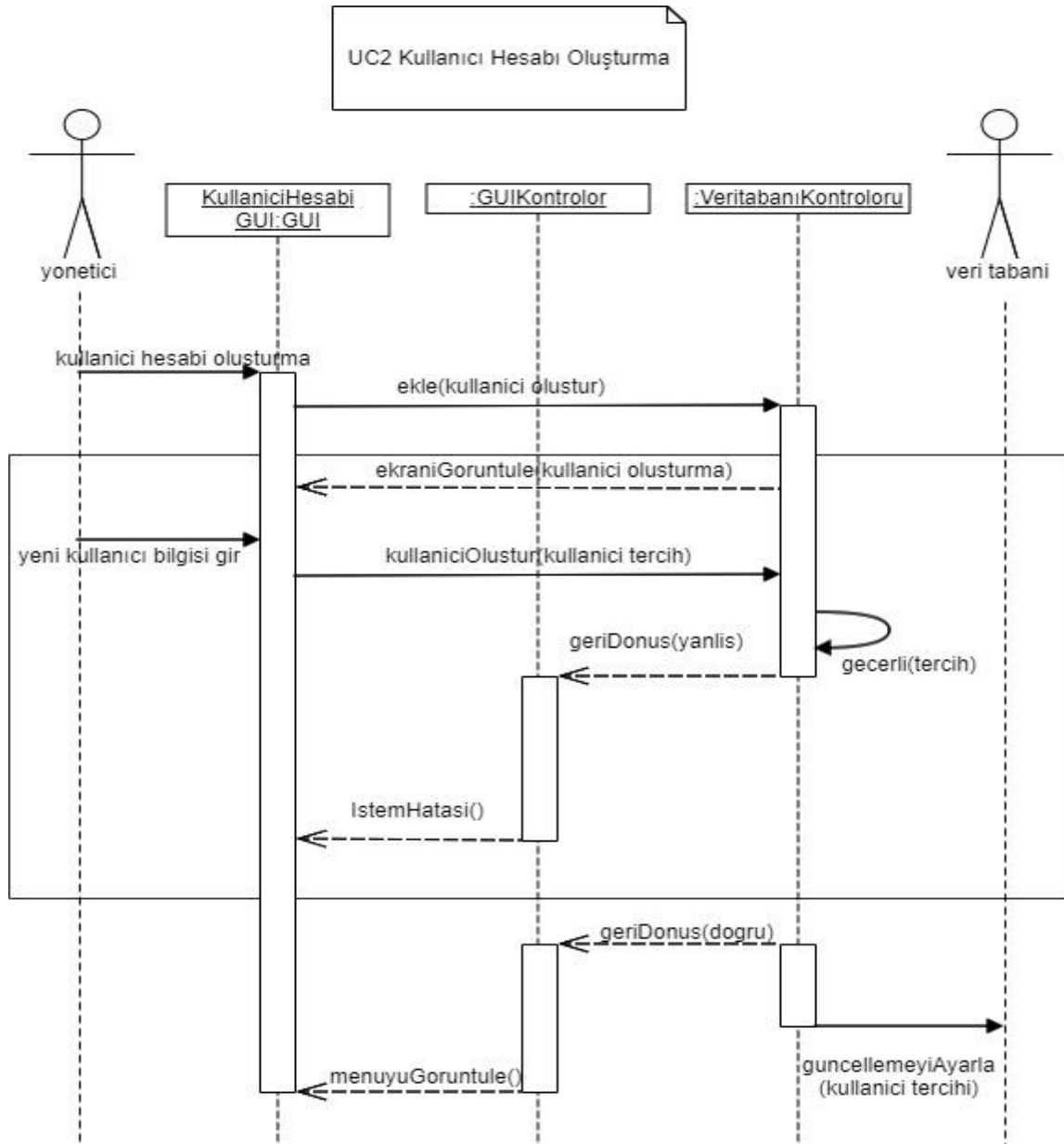
Aktörler:

Yönetici, Veri tabanı

Açıklama:

Kullanıcı sistem tercihlerini değiştirmek, sensörleri uzaktan kontrol etmek veya bilgi istemek için sisteme giriş yapmak ister. Kullanıcının giriş yapabilmesi için sistemde geçerli bir kullanıcı hesabının kimlik bilgilerine sahip olması gerekir. Sistemin ilk açılışında kullanıcıdan “parola” girilmesi istenir. Geçerli bilgiler girilmesi durumunda kullanıcı ana sayfaya yönlendirilir. Geçersiz bir giriş sistem tarafından bir hata bildirimi ve günlüğe kaydetme ile sonuçlanır. Bu durumda kullanıcıdan tekrar şifre girmesi istenir.

Use Case-2 Kullanıcı Hesabı Oluşturma



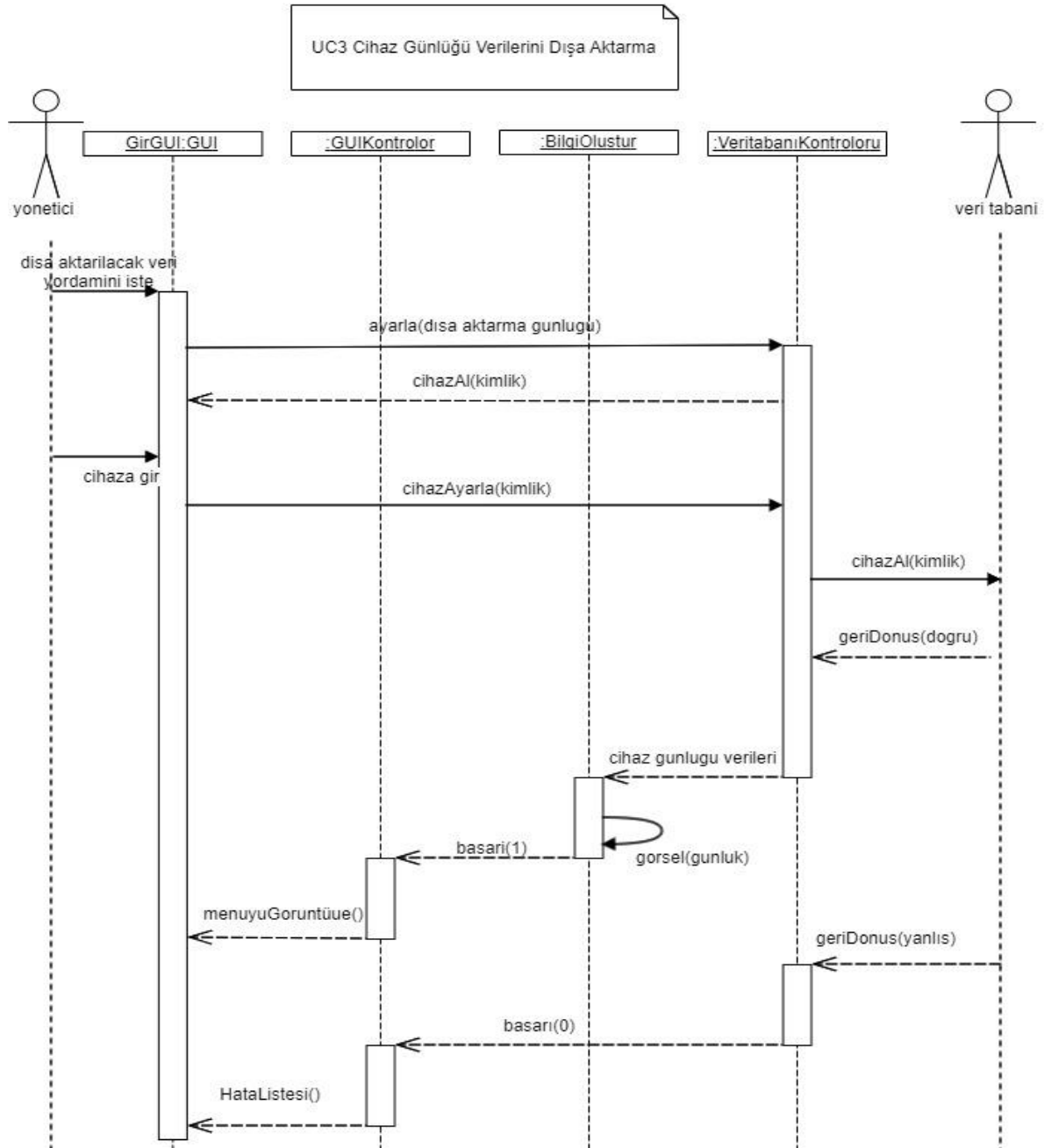
Aktörler:

Yönetici, Veri tabanı

Açıklama:

Yeni kullanıcı ekleneceği durumda yönetici, sisteme yeni kullanıcı oluşturma talebinde bulunur ve gerekli kontrollerden geçerek “yeni kullanıcı bilgi girişi” ekranına gelir. Bilgi girdisinden sonra kullanıcı ayrıcalık seviyesini belirtir. Bilgilerin eşleşmemesi durumunda sistem tarafından verilen hata mesajı ile karşılaşılır ve tekrar kullanıcı bilgi girişi sayfasına yönlendirilir. Bilgiler eşleşiyor ise yeni kullanıcı veri tabanına kaydedilerek ana sayfaya yönlendirilir.

Use Case-3 Cihaz Günlüğü Verilerini Dışa Aktarma



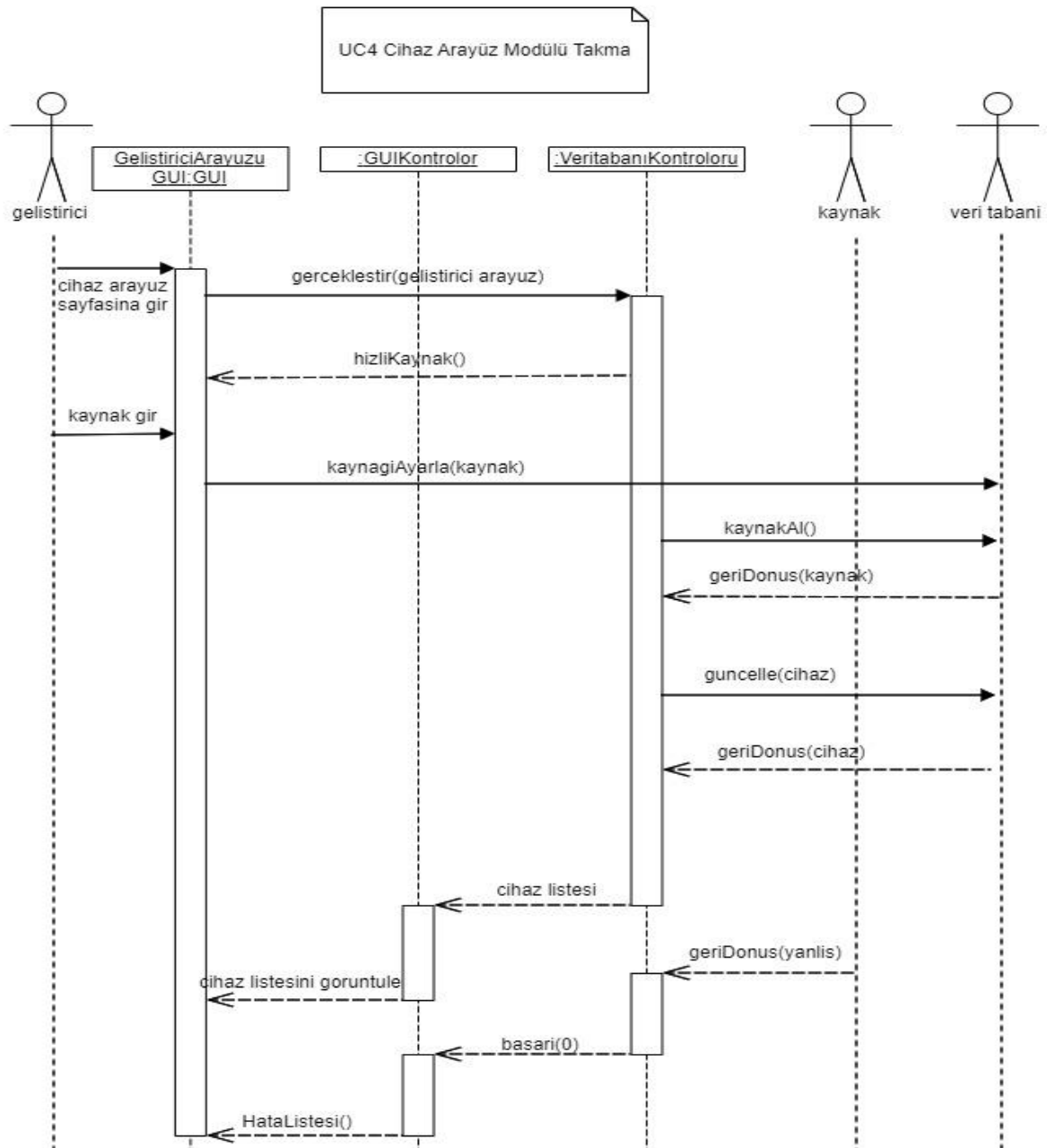
Aktörler:

Yönetici, Veri tabanı

Açıklama:

Kimliği doğrulanmış bir kullanıcı, yetkilendirilmiş aygıtlar için veri değerlerini sistemden alabilir. İlk olarak kullanıcı(yönetici) verilerin dışa aktarılması talebinde bulunur. Gerekli kontrollerden geçen isteğin değerlendirilmesi için cihaz kimliği istenir. Alınan cihaz kimliği veri tabanından kontrol edilir. İstenilen cihaz verileri doğrulandıysa cihaz günlükleri alınır dışa aktarılmadan önce görsel olarak dönüştürülür ve sunulur. İstenilen cihaz verilerinin doğrulanmaması durumunda kullanıcı hata mesajı ile karşılaşır.

Use Case-4 Cihaz Arayüz Modülünü Takma



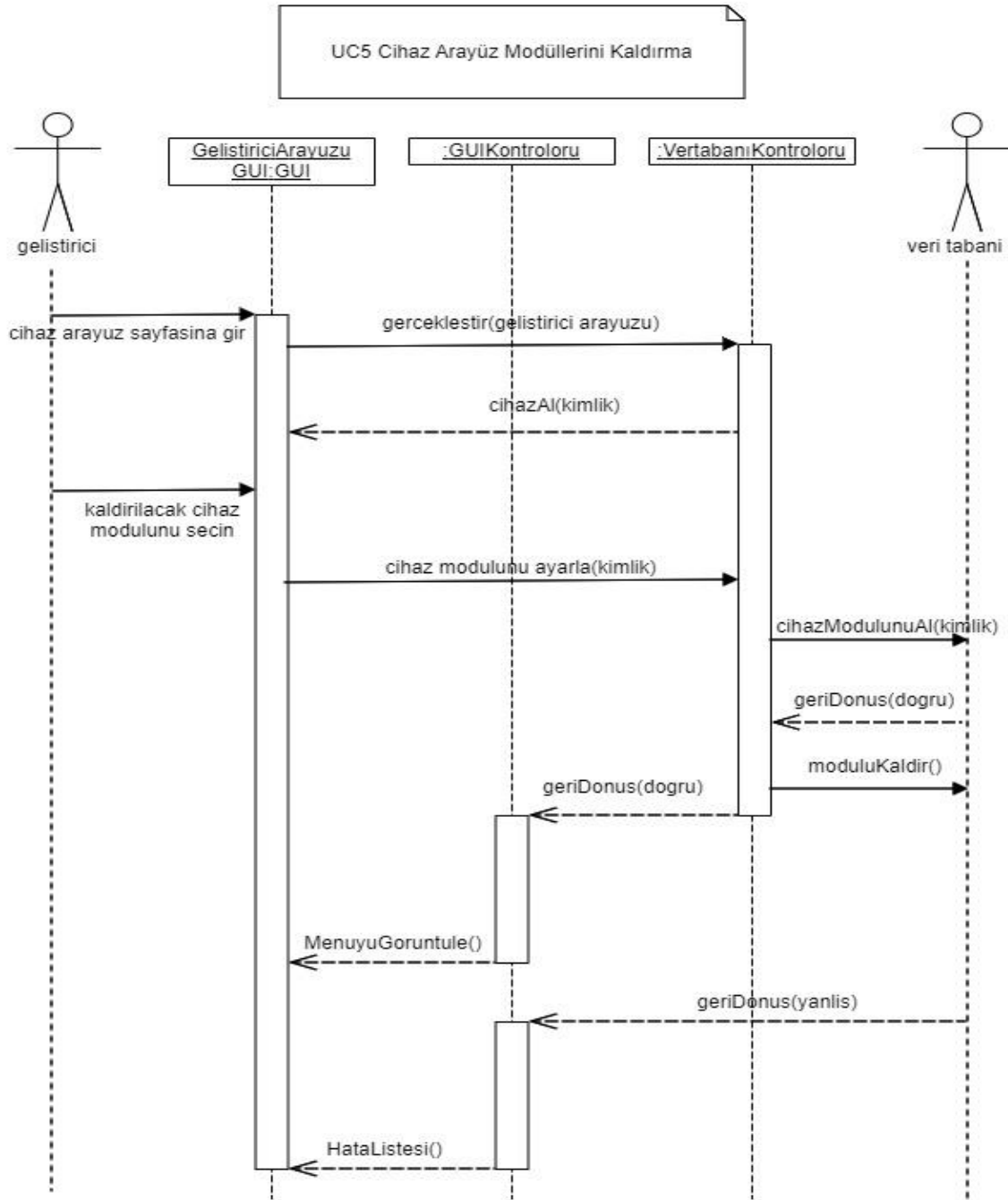
Aktörler:

Geliştirici, Veri tabanı, Kaynak

Açıklama:

Kimliği doğrulanmış bir geliştirici sistemi yeni aygıtları destekleyecek şekilde güncellemek isteyebilir. Yönetici sensör ve denetleyici verilerini makul değerlere ve birimlere değiştirmek için satıcı(geliştirici) tarafından sağlanan bir modül yükleyebilir. İlk olarak cihaz arayüz sayfasına girilir gerçekleştirilecek işlem bildirilir ve gerekli kontrollerden sonra eklenecek kaynağın girilmesi istenir. Kaynak veri tabanına eklenir. Bu şekilde veri tabanı güncellenerek kaynak eklenmiş olur. İşlemin başarılı olmasıyla birlikte ekranda cihaz listesi görüntülenir. Aksi durumda hata mesajı ile karşılaşılır.

Use Case-5 Aygıt Arabirim Modüllerini Kaldırma



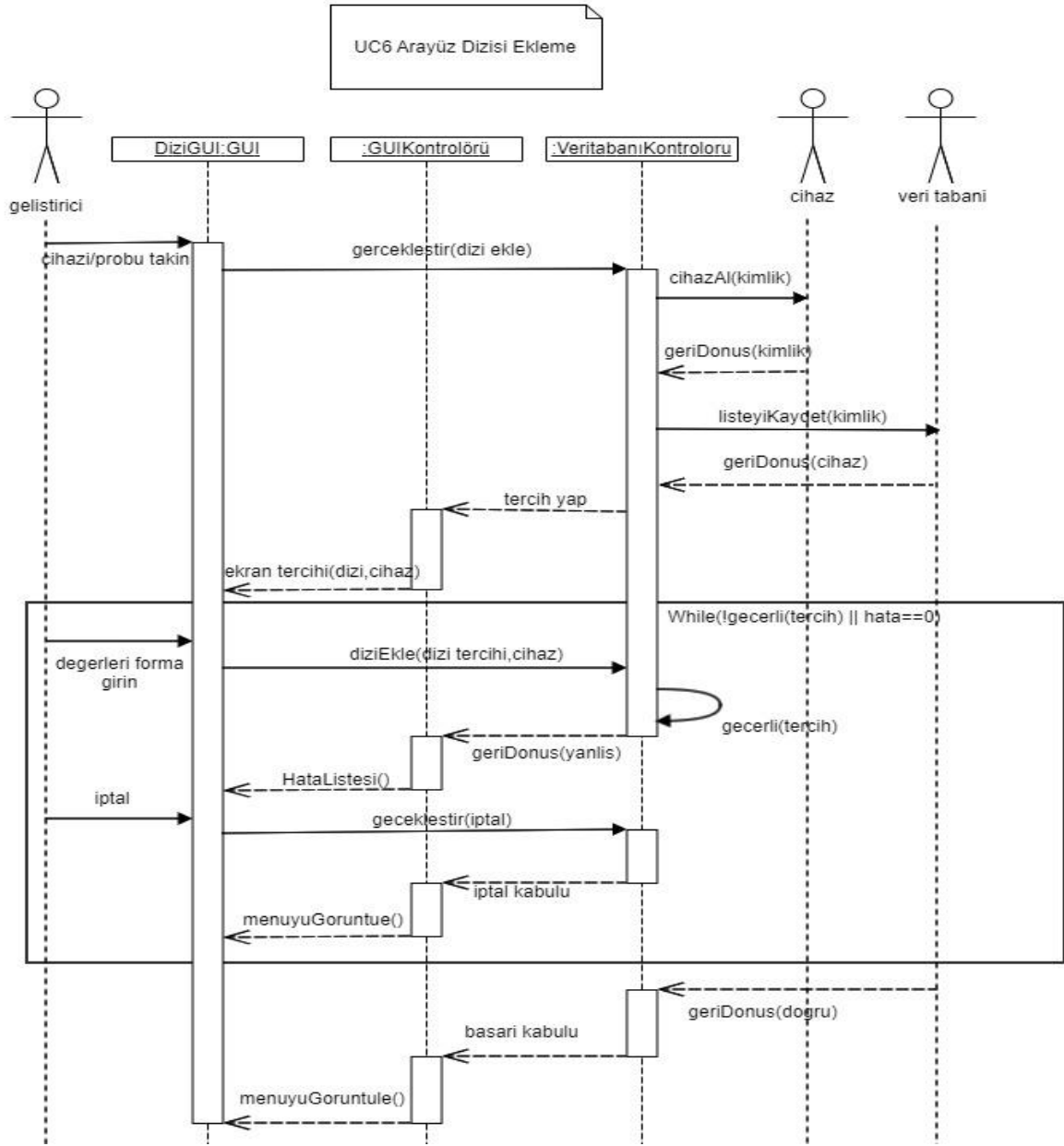
Aktörler:

Geliştirici, Veri tabanı

Açıklama:

Kimliği doğrulanmış bir geliştirici bir aygıtın arabirim modülünü kaldırmak isteyebilir. Bu durumda tıpkı modül ekleme işlemleri gibi işlemler uygulanır.

Use Case-6 Arabirim Dizisi Ekleme



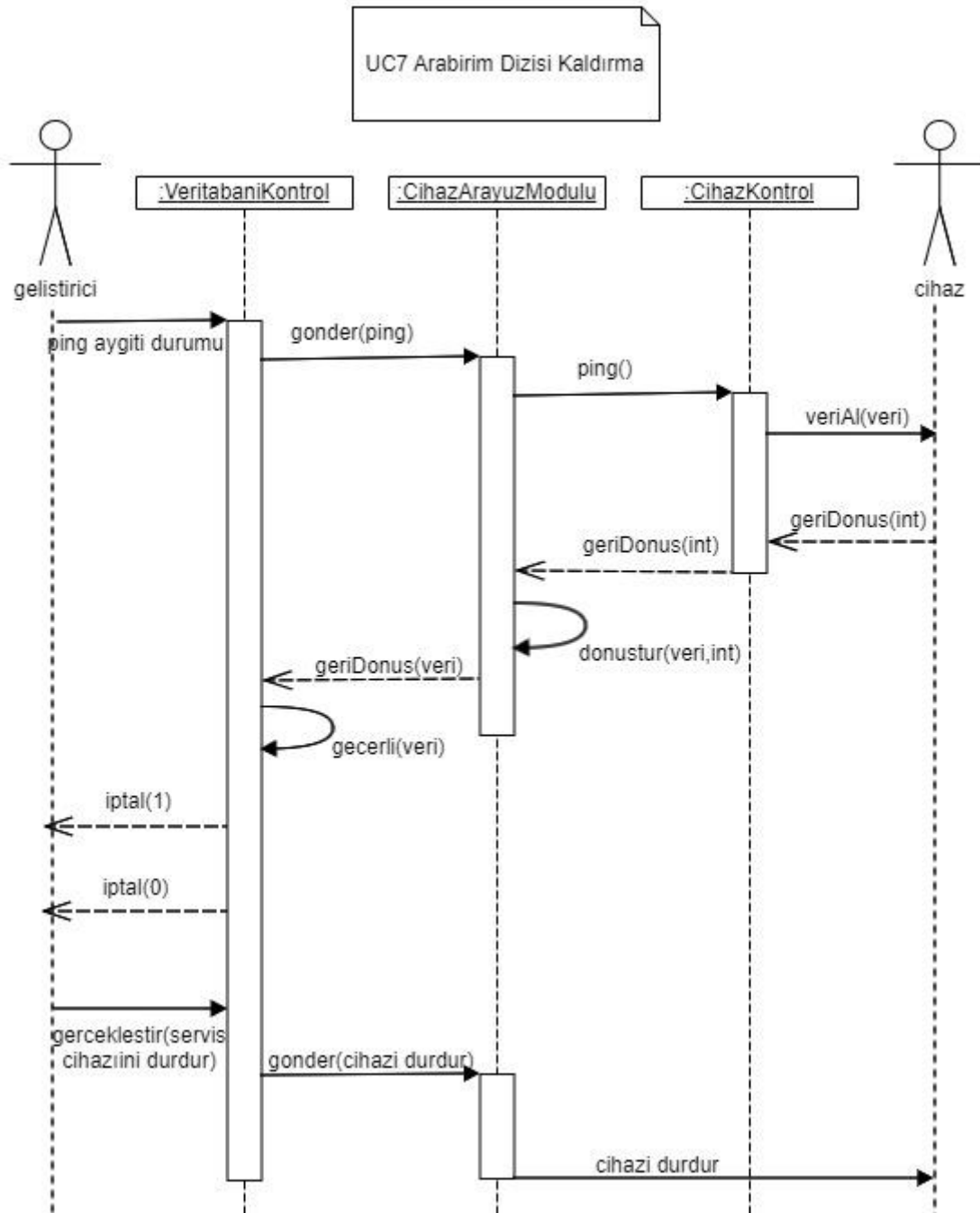
Aktörler:

Geliştirici, Veri tabanı, Cihaz

Açıklama:

Kimliği doğrulanmış bir geliştirici, aygıt değerlerini denetlemek ve okumak için ek fiziksel veya sanal aygıt yuvaları kümesi eklemek ister. İlk olarak dizi ekleme isteğinde bulunulur. Gerekli kontrollerden geçen istek cihazdan kimlik alınarak veri tabanına kayıt için gelir. Daha sonra dizi, cihaz tercihi yapılır ve döngüye girer. Döngüde değerler forma girilir tercih edilen dizi, cihaz geçerli değilse hata mesajı ile karşılaşılır. İptal işlemi için güncelleme gerçekleştirilir ve ana sayfaya yönlendirilir ve döngü sonlandırılır. Eğer geçersiz bir durum söz konusu değilse dizi veri tabanına kaydolur, işlem başarıyla sonlanır ve ana sayfaya yönlendirme yapılır.

Use Case-7 Arabirim Dizisini Kaldırma



Aktörler:

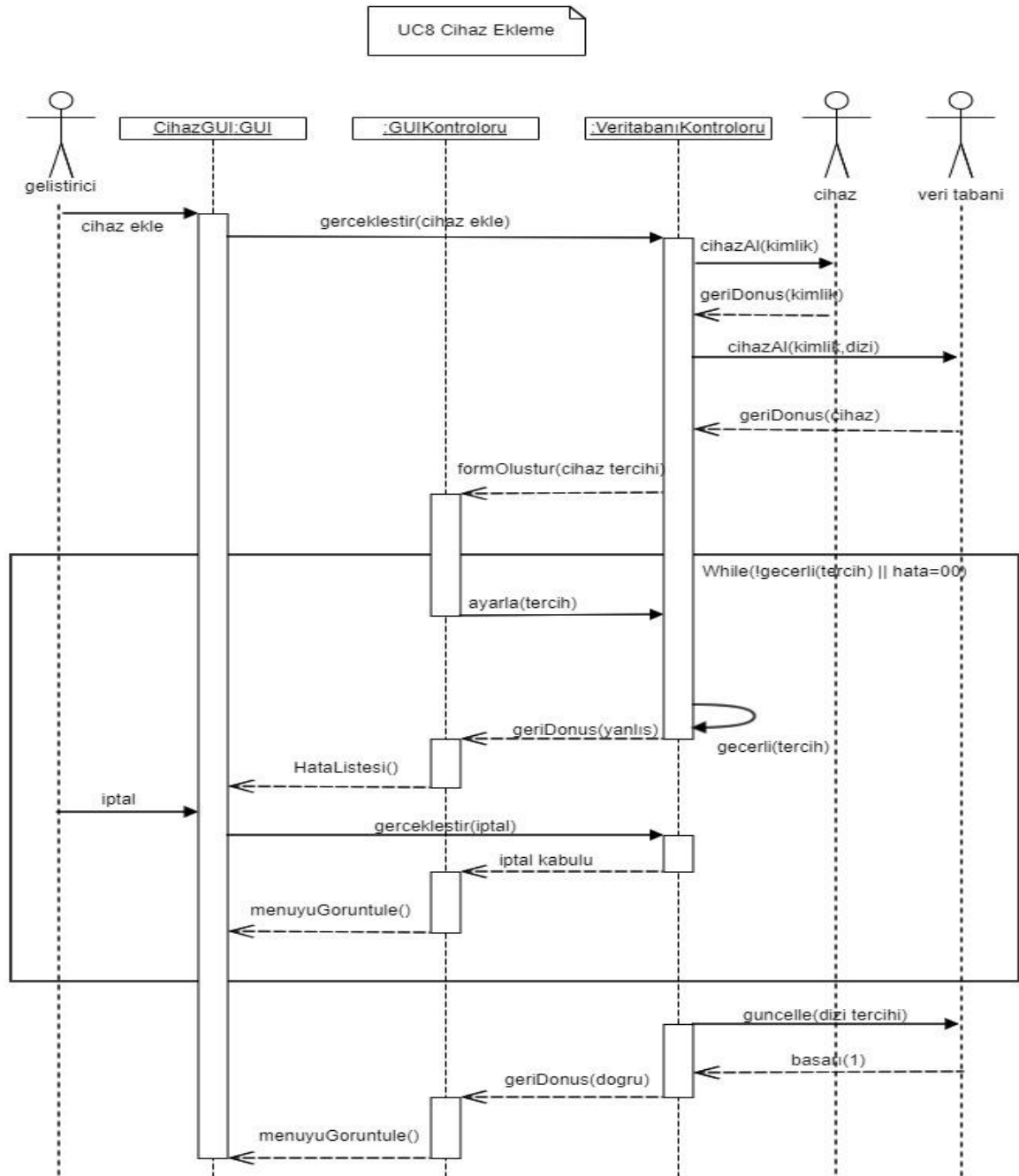
Geliştirici, Cihaz

Açıklama:

Kimliği doğrulanmış bir yönetici, cihazın arızalanması veya devre dışı bırakılması durumunda bir aygıt arabirimi dizisini kaldırmak isteyebilir. Bu durumda geliştiriciye haber verilir.

Aygıt durumundan ping gönderilir. Cihazdan değer alınır bu değerler int(tam sayı) türündedir. Int türündeki değerler dönüştürülür, geçerli veri ise iptal işlemi yani kaldırma işlemi gerçekleştirilir. Geçerli veri değilse cihazı durdur komutuyla cihaz durdurulur.

Use Case-8 Cihaz ekleme



Aktörler:

Geliştirici, Veri tabanı, Cihaz

Açıklama:

Geliştirici kullanıcının isteği üzerine sisteme yeni cihaz ekleyebilir. Bu durumda ilk olarak yeni cihaz ekleme talebinde bulunulur. Kontrolden geçilir cihazdan cihaz kimliği alınır. Alınan cihaz kimliği veri tabanına gider karşılaştırma yapılarak cihazın uygunluğu tespit edilir. Daha sonra cihaz form ekranından cihaz tercihleri girilir. Bu tercihlerin geçerli olmaması durumunda hata mesajı ile karşılaşılar ve iptal işlemi gerçekleştirilir. Tercihlerin kabulü durumunda ise gerekli güncellemeler yani veri tabanına kayıt işlemi gerçekleşir ve ana sayfaya yönlendirilir.

```

sequenceDiagram
    actor gelistirici
    participant CihazGUI as CihazGUI:GUI
    participant GUK as :GUIKontroloru
    participant VK as :VeritabanıKontroloru
    participant CAM as :CihazArayuzModulu
    actor veritabanı
    actor cihaz

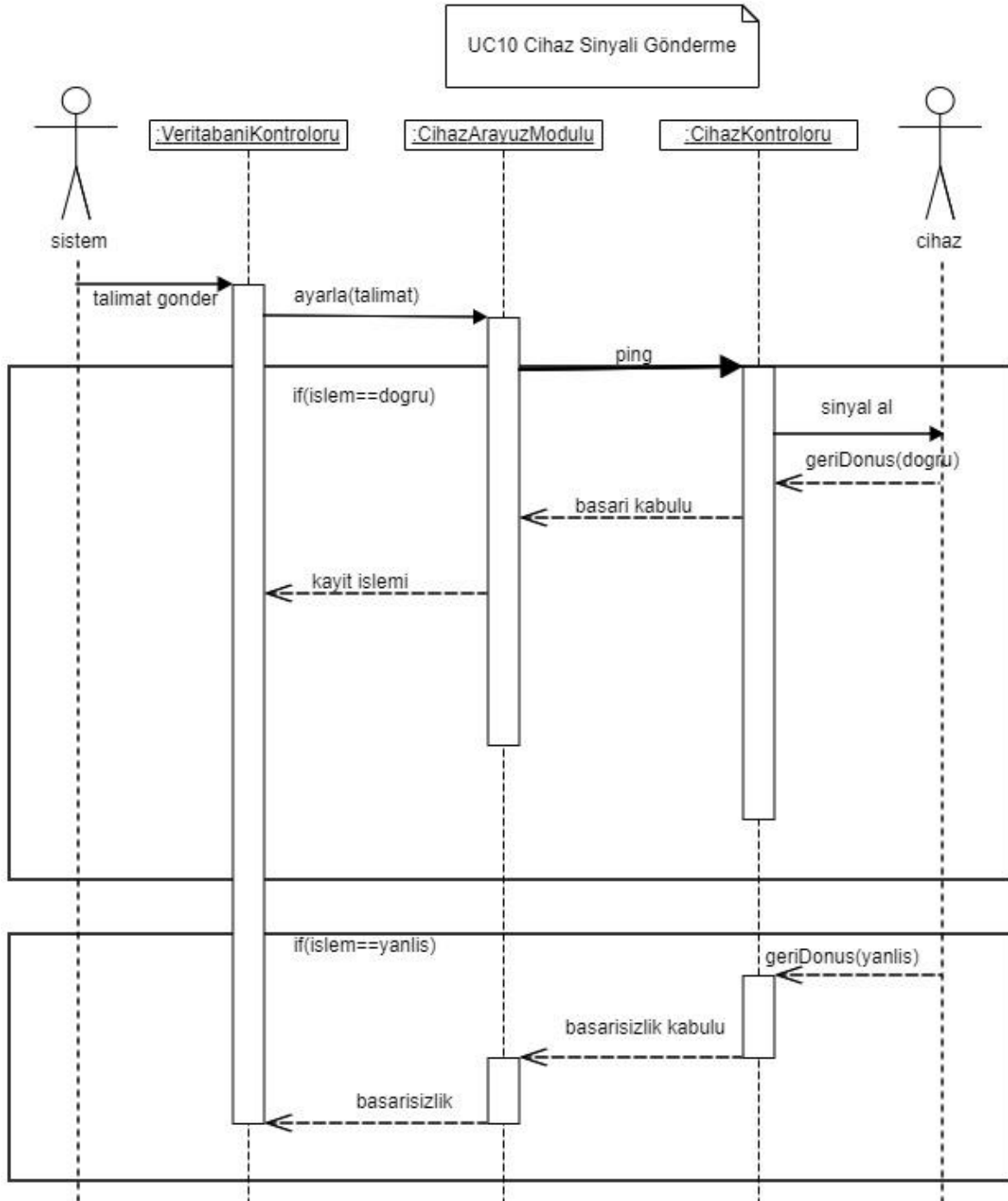
    Note over CihazGUI, VK: UC9 Cihaz Kaldırma

    gelistirici->>CihazGUI: cihaz kaldırma istegi
    activate CihazGUI
    CihazGUI->>VK: gerceklestir(cihaz kildir)
    activate VK
    VK->>CAM: cihazAl(cihaz listesi)
    activate CAM
    CAM->>VK: geriDonus(cihaz listesi)
    deactivate CAM
    VK-->>CihazGUI: cihaz listesi
    deactivate VK
    gelistirici->>CihazGUI: cihaz gir
    activate CihazGUI
    CihazGUI->>VK: cihazAyarla(kimlik)
    activate VK
    VK->>CAM: cihazKaldir(kural, cihaz)
    activate CAM
    CAM->>veritabanı: if(!iptal)
    activate veritabanı
    veritabanı->>CAM: cihazı devre dışı bırak
    deactivate veritabanı
    CAM->>VK: geriDonus(basarili)
    deactivate CAM
    VK-->>GUK: basari(1)
    deactivate VK
    GUK->>CihazGUI: tanimlama
    activate CihazGUI
    CihazGUI->>GUK: menuyuGoruntule()
    deactivate CihazGUI
    GUK->>VK: basari(0)
    deactivate GUK
    VK->>CAM: tanimlama
    activate CAM
    CAM->>GUK: HataLiatesi()
    deactivate CAM
    GUK->>CihazGUI: HataLiatesi()
    deactivate GUK
    gelistirici->>CihazGUI: kaldirmayi iptal et
    activate CihazGUI
    CihazGUI->>VK: gerceklestir(iptal)
    activate VK
    VK->>CAM: if(iptal)
    activate CAM
    CAM->>veritabanı: 
    deactivate CAM
    VK->>GUK: iptal onayi
    deactivate VK
    GUK->>CihazGUI: menuyuGoruntule()
    deactivate GUK
    deactivate CihazGUI
  
```

Geliştirici, Veri tabanı, Cihaz

Geliştirici kullanıcının isteği üzerine sisteme yeni cihaz ekleyebildiği gibi var olanı da kaldırabilir. İlk olarak kaldırma isteğinde bulunulur. Veri tabanından cihaz listesi alınır, geliştirici bu cihazlar arasından hangisini kaldırmak istediğini seçer. Kaldırma işlemini onaylama için döngü kullanılır. Veri tabanına cihaz kaldırma talimatı verilir aynı zamanda cihaz devre dışı bırakılır. Geri dönüş başarılı olursa ana sayfaya yönlendirme yapılır. Başarısız olması durumunda hata mesajı ile karşılaşılır. Bu durumda verilerin zarar görmesini engellemek için tekrar döngüye girilir güncelleme ile iptal işlemi gerçekleştirilir.

Use Case-10 Cihaz Sinyali Gönderme



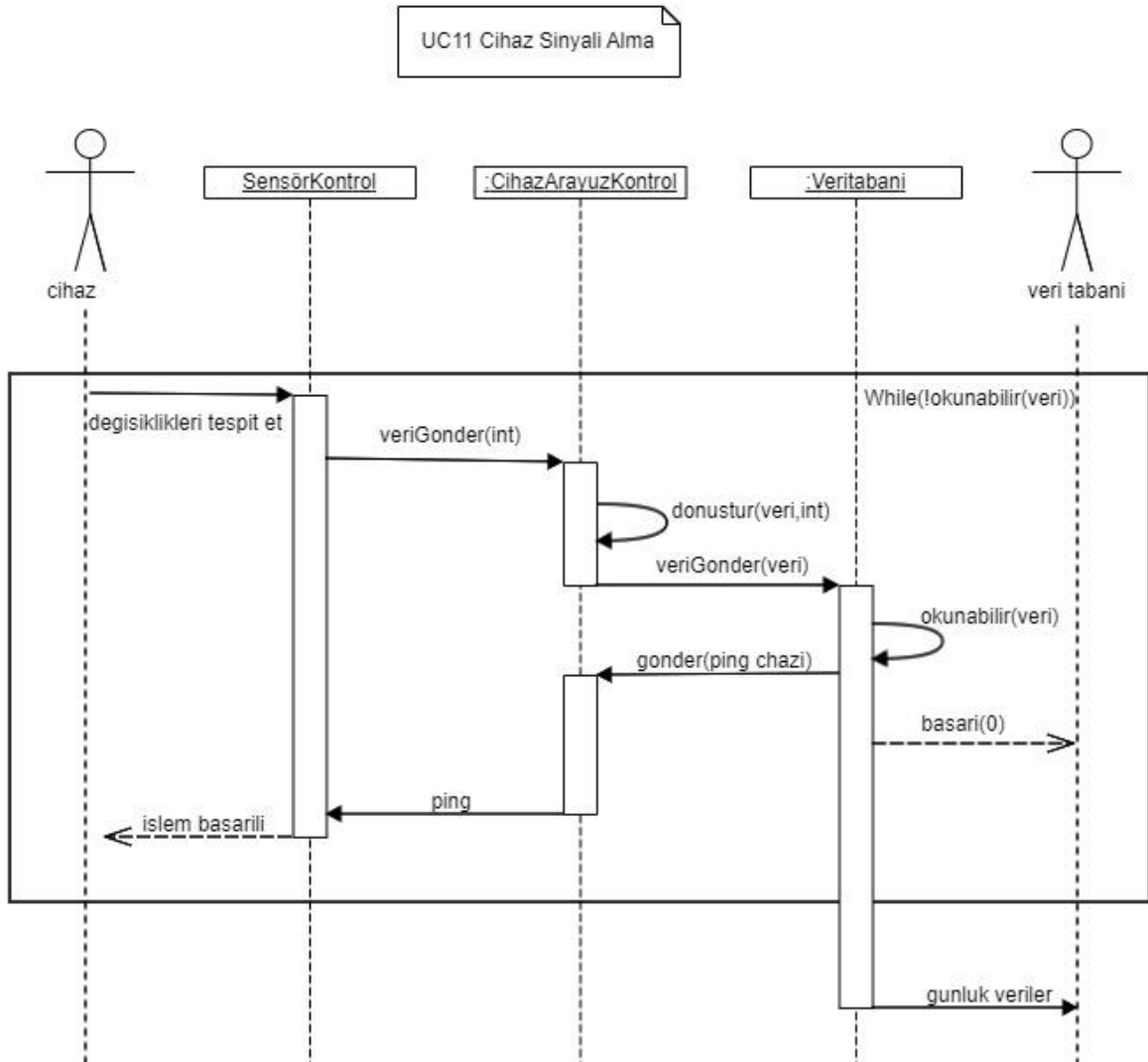
Aktörler:

Sistem, Cihaz

Açıklama:

Cihaz sinyali gönderme işlemi ilk olarak talimat göndermekle başlar. Talimat ayarlanır cihaz kontrolüne ping gönderilir. Cihaz sinyali alabildiyse olumlu geri dönüt verir ve kayıt işlemi gerçekleştirilir. Cihaz sinyali alamadığı durumda işlem başarısızlıkla sonuçlanır.

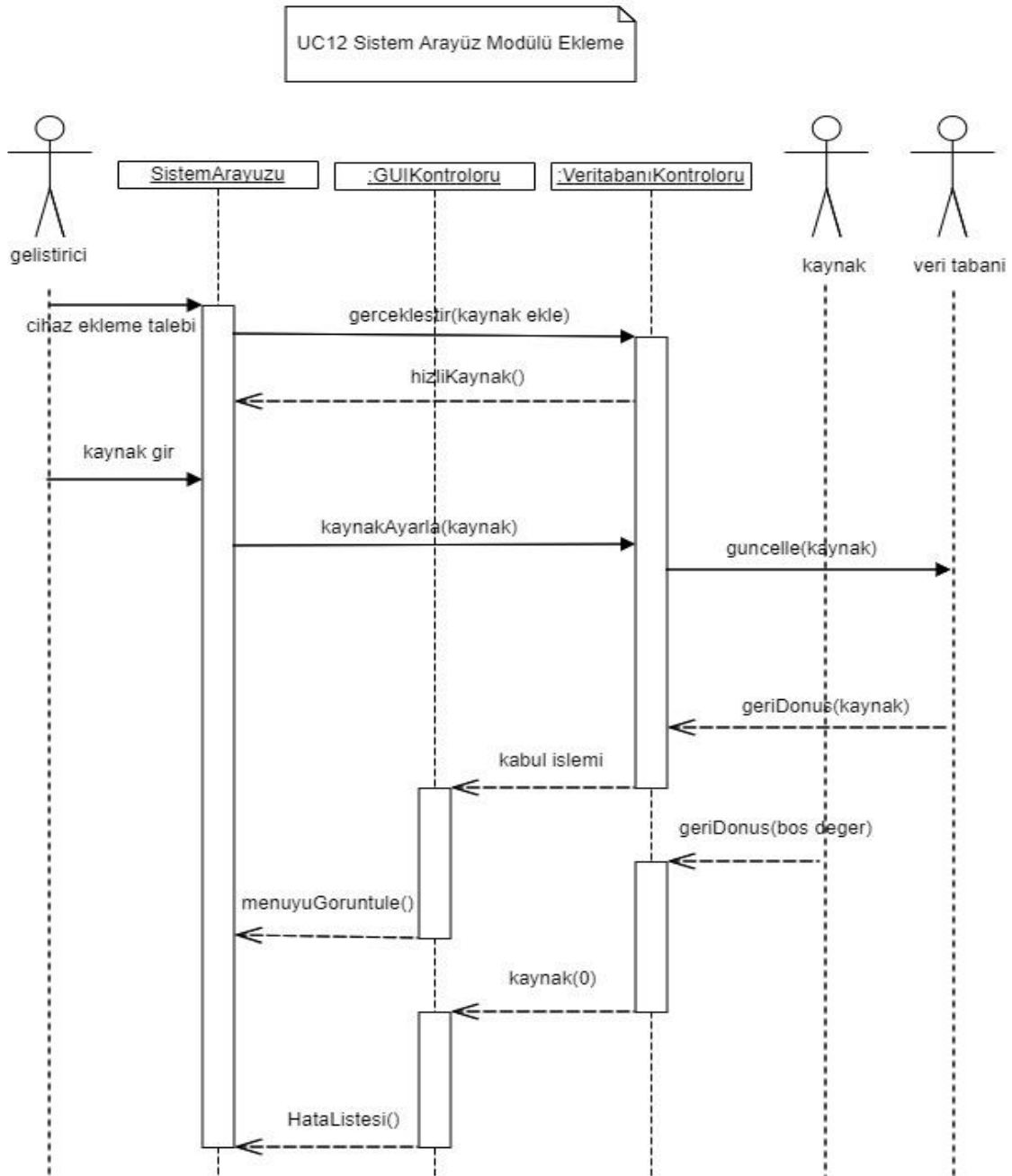
Use Case-11 Cihaz Sinyali Alma



Aktörler:

Cihaz, Veri tabanı

Use Case-12 Sistem Arayüz Modülü Ekleme



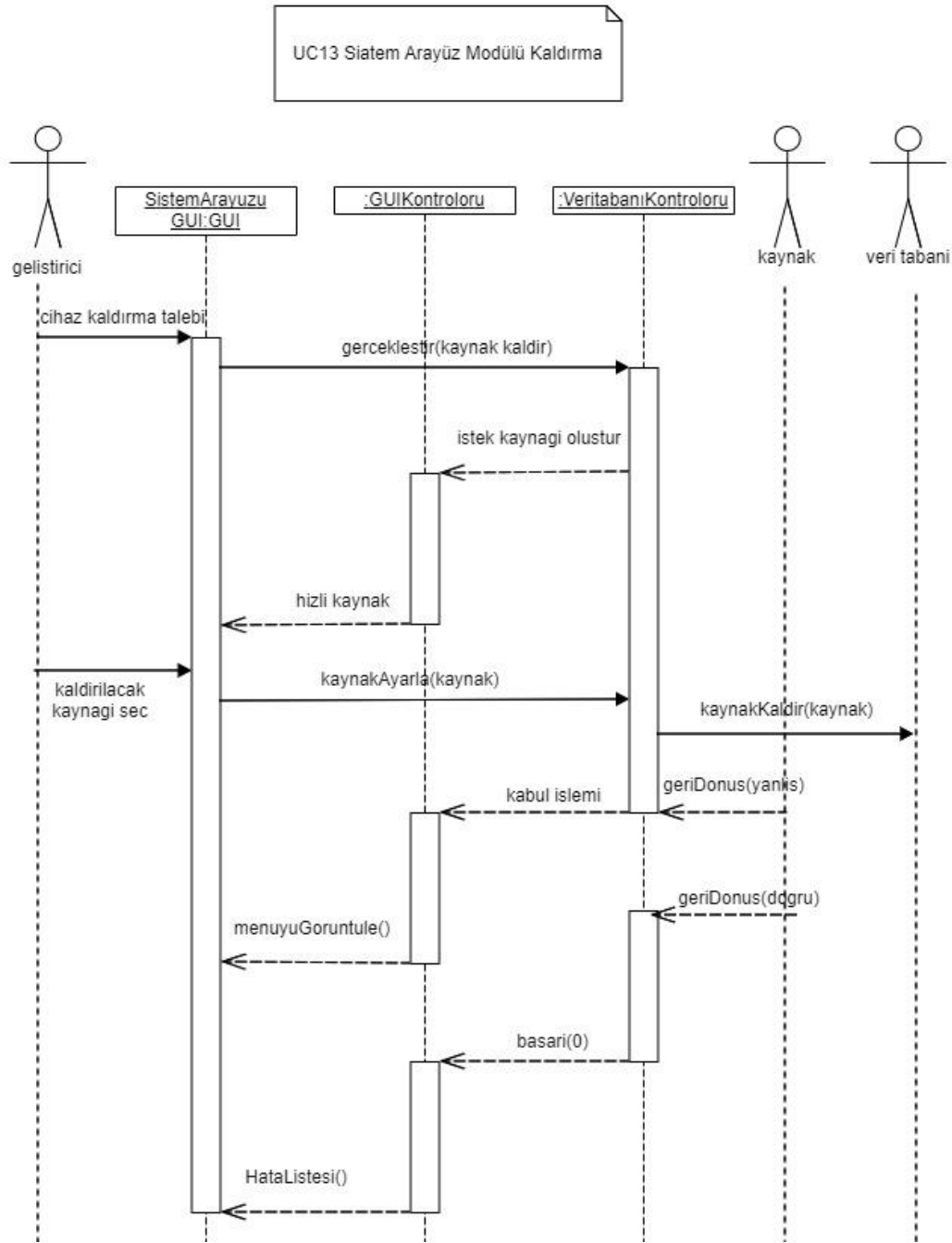
Aktörler:

Geliştirici, Veri tabanı, Kaynak

Açıklama:

Kullanıcı sisteme geliştirici aracılığıyla arayüz modülü ekleyebilir. Bu durumda geliştirici ilk olarak sistem ara yüzünde cihaz ekleme talebinde bulunur. Veri tabanı kontrolünden sonra geliştirici hangi modülü eklemek istediğini yani kaynağı girer. Girilen kaynak tekrar kontrolden geçer ve veri tabanına kaydı gerçekleştirilir. Kayıt işlemi sorunsuz gerçekleştiyse geri dönüt başarılı olur ve ana sayfaya yönlendirilme yapılır. Kayıt işlemi sorunlu ise geri dönüt hatalı olur ve hata mesajıyla karşılaşılır.

Use Case-13 Sistem Arayüz Modülünü Çıkarma



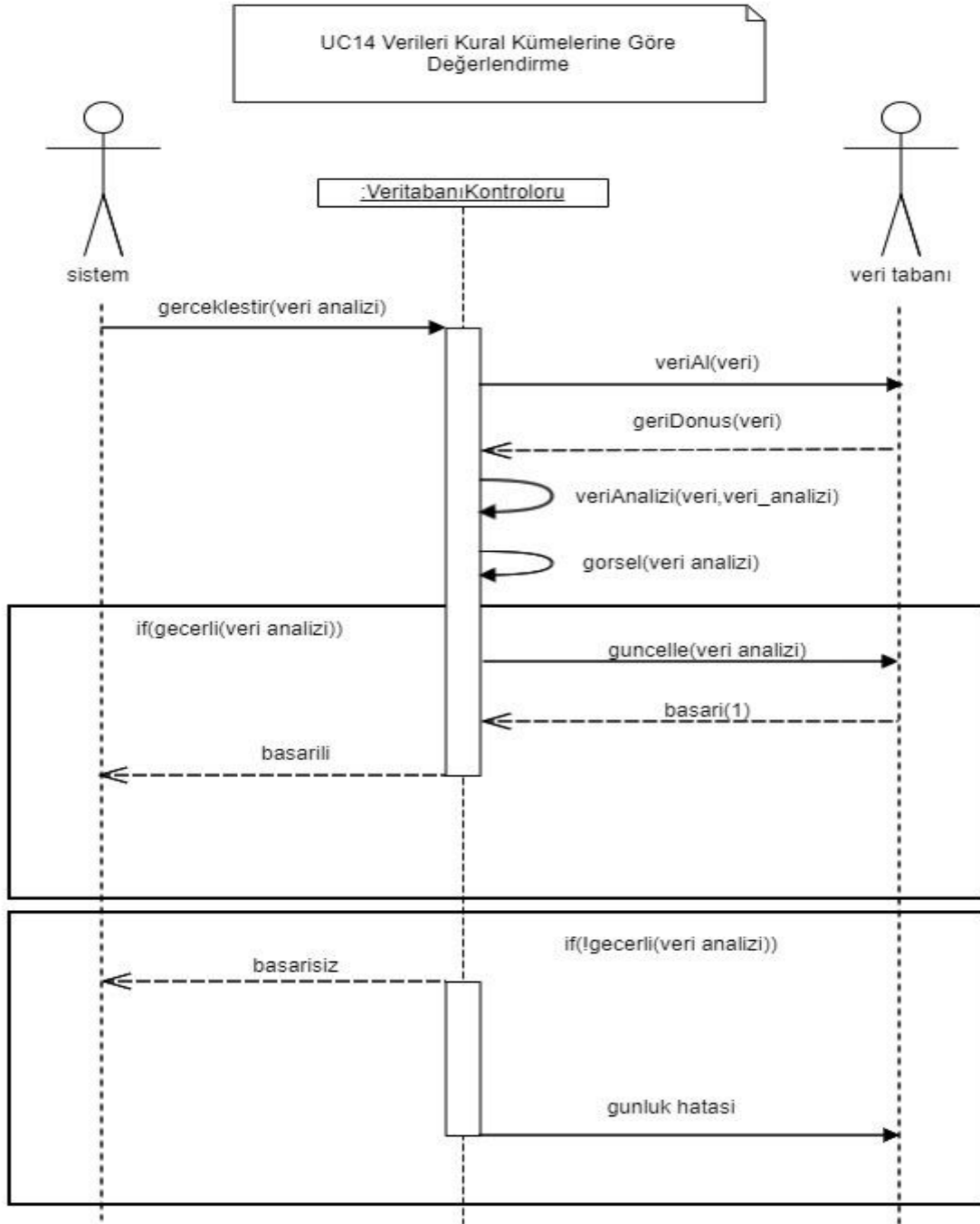
Aktörler:

Geliştirici, Veri tabanı, Kaynak

Açıklama:

Geliştirici ilk olarak sistem ara yüzünde cihaz kaldırma talebinde bulunur. Veri tabanı kontrolünden sonra geliştirici hangi modülü kaldırmak istediğini yani kaynağı girer. Girilen kaynak tekrar kontrolden geçer ve veri tabanından silinir. Kaynaktan veri tabanı kontrolü geri dönüt yanlış ise silme işlemi sorunsuz gerçekleşmiştir. Geri dönüt doğru ise silme işlemi gerçekleşmemiş demektir bu durumda hata mesajı ile karşılaşılır.

Use Case-14 Verileri Kural Kümelerine Göre Değerlendirme



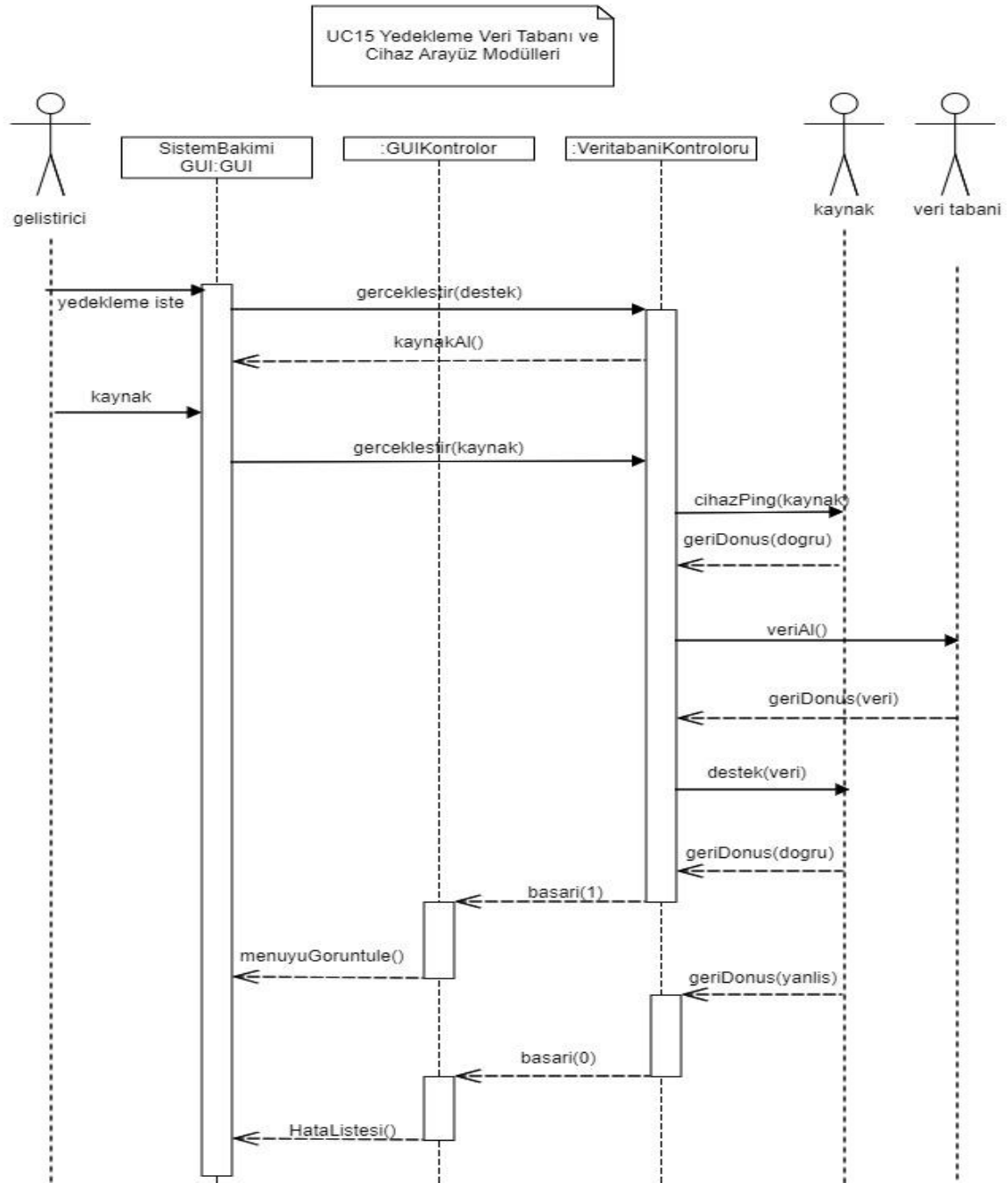
Aktörler:

Sistem, Veri tabanı

Açıklama:

Sistem tarafından veri tabanına komut verilerek veri analizi yapılmak istenir. Veri tabanından alınan verilerin kontrolde her türlü analizi yapılır. Kontrol sonucunda verilerde bir sorunla karşılaşılmazsa veri tabanına gerekli güncelleme yapılır ve işlem başarıyla sonlanır. Kontrol sonucunda verilerde sorunla karşılaşıldıysa işlem başarısızlıkla sonuçlanır ve günlük hatası oluşur.

Use Case-15 Yedekleme Veritabanı ve Cihaz Arayüz Modülleri



Aktörler:

Geliştirici, Veri tabanı, Kaynak

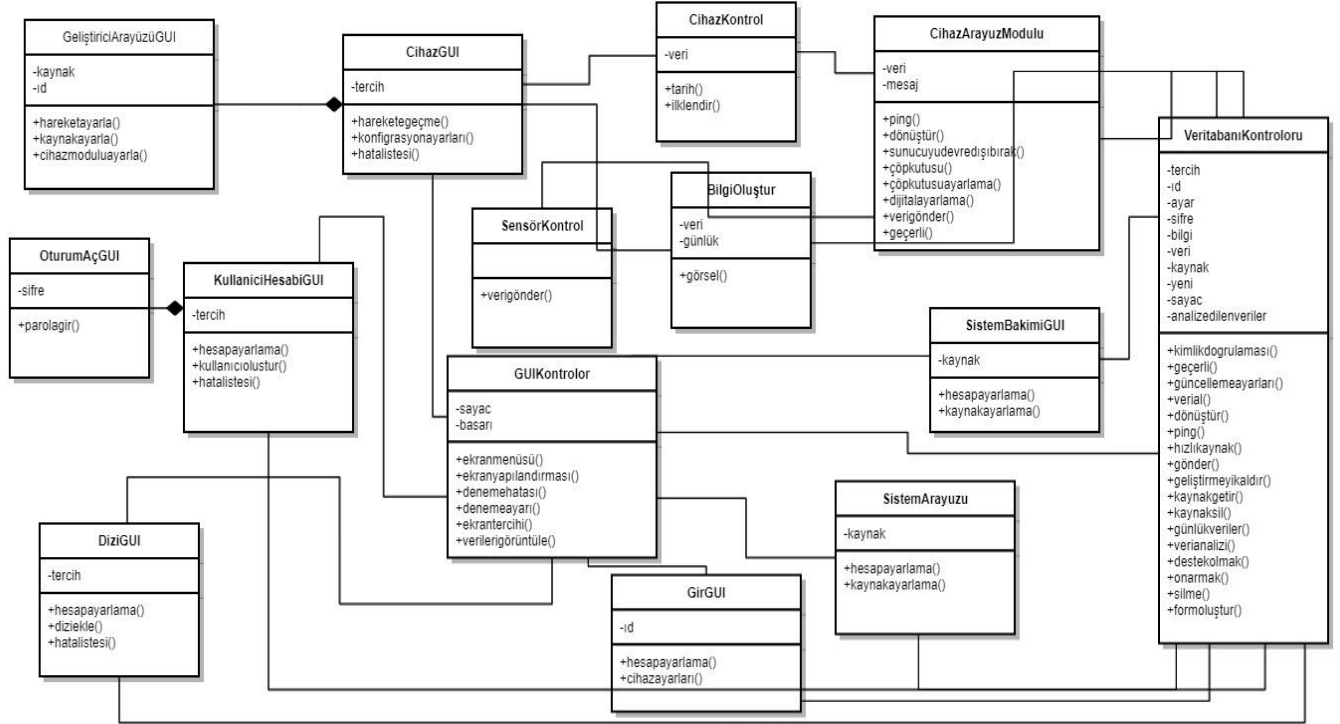
Açıklama:

Geliştirici yedekleme isteğinde bulunur. İstek gerekli kontrollerden geçtikten sonra geliştiriciden kaynak girmesi istenilir. Kontrolde kaynağa ping atılarak doğrulama sağlanır ve veri tabanından yedeklenmesi istenen veriler alınır. Alınan veriler kaynak desteğiyle karşılaştırılır. Geri dönüt doğruysa işlem başarı ile sonuçlanmış olup ana sayfaya yönlendirme yapılır. Geri dönüt yanlış ise işlem başarısız olur hata mesajıyla karşılaştırılır.

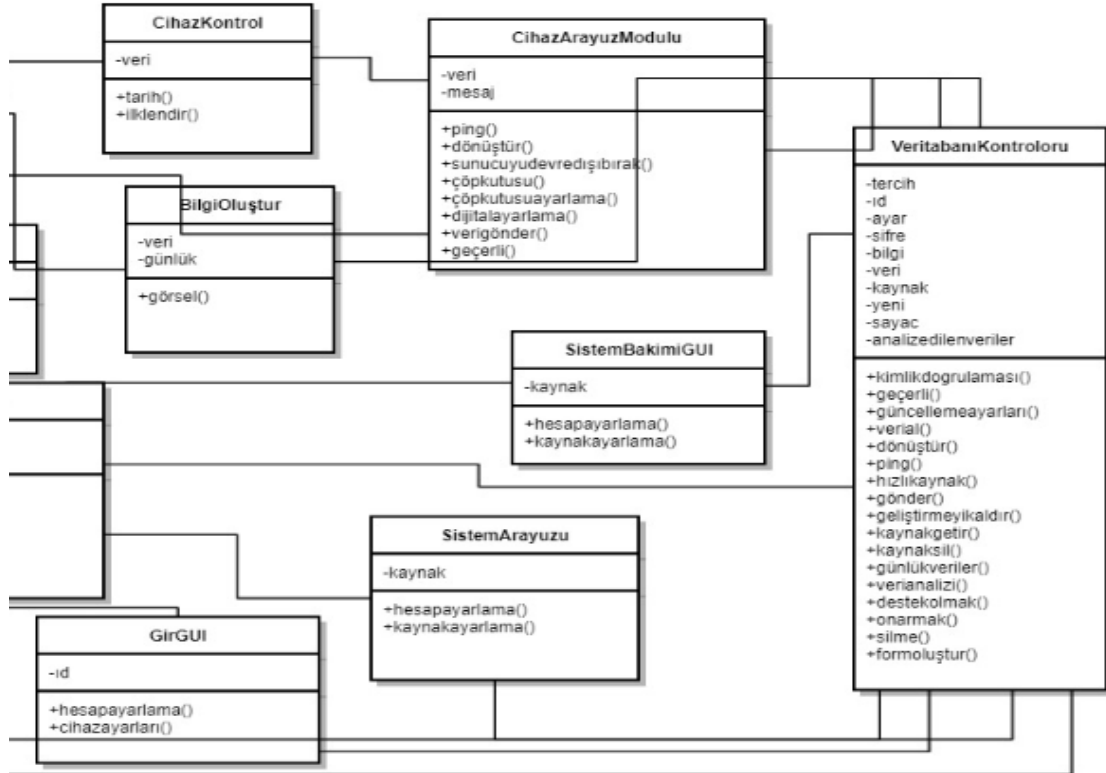
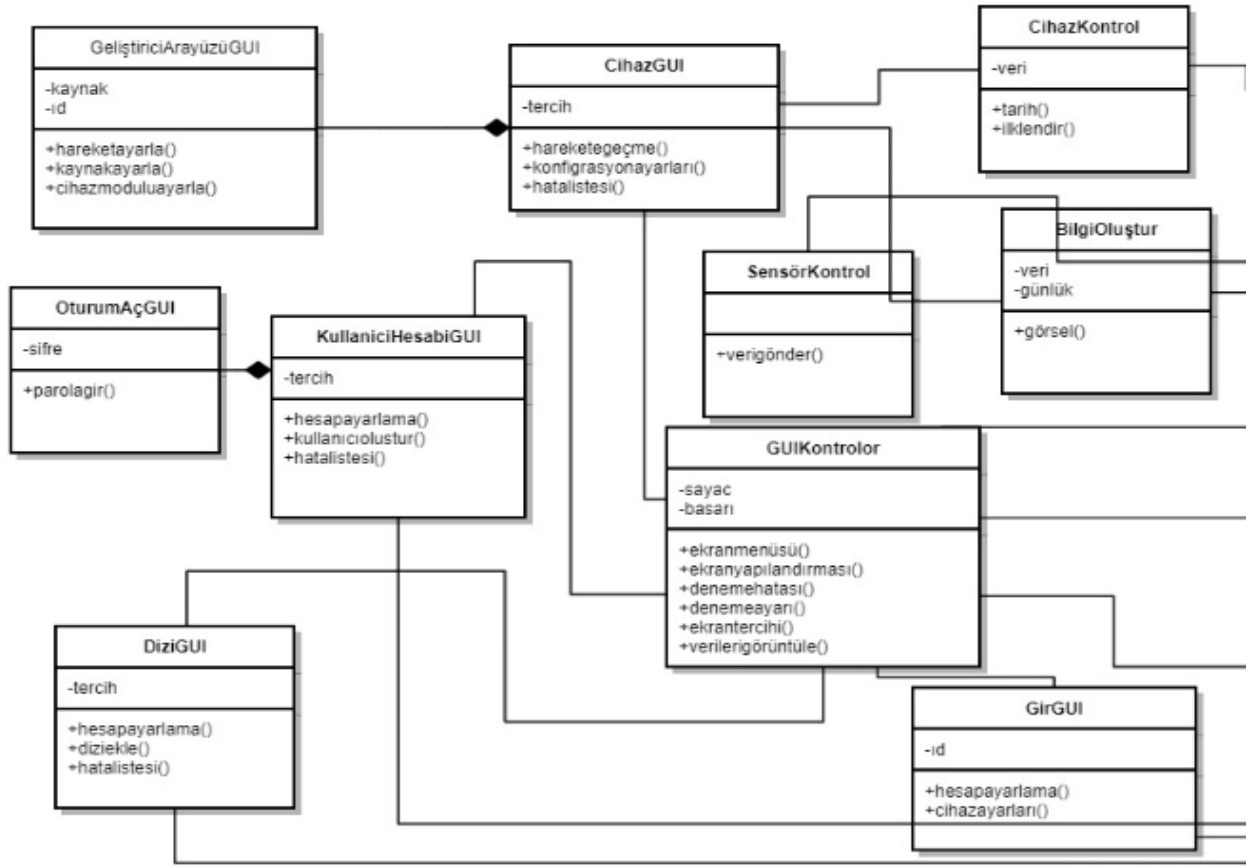
Bölüm 2

2.1. Sınıf Diyagramı ve Arayüz Özellikleri

a. Class Diyagramları



Okunabilirliği artırmak adına aşağıda parça parça class diyagramları vardır.



Yukarıda tüm sınıflar ve ilişkilendirmeler vardır. Projemize uygun classlarımız mevcut olup ilişkilendirmeler yapılmıştır.

Bu UML class diyagramı, akıllı ev sistemimizin farklı alt sistemlerini, yöntemlerini, parametrelerini ve birbirleriyle nasıl etkileştiklerini göstermektedir.

b. Veri Türleri ve İşlem İşaretleri

GeliştiriciArayüzüGUI

Attribute	Type
kaynak	string
id	int

Method	Type
hareketayarla	void
kaynakayarla	void
cihazmoduayarla	void

OturumAçGUI

Attribute	Type
sifre	string

Method	Type
parolagir	void

GUIKontrolleri

Attribute	Type
sayac	İnt
basarı	bool

Method	Type
ekranmenüsü	void
ekranyapılandırması	void
denemehatası	void
denemeayari	void
ekrantercihi	void
verilerigörüntüle	void

KullaniciHesabiGUI

Attribute	Type
tercih	string

Method	Type
hesapayarlari	void
kullaniciolustur	void
hatalistesi	void

SistemBakimiGUI

Attribute	Type
kaynak	string

Method	Type
hesapayarlama	void
kaynakayarlama	void

BilgiOlustur

Attribute	Type
veri	float
gunluk	float

Method	Type
gorsel	void

GirGUI

Attribute	Type
id	int

Method	Type
hesapayarlama	void
cihazayarlari	void

SensörKontrol

Method	Type
verigonder	void

CihazKontrol

Attribute	Type
veri	float

Method	Type
tarih	float
ilklendir	void

CihazGUI

Attribute	Type
tercih	string

Method	Type
hareketegeçme	void
konfigrasyonayarları	void
hatalistesi	void

CihazarAyüzModulu

Attribute	Type
veri	float
mesaj	string

Method	Type
ping	float
donuşturmek	float
sunucuyudevredışıbirak	void
verigonder	void
gecerli	bool

SistemArayüzüGUI

Attribute	Type
kaynak	string

Method	Type
hesapayarlama	void
kaynakayarlama	void

DiziGUI

Attribute	Type
tercih	string

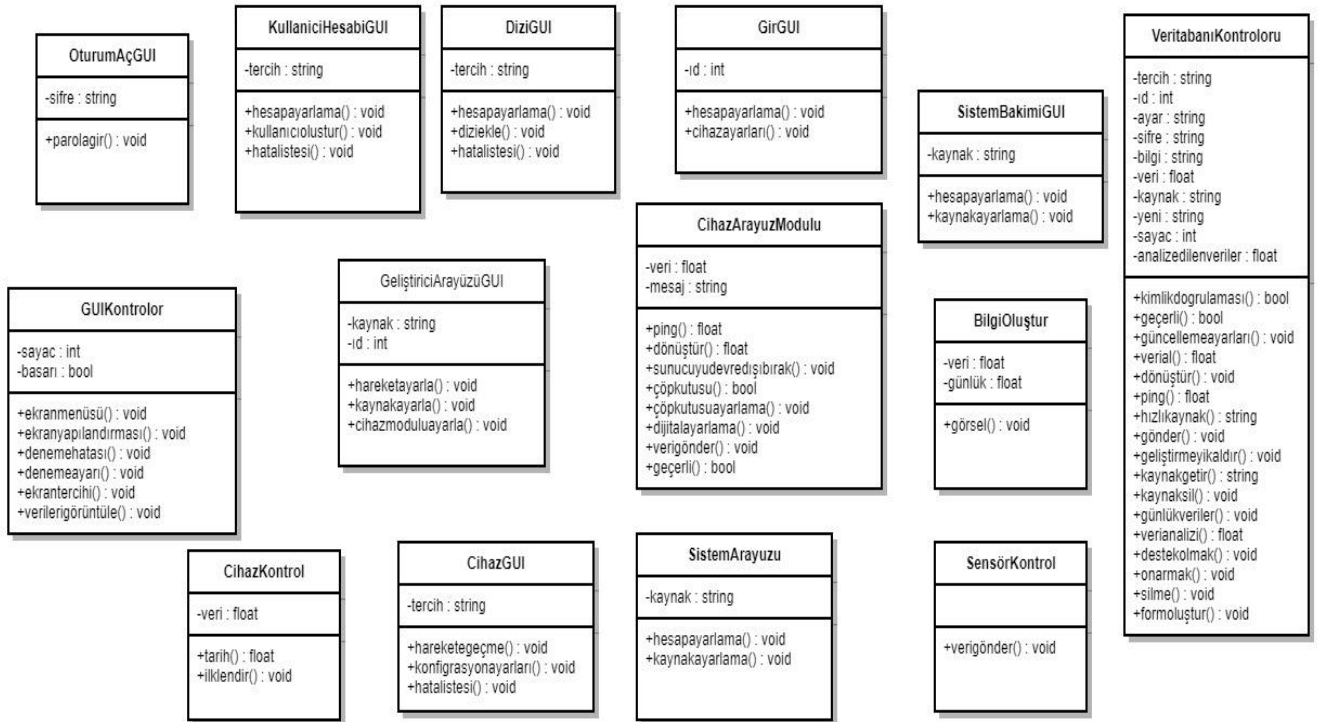
Method	Type
hesapayarlama	void
dizikle	void
hatalistesi	void

VeritabanıKontrolleri

Attribute	Type
tercih	string
id	int
ayar	string
sifre	string
bilgi	string
veri	float
kaynak	string
yeni	string
analizedilenveriler	float

Method	Type
kimlikdogrulaması	bool
gecerli	bool
güncellemeayarları	void
verial	float
donustur	void
ping	float
hızlıkaynak	string
gonder	void

gelistirmeyikaldır	void
kaynakgetir	string
kaynaksil	void
gunlukveriler	void
verianalizi	float
destekolmak	void
onarmak	void
silme	void



Her sınıf, yöntem ve nitelik aşağıda tanımlıdır.

Veritabanı İşlemleri

DiziGUI, Veritabanı Yöneticisinden üretilmiştir, çünkü cihaz dizisi kullanıcıya gösterilmektedir. Veri tabanı, sistemde hangi aygıtların bulunduğu hakkında bilgi içerdiğinden, aygıt dizisi Veri tabanına kaydedilir.

Veritabanıkontrolleri, veri tabanının kullandığı 10'dan fazla farklı işlevi olması ilkesine dayanarak Veritabanı Yöneticisinden üretilir. Veritabanıkontrolleri içindeki fonksiyonların sadece Database tarafından kullanılması çok önemlidir.

GUIkontrolleri, Veri Tabanı Yöneticisinden üretilmiştir çünkü Veri tabanı GUI'leri kullanıcıya iletmek zorundadır ve GUIkontrolleri GUI'leri ve bunların ne zaman dışarı itildiklerini düzenlemek için kullanılır. Bu önemlidir çünkü veritabanı GUI'nin eylemlerini düzenleyecek bir sınıfa sahip değilse sistem arızalanabilir.

Bilgi kaynağı veri tabanından türetilir bilgi bilgisi veri tabanından bilgi görüntüleme ilkesine dayanır.

GirGUI, Veri Tabanı Yöneticisinden üretilmiş şu anki durumuna dayalı kullanıcı için bir GUI şeklinde bir günlük görüntüleyecektir.

SistamarayüzüGUI, Veri Tabanı Yöneticisi'den sistemarayüzüGUI görüntülemesi gereken bir dizi farklı bilgiye sahip olmasına dayanarak Veri Tabanı Yöneticisinden üretilir

SistembakimiGUI, veri tabanından üretilir. Sistem bakımı, Veri tabanının kullanıcı tarafından erişimi izni olmaması gereken kısımlarına erişim izni vermesine dayanır.

Kontrol Yöneticisi

Sensörkontrolü, sistem üzerinden veri göndermesi nedeniyle Kontrol Yöneticisinden oluşur. Bu çok önemlidir, çünkü sistem tarafından veri alınacaktır ve kullanıcılar verileri geri gönderilmesini beklerler.

Cihazkontrol, sistemde veri alması sebebiyle Kontrol Yöneticisinden oluşur. Cihazkontrolü önemlidir. Çünkü sistem tarafından herhangi bir girdi almazsa düzgün şekilde çalışmaz.

Aygıt Yöneticisi

GeliştiriciarayüzüGUI, Aygıt Yöneticisinden üretilmiştir. Bu önemlidir, çünkü cihazın bilgileri neredeyse sürekli değişecektir ve GeliştiriciarayüzüGUI'nün cihazdaki en güncel durum raporlarına ihtiyacı vardır.

Cihazarayüzmodülü, GeliştiriciarayüzüGUI kullanıcılarını değiştirmek istediğinde ve tercihe göre ne girdiğine bağlı olarak veri gönderip alması nedeniyle Aygıt yöneticisinden üretilir.

CihazGUI, kullanıcının görmek ve değiştirmek istediklerini seçmesine olanak sağlayan GUI'dir. Aygıt yöneticisinden üretilir.

GUI Yöneticisi

DiziGUI, GUI Yöneticisinden üretilmiştir. Çünkü sistem tarafından kullanıcıya cihaz dizisini gösteren bir GUI üretmelidir.

GeliştiriciarayüzüGUI, GUI Yöneticisinden üretilmiştir, çünkü kullanıcıya tercihler sağlar.

GUIkontrolleri, GUI Yöneticisinden üretilmiştir, çünkü kullanıcıya bir GUI gönderildiğinde gerekli mi diye bakar ve düzenleyeceği bir sınıf olması gerekir. Bu önemlidir, çünkü bunsuz kullanıcının ekranında istenmeyen bir GUI oluşabilir ve bazı karışıklıklara yol açabilir.

GirGUI, GUI Yöneticisinden üretilmiştir, çünkü sistemin mevcut durumuna göre kullanıcıya bir GUI biçiminde bir günlük görüntü ve veriler gönderir.

OturumaçGUI, GUI Yöneticisinden üretilmiştir. Çünkü bir kullanıcı akıllı ev oturum açtığında, şifrelerini girmesi istenir.

SistमारayüzüGUI, GUI Yöneticisinden üretilmiştir. Çünkü genel sistemi göstermek ve sisteme ait yedekleme, geri yükleme, silme, güncellemeleri kontrol etme için sistem arayüzü gereklidir.

CihazGUI, kullanıcının görmek ve değiştirmek istediklerini seçmesine yardımcı olduğu için GUI Yöneticisinden üretilir.

SistembakimiGUI, GUI Yöneticisinden üretilmiştir. Çünkü sistem tarafından güncellemeleri kontrol etmesi veya güncelleniyorsa, kullanıcıdan sistemin belirli bir bölümünü güncellemesi istenmesi için gereklidir. Güncellemeler gerçekleştiğinde kullanıcıya sorulmazsa, arka planda bilinmeyen güncelleme nedeniyle arayüzleri yavaşlayabilir. KullanicihesabiGUI, GUI Yöneticisinden üretilmiştir. Çünkü kullanıcı oturum açtığında, başarılı oturum açtıktan sonra, bir ekranla yönlendirilir.

Kullanıcı Hesap Yöneticisi

OturumaçGUI, sistemin diğer kullanıcının sisteme oturum açmasına izin vermesi gerektiği için Kullanıcı Hesap Yöneticisinden üretilir. Kullanıcı Hesap Yöneticisi kullanıcı adı ve şifreyi doğruladıktan sonra ilgili hesabı alabilir.

KulanicihesabiGUI, sistemin belirli bir kullanıcı hesabını alabilmesi ve belirli hesabı görüntüleyebilmesi için Kullanıcı Hesap Yöneticisinden üretilmiştir. Bu önemlidir çünkü kullanıcı hesabındaki önemli bilgileri doğrulamak isteyebilir.

Şifre Yöneticisi

OturumaçGUI, bir kullanıcı kendi kullanıcı adı ve şifresini girdiğinde, Şifre yöneticisi kullanıcı adı ve şifresinin eşleşmesini sağlayacaktır.

c. İzlenebilirlik Matrisi

	Veri tabanı Yöneticisi	Kontrol Yöneticisi	Aygıt Yöneticisi	GUI Yöneticisi	Kullanıcı Hesap Yöneticisi	Doğrulayıcı
Geliştirici arayüzü GUI				X		
Cihaz kontrol		X				
Oturum aç GUI				X	X	X
GUI kontrolleri	X			X		
Kullanıcı hesabi GUI				X	X	
Sistem bakimi GUI	X			X		
Bilgi oluşturma	X					
Giriş GUI	X			X		
Cihaz GUI			X	X		
Veri tabanı kontrolleri	X					
Sensör kontrolü		X				
Cihaz arayüz modülü			X			
Sistem arayüz GUI	X			X		
Dizi GUI	X			X		

Burada her bir class'ın hangi yöneticide olduğu belirlendi. Örneğin Oturma aç GUI class'ı GUI yöneticisi, Kullanıcı hesap yöneticisi, Doğrulayıcı da olmalıdır. Çünkü bir kişi oturum açmak istediği zaman kullanıcı adı ve şifresi doğru ise oturum açabilir. Bunun için de bu yöneticilere bakılır.

2.2. Sistem Mimarisi ve Sistem Tasarımı

a. Mimari Tarzlar

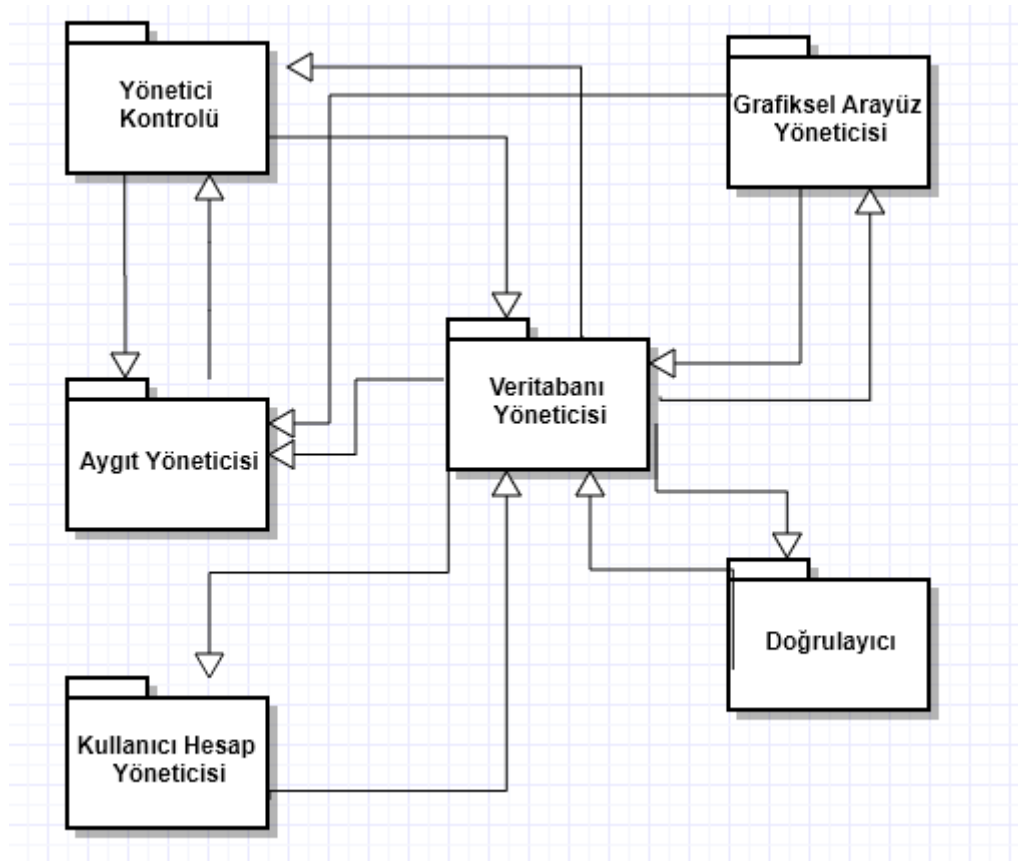
Yazılım mimarisi doğrudan sistemimizin kompozisyonunu ifade eder. Mimari, sistemin davranışını ve sistemimizin diğer görüşlerini gösteren kavramsal bir model veya modellerle gösterilebilir. Sistemimiz, aşağıdaki gibi tanımlanan bir dizi farklı yazılım mimarisi kullanır:

İstemci-sunucu modeli, İstemciyi, sunucudan ayıran bir ağ mimarisidir. Ağdaki tüm kaynaklar server'a bağlıdır ve ağ kaynaklarının tek sorumlusu server'dır. Eğer server isterse, bu sorumluluğu ağdaki diğer makinalara verebilir veya hepsini kendi kullanır. Kullanılan diğer makinalar ancak server üzerinden ağa bağlanabilir ve server'ın verdiği erişim izinleri dahilinde ağ kaynaklarından faydalanabilir. Biz projemizde mobil uygulama kullanmaktayız. Bu sayede server ile olan bağ mobil uygulama üzerinden sağlanmaktadır.

Akıllı Ev, bir tüketicinin mobil tabanlı bir arayüze erişmesine izin vererek tüm evlerine ve evin ilgili yardımcı programlarına / bileşenlerine erişmelerine izin verecek şekilde çalışır. Bu düzeyde, tüketici sisteme bluetooth bağlantısı olan bir telefonda erişebildiğinden, yazılım mimarisinin istemci-sunucu modelinin altına düşeceği anlaşıyor. Bununla birlikte, arayüzde Akıllı Ev'e bağlanacak iki farklı yetkiye sahip kullanıcı olacaktır. Bu durumlar, yalnızca birkaç tıklamayla tüketici tarafından herhangi bir zamanda değiştirilebilir. Farklı yetkiye sahip bu iki kullanıcı, ihtiyaçlarına göre minimum tıklama ile istediği aktiviteyi gerçekleştirecektir. Bu artık sistemimizi olay güdümlü bir yazılım mimarisinin altına alacak gibi görünecektir.

b. Alt Sistemlerin Tanımlanması

Tam paket etkileşim diyagramı



Yukarıdaki izlenebilirlik matrisinden, Akıllı Ev sisteminde 6 ayrı paketin mevcut olduğu görülebilir.

Bu alt sistemler aşağıdaki gibidir; Yönetici Kontrolü, Grafikselsel Arayüz Yöneticisi, Aygıt Yöneticisi, Veritabanı Yöneticisi, Kullanıcı Hesap Yöneticisi ve Kimlik Doğrulayıcı. Bu alt sistemler ve içerilen sınıfların bir sonucu olarak, sınıflarla eşleşen alt sistemler arasında bir ilişki görülebilir.

c. Alt Sistemleri Donanıma Eşleme

Akıllı Ev üzerinde fiziksel sunucuda ve evdeki aygıtlarla birlikte çalışmak üzere uyarlanmış yerleşik donanım bulunur.

Kullanıcılar evindeki sunucu, tüm cihazlara bağlamak ve herhangi bir ışık açık olup olmadığını veya garaj kapısını kapatmak için bir talimat gönderilip gönderilmediği gibi sistem durumunu ve özelliklerini kaydetmek için kullanılacaktır. Sunucu, veritabanında depolanan içeriği görüntülemek için bu verilerin tümünü bir veritabanına depolar. Kullanıcı ayrıca, farklı özelliklerle yetkilendirilmiş çocuk ve yetişkin olmak üzere iki farklı giriş seçeneğiyle karşılaşacaktır. Bu giriş seçeneklerini kullanacak kişiye göre yetkilerle donatılmış bir sistemdir.

Gömülü sistemler donanım tabanlı kod çalıştırmak için kullanılacaktır, kullanılan cihazlar ile sistem arasında veri iletecek ve bu da bir anlamda cihazlara konuşma becerisi verecektir. Bu sayede sistem kullanım kolaylığı hedeflenmiştir.


d. Kalıcı Veri Depolama

Uygulamamızda kalıcı depolama ihtiyacı, uygulamanın bir bütün olarak başarısı için çok önemlidir. Bizim durumumuzda, kullanıcı deneyiminin uygulamamızın önemli bir parçası olan kullanıcı bazında tutarlı olmasını sağlamak için kalıcı veri depolama alanı kullanılacaktır. Kullanıcının uygulamaya giriş yapabilmesi için veri tabanındaki kalıcılığın sağlanması gerekiyor. Aksi durumda sistem giriş yapmaya çalışan kullanıcıyı hatırlayamaz ve giriş sağlanamaz.


Çok sayıda kalıcı veri depolama sistemi mevcuttur. Uygulamamız için en yararlı olacak sistem yaygın olarak kullanılan ilişkisel veri tabanıdır. Bu sayede tüm kullanıcılar esas olarak kendileriyle ilişkili veri türlerine sahip olacak ve daha esnek tasarımlar oluşturmak için ilişkisel veri tabanlarının sınırlamaları etrafında çalışmak kolaylık sağlar.

cihaz			
	Column Name	Data Type	Allow Nulls
🔑	id	int	<input type="checkbox"/>
	c_adi	nvarchar(50)	<input checked="" type="checkbox"/>
	aktif	tinyint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“cihaz” Veri Tabanı Tablosu

cihaz_izni			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	k_id	int	<input checked="" type="checkbox"/>
	cihaz_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“cihaz_izni” Veri Tabanı Tablosu

eylem_iliski			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	eylem_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“eylem_iliski” Veri Tabanı Tablosu

eylem			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Tarih	datetime	<input checked="" type="checkbox"/>
	cihaz_id	int	<input checked="" type="checkbox"/>
	k_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“eylem” Veri Tabanı Tablosu

kosul			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Tarih	datetime	<input checked="" type="checkbox"/>
	cihaz_id	int	<input checked="" type="checkbox"/>
	k_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“kosul” Veri Tabanı Tablosu

kosul_kumesi			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Tarih	datetime	<input checked="" type="checkbox"/>
	Aciklama	nvarchar(MAX)	<input checked="" type="checkbox"/>
	k_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“kosul_kumesi” Veri Tabanı Tablosu

kosul_iliski			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	kosul_id	int	<input checked="" type="checkbox"/>
	kosul_kumesi_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>


“kosul_iliski” Veri Tabanı Tablosu

kullanici			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	k_adi	nvarchar(50)	<input checked="" type="checkbox"/>
	Tarih	datetime	<input checked="" type="checkbox"/>
	Son_Giris	datetime	<input checked="" type="checkbox"/>
	Yonetici	tinyint	<input checked="" type="checkbox"/>
	k_sifre	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

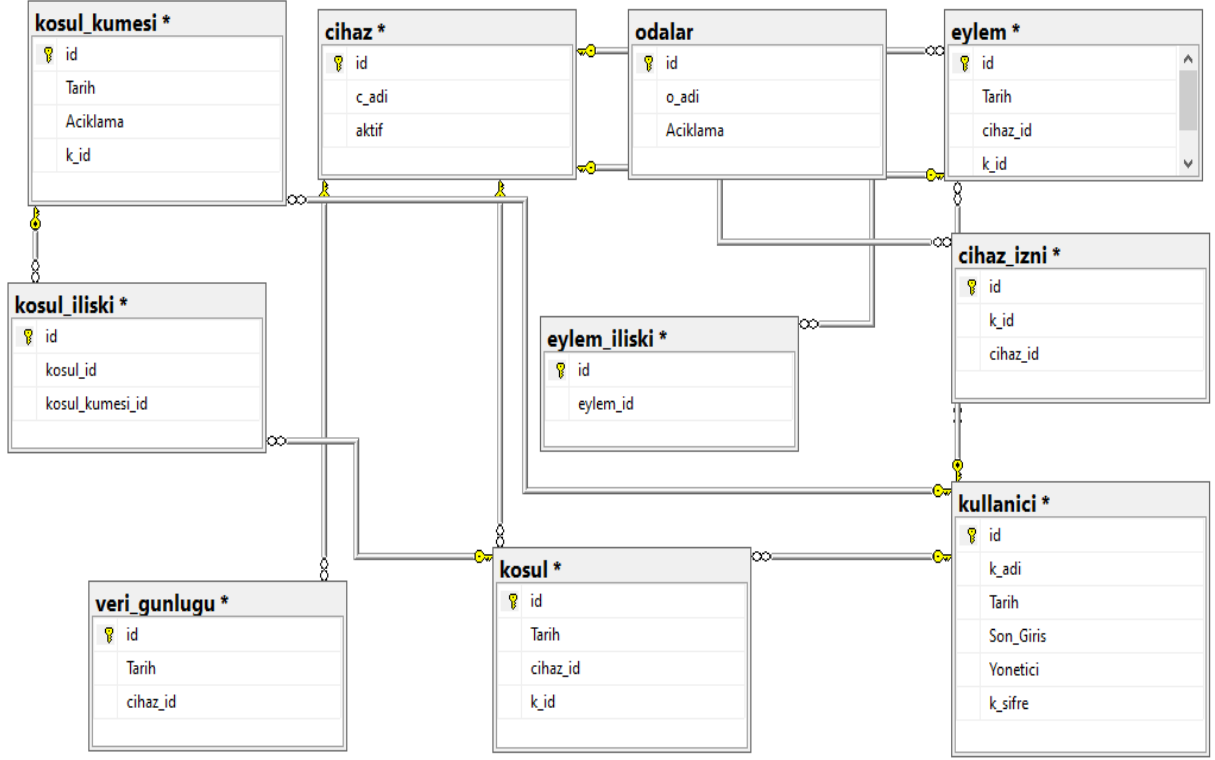
“kullanici” Veri Tabanı Tablosu

odalar			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	o_adi	nvarchar(50)	<input checked="" type="checkbox"/>
	Aciklama	nvarchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

“odalar” Veri Tabanı Tablosu

veri_gunlugu			
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	Tarih	datetime	<input checked="" type="checkbox"/>
	cihaz_id	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

“veri_gunlugu” Veri Tabanı Tablosu



Tablolar Arası İlişki Diyagramı

e. Ağ Protokolü

Tüm sistemi birden fazla cihazın onunla etkileşime girebileceği şekilde bağlamak için bir sunucu istemci ağ sistemi uygulanacaktır. Sunucu istemci ağ sistemi için http kullanılacaktır.

HTTP protokolü server-client(sunucu-istemci) yapısında çalışır. İstemcilerin sayfaları sunuculardan nasıl isteyeceğini ve sunucuların bu sayfaları istemcilere nasıl aktaracağını HTTP tanımlar. Bu yapıda istemciler genellikle bilgisayarlarımızda çalıştırdığımız tarayıcılardır, sunucular ise uzak bilgisayarlarda sürekli olarak açık tutulan programlardır. Bu durumda akıllı ev projemizde istemci mobil uygulama, sunucu ise Arduino Ana Ünitesidir.

Bir kullanıcı mobil uygulamada oturum açtığı anda, istemci olarak sunucuya (Raspberry Pi) ping atıp onunla bağlantı kurmaya çalışır. Çoklu kullanım özelliğini kullanarak, birden çok kullanıcı mobil uygulamasını ve sıra bazında işlenecek tüm gönderme isteklerini kullanarak yukarıda belirtilen sunucuya aynı anda bağlanabilir. Bir cihaz bağlandıktan sonra, kullanıcı Raspberry Pi'ye HTTP istekleri gönderebilir ve Arduino Ana Ünitesine donanım bileşenlerini açmasını veya kapatmasını söyleyebilir. Kullanıcı, örneğin mobil uygulamada ışıkları değiştirdiğinde, istek istemciden sunucuya gönderilir. Raspberry Pi sunucusu daha sonra açma / kapama değerlerini düzenlemek için verileri Arduino Ana Ünitesine gönderir.

f. Global Kontrol Akışı

Programımız olaya bağımlıdır, çünkü onu tetikleyen kullanıcıya bağlı olarak belirli olaylar meydana gelir. Örneğin, ışık yalnızca kullanıcı tarafından açılırsa ve kapatılmadığı sürece açık kalacaktır. Ayrıca, olayların gerçekleşme sırası değişkendir, çünkü bunların tümü kullanıcının başlattığı eyleme bağlıdır. Döngülerimiz ve beklemlerimiz olsa da, yürütülme sırasına göre farklı eylemler üretebilirler.

Zamana bağlı olarak, sistemimizde birkaç zamanlayıcı vardır. Alarm tetiklendiğinde, kullanıcının mobil uygulama aracılığıyla alarmı devre dışı bırakması için 30 saniye olacaktır. Aynı zamanda sistemimiz bir olay müdahale sistemidir. Örneğin, toprak nemi azalırsa sistem tarafından sulama yapılır.

İşletim sisteminin eş zamanlı çalıştırdığı iş parçacıkları prosesler olarak adlandırılır. Prosesler de kendi içlerinde eş zamanlı çalışan iş parçacıklarına sahip olabilir ve bunlara da Thread denir.

Farklı prosesler arasında ya da aynı proses içinde yer alan Thread'ler arasında zaman zaman senkronizasyon ihtiyacı duyulur çünkü bu Thread'ler görevlerini yerine getirebilmek için işletim sistemi tarafından sağlanan veya prosesin kendisinin tuttuğu paylaşılan bir kaynağa erişmek isterler. Örnek olarak Thread'lerin bir log dosyasına veya ekrana log yazmak istedikleri durumu ele alalım. İki Thread aynı anda log dosyasına yazmaya çalışırsa dosyaya yazılan loglar birbirine karışacak ve okunmaz hale gelecektir. Burada Thread'leri log dosyasına yazma konusunda sıraya sokacak bir mekanizmaya ihtiyaç vardır.

Mutex (MUTual EXclusion)'ler tam da bu mekanizmayı sağlamak için tasarlanmıştır. Mutex'ler uygulamanın yazıldığı dil ve Runtime tarafından sağlanan basit yapılardır. Farklı Thread'ler tarafından paylaşılan her bir kaynak için kaynağa olan erişimi düzenlemek üzere bir Mutex yaratılır. Paylaşılan kaynağa erişim yapılan kod bölgesi critical section olarak adlandırılır. Kaynakla işi olan Thread, Mutex'in sahipliğini almaya (Acquire) çalışır. Mutex o anda başka bir Thread tarafından tutulmuyorsa Thread Mutex'i alır, critical section'a girerek ilgili kaynağı kullanır. Diğer durumda, yani Mutex o anda başka bir Thread tarafından kullanılıyor ise, ikinci Thread işlemci tarafından beklemeye alınır. Mutex'i tutan Thread critical section'ı bitirip Mutex'i bırakırken, halihazırda Mutex'in bırakılmasını bekleyen Thread uyandırılır ve Mutex'in sahipliğini alarak critical section'a girer ve paylaşılan kaynağa erişim sağlar.

Eşzamanlılık için, farklı kullanıcıları ana birime bağlandıklarında başlattığımızda birden çok iş parçacığı kullanmayı planlıyoruz. Bu, birden fazla kullanıcının aynı anda sistemi değiştirmesini ve sorunlara neden olmasını önler. Mutex kilitlerini kullanarak bu sorunun önüne geçmeyi planlıyoruz, böylece belirli cihazları kontrol eden değişkenler aynı anda değiştirilemez.

g. Donanım Gereksinimleri

Sunucu donanımımızın sadece SQL Server, Arduino bulunmaktadır. Mobil uygulama ve arduino HC05 arduino bluetooth sensörü ile iletişim kurmaktadır. Mobil uygulama arduino'ya bağlı sensörlerimizi ve olayları kontrol etmemizi sağlar. Veriler ile ilgili bilgiler veritabanından SQL de bilgiler mevcuttur.

Akıllı ev sistemimiz için tasarladığımız mobil uygulamada android işletim sistemini kullanabilen cihazlar üzerinden erişim sağlanabilmektedir. Bu cihazlar ile uygulamanın arayüzüne erişebilmek mümkündür. Bu arayüz erişimi sayesinde cihazlar ile bağlantı kurup gerekli işlemleri gerçekleştirmek mümkün olacaktır.

Android yazılımı için gereken donanım özellikleri:

Ekran çözünürlüğünde herhangi bir kısıtlama yoktur ve minimum düzeyde kurulum alanı(<15 MB) ile erişim sağlanabilmektedir. Android sürümleri üzerinde uyumluluk problemi yaşanmaksızın kullanılmaktadır.

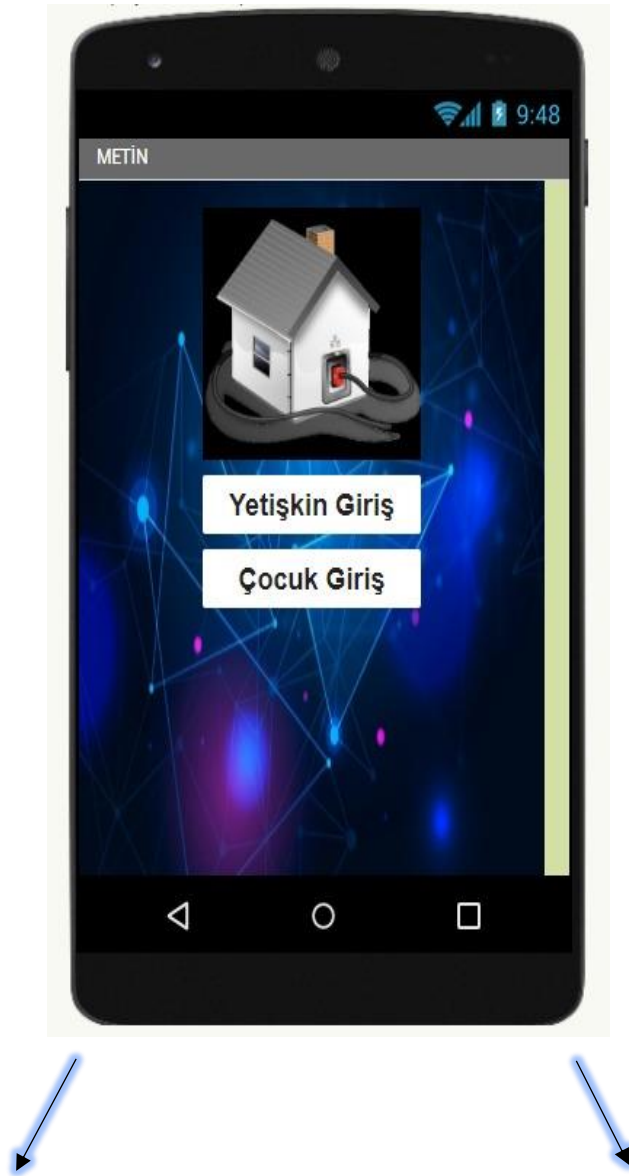
Bölüm3

3.1. Kullanıcı Arayüz Tasarımı ve Uygulama

a. Tasarım ve Uygulamalar

Aşağıdaki sayfalarda arayüz tasarımını kısa açıklamalarla görebilirsiniz.

Kullanıcıya, ilk olarak “Yetişkin Giriş” ve “Çocuk Giriş” diye iki seçenek sunulmaktadır. Kullanıcı, sahip olduğu kullanıcı kimliğine göre seçim yapar.





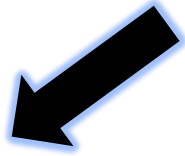
Kullanıcı adı ve parolası ile başarılı bir şekilde oturum açıldıktan sonra kullanıcıyı mümkün olduğunca kullanıcı dostu bir arayüz karşılar.

Yetişkin girişinden sonra ana sayfada uygulamanın sunduğu tüm işlemleri yapabilecektir. Evdeki tüm sensör sistemleri kontrol edilebilecektir.

Çocuk girişinden sonra ise ana sayfada yetişkin sayfasına göre daha kısıtlı işlem yer alır.

Burada amaç çocuğun güvenliğini sağlamaktır.

Yetişkin Giriş Sayfası:



Işıkları kontrol edebilir. Butonlara basıp ışık açıp-kapatma işlemlerini yapılabilir.

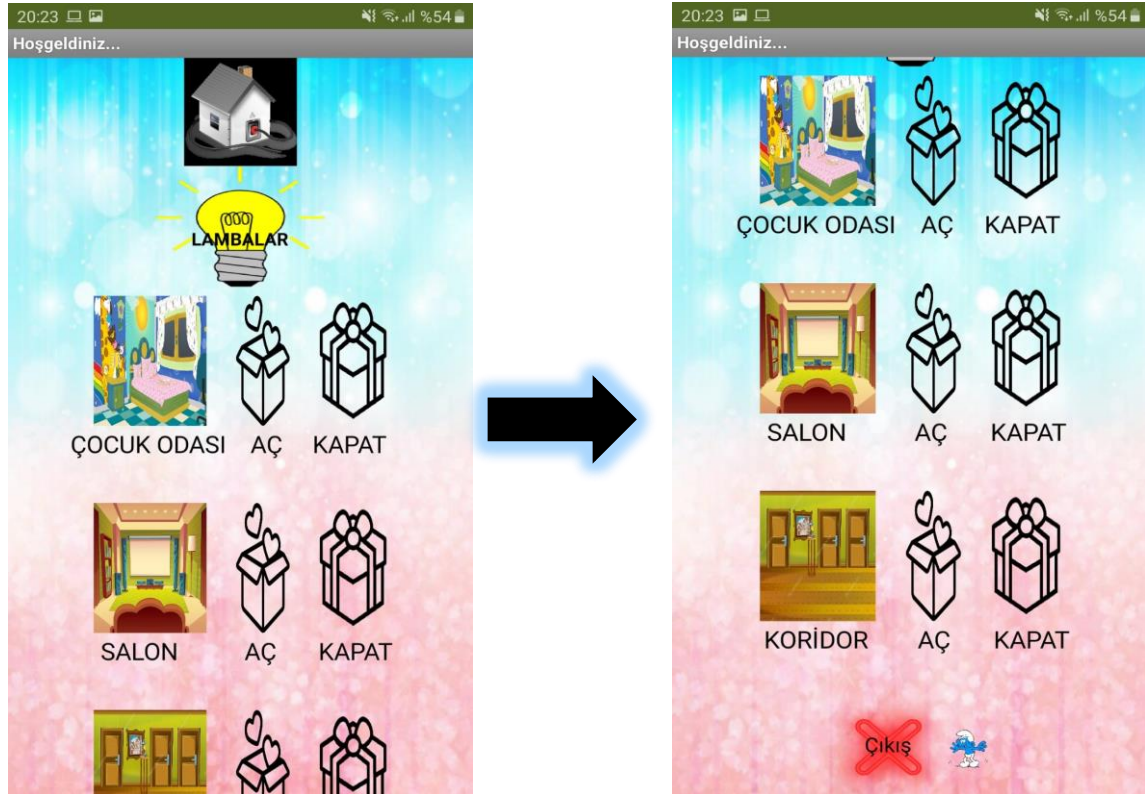
(Kontrol edilen yerler: Salon, çocuk odası, yatak odası, mutfak, koridor ve bahçe ışıklarıdır.)

- Elektronik cihazlar ile kapı açma-kapatma ve perde açma-kapatma işlemleri gerçekleştirilir.
- Kamera sistemi ile güvenlik kontrol altına alınır.
- Evin sıcaklık değeri görülebilir.
- Uygulama içinde gezinti yukarıdan aşağıya doğrudur.

Bunun dışında,

- Yetkisiz biri tarafından kapı açılması vb. durumlarda alarm sistemi devreye girer ve alarm çalmaya başlar.
- Bahçedeki bitkilerin suyu bittiği an otomatik olarak nem sensörü sayesinde kendiliğinden sulama işlemini gerçekleştirir.
- Garaja yerleştirilen sensör sayesinde park işlemleri çok daha kolay hale gelmektedir.
- Evde yabancı gaz olma durumunda kullanıcıya bildirme işlemi yapılır.

Çocuk Giriş Sayfası:

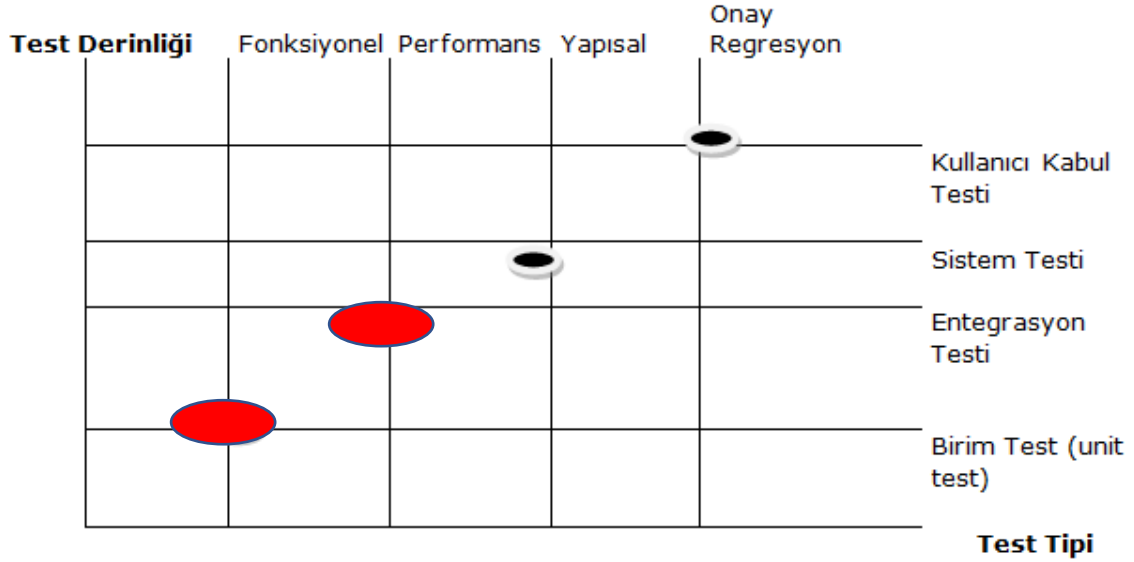


Çocuk giriş ekranında daha önce de belirtildiği gibi çocuğun güvenliği açısından işlem kısıtı yer almaktadır. Bu durumda çocuk kullanıcı sadece lambaları kontrol edebilmektedir.

Bu sayede yetişkin bireyler evde yokken izinsiz dışarı çıkma, uzak durması gereken tehlikeli durumlardan korunma ve eve gelen kimliği belirsiz kişileri eve alamama sonuçlarını ortaya çıkarır. Yetişkin bireyler evde bulunmadıkları halde içleri rahat bir şekilde işlerine odaklanabileceklerdir.

3.2. Test Tasarım

Uygulanması gereken test safhaları:



(Uygulanan testler kırmızı ile ifade edilmiştir.)

(Test tasarımıda kullanacak “Kullanım Örnekleri “aşağıda listelenmektedir)

Günlük Kullanımlar:

Kullanım Örneği 1. Kullanıcı Kimlik Doğrulaması:

Bir kullanıcı, sistem tercihlerini değiştirmek, yapılandırmak için sisteme giriş yapmak ister. Şifreler ve kullanıcı adları, doğru hesaba giriş yapılmasına izin verir.

Kullanım Örneği 2. Cihaz Günlüğü Verilerini Dışa Aktar

Bir kullanıcı, kullanıcının erişebildiği bir aygıtla ilgili bilgileri görüntüler. Cihaz adı, cihaz kimliği ve cihazın verileri hakkındaki veri günlüğü getirir.

Kullanım Örneği 6. Cihaz Ekle

Kullanıcı isteği üzerine deneme kullanımlarında geliştirici tarafından arayüze cihaz dizi modül ekleme yapılabilir.

Kullanım Örneği 7. Cihaz Sinyali Gönderme

Doğru gerilime dönüştürülmek üzere bir aygıt sürücüsüne bir değer gönderilir ve daha sonra kontrol iletkenleri yoluyla gönderilir cihaz değerlerini ayarlamak için Cihaz ayarı olmadığında, arabirim modülleri tarafından denetleyicileri tutmak için varsayılan değerler sağlanır. Testler, yazılım ayarlı değerlerin değerleri doğru bir şekilde yansıtmasını sağlar.

Kullanım Örneği 8. Cihaz Sinyali Alma

Sistem, bağlı cihazlardan değerler alır. Cihaz voltajları anlamlı değerlere dönüştürür ve insanlar tarafından kullanılabilen geçerli bir değere dönüştürür. Testler, yazılımdaki değerler sensörlerin değerlerini yansıtırlar.

Birim Testi

1.Kullanıcı Kimlik Doğrulaması:

Başarı: Kullanıcı, kullanıcı adını ve şifreyi başarıyla girer. Sistem, giriş başarılı olur ve kullanıcı günlüğe kaydedilir

Test Durumları	Beklenen Sonuç
<p>-Kullanıcı, kullanıcı adını ve şifreyi girer. Sistem doğrulamaya gittiğinde kullanıcı adı yanlıştır ve veri tabanındaki hiçbir kullanıcı adıyla eşleşmez</p> <p>-Kullanıcı, kullanıcı adını ve şifreyi girer, ancak sistem doğrulamaya gittiğinde şifre yanlıştır ve veri tabanındaki hiçbir şifre ile eşleşmez</p> <p>-Kullanıcı, kullanıcı adı ve şifre girer, ancak sistem doğrulama yapmaya gittiğinde, kullanıcı adı da parolada veri tabanına kaydedilen parola ile uyuşmuyorsa eşleşmez.</p>	<p>ve " Kullanıcı Adı veya Şifre Hatalıdır" hata mesajını alır.</p>

2. Cihaz Günlüğü Verilerini Dışa Aktar

Başarı: Kimliği doğrulanan kullanıcı hesabında başarıyla oturum açar. Akıllı Ev uygulama menüsüne kolaylıkla gidebilir. Veri tabanındaki olası bir arızada bellek saklanan verileri tutacaktır.

Test Durumları	Beklenen Sonuç
<ul style="list-style-type: none">-Başarı ile oturum açılır fakat arayüz farklı Akıllı Ev arayüzü değildir.-Kimliği doğrulanan kullanıcı hesabında başarıyla oturum açar sensör değerleri görünmüyor.-Kimliği doğrulanan kullanıcı hesabında başarıyla oturum açar arayüz var fakat herhangi veri bilgisi yok.-Kimliği doğrulanan kullanıcı hesabında başarıyla oturum açar arayüzde düzgündür fakat göstermesi gereken veriyi göstermiyor.	Veri tabanından doğru bilgilerin çekilmesi

6.Cihaz ekle

Başarı: Yeni bir sensör eklenirse testi geçer ve uygulamanın kontrol panelinde görünür

Test Durumları	Beklenen Sonuç
<ul style="list-style-type: none">-Hesaba yeni bir sensör ekleyin-Sensörün durumları ve kullanım amacı kontrol ederek sensör eklenir.	Sistem yeni sensörü hesaba kaydeder. Sensör başarılı bir şekilde işlev gösterir.

7. Cihaz Sinyali Gönderme

Başarı: Sisteme giriş yapıldıktan sonra sistem o anda bağlı olan cihaza değerleri belirtir. Daha sonra cihaz arayüz modülleri değerleri anlamlı voltajlara dönüştürür. Bir sorun olduğunda varsayılan değer kullanılır.

Test Durumları	Beklenen Sonuç
-Arabirim modülleri değerleri anlamlı gerilimlere dönüştürmez veya değerleri belirtilen değeri için gereken voltajdan daha yüksek veya daha düşük bir değere dönüştürün.	Varsayılan değerlerin aktarılmasıdır.

8. Cihaz Sinyali Alma

Başarı: Bir sensör üzerinden sinyal gönderirse test geçer veri tabanı kayıtlarını alır.

Test Durumları	Beklenen Sonuç
-Herhangi biri için bir tetikleyici bir olay oluşturulur.	Sensör sinyali veri tabanına gönderir
-Veri tabanının kontrol edilir.	Veri tabanı kaydeder.

Entegrasyon Testi:

(Uygulamanın diğ er mod llerle, farklı iřletim sistemleri ve uygulamalarla  alıřabilirliđinin test edilmesidir.)

Entegrasyon testi, bađımsız birimlerin birleřtirildiđi ve test edildiđi bir yazılım testi seviyesidir. Bu test seviyesinin amacı, arasındaki etkileřimdeki hataları ortaya koymaktır. Sistemimiz i in test edilmesi gereken 3 temel entegrasyon vardır. Birincisi t m sens rler, veri tabanı ve uygulama arasındaki entegrasyon.

Kullanıcının acil durumlar i in kusursuz ve hızlı olacak řekilde ayarlanmalıdır. Sistemimizi kullanıcı kullanmadan  nce her sens r  ayrı ayrı titizlikle test etmeliyiz ve veri tabanı ile aralarındaki etkileřimin sorunsuz olduđuna bakılmalı uygulama ve veri tabanı arasındaki etkileřim ve verilerin d zg n okunup okunmadıđına bakılmalıdır.

Acil durumlar i in belirli sabit deđerler verilmelidir t m sistemin teknik b l m n n nasıl olması gerektiđi belirlenmelidir. Farklı sens r sinyali t rlerinin arasındaki entegrasyon d zenlemelidir.

Bir kullanıcı ara y z nden bařka kullanıcı ara y z ne ge iřin yanı sıra kullanıcı ara y z  ile kullanıcı aray z  arasındaki etkileřim sens rler, kullanıcının sorunuzuz kullanabilmesi i in m mk n olduđunca fazla gecikme olmamalıdır buna dikkat edilmelidir.

3.3. Proje Yönetimi ve Çalışma Planı

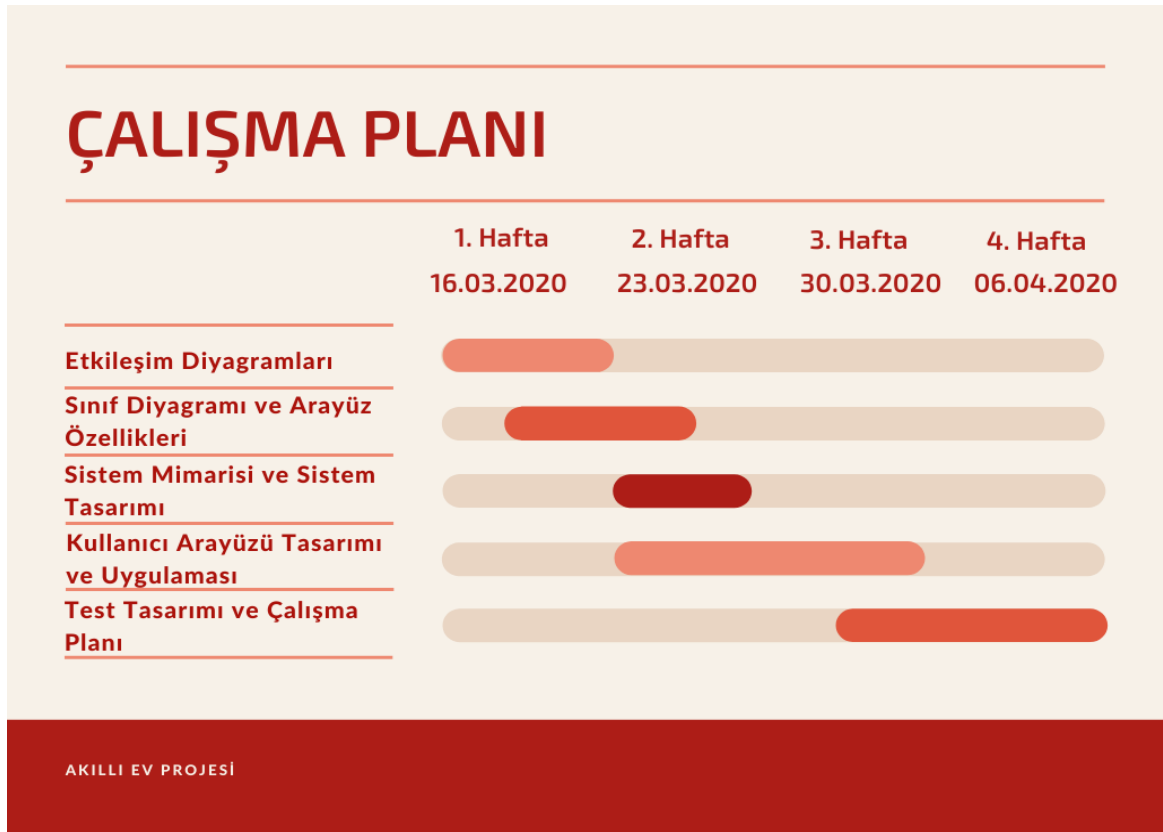
a. Ekip Üyelerinin Bireysel Katkılarını Birleştirme

Bütün ekip arkadaşlarımızın katkılarını birleştirmek görüldüğü kadar kolay değildir. Bunun nedeni, okulumuza ara verilerek uzak yerlerde oturmamızdır. Aynı zamanda herkesin projede bir bölümü yapması için farklı dosya, çizim ve kod yazılmasıdır. Arduino'dan App Inventor'a, hatta Microsoft Word'den Not defterine kadar her şey. Bu katkıları tek bir belgede birleştirmek için Microsoft Word kullanıldı. Oradan, WhatsApp Web uygulamasından katkılar paylaşıldı ve tek bir Microsoft Word belgesine yapıştırıldı. Yazı tipi boyutu ayarlandı. Karşılaştığımız bir diğer sorun ise, uzaktan Arduino'yu nasıl kodlayıp çalıştıracağımızdı. Bu kısımda ise çeşitli alternatifler geliştirmeye çalıştık ve bir çözüme ulaştırdık.

b. Proje Koordinasyon ve İlerleme Raporu

Şu anda mobil arayüz tasarlanıp veritabanı oluşturulmaktadır. Aynı şekilde Arduino sensörlerinin kodlama aşaması çalışmalarınada başlandı. Mobil kısım oluşturulduktan sonra, veritabanı Arduino'ya bağlanacaktır. Böyle bir durumda mobilde yapılan işlemler Arduino ya yansıtılacak ve oradan veri tabanına kaydı sağlanacak. İlk demo için gereken her şey etkili bir şekilde yapılacaktır. Mobilde sensörlere müdahale edilecek ve cihazlara veri gönderilebilecektir. Aynı zamanda Google Hangouts Meet üzerinden toplantılar yapıp projemizde nasıl ilerleyeceğimiz hakkında fikir alış verişinde bulunmaktadır. Alternatifler üretip o şekilde ilerleme sağlanacaktır.

c. Çalışma Planı



Planlama	Tarih	Görevler/Katılan kişiler
Etkileşim Diyagramları	09/03/2020	Etkileşim diyagramları oluşturuldu. Ve gliffty uygulamasından diyagramlar tek tek çizildi.
Sınıf Diyagramı ve Arayüz Özelliği	16/03/2020	Class sınıf diyagramları tespit edilip, gliffty uygulamasından çizildi.
Sistem Mimarisi ve Sistem Tasarımı	23/03/2020	Bu konuya dair gerekli araştırmalar yapıldı ve ne yapılacağına dair tespitler yapıldı. Projemize dair en ince ayrıntıya kadar düşünülüp kod kısmı hazırlanmaya başlandı ve bunu tablolarla ifade ettik.
Kullanıcı Arayüzü Tasarımı ve Uygulaması	06/04/2020	Kullanıcı arayüzü geliştirildi. Daha estetik ve kullanışlı yapılmaya başladık. Ara ara da testler yapıldı.
Test Tasarımı	30/03/2020	Projemizi yaparken ara ara test edip kontroller yaptık. Bu konu daha derinlemesine araştırılıp projemizin daha etkili test yapılması sağlanmaya başlandı.
Proje Yönetimi ve Planı işin	06/04/2020	Projemizi nasıl hazırladığımızı, planlamalarımızı nasıl yaptığımızı ve araştırmalarımızın nasıl yapıldığı hakkında planlar yapıp Google Hangouts Meet uygulamasından toplantılar yaparak planlarımızı oluşturduk.

d. Sorumlulukların Dağılımı

4 grup üyemiz var, bu yüzden her birimizi eşit katkıda bulmayı hedefledik ve bu doğrultuda hareket etmekteyiz.

Şimdiki durumumuz:

Esra YÜCE, Halil SEÇİLMİŞ Mit App Inventor kullanarak uygulama geliştirmeye başladı.

Özlem ÖZKAYA, Feyza YILMAZ Microsoft Sql Server kullanarak veritabanı işlemlerine başladı.

Tüm grup üyeleri olarak arduino işlemlerinde kullanılacak sensörler ile detaylı araştırmalar devam etmekte.

GitHub, bu projenin tüm yönlerini birbirine bağlamak için kullanıyoruz.

Sürekli olarak birbirimizle iletişim halinde bulunmaktayız.

3.4. Kaynakça

Etkileşim Diyagramları:

[1] UML ile "Sequence" Diyagramları, *Web blog*

<http://www.csharpnedir.com/articles/read/?id=402>

[2] UML Şeması Anlatım, *Web blog*

<https://www.smartdraw.com/uml-diagram/>

Sınıf Diyagramı ve Arayüz Özellikleri:

[3] Sınıf Diyagramları Nedir? Nasıl Çizilir?

<http://muhammetbaykara.com/wp-content/uploads/2017/03/S%C4%B1n%C4%B1f-Diyagramlar%C4%B1.pdf>

[4] UML Modelleme ve Class Diyagramları

<http://univera-ng.blogspot.com/2009/12/uml-ve-modelleme-bolum-4-class-snf.html>

[5] UML Diyagramlar ve İlişkiler

<https://www.youtube.com/watch?v=eI5RA45OQG8>

[6] Diyagramları İlişkilendirme

<https://sites.google.com/site/modernyazilimgelistirme/uml/uml-diyagramlari/sinif-diyagramlari>

[7] UML Class Diyagramları

<https://www.youtube.com/watch?v=UI6lqHOVHic>

[8] UML Temelleri

<http://www.csharpnedir.com/articles/read/?id=5>

[9] UML Class Diyagramlar Nasıl Kullanılır?

<https://www.youtube.com/watch?v=rszY9x3TvU8>

[10] Class Diyagramlar nedir?

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>

Sistem Mimarisi ve Sistem Tasarım:

[11] İlişkisel Veritabanı Yönetim Sistemleri

<https://kod5.org/iliskisel-veritabani-yonetim-sistemleri-rdbms/>

[12] İlişkisel Veritabanı Nedir?

<https://www.oracle.com/tr/database/what-is-a-relational-database/>

[13] Mutex ve Semaphore

<https://yazilimdnyasi.wordpress.com/2019/02/25/mutex-vs-semaphore/>

Test Tasarım:

[14]Yazılım Testi ve Test Süreçleri Nelerdir? *Web blog*

<https://bidb.itu.edu.tr/sevir-defteri/blog/2013/09/08/yaz%C4%B1%C4%B1m-testi-ve-test-s%C3%BCre%C3%A7leri>

[15]Yazılım Tasarımı Nedir? *Web blog*

<https://medium.com/t%C3%BCrkiye/yaz%C4%B1%C4%B1m-tasar%C4%B1m%C4%B1-nedir-cd8aad12c8ae>

[16] Test Derinlikleri, Birim testi Entegrasyon Testi Nedir? *Web bolg*

<https://classroom.google.com/u/5/c/MzA3MTQ5NjkxOTBa/m/NzQxNDU1NTY5OTNa/details>