



Akıllı Armatür Tasarımı ve Gerçekleştirme

Yazılım Mühendisliği Ana Bilim Dalı

Dönem Projesi

Eray ADIYOKLAR

ORCID 0009-0007-4769-8986

Proje Danışmanı: Prof. Dr. Doğan AYDIN

Ocak 2024

Akıllı Armatür Tasarımı ve Gerçekleştirme

Öz

Bu proje, fotoselli armatürlerin akıllı ev sistemleri ve IoT teknolojisiyle entegre edilerek daha etkin bir yönetim sistemine dönüştürülmesini amaçlamaktadır. Fotoselli armatürler, çeşitli çevresel koşulları algılayabilen sensörlerle donatılmış, otomatik açılıp kapanabilen su akış üniteleridir. Bu armatürleri merkezi bir kontrol birimi üzerinden yöneterek su tüketim miktarı, arıza tespiti, bakım ihtiyacı gibi kritik bilgileri anlık olarak takip edip mobil uygulama aracılığıyla kullanıcılara iletmeyi hedeflemektedir.

Akıllı ev sistemleri, WiFi haberleşme ve merkezi kontrol birimi kullanılarak bu armatürlerin proaktif bakımını sağlamayı, enerji verimliliğini artırmayı ve kullanıcı memnuniyetini yükseltmeyi amaçlamaktadır. Bu ölçeklenebilir yapı, farklı kullanım alanlarına ve kurumlara uyum sağlayarak geniş bir kullanıcı kitlesine hitap etmeyi amaçlamaktadır. Siri gibi sesli komut özelliği ile kullanıcıların kolay bir şekilde kontrol sağlaması ve Firebase altyapısıyla verilerin güvenli bir şekilde depolanması da projenin avantajları arasında yer almaktadır. Bu inovatif proje, toplu kullanım alanlarında fotoselli armatürlerin yönetimini optimize ederek su israfını önlemeyi, enerji verimliliğini artırmayı ve çevre dostu bir çözüm sunmayı hedeflemektedir.

Anahtar Sözcükler: Fotoselli Armatür, Akıllı Ev Sistemi, IoT Teknolojisi, Proaktif Bakım, Su Tüketim Miktarı, Mobil Uygulama, Merkezi Kontrol Birimi

Smart Armature Design and Realization

Abstract

This project aims to transform photocell luminaires into a more efficient management system by integrating them with smart home systems and IoT technology. Photocell faucets are automatic on and off water flow units equipped with sensors that can detect various environmental conditions. By managing these faucets through a central control unit, it aims to instantly monitor critical information such as water consumption amount, fault detection, and maintenance needs and communicate them to users via mobile application.

Smart home systems aim to provide proactive maintenance of these luminaires, increase energy efficiency and increase user satisfaction by using WiFi communication and a central control unit. This scalable structure aims to appeal to a wide range of users by adapting to different areas of use and institutions. Among the advantages of the project are the easy control of users with voice command features such as Siri and the secure storage of data with Firebase infrastructure. This innovative project aims to prevent water waste, increase energy efficiency and provide an environmentally friendly solution by optimizing the management of photocell luminaires in public areas.

Keywords: Photocell Luminaire, Smart Home System, IoT Technology, Proactive Maintenance, Water Consumption Amount, Mobile Application, Central Control Unit.

Teşekkür

Bu çalışmanın yürütülmesi sırasında, proje konusunun belirlenmesinden son aşamaya kadar her konuda desteğini ve yardımlarını esirgemeyen danışman hocam Sayın Prof. Dr. Dođan AYDIN'a ve İKÇÜ akademi kadrosuna şükranlarımı sunarım.

Bu araştırma ve geliştirme boyunca gerek maddi ve manevi anlamda yanımda olan ailem, dostlarım ve iş hayatımdaki çalışma arkadaşlarıma teşekkür ederim.

İçindekiler

Öz	i
Abstract	ii
Teşekkür	iii
Şekiller Listesi.....	vii
Tablolar Listesi.....	ix
Kısaltmalar Listesi	x
1 Giriş	11
1.1 Araştırma Amacı.....	12
1.2 Araştırma Sonucu.....	13
1.3 Araştırma Avantajları.....	14
1.4 Terimler Sözlüğü	15
2 Yöntemler.....	18
2.1 Sensör Teknolojisi ve Elektronik Panel Tasarımı.....	18
2.1.1 Dokunmatik Sensör Entegrasyonu	19
2.1.2 Elektronik Panel Tasarımı	19
2.1.3 Su Sıcaklığı ve Debi Ölçümü	19
2.1.4 Akış Yönü Kontrolü	20
2.2 IoT Entegrasyonu ve Cloud Servis Kullanımı	20
2.2.1 WiFi Haberleşme Altyapısı	20
2.2.2 IoT Cloud Servis Entegrasyonu.....	21
2.2.2.1 IFTTT Entegrasyonu	21
2.2.2.2 Google Firebase Entegrasyonu.....	22

2.2.3	Veri Analizi ve Raporlama	23
2.3	Mobil Uygulama Geliştirme	23
2.3.1	Uzaktan Kontrol ve İzleme	23
2.3.2	Anlık Bildirimler ve Uyarılar	24
2.3.3	Sesle Komut Entegrasyonu (Siri)	24
2.3.4	Güvenlik ve Veri Gizliliği	24
2.3.5	Kullanıcı Geri Bildirimi ve İyileştirme	25
2.3.6	Kullanıcı Dostu Arayüz Tasarımı	25
2.4	Arıza Tespiti ve Önleyici Bakım	26
2.4.1	Anlık Bildirimler ve Kullanıcı Etkileşimi	26
2.4.2	Servis Ekibi ile İletişim	26
2.4.3	Önleyici Bakım Planlaması	26
3	Literatür Taraması.....	27
4	Sistem Gereksinimleri	29
4.1	Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Gereksinimler	29
4.2	Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Olmayan Gereksinimler	30
4.3	Kullanıcı Arayüzü Gereksinimleri	31
5	Gereksinimlerin Belirlenmesi.....	33
5.1	Paydaşlar	33
5.2	Aktörler ve Hedefler	33
5.3	Kullanım Durumları (Use Cases).....	34
5.3.1	Use Case UC-1: GirişYap.....	34
5.3.2	Use Case UC-2: KayıtOl	35
5.3.3	Use Case UC-3: CihazEkle.....	36
5.3.4	Use Case UC-4: CihazYonet	37
5.3.5	Use Case UC-5: ArızaBildir	38

5.3.6	Use Case UC-6: BirimYonet	39
5.3.7	Use Case UC-7: LimitYonet.....	40
5.3.8	Use Case UC-8: EylemYonet	41
5.3.9	Use Case UC-9: SesliKomut	42
5.3.10	Use Case UC-10: HataAyikla.....	43
5.4	Sistem Sequence Diyagramları (SSD).....	45
5.4.1	SSD-1: GirisYap.....	45
5.4.2	SSD-2: KayitOl.....	45
5.4.3	SSD-3: CihazEkle.....	46
5.4.4	SSD-4: CihazYonet	46
5.4.5	SSD-5: ArizaBildir	47
5.4.6	SSD-6: BirimYonet	47
5.4.7	SSD-7: LimitYonet.....	48
5.4.8	SSD-8: EylemYonet	48
5.4.9	SSD-9: SesliKomut.....	49
5.4.10	SSD-10: HataAyikla.....	49
6	Uygulama Ekran Görüntüleri	50
6.1	Android Ekran Görüntüleri	50
6.2	Ios Ekran Görüntüleri	52
6.3	Firestore Veritabanı Ekran Görüntüleri.....	53
6.4	Geliştirme Ortamı Ekran Görüntüleri	54
7	Sonuç ve Bulgular	58
7.1	Sistemin Sağladığı Avantajlar.....	58
7.2	Gelecek Çalışmalar ve Geliştirmeler	59
	Kaynaklar	60

Şekiller Listesi

Şekil 2.1	Fotosel Sensörü ve Fotoselli Armatür	19
Şekil 2.2	WiFi Router Haberleşme Şeması	21
Şekil 2.3	IFTTT ve Arduino Atmel Haberleşme Protokolü Şeması	22
Şekil 2.4	Firestore Cloud Platform Haberleşme Protokolü Şeması	22
Şekil 2.5	Firestore Veri Analizi ve Kayıt Verileri Şeması.....	23
Şekil 2.6	Uygulama Arayüz Tasarım Özet Ekranı Şeması.....	25
Şekil 5.4.1	SSD-1: GirişYap Diagram	45
Şekil 5.4.2	SSD-2: KayıtOl Diagram	45
Şekil 5.4.3	SSD-3: CihazEkle Diagram	46
Şekil 5.4.4	SSD-4: CihazYonet Diagram	46
Şekil 5.4.5	SSD-5: ArızaBildir Diagram.....	47
Şekil 5.4.6	SSD-6: BirimYonet Diagram	47
Şekil 5.4.7	SSD-7: LimitYonet Diagram	48
Şekil 5.4.8	SSD-8: EylemYonet Diagram.....	48
Şekil 5.4.9	SSD-9: SesliKomut Diagram	49
Şekil 5.4.10	SSD-10: HataAyıkla Diagram	49
Şekil 6.1.1:	Android Ana Ekran Sekmeleri Ekran Görüntüleri	50
Şekil 6.1.2:	Android Ayarlar Sekmesi Birim Düzenleme Ekran Görüntüleri-1	50
Şekil 6.1.3:	Android Ayarlar Sekmesi Birim Düzenleme Ekran Görüntüleri-2	51
Şekil 6.1.4:	Android Cihaz Ekleme ve Servis Kayıt Oluşturma Ekran Görüntüleri..	51
Şekil 6.1.5:	Android Profil, Giriş Yap, Kayıt Ol Ekran Görüntüleri	51
Şekil 6.2.1:	Ios Ana Ekran Sekmeleri Ekran Görüntüleri.....	52
Şekil 6.2.2:	Ios Ayarlar, Profil, Sesli Komut Ekran Görüntüleri	52
Şekil 6.2.3:	Ios Arıza Tespit ve Bildirim Örnek Ekran Görüntüleri	53
Şekil 6.3.1:	Firestore Parametre Yönetim Ekran Görüntüsü	53
Şekil 6.3.2:	Firestore Geçmiş Kayıt Log Görüntüleme Ekran Görüntüsü.....	54
Şekil 6.4.1:	Xcode Geliştirme Ortamı Ekran Görüntüsü	54
Şekil 6.4.2:	Google Firestore Servis Ekran Görüntüsü.....	55

Şekil 6.4.3: Ios Uygulama Kullanılan Kütüphane Paketleri Ekran Görüntüsü.....	55
Şekil 6.4.4: Xcode Uygulama Arayüz Tasarım Özet Ekranı Şeması.....	55
Şekil 6.4.5: Google Firebase Servis Veritabanı Servisi Ekran Görüntüsü.....	56
Şekil 6.4.6: Google Firebase Servis Veritabanı Sınıfı Ekran Görüntüsü.....	56
Şekil 6.4.7: Sesli Komut ile Su Akışı Sağlayan Foksiyon Kodu Ekran Görüntüsü...	57
Şekil 6.4.8: Arıza Bildirimi Sızıntı Tespit Edilen Fonksiyon Kodu Ekran Görüntüsü.	57

Tablolar Listesi

Tablo 4.1	Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Gereksinimler Tablosu	29
Tablo 4.2	Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Olmayan Gereksinimler Tablosu	30
Tablo 4.3	Kullanıcı Arayüzü Gereksinimleri Tablosu	31
Tablo 5.1	Paydaşlar Tablosu	32
Tablo 5.2	Aktörler ve Hedefler Tablosu.....	33
Tablo 5.3.1	Use Case UC-1: GirişYap Tablosu	34
Tablo 5.3.2	Use Case UC-2: KayıtOl Tablosu	35
Tablo 5.3.3	Use Case UC-3: CihazEkle Tablosu	36
Tablo 5.3.4	Use Case UC-4: CihazYonet Tablosu.....	37
Tablo 5.3.5	Use Case UC-5: ArızaBildir Tablosu.....	38
Tablo 5.3.6	Use Case UC-6: BirimYonet Tablosu.....	39
Tablo 5.3.7	Use Case UC-7: LimitYonet Tablosu	40
Tablo 5.3.8	Use Case UC-8: EylemYonet Tablosu.....	41
Tablo 5.3.9	Use Case UC-9: SesliKomut Tablosu	42
Tablo 5.3.10	Use Case UC-10: HataAyikla Tablosu	43

Kısaltmalar Listesi

UC	Use Case
SSD	System Sequence Diagram
İKÇÜ	İzmir Kâtip Çelebi Üniversitesi
ORCID	Open Researcher and Contributor ID
TÜBİTAK	Türkiye Bilimsel ve Teknolojik Araştırma Kurumu

Bölüm 1

Giriş

Mobil uygulamalar, günümüzde hızla değişen teknoloji ve bilgi toplumu ile birlikte hayatımızın ayrılmaz bir parçası haline gelmiştir. Akıllı telefonlar ve tabletler gibi mobil cihazlar aracılığıyla erişilebilen bu uygulamalar, kullanıcılara geniş bir yelpazede işlevsellik sunarak günlük yaşamlarını önemli ölçüde kolaylaştırmaktadır. Mobil uygulamaların popülerliği, kullanım kolaylığı, gelişmiş özellikleri ve hızlı erişim imkanlarından kaynaklanmaktadır. Günümüzde bireylere geniş bir kullanım yelpazesi sunarak iş, sosyal yaşantı ve eğitim gibi birçok alanda etkin bir şekilde kullanılmaktadır. Bu çok yönlü kullanım alanlarından biri Internet of Things (IoT)'dir.

IoT, nesnelerin interneti olarak adlandırılan bir teknoloji trendidir ve mobil uygulamalar, bu alandaki yenilikleri destekleyerek kullanıcılara akıllı armatürlerin etkileşimini kolaylaştırma ve yönetme imkanı sunmaktadır. Bu sayede bireyler, mobil uygulamalar aracılığıyla su akış yönetimini uzaktan kontrol edebilir, enerji verimliliğini artırabilir ve çevresel sürdürülebilirlik hedeflerine katkıda bulunabilir. İşte bu noktada mobil uygulamalar, IoT ve armatür sektörünün birleşiminde kullanıcıların yaşam kalitesini artırmak için önemli bir rol oynamaktadır.

Armatür, suyun akışını kontrol etmek, yönlendirmek ve düzenlemek amacıyla kullanılan mekanik veya elektronik cihazlardır. Bu kapsamda musluklar, lavabo bataryaları, duş başlıkları, vanalar ve benzeri donanımlar, su tesisat sistemlerinde armatür olarak adlandırılır. Armatürler, kullanıcıların suyun sıcaklık ayarlarını kontrol etmelerini, su akışını düzenlemelerini ve ihtiyaçlarına uygun olarak su kaynaklarını kullanmalarını sağlar. Su tesisatındaki armatürler genellikle dayanıklı malzemelerden üretilir ve uzun ömürlü, güvenilir performans sağlamak üzere tasarlanır.

1.1 Araştırma Amacı

Mevcut durumda armatürlerin geneli akıllı sistemler olarak üretilmemektedir. Ürün gamında bulunan fotoselli ürünler akıllı bir sistemin parçası olabilecek kabiliyette değildir. Bu sebeple su tüketimi, arıza, bakım gibi durumlar kullanıcılar tarafından manuel olarak takip edilmek durumundadır.

Kullanıcı tarafından istenmeyen durumlarda (su kaçağı, arıza vb.) su tüketiminin arttığı fark edilmez ve istenmeyen finansal zararlar karşılanmaktadır. Ayrıca kullanıcı tarafından fark edilen bir problemle karşılaşıldığı zaman ise, bireysel olarak müdahalesi söz konusu olmadığından dolayı, o birime bakan servisi çağırarak durumu kalınmaktadır.

Projenin sunduğu çözüm, IoT teknolojisinin gücünü kullanarak fotoselli armatürleri merkezi bir kontrol birimi üzerinden yönetmeyi sağlamaktır. Bu sayede, armatürlerin su tüketim miktarı, arıza durumları ve bakım ihtiyaçları gibi kritik bilgiler anlık olarak takip edilebilecek ve bu bilgiler mobil uygulama üzerinden iletilerek sorunlara hızlı bir şekilde müdahale edilebilecektir.

Geliştirilecek sistem bunu ön görerek kullanıcıya ya da servise bilgi verebilir. Hali hazırda dünya genelinde yaygın olarak kullanılan akıllı ev sistemine adapte armatürler bulunmamaktadır. Bu nedenle geliştirilecek olan ürünlerle birlikte bu pazar içerisinde aktif olarak yer alınacaktır. Bu sistemde sıcaklık ve debi kontrolünün dışında anlık ve toplam tüketim, arıza durumu, montaj tarihi gibi raporlamaları da kullanıcı tarafından görüntülenmesi sağlanacaktır.

Hedefimiz armatür grubunu zenginleştirerek ürünlerin bu saymış olduğumuz özelliklere sahip olarak akıllı ev sistemindeki ürünlerin oluşturduğu ekosistemin bir parçası haline dönüştürmektedir. Bunun için yapılan arge çalışmalarında kullanıcıların mekanik aksam yerine elektronik aksam olarak dokunmatik sensörlerin yerleştirildiği elektronik bir panel vasıtasıyla suyun sıcaklığını, debisini ve akış yönünü değiştirmesi için tasarım, donanım ve yazılım çalışmaları yapılacaktır.

Sesle komut özelliği için internet ortamında sıklıkla kullanılan IOT cloud servislerinden faydalanılarak mevcut fotoselli ürünlere sesle komut özelliği dahil

edilecektir. Bu özellik sayesinde kullanıcılar belirlenen komut seçeneklerinden istediklerini seçerek armatürlerini mevcut kullanımlarının yanı sıra sesle komut ile de çalıştırabileceklerdir. Bu sayılan özellikler için tasarlanan ve programlanan monil uygulama ile yönetimi sağlanacaktır.

Çeşitli uygulamalarla uzaktan kontrol edilebilen ve akıllı ev sisteminin bir parçası olabilecek armatürleri içerisinde barındıran bir sistem geliştirilecektir. Bu sayede akıllı ev otomasyonu sistemlerinde yaygınlaşan haberleşme protokollerinin bir parçasıda armatürlerde yer alacağı gibi geliştirilecek ürünlerin avm, havaalanı, hastane, hotel vb. ortamlarda kullanılması yaygınlaştırılarak kullanıcıya yeni bir deneyim kazandırmayı hedeflemektedir. Bunların neticesinde sektörde öncü olarak örnek alınacak bir proje ortaya konulması hedeflenmektedir. Toplu kullanım alanlarında sıkça karşılaşılan sorunları çözmek ve kullanıcı deneyimini iyileştirmek için bir çözüm sunmaktadır. Çoğunlukla kamu ve özel alanlarda kullanılan fotoselli armatürlerin yönetimi ve bakımı, günümüzde sıklıkla karşılaşılan sorunlardan biridir. Bu projede, sizin karşılaştığınız günlük sorunları anlamak ve çözmek için bir adım atmak hedeflenmektedir.

1.2 Araştırma Sonucu

Bu projenin çıktıları sonucunda israf olan suyun önlenmesi için yapılan çalışmalara katkıda bulunmuş olacaktır. Bu sistemi kullanan herkes suyun önemini bir kez daha anlamış olacaktır. Buna ek olarak ürün özelliklerinin interaktif olması nedeniyle kullanıcının ürünle ilgili şikayetlerine, düşündükleri özelliklerin iletilmesine, gerekli ise sistem güncellemesine anlık olarak erişimine imkân sağlanacaktır. Yapılacak bu geribildirim sistemi sonucunda, ürünlerin akıllı olmasının yanı sıra yapı sektöründeki taleplerin anında üreticiye ulaşması yönünde bir adım atılmış olacaktır. Bu sistemlerin teknoloji trendlerini takip edenlere hitap edeceği düşünüldüğünde, bundan sonraki trendler için de müşteri tarafından geri bildirimlerin alınması proje açısından ileri dönük bir kazanım sağlayacaktır.

Günümüzde, toplu kullanım alanlarında kullanılan fotoselli armatürlerin yönetimi ve bakımı sürekli bir problem kaynağı oluşturmaktadır. Kullanıcılar arasında şikayetlerin artması, arızaların gecikmeli fark edilmesi, enerji israfı ve bakım ihtiyaçlarının

belirlenememesi gibi sorunlar günlük hayatta sıkça karşılaşılan durumlar haline gelmiştir.

Armatürler arasında haberleşmeyi sağlamak için bir WiFi ağı üzerinden iletişim kurmaktadır. Bu sayede, her bir armatürden elde edilen veriler merkezi bir veritabanına aktararak analiz edilmekte ve belirlenen kriterlere göre raporlar oluşturulmaktadır. Örneğin, bir armatürün su tüketim miktarının anormal derecede yüksek olduğu durumlar otomatik olarak tespit edilerek bildirim gönderilecektir.

Proje özeti kapsamında amaç günlük yaşantıyı kolaylaştırmak ve fotoselli armatürlerin yönetimini daha etkin bir hale getirmektir.

1.3 Araştırma Avantajları

Anlık Takip ve Kontrol: Armatürlerin çalışma durumu, su tüketimi ve enerji kullanımı anlık olarak takip edilebilecektir.

Arıza Tespiti ve Önleme: Proaktif bir yaklaşım ile armatürlerde oluşabilecek arızalar önceden tespit edilerek önleyici bakım sağlanacaktır.

Enerji Verimliliği: Su tüketim miktarının doğru bir şekilde takip edilmesi, enerji israfını minimuma indirecek ve sürdürülebilir bir çözüm sunacaktır.

Kullanıcı Memnuniyetinin Artması: Fotoselli armatürlerin daha etkin bir şekilde yönetilmesi, kullanıcıların olası sorunlardan etkilenmemesini sağlayacak ve memnuniyetlerini artıracaktır.

Maliyet Tasarrufu: Proaktif bakım ve arıza önleme sayesinde, beklenmeyen durumlar nedeniyle oluşabilecek maliyetler önceden engellenecek ve bütçenizi koruyacaksınız.

Sürdürülebilirlik ve Çevre Dostu Çözüm: Enerji verimliliğinin artırılması ve su tüketiminin kontrol altında tutulması, çevre dostu bir uygulama sunmaktadır.

Esnek ve Ölçeklenebilir Yapı: Proje farklı ölçeklerdeki kurumlar ve kullanım alanları için uyarlanabilir ve ölçeklenebilir bir yapıya sahiptir.

Kolay Kullanım ve Entegrasyon: Mobil uygulama kullanıcı dostu arayüzleri ile kolay bir şekilde entegre edilebilir ve kullanılabilir.

Geliştirilebilirlik: Proje, gelecekteki teknolojik gelişmelere uyum sağlamak üzere geliştirilebilir ve güncellenebilir bir yapıya sahiptir.

1.4 Terimler Sözlüğü

Fotoselli Armatür: Problem alanımızdaki temel ürün; otomatik olarak çalışan, ışığı algılayarak açılıp kapanan aydınlatma üniteleridir. Ellerin temasına gerek olmadan otomatik olarak açılıp kapanabilen bir armatür türünü ifade eder. Bu tip armatürler, genellikle kamusal alanlarda, ofislerde, tuvaletlerde ve benzeri yerlerde kullanılır. Fotoselli armatürler, içerdikleri sensörler aracılığıyla çevresel koşulları algılar ve kullanıcıların ellerini yaklaştırdığında otomatik olarak su, sabun, ya da başka bir akışkanı devreye sokar.

Akıllı Ev Sistemi: Sensörler, cihazlar veya aktüatörler gibi kendi başlarına akıllı olması gerekmeyen en az birkaç cihazı içermektedir. Bir sensörden üretilen veri kendi başına ev ortamına herhangi bir değer katmaz. Benzer şekilde evin sıcaklığı, dış sıcaklık, nem ve diğer faktörlere göre düzenlemek zorundaysa, ev içerisinde kullanılan termostat akıllı kabul edilmez. Sabit bir sıcaklığı koruyabilir, ancak bu “akıllılık” değil, otomasyondur. Çevre ile ilgili tüm veriler toplu olarak saklanır, analiz edilir ve kullanıcının müdahalesi olmadan sistem kendi başına karar alırsa, bu sistem akıllı olarak adlandırılabilir. Akıllı bir evin mimarisi, cihazların birbirleriyle iletişim kurma şekline, sensörlerden ve cihaz kullanım alışkanlıklarından gelen bilgilerin nasıl ve nerede saklanacağına, bu bilgilerin nasıl işlendiğine ve kullanıcının cihazlarla nasıl etkileşime girebileceği gibi kriterlerle belirlenir.

IoT (Nesnelerin İnterneti): Nesnelerin birbirleri ile iletişim kurabildiği, veri toplayabildiği ve merkezi bir kontrol birimi tarafından yönetilebildiği internet tabanlı sistemdir. IoT'nin temel amacı, fiziksel dünyadaki nesnelerin, cihazların ve sistemlerin birbirleriyle etkileşimde bulunabilmesi ve veri paylaşabilmesidir. Bu sayede daha akıllı, daha etkili ve daha verimli sistemler oluşturulabilir.

WiFi Haberleşme: Kablosuz haberleşme standardını ifade eder. Wi-Fi, bilgisayarlar, akıllı telefonlar, tabletler ve diğer çeşitli cihazların kablosuz ağlara bağlanabilmesini sağlayan bir teknolojidir. Bu teknoloji, belirli bir alanda (genellikle bir ev, ofis veya genel bir alan) internet erişimini paylaşmak veya cihazlar arasında veri iletimini sağlamak amacıyla kullanılır. Wi-Fi, IEEE 802.11 ailesi standartlarına dayanır. Bu standartlar, kablosuz iletişimde kullanılan frekans bantlarını, veri iletim hızlarını ve diğer teknik özellikleri belirler. Wi-Fi, genellikle 2.4 GHz ve 5 GHz frekans bantları üzerinden iletişim kurar. Bu bantlar, kablosuz cihazların birbirleriyle ve kablosuz erişim noktalarıyla (access point) bağlantı kurmalarına izin verir.

Merkezi Kontrol Birimi: Projenin ana kontrol noktası; armatürlerin yönetimini sağlayan, veri toplama, analiz ve kullanıcı arayüzü sunan birimdir. Kullanıcıya sunulan mobil uygulama olarak nitelendirebiliriz.

Su Tüketim Miktarı: Armatürlerin belirli bir zaman diliminde tükettiği su miktarıdır. Su tüketim miktarı genellikle litre veya metreküp cinsinden ölçülür ve geniş bir yelpazede kullanım alanlarına sahiptir.

Arıza Tespiti: Sistem tarafından algılanan ve bildirilen armatürlerdeki olası problemlerin tanımlanmasını kapsar. Genellikle bir cihazın, makinenin, yazılımın veya başka bir sistem bileşeninin beklenen şekilde çalışmaması durumunda uygulanan bir yöntemdir.

Proaktif Bakım: Arıza olmadan önce sorunları tespit ederek önleyici müdahale sağlama prensibini ifade etmektedir. Armatürlerin belirli bir periyotta geçmesi gereken bakım sürecini de kapsamaktadır.

Ölçeklenebilir Yapı: Sistemin büyüklüğünü, kapsamını ve kullanıcı sayısını esnek bir şekilde artırabilme yeteneğidir. İlerleyen zamanda bu ürün ailesine akılsız olarak ifade edilebilecek ürün ya da ürün gruplarının eklenebilmesi sağlanacaktır.

Mobil Uygulama: Akıllı telefon veya tablet gibi taşınabilir cihazlarda çalışan uygulamadır. Kullanıcı bu uygulama üzerinden sistemin analiz edilebilmesi ve istenilen parametrelere göre cihaz yönetimlerinin gerçekleştirilmesi sağlamaktadır.

Siri: Apple tarafından geliştirilen bir sanal asistan ve sesli komut teknolojisidir. Siri, kullanıcının sesli komutlarını algılayarak çeşitli görevleri yerine getirmek veya bilgi sağlamak için tasarlanmıştır. Uygulama içinde belirlenmiş sesli komutları uygulamaya ileterek su akışı sağlanmaktadır.

Veritabanı: Sistem tarafından toplanan verilerin depolandığı elektronik ortamdır. Firebase altyapısı kullanılmıştır.

Sistem: Uygulamanın tüm bileşenlerini içeren genel yapıya denir.

Kullanıcı: Uygulama üzerinden akıllı sistemlerin yönetimini sağlayacak kişidir.

Yönetici: Sistemin yönetimi sağlayacak kişidir.

Bölüm 2

Yöntemler

Proje kapsamında, mevcut durumda kullanılan fotoselli armatürlerin akıllı sistemlerle entegre edilmesi ve IoT teknolojisinin gücünden yararlanılması amaçlanmaktadır. Bu projede geliştirilecek olan sistem, su tüketimi, arıza durumları, bakım ihtiyaçları gibi kritik bilgileri anlık olarak takip edebilmek için mobil uygulama aracılığıyla merkezi bir kontrol birimi üzerinden yönetim sağlayacaktır. Kullanıcılara su kaçağı gibi istenmeyen durumlar hakkında anında bildirimler gönderilerek finansal zararların önüne geçilecektir. Ayrıca, geliştirilen sistem sayesinde armatürlerin sıcaklık, debi, arıza durumu gibi detaylı raporları kullanıcılar tarafından görüntülenebilecektir. Sesle komut özelliği, internet ortamında sıklıkla kullanılan IoT cloud servisleri ile entegre edilerek, kullanıcılara sesli komutlarla armatürlerini yönetme imkanı sağlanacaktır. Bu yenilikçi yaklaşım, fotoselli armatürleri geleneksel sınırlarının ötesine taşıyarak, akıllı ev sistemlerinin bir parçası haline getirerek sektörde öncü bir projenin ortaya konulmasını amaçlamaktadır.

2.1 Sensör Teknolojisi ve Elektronik Panel Tasarımı

Armatürlerin mekanik aksam yerine dokunmatik sensörlerin entegrasyonu için gerekli tasarım çalışmaları yapılacaktır. Elektronik panel üzerine yerleştirilecek sensörler, su sıcaklığı, debisi ve akış yönü gibi parametreleri ölçebilecek şekilde tasarlanacaktır. Sensör teknolojisi sayesinde kullanıcı, armatürleri dokunmadan kontrol edebilecek ve istediği ayarlamaları gerçekleştirebilecektir.



Şekil 2.1: Fotosel Sensörü ve Fotoselli Armatür

2.1.1 Dokunmatik Sensör Entegrasyonu

Armatürlerin mekanik aksamalarının yerine dokunmatik sensörler entegre edilecektir. Kullanıcı, mobil uygulama üzerinden armatürleri dokunmadan kontrol edebilecek ve dokunmatik sensörler sayesinde istediği ayarlamaları gerçekleştirebilecektir.

2.1.2 Elektronik Panel Tasarımı

Elektronik panel, armatür üzerine yerleştirilecek ve mobil uygulama aracılığıyla kontrol edilebilecek bir arayüz sunacaktır. Panel, kullanıcının su sıcaklığı, debisi ve akış yönü gibi parametreleri anlık olarak görmesini sağlayacak bilgileri içerecektir.

2.1.3 Su Sıcaklığı ve Debi Ölçümü

Elektronik panelde yer alan sensörler, su sıcaklığını ve debiyi ölçerek kullanıcıya anlık bilgi sunacaktır. Mobil uygulama üzerinden kullanıcı, su sıcaklığını ve debiyi istediği şekilde ayarlayabilecek ve kontrol edebilecektir.

2.1.4 Akış Yönü Kontrolü

Sensör teknolojisi, armatürlerin akış yönünü algılayabilecek ve kullanıcı, mobil uygulama aracılığıyla akış yönünü istediği gibi kontrol edebilecektir. Bu özellik, su tasarrufu sağlamak ve kullanıcıya özelleştirilebilir bir deneyim sunmak amacıyla tasarlanmıştır.

2.2 IoT Entegrasyonu ve Cloud Servis Kullanımı

Armatürler arasında haberleşmeyi sağlamak amacıyla bir WiFi ağı üzerinden iletişim kurulacaktır. Internet of Things (IoT) cloud servislerinden faydalanarak, armatürlerden elde edilen veriler merkezi bir veritabanına aktarılacak ve analiz edilecektir. Sesle komut özelliği için yaygın olarak kullanılan IoT cloud servisleri entegre edilecek ve kullanıcılara sesle kontrol imkânı sunulacaktır.

2.2.1 WiFi Haberleşme Altyapısı

Armatürler arasında etkili bir iletişim sağlamak amacıyla, bir WiFi ağı üzerinden haberleşme altyapısı oluşturulacaktır. WiFi bağlantısı, kullanıcıların mobil uygulama aracılığıyla armatürleri uzaktan kontrol etmelerini sağlayacak bir temel oluşturacaktır.

Wi-Fi veya Kablosuz Bağlantı, IEEE 801.11 standardını temel alan bir kablosuz iletişim protokolüdür. Wi-Fi bir lisans gerektirmez ve bu nedenle günümüzde kullanılan en popüler kablosuz teknolojilerden biri haline gelmiştir. İç mekânda 45 m'lik teorik bir kapsama alanına sahiptir. Wi-Fi menzili, tekrarlayıcıları ve yedek erişim noktaları ile genişletilebilir. Bu da onu WLAN'lar için uygun hale getirmektedir. Bu teknoloji WPA2 şifrelemesini destekler ve 2.4–5.8 GHz frekans spektrumunda çalışır. Wi-Fi teknolojisinin dezavantajları arasında yüksek güç talebi ve elektronik girişimlere yatkınlık sayılabilir. Ayrıca, iç mekân içerisinde bulunan engeller ağ hızını ve güvenilirliğini etkileyebilmektedir (Poulakis et al., 2016).



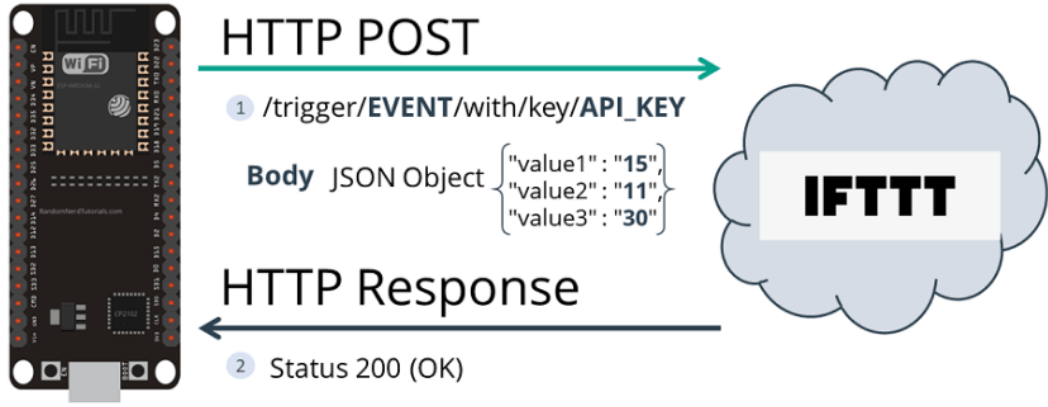
Şekil 2.2: WiFi Router Haberleşme Şeması

2.2.2 IoT Cloud Servis Entegrasyonu

Internet of Things (IoT) cloud servisleri, amatörlerden elde edilen verilerin merkezi bir veritabanına aktarılmasını sağlayacaktır. Bu servisler, verilerin güvenli ve hızlı bir şekilde depolanması, analiz edilmesi ve raporlanması için kullanılacaktır. Bu entegrasyonda, IFTTT ve Firebase gibi öne çıkan servisler de kullanılacaktır.

2.2.2.1 IFTTT Entegrasyonu

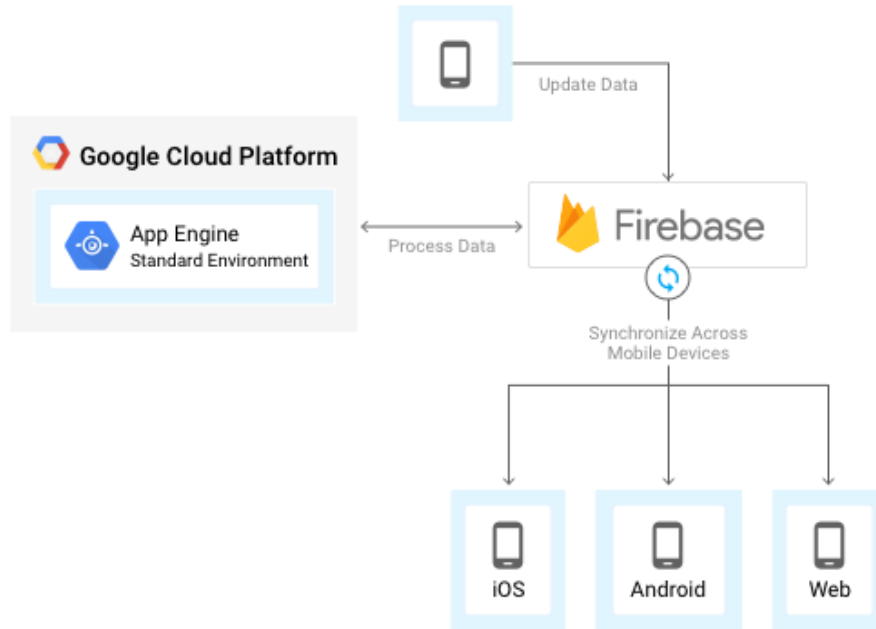
IFTTT Entegrasyonu: Amatörlerden gelen belirli durumlar (örneğin, anormal su tüketimi, arıza durumları) tetikleyici olarak kullanılarak IFTTT entegrasyonu gerçekleştirilecektir. IFTTT servisi, bu tetikleyici durumlar üzerine belirlenen eylemleri otomatik olarak gerçekleştirecek ve ilgili verilerin IoT cloud servisine iletilmesini sağlayacaktır. Örneğin, bir arıza durumu tespit edildiğinde IFTTT, belirlenen bir eylemi tetikleyerek ilgili verilerin analiz için IoT cloud servisine aktarılmasını sağlayacaktır.



Şekil 2.3: IFTTT ve Arduino Atmel Haberleşme Protokolü Şeması

2.2.2.2 Google Firebase Entegrasyonu

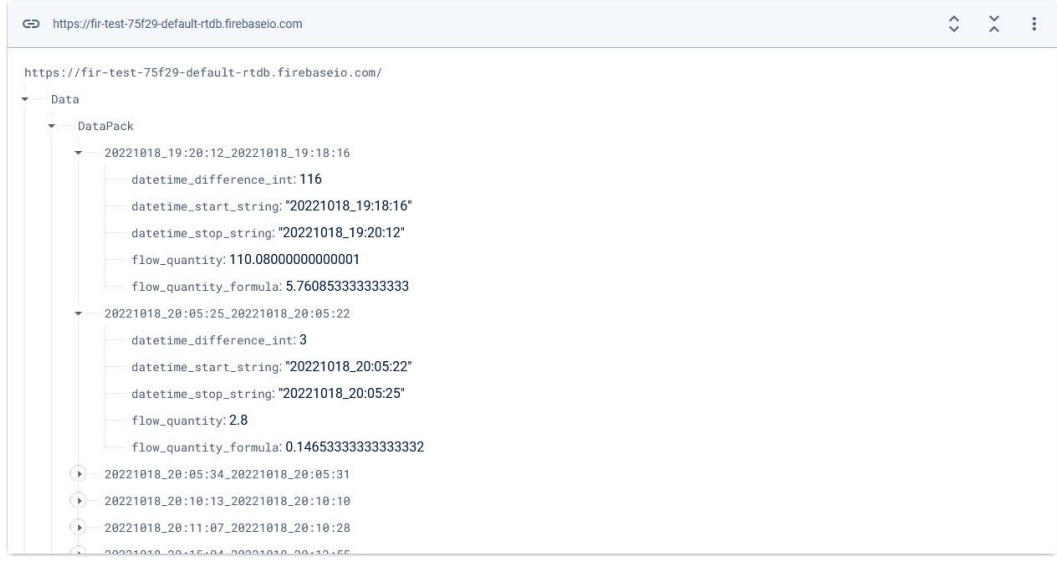
Firestore, armatürlerden gelen verilerin anlık olarak depolanması ve gerçek zamanlı analiz edilmesi için kullanılacaktır. Firestore Realtime Database, kullanıcıların mobil uygulama üzerinden anlık olarak armatür durumlarını takip etmelerini sağlayacak ve hızlı bir şekilde güncel verilere erişim imkânı sunacaktır. Firestore Cloud Functions kullanılarak, verilerin güvenli bir şekilde depolanması ve istenilen formatta raporlanması sağlanacaktır.



Şekil 2.4: Firestore Cloud Platform Haberleşme Protokolü Şeması

2.2.3 Veri Analizi ve Raporlama

IoT cloud servisleri, armatürlerden gelen verileri analiz ederek su tüketimi, enerji kullanımı, arıza durumları gibi kritik bilgileri belirleyecek ve raporlar oluşturacaktır. Mobil uygulama üzerinden kullanıcılara, bu raporlar aracılığıyla anlık bilgiler sunulacak ve kullanıcılar su tüketimi gibi önemli verileri takip edebilecektir.



Şekil 2.5: Firebase Veri Analizi ve Kayıt Verileri Şeması

2.3 Mobil Uygulama Geliştirme

Kullanıcıların armatürleri uzaktan yönetebilmeleri için bir mobil uygulama tasarlanacaktır. Mobil uygulama, anlık takip ve kontrol imkânı sağlayacak, su tüketimi, arıza durumları, enerji kullanımı gibi kritik bilgileri kullanıcılara iletacaktır. Kullanıcılar, mobil uygulama üzerinden armatürleriyle etkileşimde bulunabilecek, gerekirse geri bildirimde bulunabilecek ve istedikleri ayarlamaları yapabilecektir.

2.3.1 Uzaktan Kontrol ve İzleme

Mobil uygulama, kullanıcılara armatürleri uzaktan kontrol etme imkânı sağlayacaktır. Anlık takip özelliği, kullanıcılara su tüketimi, arıza durumları, enerji kullanımı gibi kritik bilgileri sağlayarak kullanıcıların armatürleri etkin bir şekilde yönetmelerini mümkün kılacaktır.

2.3.2 Anlık Bildirimler ve Uyarılar

Mobil uygulama, armatürlerden alınan verilere dayalı olarak kullanıcılara anlık bildirimler ve uyarılar gönderecektir. Örneğin, anormal su tüketimi, potansiyel arıza durumları veya bakım ihtiyaçları gibi önemli durumlar hakkında kullanıcılar bilgilendirilecek ve hızlı bir müdahale imkânı sağlanacaktır.

2.3.3 Sesle Komut Entegrasyonu (Siri)

Sesle komut özelliği için yaygın olarak kullanılan IoT cloud servisleri, mobil uygulama üzerinden armatürleri sesle kontrol etme imkânı sunacaktır. Kullanıcılar, mobil uygulama aracılığıyla belirlenen komutları kullanarak su açma/kapama, istenilen miktarda su akışının sağlanması gibi işlemleri sesle gerçekleştirebilecektir.

Siri entegrasyonu, Siri'nin doğal dil anlama yeteneği, kullanıcıların armatürleriyle etkileşimde bulunurken belirlenen komutları hatırlamak zorunda kalmadan istedikleri değişiklikleri yapmalarına olanak sağlayacaktır. Bu entegrasyon, eller serbest kullanımı destekleyerek kullanıcılara yoğun anlarda veya ellerin meşgul olduğu durumlarda armatürleri rahatça kontrol etme imkânı sunacaktır. Ayrıca, Siri'nin hızlı tepki süreleri, kullanıcıların anında istedikleri komutları ileterek armatürleri hızlı bir şekilde yönetmelerini sağlayacaktır.

Projedeki IoT cloud servisleri ile birleştirildiğinde, Siri ile entegre edilen sesle komut özelliği, projenin kullanıcı dostu ve etkileşimli bir deneyim sunma hedefine önemli bir katkı sağlayacaktır.

2.3.4 Güvenlik ve Veri Gizliliği

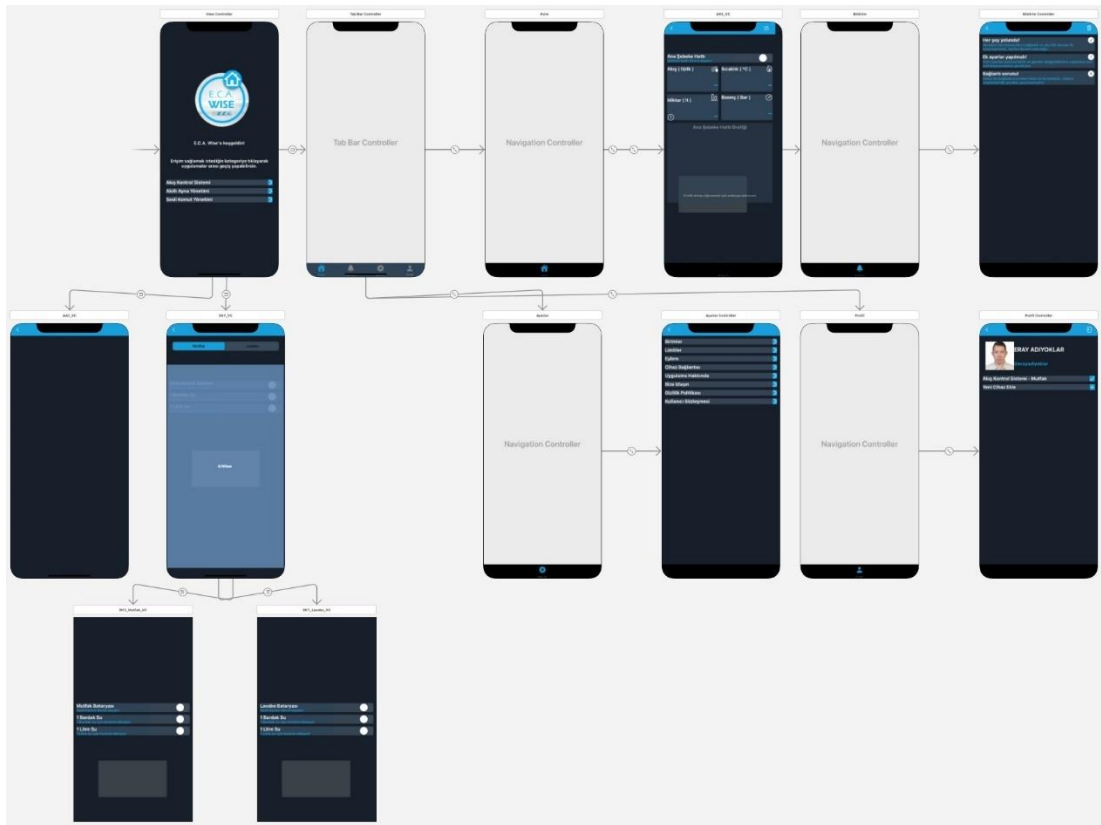
IoT cloud servisleri başta olmak üzere uygulama içinde iletilen veriler, endüstri standardına uygun güvenlik protokollerine tabi tutularak gizlilik sağlanacaktır. Mobil uygulama üzerinden yapılan tüm işlemler, güvenli bir bağlantı üzerinden gerçekleştirilecek ve kullanıcı verileri korunacaktır.

2.3.5 Kullanıcı Geri Bildirimi ve İyileştirme

IoT cloud servisleri, kullanıcıların mobil uygulama üzerinden veri analizi ve raporlama ile ilgili geri bildirimlerini alacak ve gerekirse sistem güncellemeleri için kullanıcılarla etkileşime geçecektir. Bu sayede sistem, kullanıcı beklentilerine uyum sağlayacak ve sürekli olarak iyileştirilecektir.

2.3.6 Kullanıcı Dostu Arayüz Tasarımı

Mobil uygulama ve elektronik panel arayüzleri, kullanıcı dostu bir tasarıma sahip olacak ve kullanıcının kolayca anlayabileceği bir arayüz sunacaktır. Ayarlamaların yapılması ve bilgilerin takip edilmesi, mobil uygulama ve panel üzerinden kolaylıkla gerçekleştirilebilecektir.



Şekil 2.6: Uygulama Arayüz Tasarım Özet Ekranı Şeması

2.4 Arıza Tespiti ve Önleyici Bakım

Armatürlerden elde edilen veriler, belirlenen kriterlere göre analiz edilerek olası arızalar proaktif bir şekilde tespit edilecektir. Arıza durumlarında kullanıcılara anlık bildirimler gönderilerek, servis ekipleri müdahale edebilecek ve önleyici bakım sağlanacaktır.

2.4.1 Anlık Bildirimler ve Kullanıcı Etkileşimi

Analiz sonuçlarına dayanarak tespit edilen potansiyel arıza durumlarında, mobil uygulama üzerinden kullanıcılara anlık bildirimler gönderilecektir. Kullanıcılar, bu bildirimler aracılığıyla hızlı bir şekilde arıza durumlarından haberdar olacak ve gerektiğinde mobil uygulama üzerinden detaylı bilgilere erişebileceklerdir.

2.4.2 Servis Ekibi ile İletişim

Kullanıcılara iletilen arıza bildirimleri aynı zamanda servis ekiplerine de yönlendirilecektir. Mobil uygulama, kullanıcının onayı ile servis ekiplerine otomatik olarak arıza detaylarını ileticek ve böylece hızlı bir müdahale sağlanacaktır.

2.4.3 Önleyici Bakım Planlaması

Analiz sonuçlarına dayanarak belirlenen arıza durumları, önleyici bakım planlamasına dahil edilecektir. Mobil uygulama üzerinden kullanıcılara, önleyici bakımın ne zaman ve nasıl gerçekleştirileceği konusunda bilgilendirme yapılacak ve gerekirse kullanıcıdan onay alınacaktır.

Bölüm 3

Literatür Taraması

Soliman ve arkadaşları (2013) tarafından yapılan çalışmada, akıllı bir ev için bir sistem mimarisi yaratılmıştır. Sensör ve aktüatör bir mikro denetleyiciye bağlanır ve çevre hakkında veri toplamak ve belirli eylemleri gerçekleştirmek için kullanılır. Veriler, mikrodenetleyici özellikli sensörlerden kablosuz ZigBee teknolojisi kullanılarak merkezi bir sunucuya iletilir. Sunucu daha sonra verileri bir uygulama programlama arabirimi (API) aracılığıyla bir bulut depolama alanına yükler. Cloud çözümü bir back-end uygulaması, depolama alanı ve bir front-end uygulamasından oluşur. Veriler back-end'de (Google App Engine) işlenir ve analiz edilir. Kullanıcı bir Web uygulaması kullanarak ortamı görselleştirebilir ve cihazları kontrol edebilir.

Guoqiang ve arkadaşları (2013) tarafından yapılan çalışmada, desteklenen iletişim protokolleri açısından uygun olan ve heterojen sensör verilerini muntazam bir formata çevirebilen akıllı bir ev ağ geçidi tasarımı önerilmektedir. Ağ geçidi genellikle uç cihazlarla birlikte çalışabilirlik için çoklu iletişim protokollerini destekleyen bir cihazdır. Verileri buluta göndermeden önce ağın kenarında bazı işlemler yapmak için yeterince güçlüdür. Ağ geçidi ayrıca akıllı ev ağına bir güvenlik katmanı ekler, çünkü son cihazlar ve dış dünya arasındaki iletişimi köprülemektedir ve böylece komutlar son aygıtlara ulaşmadan önce tüm iletişim değiştirilebilir (kaynak kısıtlamalı, düşük güçlü, daha düşük güvenlik katmanlarına sahip olan ve bu nedenle saldırılara karşı daha savunmasız olan). Bulut, temelde kullanılabilirliği, güvenilirliği ve güvenliği artırmak için araçlar sağlarken aynı zamanda daha yüksek hesaplama gücü ve ölçeklenebilir mimariden faydalanır. Bulut, veri görselleştirme, akıllı ev cihazı yönetimi veya kullanıcı erişimi ve rol yönetimi gibi birçok üçüncü taraf hizmetiyle entegre olabilir.

Amir ve John (2019) yaptıkları çalışmada akıllı evler için ağ modeli tanıtmışlardır. Heterojen teknolojilerden oluşan ağların temsili karmaşıklığını basitleştirmek amacıyla yaygın olarak kullanılan teknolojileri ve ağ topolojilerini dikkate almaktadırlar. Modelilerini normal durumdayken ve ana İnternet bağlantısı hatası sırasında çeşitli grafik merkezi metrikleri aracılığıyla star ve mesh gibi diğer temel topolojilerle karşılaştırmışlardır. Yaptıkları modelin düğümleri çeşitli işlevlere ve farklı bit hızı bağlantılarına sahip çok teknoloji bir ağı temsil etmektedir.

Liu (2014) tarafından yapılan çalışmada ZigBee teknolojisi ve GSM/GPRS şebekesine dayalı akıllı ev sistemi, sistem fonksiyonları, donanım tasarımı, ZigBee teknolojisinin sisteme uygulanmasının analizini tasarlanmıştır. ZigBee teknolojisi akıllı evlerde hızlı, düşük maliyetli ve güçlü kablosuz ağ iletişimi sağlamak için kullanılmaktadır. CC2430 ZigBee kablosuz sensör ağlarını kullanarak, akıllı ev sistemi içerisinde ortam sıcaklığı, nem, kızılötesi, gaz, yangın, hırsızlık alarmı sensörleriyle birlikte ev ortamının gerçek zamanlı olarak bilgi akışını sağlamaktadır. Ev ortamında kullanılan cihazlar, geleneksel sensör alarm sistemi ve görüntü izleme sistemini birleştirerek yeni tip akıllı ev sistemi türünün oluşmasını sağlıyor. Kullanıcılar, esnek ve kullanışlı bir ev güvenlik izlemesi sağlamak için uzaktan telefon ayarları seçimine göre cep telefonu veya PC üzerinden uyarıları MMS mesajları olarak almaktadır.

Koo ve arkadaşları (2016) tarafından yapılan çalışmada yaşlı insanların güvenliği hedef alınarak IoT sistemlerin yardımıyla banyoyu akıllı hale getirmişlerdir. Bu makalede sunulan IoT ve büyük veri kavramı dört hedef önermektedir. Bunlar sırayla 1) Sensör seçimi ve uygulaması, 2) Kablosuz sensörün yerel alan ağ sistemine entegrasyonu, 3) IoT uygulaması için tasarım konsepti ve 4) Büyük bir veri analiz sistemi modeli. Sistem banyo faaliyetlerini algılayan sensörlerden elde edilen verilere dayanan büyük ölçekli verilerden oluşacak şekilde test edilecektir. Basınç sensörleri dahil olmak üzere çeşitli sensörler, konsepti küçük ölçekli bir örnekte doğrulamak için test edilmiştir. Bu çalışmanın potansiyel etkisi sadece özel olayları zamanında bilgilendirmek değil, aynı zamanda banyo ortamındaki tehlikeli koşulları ve özellikle yaşlı insanlar için banyo yaralanmaları nedeniyle ciddi çarpışma olasılığını azaltmaktır. Dahası, bu çalışma IoT'nin ve büyük veri uygulamalarının muhtelif yönlerde kullanıcı sürdürülebilirliğini artırmak için muazzam potansiyelini göstermektedir.

Bölüm 4

Sistem Gereksinimleri

4.1 Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Gereksinimler

Tablo 4.1: Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Gereksinimler
Tablosu

Gereksinim Etiketi	Öncelik Ağırlığı	Gereksinim Açıklaması
REQ-1	9	Sistem güvenli oturum açma özelliğine sahip olmalıdır ve parametrelerin güncellenmesine izin vermeli
REQ-2	10	Merkezi kontrol birimi üzerinden armatürleri uzaktan açma ve kapatma yeteneğinin kullanılması
REQ-3	8	IoT sistemi aracılığıyla fotoselli armatürlerin su tüketiminin anlık olarak takip edilmesi ve kayıt altına alınması
REQ-4	10	Armatürlerde oluşabilecek su kaçağı durumlarını tespit edip anlık bildirim gönderilmesi
REQ-5	8	Fotoselli armatürlerdeki arızaları önceden tespit edip proaktif bir bakım süreci başlatılması
REQ-6	7	Fotoselli armatürlerin enerji tüketimini anlık olarak izleyerek enerji verimliliğinin artırılması
REQ-7	7	Uygulama aracılığıyla armatürlerin yeni cihaz eklenmesi, çalışma zamanlarını ve modlarını kullanıcının düzenleyebilmesi

REQ-8	8	Sesli komut ile fotoselli armatürlerden istenilen miktarda su akışının sağlanabilmesi
REQ-9	9	Fotoselli armatürlerden yapay zeka desteği ile kullanıcı kullanımlarını analiz ederek hazır hale geçmesi
REQ-10	8	Kişiselleştirilmiş kullanıcı profili ile kullanıcı su tüketim hedeflerini belirleyebilmeli

4.2 Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Olmayan Gereksinimler

Tablo 4.2: Numaralandırılmış ve Önceliklendirilmiş Fonksiyonel Olmayan Gereksinimler Tablosu

Gereksinim Etiketi	Öncelik Ağırlığı	Gereksinim Açıklaması
NREQ-1	9	Sistem güvenliği için endüstri standardı şifreleme yöntemlerinin kullanılması
NREQ-2	10	Mobil uygulamanın hızlı tepki verme süresinin (örneğin, 2 saniye içinde) sağlanması
NREQ-3	9	Sistem veritabanının yedeklenmesi için otomatik bir mekanizmanın bulunması
NREQ-4	10	Kullanıcı verilerinin gizliliğini sağlamak adına güçlü kimlik doğrulama yöntemlerinin kullanılması
NREQ-5	8	Sistem güncellemeleri için kullanıcıya anlık bildirim gönderilmesi ve güncel sürekliliğinin sağlanması
NREQ-6	9	Merkezi kontrol biriminin %99.99 sürekli çalışabilirliğinin sağlanması
NREQ-7	8	Sistem olarak kullanıcı talepleri/sorunları desteklenmeli ve ele alınmalı
NREQ-8	9	Uygun Dokümantasyon ile birlikte Kullanım Kılavuzu ve Mimari Diyagram sağlanmalı

4.3 Kullanıcı Arayüzü Gereksinimleri

Tablo 4.3: Kullanıcı Arayüzü Gereksinimleri Tablosu

Gereksinim Etiketi	Öncelik Ağırlığı	Gereksinim Açıklaması
UI-1	10	Mobil uygulamada kullanıcıya anlık su tüketimi, akış miktarı, su sıcaklığı, basınç değerlerinin gösterilmesi
UI-2	9	Tüm zamanlardaki ilgili cihaz için su tüketiminin grafiksel olarak gösterilmesi
UI-3	9	Merkezi kontrol biriminde kullanıcının belirli bir armatürü uzaktan kontrol etmesini sağlayan basit bir arayüzün tasarlanması
UI-4	10	Su kaçağı veya arıza durumlarında kullanıcıya anında bildirim gönderilmesi ve bu bilgilerin görsel olarak sunulması
UI-5	9	Fotoselli armatürlerin çalışma modlarını seçmek için kullanıcı dostu bir arayüz tasarımının sağlanması
UI-6	9	Enerji tüketiminin günlük, haftalık ve aylık periyotlarla gösterilmesi
UI-7	9	Sesli komut sistemine paralel çalışacak kullanıcı arayüzü ile yönetimin kolaylaştırılması
UI-8	8	Kullanıcı verilerinin güvenli bir şekilde görüntülenmesi ve yönetilmesi için şifreleme yöntemlerinin kullanılması
UI-9	10	Kullanıcıların giriş/kayıt yapabilmesi, ilgili akıllı cihazlarının yönetiminin sağlanabilmesi için ekranların oluşturulması
UI-10	10	Kullanıcı akıllı ev sisteminde kullanacağı Fotoselli armatürde bulunan QR kodu okutarak uygulamaya ekleyebilmeli

Bölüm 5

Gereksinimlerin Belirlenmesi

5.1 Paydaşlar

Tablo 5.1: Paydaşlar Tablosu

Gereksinim Etiketi	Gereksinim Açıklaması
Kullanıcı	Uygulama üzerinden akıllı sistemlerin yönetimini sağlayacak kişidir.
Sistem	Uygulamanın tüm bileşenlerini içeren genel yapıya denir.
Yönetici	Sistemin yönetimi sağlayacak uygulama geliştiricilerini kapsar.
Veritabanı	Sistem tarafından toplanan verilerin depolandığı elektronik ortamdır. Firebase altyapısı kullanılmıştır.
Arayüz	Kullanıcıya sunulan ekranların genel isimlendirilmesidir.
Siri	Uygulama içinde belirlenmiş sesli komutları uygulamaya ileterek su akışı sağlanmaktadır.

5.2 Aktörler ve Hedefler

Tablo 5.2: Aktörler ve Hedefler Tablosu

Actor	Actor's Goal	Use Case Name
Kullanıcı	Kullanıcı uygulamaya giriş yapar.	GirisYap (UC-1)
Kullanıcı	Kullanıcı uygulamaya kayıt olur.	KayitOl (UC-2)
Kullanıcı	Kullanıcı akıllı ev sisteminde kullanacağı Fotoselli armatürde bulunan QR kodu okutarak uygulamaya ekler.	CihazEkle (UC-3)
Kullanıcı	Kullanıcı fotoselli armatürleri açma, kapatma, su tüketimini kontrol etmesini sağlar.	CihazYonet (UC-4)
Sistem/Arayüz	Sistem arıza ya da bakım durumu olan fotoselli armatür için anlık bildirim gönderir.	ArizaBildir (UC-5)
Kullanıcı/Sistem	Sisteme eklenmiş olan fotoselli armatürlerin cihaz yönetiminde kullanıcı arayüzünde gösterilmesini istediği birimlerin parametre yönetimini sağlar.	BirimYonet (UC-6)
Kullanıcı/Sistem	Sisteme eklenmiş olan fotoselli armatürlerin arıza/bakım uyarı bildiriminde limitlerin parametre yönetimini sağlar.	LimitYonet (UC-7)
Kullanıcı/Sistem	Sisteme eklenmiş olan fotoselli armatürlerin cihaz yönetiminin kullanıcı tarafından düzenlenebilmesi için parametre yönetimini sağlar.	EylemYonet (UC-8)
Kullanıcı/Siri	Kullanıcı sesli komut ile sistemde tanımlı ilgili fotoselli armatür üzerinden su akışı sağlayabilir.	SesliKomut (UC-9)
Yönetici	Sistem arkaplanda oluşan hataların gideilmesini sağlar.	HataAyikla (UC-10)
Veritabanı	Kullanıcı bilgilerini, oturum açma ayrıntılarını, verileri, parametreleri, eklenen cihaz listesi, geçmiş ve karşılaştırma verilerini depolamak için kullanılması sağlar.	UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, UC-7, UC-8, UC-9, UC-10

5.3 Kullanım Durumları (Use Cases)

5.3.1 Use Case UC-1: GirişYap

Tablo 5.3.1: Use Case UC-1: GirişYap Tablosu

Use Case UC-1:	GirisYap
Related Requirements:	REQ-1: Sistem güvenli oturum açma özelliğine sahip olmalıdır ve parametrelerin güncellenmesine izin vermeli.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, uygulamaya güvenli bir şekilde giriş yapmak istiyor.
Participating Actors:	Sistem
Preconditions:	1. Kullanıcının kayıtlı bir hesabı olmalıdır. 2. Kullanıcı uygulamayı açmış olmalıdır.
Postconditions:	1. Kullanıcı başarılı bir şekilde oturum açmış olmalıdır.
Flow of Events for Main Success Scenario:	1. Kullanıcı uygulamayı açar. 2. Kullanıcı "Giriş Yap" seçeneğine tıklar. 3. Sistem, kullanıcıdan kullanıcı adı ve şifre bilgilerini ister. 4. Kullanıcı, gerekli bilgileri girer. 5. Sistem, girilen bilgileri kontrol eder. 6. Eğer bilgiler doğru ise, sistem kullanıcıyı uygulamaya yönlendirir ve oturum açma işlemi tamamlanır. 7. Kullanıcı başarılı bir şekilde uygulamaya giriş yapmış olur.
Flow of Events for Extensions (Alternate Scenarios):	6a. Kullanıcı adı veya şifre yanlış girilirse: 1. Sistem, kullanıcıyı hatalı giriş konusunda bilgilendirir. 2. Kullanıcı, doğru bilgileri girene kadar adım 3'ten başlayarak işlemi tekrarlar. 6b. Kullanıcı, "Şifremi Unuttum" seçeneğine tıklarsa: 1. Sistem, kullanıcıya şifre sıfırlama talimatları gönderir. 2. Kullanıcı, talimatları takip eder ve şifresini sıfırlar. 3. Kullanıcı, yeni şifre ile adım 4'ten başlayarak işlemi tamamlar.

5.3.2 Use Case UC-2: KayitOl

Tablo 5.3.2: Use Case UC-2: KayitOl Tablosu

Use Case UC-2:	KayitOl
Related Requirements:	REQ-1: Sistem güvenli oturum açma özelliğine sahip olmalıdır ve parametrelerin güncellenmesine izin vermeli.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, uygulamaya yeni bir hesap oluşturmak istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none">1. Kullanıcı daha önce uygulamaya kayıt olmamış olmalıdır.2. Kullanıcı uygulamayı açmış olmalıdır.
Postconditions:	<ol style="list-style-type: none">1. Kullanıcı başarılı bir şekilde yeni bir hesap oluşturmuş olmalıdır ve oturum açmış olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none">1. Kullanıcı uygulamayı açar.2. Kullanıcı "Kayıt Ol" seçeneğine tıklar.3. Sistem, kullanıcıdan gerekli bilgileri (örneğin: ad, soyad, e-posta, şifre) girmesini ister.4. Kullanıcı, gerekli bilgileri girer.5. Sistem, kullanıcının girdiği bilgileri kontrol eder ve eğer uygunsa yeni bir hesap oluşturur.6. Sistem, kullanıcıya başarıyla kaydolduğuna dair bir bildirim gönderir.7. Kullanıcı, otomatik olarak uygulamaya yönlendirilir ve oturum açma işlemi tamamlanır.
Flow of Events for Extensions (Alternate Scenarios):	<p>5a. Kullanıcı, gerekli bilgileri eksik veya hatalı girerse:</p> <ol style="list-style-type: none">1. Sistem, kullanıcıyı doğru bilgileri girmesi konusunda bilgilendirir.2. Kullanıcı, adım 3'ten başlayarak işlemi tekrarlar. <p>5b. Kullanıcı, daha önce kullanılan bir e-posta adresini kullanmaya çalışırsa:</p> <ol style="list-style-type: none">1. Sistem, bu e-posta adresinin zaten kayıtlı olduğunu bildirir.2. Kullanıcı, yeni bir e-posta adresi girmesi konusunda bilgilendirilir ve adım 3'ten başlayarak işlemi tekrarlar.

5.3.3 Use Case UC-3: CihazEkle

Tablo 5.3.3: Use Case UC-3: CihazEkle Tablosu

Use Case UC-3:	CihazEkle
Related Requirements:	REQ-7: Uygulama aracılığıyla armatürlerin yeni cihaz eklenmesi, çalışma zamanlarını ve modlarını kullanıcının düzenleyebilmesi.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, akıllı ev sistemine yeni bir fotoselli armatür eklemek istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none">1. Kullanıcı uygulamaya giriş yapmış olmalıdır.2. Kullanıcı daha önce uygulama üzerinden aynı cihazı eklememiş olmalıdır.
Postconditions:	<ol style="list-style-type: none">1. Kullanıcı, başarılı bir şekilde yeni bir cihazı sistemine eklemiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none">1. Kullanıcı uygulamayı açar.2. Kullanıcı "Cihaz Ekle" seçeneğine tıklar.3. Sistem, kullanıcıya yeni cihazın eklenebilmesi için bir yönergeler listesi gösterir (örneğin: QR kodunu okutma).4. Kullanıcı, talimatları takip ederek yeni cihazı sisteme ekler.5. Sistem, yeni cihazı başarıyla sisteme eklediğini bildirir.6. Kullanıcı, otomatik olarak uygulamaya yönlendirilir ve cihaz ekleme işlemi tamamlanır.
Flow of Events for Extensions (Alternate Scenarios):	<p>4a. Kullanıcı, talimatları takip edemez veya QR kodu okutamazsa:</p> <ol style="list-style-type: none">1. Sistem, alternatif bir cihaz ekleme yöntemi sunar (örneğin: manuel olarak seri numarasını girmek).2. Kullanıcı, alternatif yöntemi kullanarak adım 4'e devam eder. <p>4b. Kullanıcı, daha önce eklenmiş bir cihazı tekrar eklemeye çalışırsa:</p> <ol style="list-style-type: none">1. Sistem, bu cihazın zaten ekli olduğunu bildirir.2. Kullanıcı, yeni bir cihaz eklemesi konusunda bilgilendirilir ve adım 3'e geri döner.

5.3.4 Use Case UC-4: CihazYonet

Tablo 5.3.4: Use Case UC-4: CihazYonet Tablosu

Use Case UC-4:	CihazYonet
Related Requirements:	REQ-7: Uygulama aracılığıyla armatürlerin yeni cihaz eklenmesi, çalışma zamanlarını ve modlarını kullanıcının düzenleyebilmesi.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, akıllı ev sistemine eklenmiş fotoselli armatürleri kontrol etmek ve yönetmek istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none">1. Kullanıcı uygulamaya giriş yapmış olmalıdır.2. Kullanıcı, sistemine en az bir fotoselli armatür eklemiş olmalıdır.
Postconditions:	<ol style="list-style-type: none">1. Kullanıcı, başarılı bir şekilde fotoselli armatürleri kontrol etmiş ve yönetmiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none">1. Kullanıcı uygulamayı açar.2. Kullanıcı "Cihazları Yönet" seçeneğine tıklar.3. Sistem, kullanıcıya eklenmiş olan tüm fotoselli armatürleri listeleterek gösterir.4. Kullanıcı, kontrol etmek istediği armatürü seçer.5. Sistem, kullanıcıya seçilen armatürü kontrol etme ve yönetme seçeneklerini sunar (açma, kapama, su tüketimini kontrol etme, birim değiştirme vb.).6. Kullanıcı, istediği işlemi seçer ve uygular.7. Sistem, kullanıcıya işlemin başarıyla gerçekleştiğini bildirir.
Flow of Events for Extensions (Alternate Scenarios):	<p>5a. Kullanıcı, kontrol etmek istediği armatürü seçmezse:</p> <ol style="list-style-type: none">1. Sistem, kullanıcıya en az bir armatür seçmesi gerektiğini bildirir.2. Kullanıcı, adım 4'e geri döner. <p>6a. Kullanıcı, belirli bir işlemi gerçekleştiremezse (örneğin: armatür arızalı ise):</p> <ol style="list-style-type: none">1. Sistem, kullanıcıya işlemin gerçekleştirilemediğini bildirir.2. Kullanıcı, alternatif bir işlem seçeneğini değerlendirir veya destek almak için yardım talebinde bulunur.

5.3.5 Use Case UC-5: ArızaBildir

Tablo 5.3.5: Use Case UC-5: ArızaBildir Tablosu

Use Case UC-5:	ArızaBildir
Related Requirements:	REQ-4: Armatürlerde oluşabilecek su kaçağı durumlarını tespit edip anlık bildirim gönderilmesi.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, sisteme eklenmiş olan fotoselli armatürlerde oluşan arıza veya su kaçağı durumlarını bilmek istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none">1. Kullanıcı uygulamaya giriş yapmış olmalıdır.2. Kullanıcı, sistemine en az bir fotoselli armatür eklemiş olmalıdır.3. Sistem, arıza veya su kaçağı durumunu kullanıcıya bildirim ile iletmış olmalıdır.
Postconditions:	<ol style="list-style-type: none">1. Kullanıcı, başarılı bir şekilde arıza veya su kaçağı durumunu öğrenmiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none">1. Kullanıcı uygulamayı açar.2. Sistem, arıza tespiti yapılan armatür için kullanıcıya arıza türünü ileten bir bildirim gönderir.3. Kullanıcı, arıza türüne (su kaçağı, elektrik arızası vb.) göre alınacak aksiyonu uygulama üzerinden belirtir.4. Sistem, kullanıcıya arıza için alınan aksiyon için bilgilendirme yapar.5. Sistem, kullanıcının çözemeyeceği arıza tespitleri için servise ilgili arızayı iletilmiş olduğunu bildirir.
Flow of Events for Extensions (Alternate Scenarios):	<p>2a. Kullanıcı, sistemden gelen bildirim kaçırır veya yanlış anlarsa:</p> <ol style="list-style-type: none">1. Sistem, bildirim kullanıcı tarafından görülmediğini veya yanlış anlaşıldığını tespit eder.2. Sistem, bildirim tekrar gönderir veya kullanıcıya arızanın detaylarını sormak için bir iletişim başlatır. <p>3a. Kullanıcı, arıza türüne uygun olmayan bir aksiyon seçerse:</p> <ol style="list-style-type: none">1. Sistem, uygun olmayan aksiyon seçimi konusunda kullanıcıyı uyarır.

	<p>2. Kullanıcı, doğru aksiyonu seçerek adıma devam eder.</p> <p>4a. Sistem, servise iletilemeyen arızayı tespit ederse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıya arızayı çözemeyeceğini ve servise ilettiğini bildirir. 2. Kullanıcıya, servis iletişim bilgileri ve süreci hakkında bilgi verilir.
--	---

5.3.6 Use Case UC-6: BirimYonet

Tablo 5.3.6: Use Case UC-6: BirimYonet Tablosu

Use Case UC-6:	BirimYonet
Related Requirements:	REQ-6: Fotoselli armatürlerin enerji tüketimini anlık olarak izleyerek enerji verimliliğinin artırılması.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, sisteme eklenmiş olan fotoselli armatürlerin birim (örneğin, enerji tüketimi) parametrelerini yönetmek ve izlemek istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none"> 1. Kullanıcı uygulamaya giriş yapmış olmalıdır. 2. Kullanıcı, sistemine en az bir fotoselli armatür eklemiş olmalıdır.
Postconditions:	<ol style="list-style-type: none"> 1. Kullanıcı, başarılı bir şekilde birim parametrelerini yönetmiş ve izlemiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none"> 1. Kullanıcı uygulamayı açar. 2. Kullanıcı "Birimleri Yönet" seçeneğine tıklar. 3. Sistem, kullanıcıya eklenmiş olan tüm fotoselli armatür birimlerini (örneğin, enerji tüketimi) listeleterek gösterir. 4. Kullanıcı, izlemek veya yönetmek istediği birimi seçer. 5. Sistem, seçilen birimin detaylarını (örneğin, anlık enerji tüketimi) kullanıcıya gösterir. 6. Kullanıcı, birim parametrelerini istediği şekilde yönetir veya izler. 7. Sistem, kullanıcıya işlemin başarıyla gerçekleştiğini bildirir.

Flow of Events for Extensions (Alternate Scenarios):	<p>4a. Kullanıcı, birim seçmezse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıya en az bir birim seçmesi gerektiğini bildirir. 2. Kullanıcı, adım 3'e geri döner. <p>6a. Kullanıcı, birim parametrelerini yanlış yönetirse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıyı uyarır ve doğru parametreleri girmesi konusunda bilgilendirir. 2. Kullanıcı, doğru parametreleri girmek için adım 6'ya geri döner.
--	---

5.3.7 Use Case UC-7: LimitYonet

Tablo 5.3.7: Use Case UC-7: LimitYonet Tablosu

Use Case UC-7:	LimitYonet
Related Requirements:	REQ-7: Uygulama aracılığıyla armatürlerin yeni cihaz eklenmesi, çalışma zamanlarını ve modlarını kullanıcının düzenleyebilmesi.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, sisteme eklenmiş olan fotoselli armatürlerin limit (örneğin, su tüketim limiti) parametrelerini yönetmek ve izlemek istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none"> 1. Kullanıcı uygulamaya giriş yapmış olmalıdır. 2. Kullanıcı, sistemine en az bir fotoselli armatür eklemiş olmalıdır.
Postconditions:	<ol style="list-style-type: none"> 1. Kullanıcı, başarılı bir şekilde limit parametrelerini yönetmiş ve izlemiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none"> 1. Kullanıcı uygulamayı açar. 2. Kullanıcı "Limitleri Yönet" seçeneğine tıklar. 3. Sistem, kullanıcıya eklenmiş olan tüm fotoselli armatür limit parametrelerini listeleterek gösterir. 4. Kullanıcı, izlemek veya yönetmek istediği limiti seçer. 5. Sistem, seçilen limitin detaylarını (örneğin, su tüketim limiti, sızıntı tespit limiti vb.) kullanıcıya gösterir. 6. Kullanıcı, limit parametrelerini istediği şekilde yönetir veya izler.

	7. Sistem, kullanıcıya işlemin başarıyla gerçekleştiğini bildirir.
Flow of Events for Extensions (Alternate Scenarios):	<p>4a. Kullanıcı, uyarılar için limit seçmezse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıya en az bir limit seçmesi gerektiğini bildirir. 2. Kullanıcı, limit belirlemediği takdirde varsayılan değer olarak sistem tarafından seçilir ve kullanıcıya değer bildirilir. 3. Kullanıcı, adım 3'e geri döner. <p>6a. Kullanıcı, limit parametrelerini yanlış yönetirse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıyı uyarır ve doğru parametreleri seçilmesi konusunda bilgilendirir. 2. Kullanıcı, doğru parametreleri girmek için adım 6'ya geri döner.

5.3.8 Use Case UC-8: EylemYonet

Tablo 5.3.8: Use Case UC-8: EylemYonet Tablosu

Use Case UC-8:	EylemYonet
Related Requirements:	REQ-7: Uygulama aracılığıyla armatürlerin yeni cihaz eklenmesi, çalışma zamanlarını ve modlarını kullanıcının düzenleyebilmesi.
Initiating Actor:	Kullanıcı
Actor's Goal:	Kullanıcı, sisteme eklenmiş olan fotoselli armatürlerin eylemlerini (örneğin, uyarı bildirimleri gönderme, cihazı kapatma, kullanıcı yönetiminde hiçbir şey yapılmaması) yönetmek ve izlemek istiyor.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none"> 1. Kullanıcı uygulamaya giriş yapmış olmalıdır. 2. Kullanıcı, sistemine en az bir fotoselli armatür eklemiş olmalıdır.
Postconditions:	<ol style="list-style-type: none"> 1. Kullanıcı, başarılı bir şekilde armatür eylemlerini yönetmiş ve izlemiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none"> 1. Kullanıcı uygulamayı açar. 2. Kullanıcı "Eylemleri Yönet" seçeneğine tıklar. 3. Sistem, kullanıcıya eklenmiş olan tüm fotoselli armatür eylemlerini listeleterek gösterir. 4. Kullanıcı, izlemek veya yönetmek istediği eylemi seçer. 5. Sistem, seçilen eylemin detaylarını (örneğin, uyarı bildirimleri gönderme, cihazı kapatma, kullanıcı

	<p>yönetiminde hiçbir şey yapılmaması) kullanıcıya gösterir.</p> <p>6. Kullanıcı, eylemi istediği şekilde yönetir.</p> <p>7. Sistem, kullanıcıya işlemin başarıyla gerçekleştiğini bildirir.</p>
Flow of Events for Extensions (Alternate Scenarios):	<p>4a. Kullanıcı, eylem seçmezse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıya en az bir eylem seçmesi gerektiğini bildirir. 2. Kullanıcı, eylem seçmediği takdirde varsayılan değer olarak sistem tarafından seçilir ve kullanıcıya eylem bildirilir. 3. Kullanıcı, adım 3'e geri döner. <p>6a. Kullanıcı, eylemi yanlış yönetirse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıyı uyarır ve doğru eylemi gerçekleştirmesi konusunda bilgilendirir. 2. Kullanıcı, doğru eylemi gerçekleştirmek için adım 6'ya geri döner.

5.3.9 Use Case UC-9: SesliKomut

Tablo 5.3.9: Use Case UC-9: SesliKomut Tablosu

Use Case UC-9:	SesliKomut
Related Requirements:	REQ-8: Sesli komut ile fotoselli armatürlerden istenilen miktarda su akışının sağlanabilmesi.
Initiating Actor:	Kullanıcı ve Siri (Sesli Asistan)
Actor's Goal:	Kullanıcı, fotoselli armatürler üzerinden sesli komut kullanarak su akışını kontrol etmek istiyor.
Participating Actors:	Sistem, Fotoselli Armatürler
Preconditions:	<ol style="list-style-type: none"> 1. Kullanıcı, Siri ile sesli komutları algılayabilen bir cihaza sahip olmalıdır. 2. Kullanıcı, uygulamaya giriş yapmış olmalıdır. 3. Fotoselli armatürler, sesli komutları algılayabilecek bir konfigürasyona sahip olmalıdır.
Postconditions:	<ol style="list-style-type: none"> 1. Kullanıcı, Siri ile başarılı bir şekilde fotoselli armatürler üzerinden su akışını kontrol etmiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none"> 1. Kullanıcı, Siri ile sesli komutu verir (örneğin, "Mutfak bataryasından 1 bardak su ver.") 2. Sistem, sesli komutu algılar ve doğru armatürü belirler. 3. Sistem, belirlenen armatürdeki su akışını kullanıcının isteğine uygun olarak açar veya kapatır.

	4. Sistem, kullanıcıya Siri'yi kullanarak işlemin başarıyla gerçekleştiğini sesli olarak bildirir.
Flow of Events for Extensions (Alternate Scenarios):	<p>1a. Kullanıcı, Siri'ye anlaşılabilir veya geçersiz bir komut verirse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıya Siri ile komutun anlaşılmadığını veya geçersiz olduğunu sesli olarak bildirir. 2. Kullanıcı, Siri'ye doğru ve anlaşılabilir bir komut verene kadar adım 1'e geri döner. <p>3a. Sistem, belirlenen armatürde su akışını kontrol edemezse:</p> <ol style="list-style-type: none"> 1. Sistem, kullanıcıya su akışını kontrol edemediğini bildirir. 2. Kullanıcı, doğru komutu söyleyip sorunu çözüne kadar adım 1'e geri döner.

5.3.10 Use Case UC-10: HataAyıkla

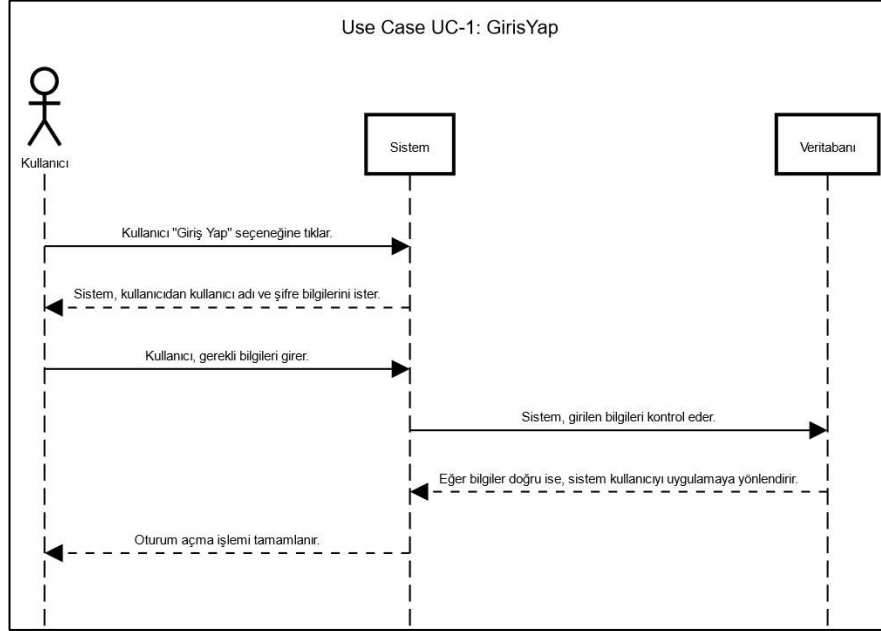
Tablo 5.3.10: Use Case UC-10: HataAyıkla Tablosu

Use Case UC-10:	HataAyıkla
Related Requirements:	REQ-6: Merkezi kontrol biriminin %99.99 sürekli çalışabilirliğinin sağlanması.
Initiating Actor:	Yönetici
Actor's Goal:	Yönetici, sistemde meydana gelen hataları tespit ederek ve gidererek sistem performansını iyileştirmek ister.
Participating Actors:	Sistem
Preconditions:	<ol style="list-style-type: none"> 1. Yönetici uygulamaya giriş yapmış olmalıdır. 2. Sistemde bir hata veya performans sorunu meydana gelmiş olmalıdır.
Postconditions:	<ol style="list-style-type: none"> 1. Yönetici, başarılı bir şekilde hata veya performans sorunlarını tespit etmiş ve gidermiş olmalıdır.
Flow of Events for Main Success Scenario:	<ol style="list-style-type: none"> 1. Yönetici uygulamayı açar. 2. Yönetici, "Hataları Ayıkla" seçeneğine tıklar. 3. Sistem, mevcut hataları ve performans sorunlarını listelerek gösterir. 4. Yönetici, tespit ettiği bir hata veya soruna odaklanır ve sorunun detaylarını inceleyerek çözümü arar. 5. Yönetici, sorunu çözmek için gerekli aksiyonları alır. 6. Sistem, yöneticiye sorunun başarıyla çözüldüğünü bildirir.

<p>Flow of Events for Extensions (Alternate Scenarios):</p>	<p>3a. Sistem, mevcut hataları ve performans sorunlarını tespit edemezse:</p> <ol style="list-style-type: none">1. Sistem, yöneticiye bir hata olmadığını veya tespit edilemeyen bir performans sorunu olduğunu bildirir.2. Yönetici, sistemi daha ayrıntılı bir şekilde kontrol etmek için adım 1'e geri döner. <p>5a. Yönetici, sorunu çözmek için gerekli aksiyonları alamazsa:</p> <ol style="list-style-type: none">1. Sistem, yöneticiye çözüm adımları konusunda yardımcı olur ve alternatif çözüm yollarını önerir.2. Yönetici, sorunu çözmek için alternatif çözüm yollarını denemek üzere adım 5'e geri döner.
---	--

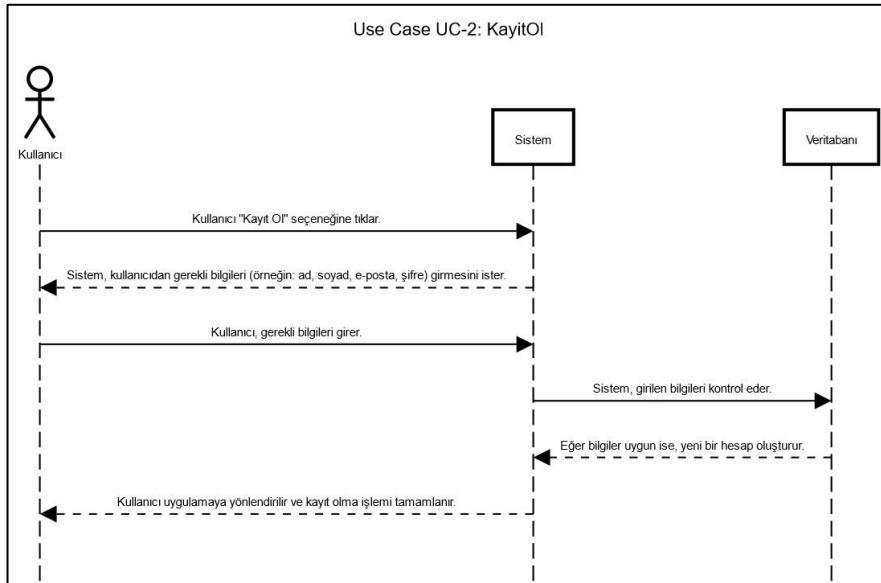
5.4 Sistem Sequence Diyagramları (SSD)

5.4.1 SSD-1: GirişYap



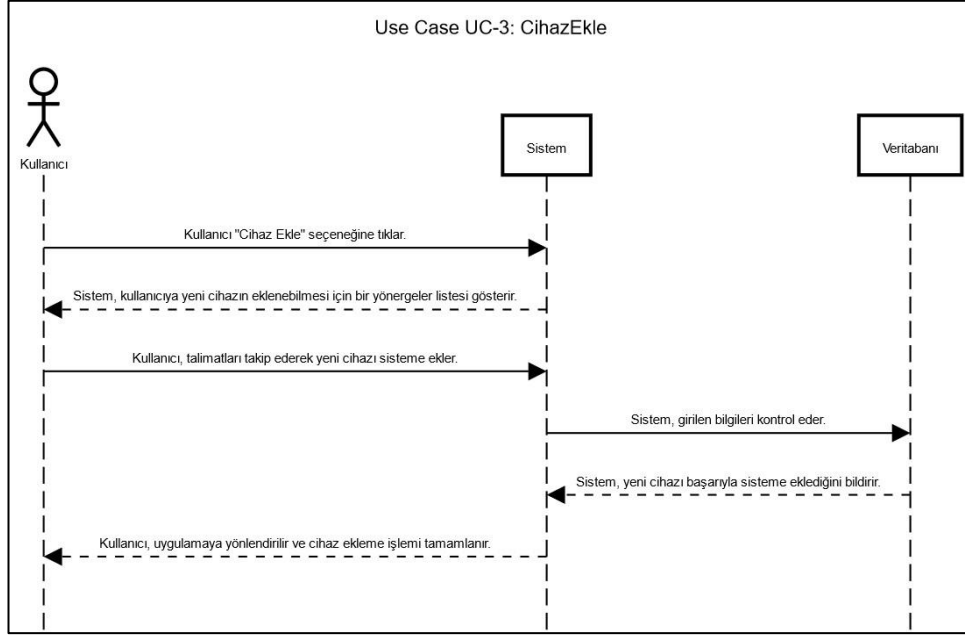
Şekil 5.4.1: SSD-1: GirişYap Diagram

5.4.2 SSD-2: KayıtOl



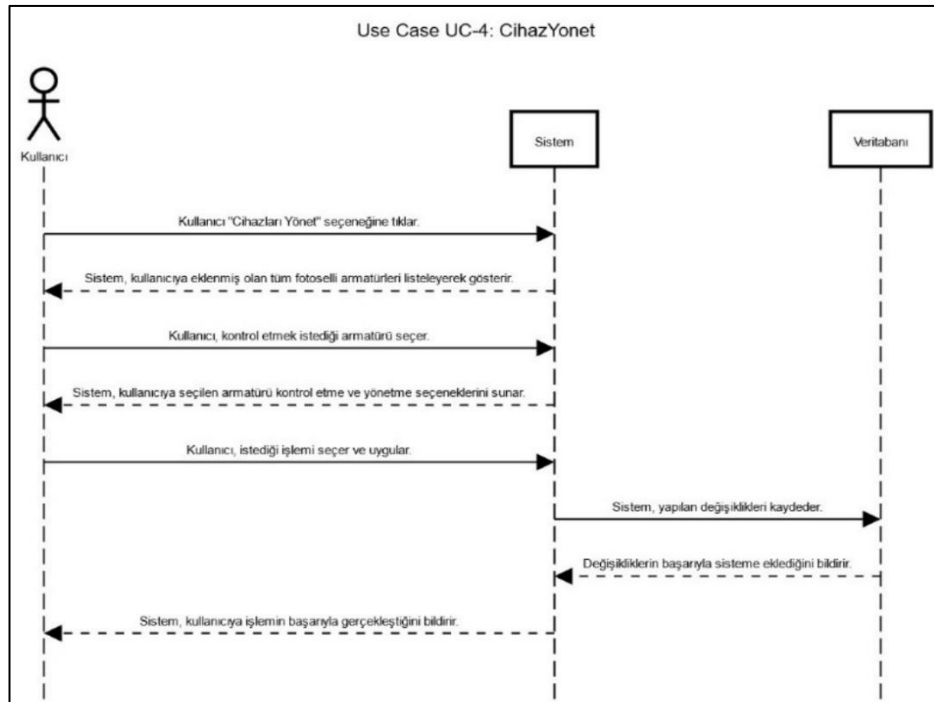
Şekil 5.4.2: SSD-2: KayıtOl Diagram

5.4.3 SSD-3: CihazEkle



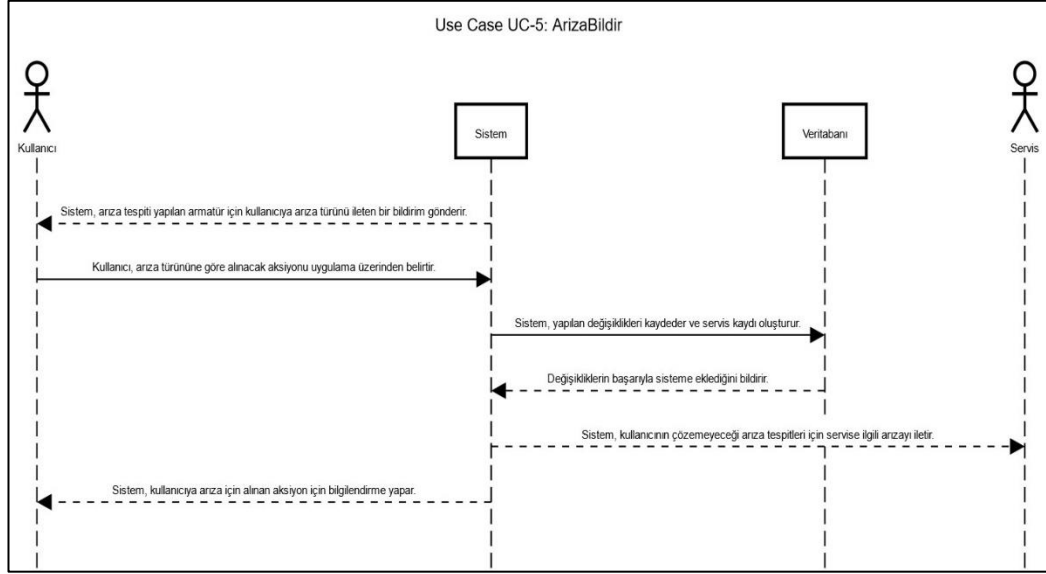
Şekil 5.4.3: SSD-3: CihazEkle Diagram

5.4.4 SSD-4: CihazYonet



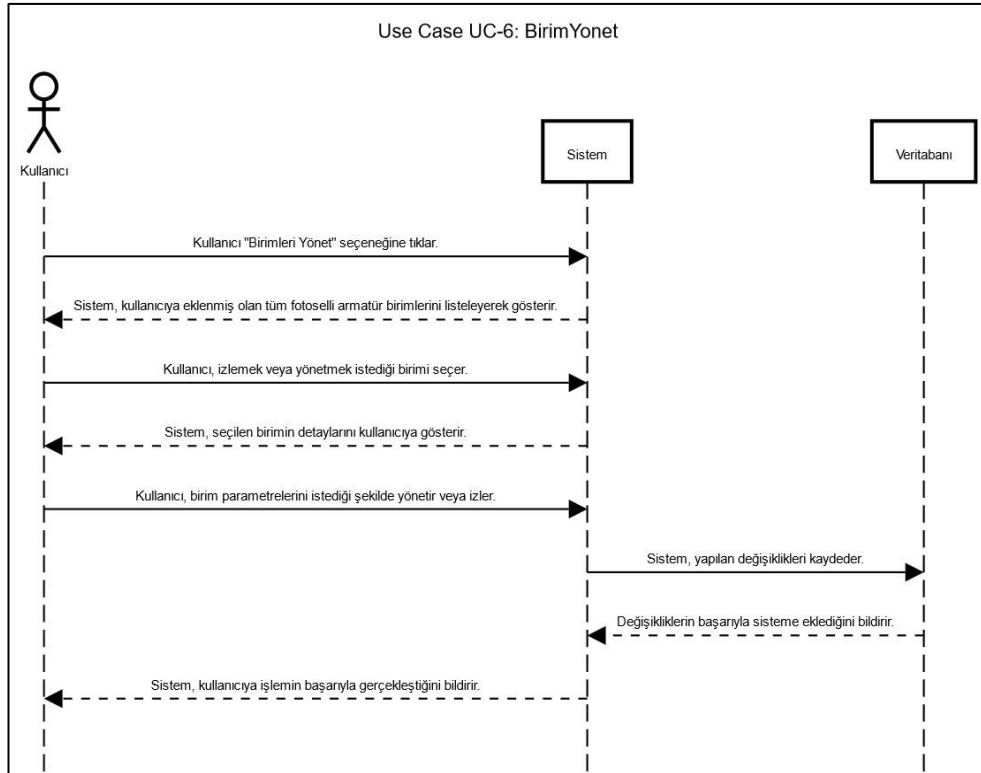
Şekil 5.4.4: SSD-4: CihazYonet Diagram

5.4.5 SSD-5: ArizaBildir



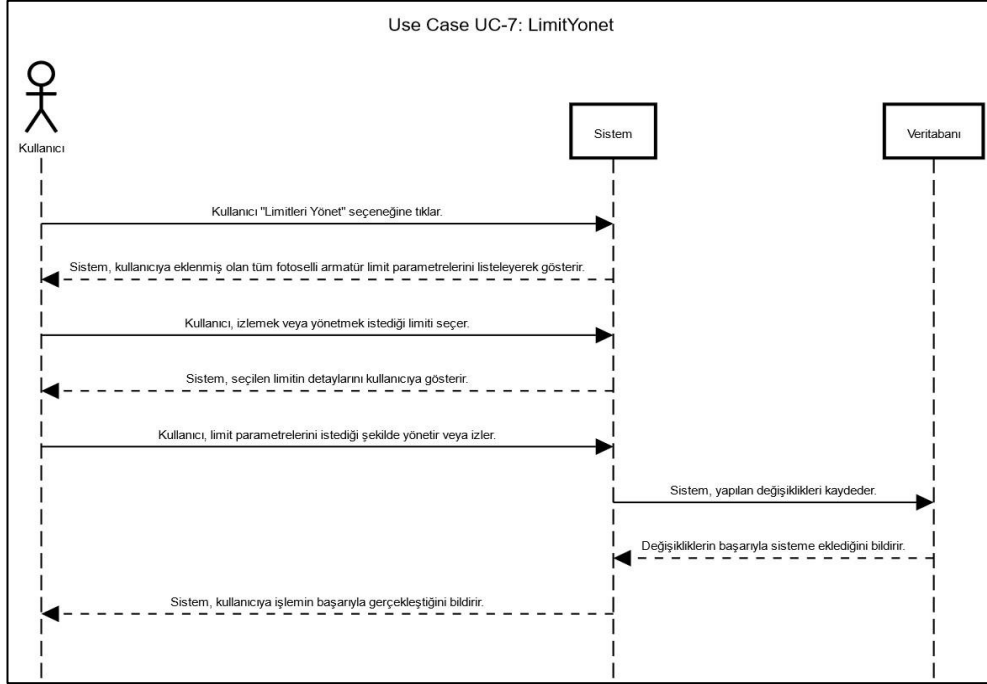
Şekil 5.4.5: SSD-5: ArizaBildir Diagram

5.4.6 SSD-6: BirimYonet



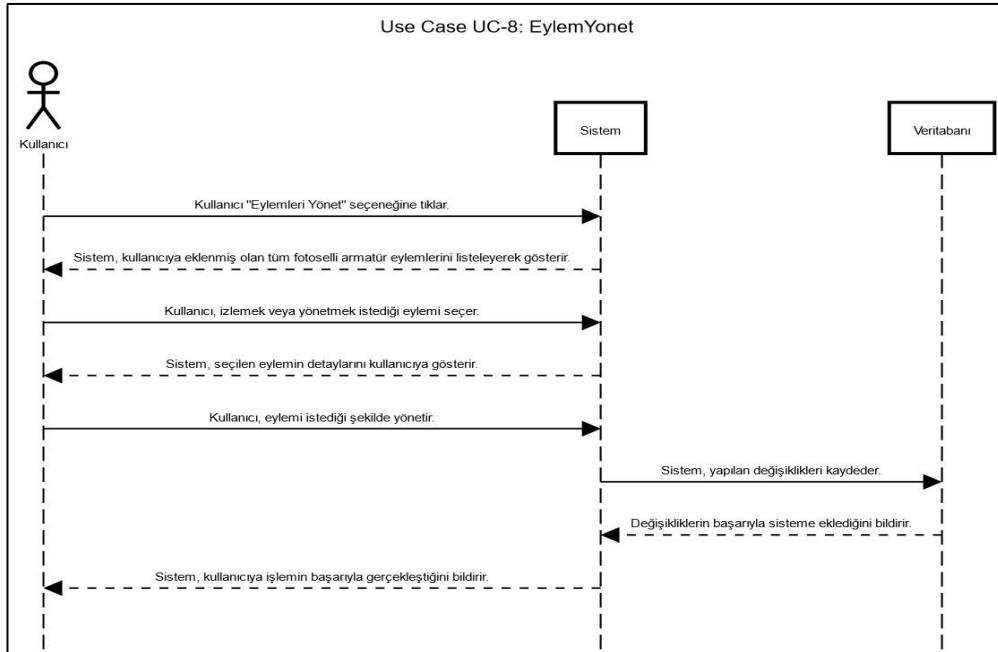
Şekil 5.4.6: SSD-6: BirimYonet Diagram

5.4.7 SSD-7: LimitYonet



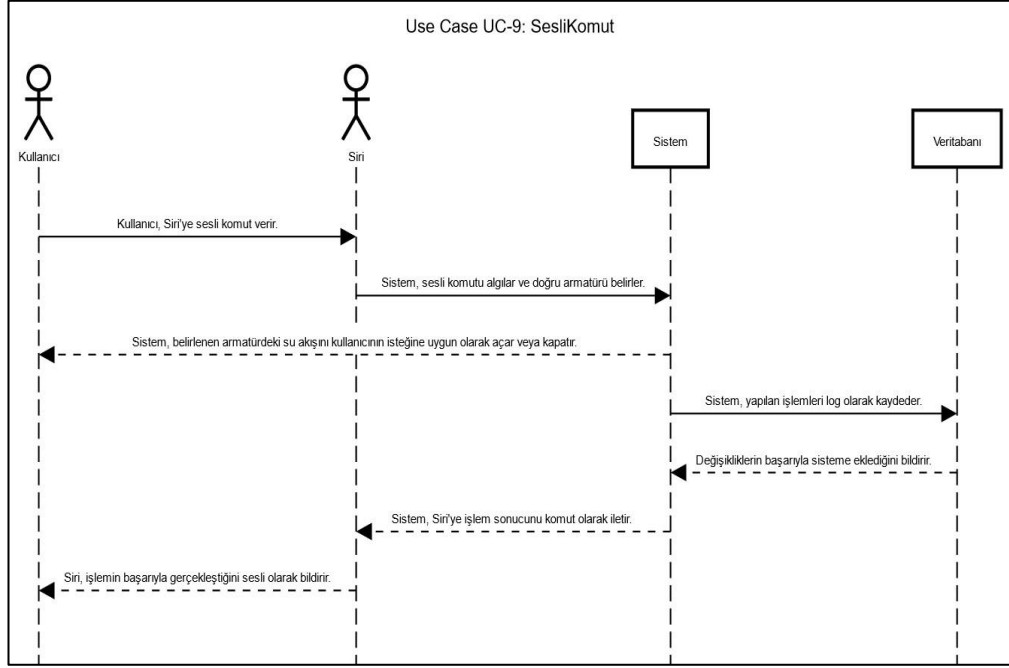
Şekil 5.4.7: SSD-7: LimitYonet Diagram

5.4.8 SSD-8: EylemYonet



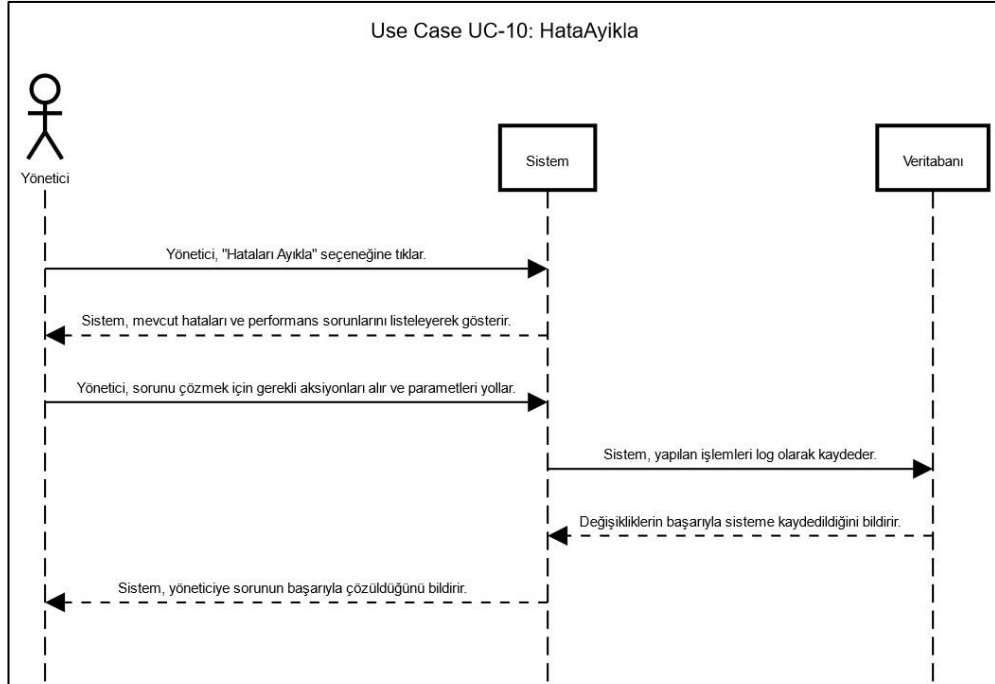
Şekil 5.4.8: SSD-8: EylemYonet Diagram

5.4.9 SSD-9: SesliKomut



Şekil 5.4.9: SSD-9: SesliKomut Diagram

5.4.10 SSD-10: HataAyıkla



Şekil 5.4.10: SSD-10: HataAyıkla Diagram

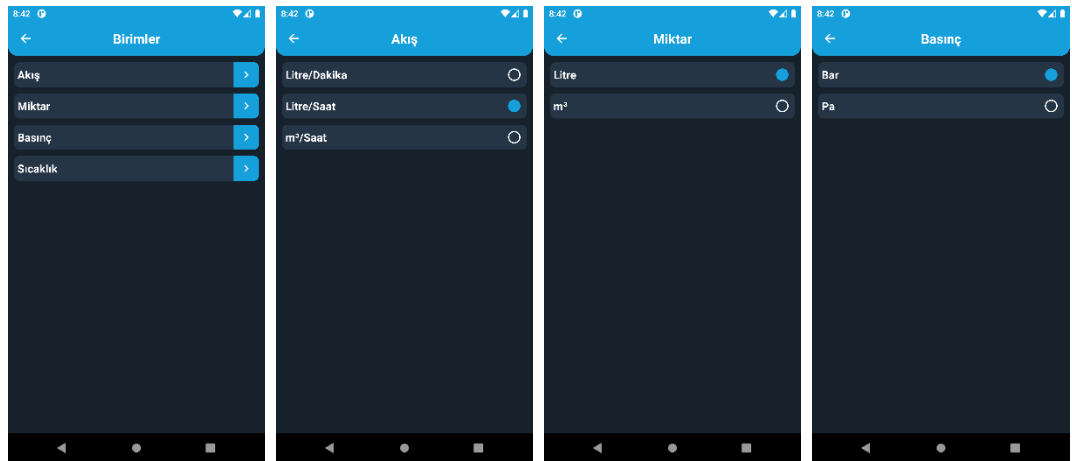
Bölüm 6

Uygulama Ekran Görüntüleri

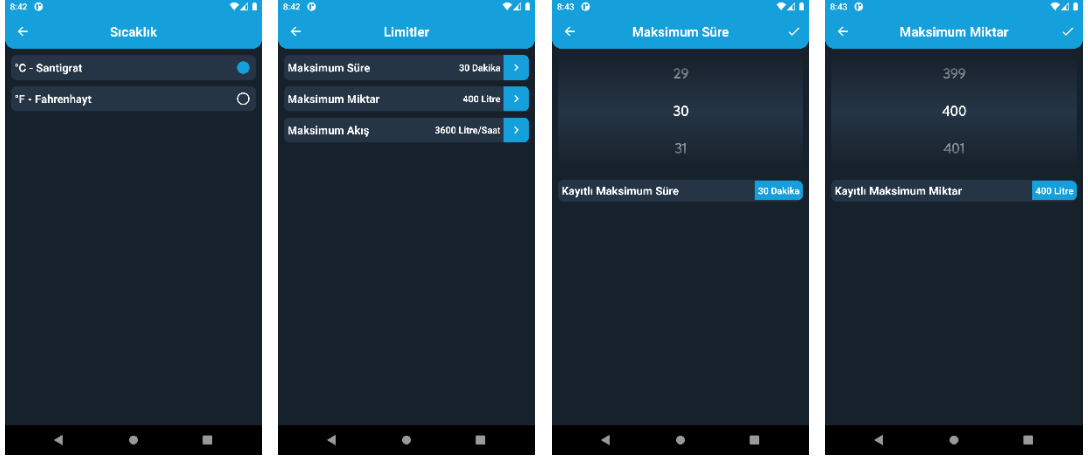
6.1 Android Ekran Görüntüleri



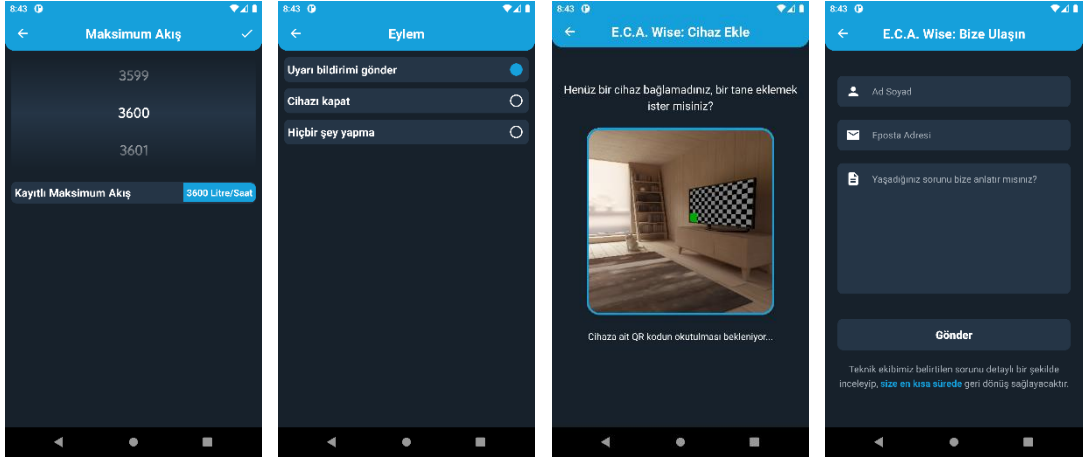
Şekil 6.1.1: Android Ana Ekran Sekmeleri Ekran Görüntüleri



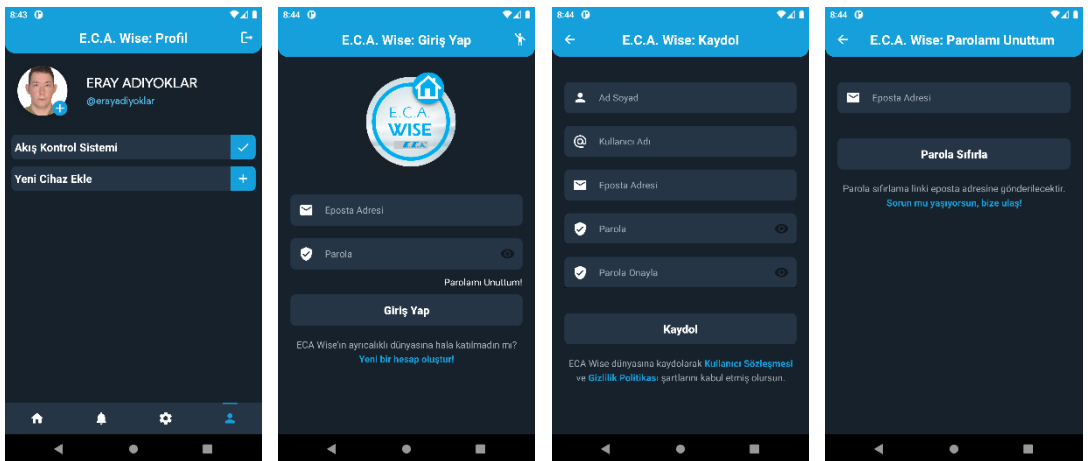
Şekil 6.1.2: Android Ayarlar Sekmesi Birim Düzenleme Ekran Görüntüleri-1



Şekil 6.1.3: Android Ayarlar Sekmesi Birim Düzenleme Ekran Görüntüleri-2

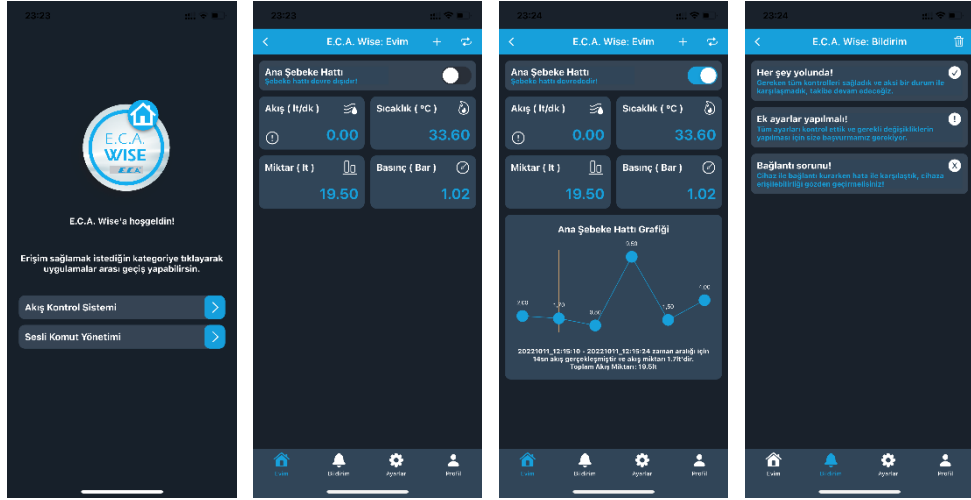


Şekil 6.1.4: Android Cihaz Ekleme ve Servis Kayıt Oluşturma Ekran Görüntüleri

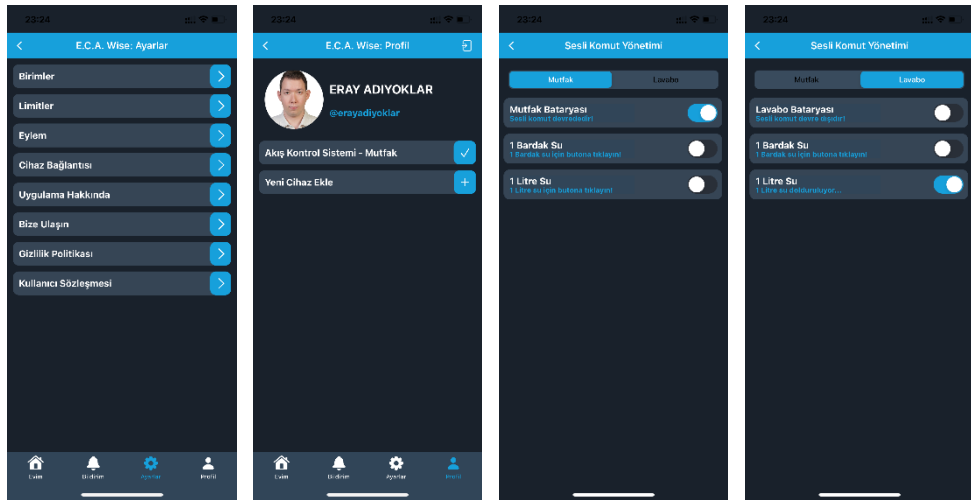


Şekil 6.1.5: Android Profil, Giriş Yap, Kayıt Ol Ekran Görüntüleri

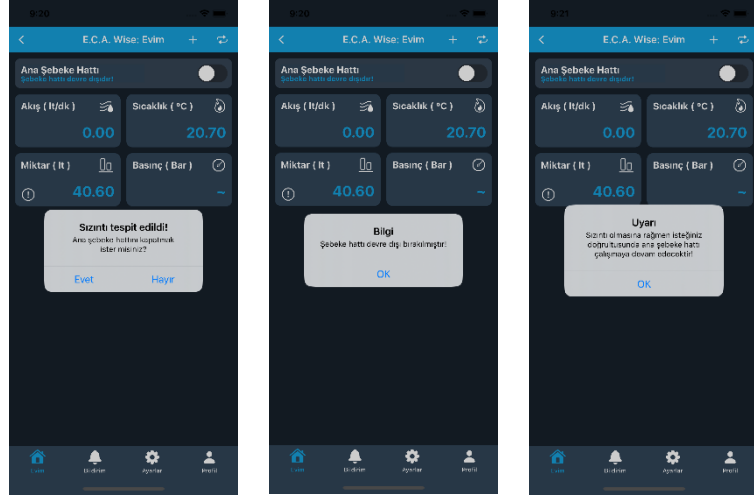
6.2 Ios Ekran Görüntüleri



Şekil 6.2.1: Ios Ana Ekran Sekmeleri Ekran Görüntüleri

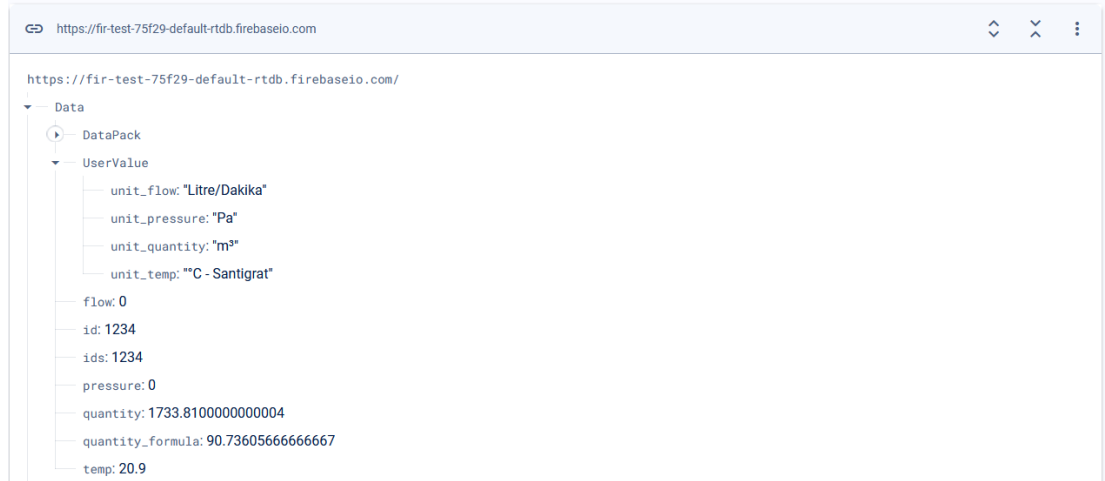


Şekil 6.2.2: Ios Ayarlar, Profil, Sesli Komut Ekran Görüntüleri

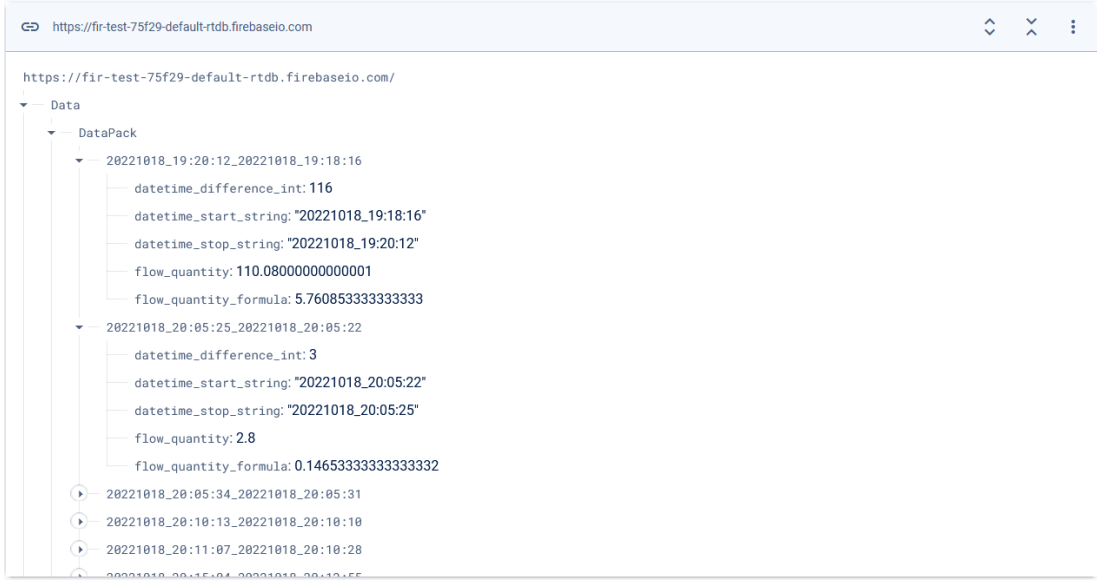


Şekil 6.2.3: Ios Arıza Tespit ve Bildirim Örnek Ekran Görüntüleri

6.3 Firebase Veritabanı Ekran Görüntüleri

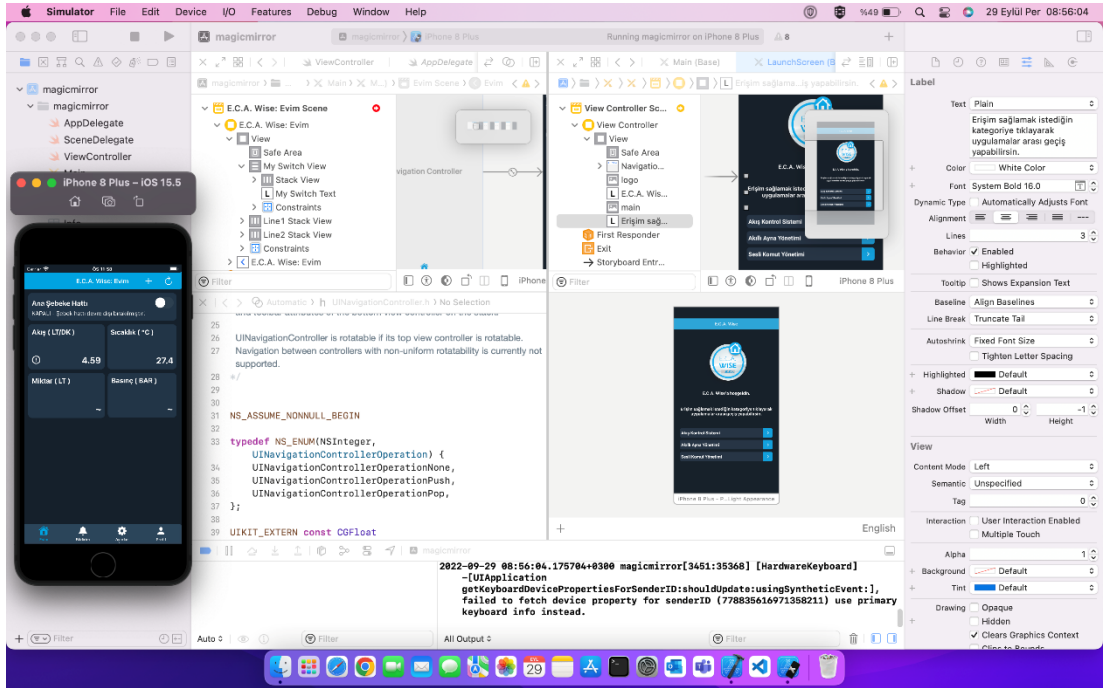


Şekil 6.3.1: Firebase Parametre Yönetim Ekran Görüntüsü



Şekil 6.3.2: Firebase Geçmiş Kayıt Log Görüntüleme Ekran Görüntüsü

6.4 Geliştirme Ortamı Ekran Görüntüleri



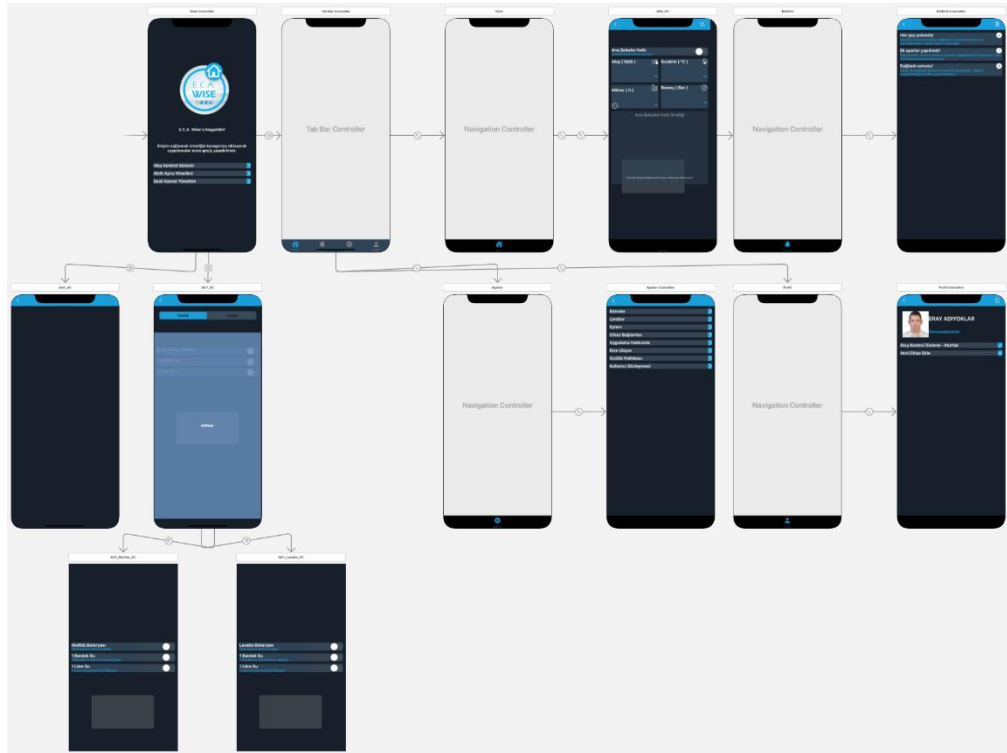
Şekil 6.4.1: Xcode Geliştirme Ortamı Ekran Görüntüsü

Key	Type	Value
Information Property List (15 items)		
CLIENT_ID	String	67744237135-cltavg2e386utvui5tuj18ipc6mcb19o.apps.googleusercontent.com
REVERSED_CLIENT_ID	String	com.googleusercontent.apps.67744237135-cltavg2e386utvui5tuj18ipc6mcb19o
API_KEY	String	AlzaSyCzcZ140JyVfY8QszL_u6T40V6nNCbBvKs
GCM_SENDER_ID	String	67744237135
PLIST_VERSION	String	1
BUNDLE_ID	String	com.ecatg.tubitak
PROJECT_ID	String	fir-test-75f29
STORAGE_BUCKET	String	fir-test-75f29.appspot.com
IS_ADS_ENABLED	Boolean	NO
IS_ANALYTICS_ENABLED	Boolean	NO
IS_APPINVITE_ENABLED	Boolean	YES
IS_GCM_ENABLED	Boolean	YES
IS_SIGNIN_ENABLED	Boolean	YES
GOOGLE_APP_ID	String	1:67744237135:ios:ef60408dcea1a3b02a8c3a
DATABASE_URL	String	https://fir-test-75f29-default-rtdb.firebaseio.com

Şekil 6.4.2: Google Firebase Servis Ekran Görüntüsü

Name	Location	Dependency Rule
Firestore	https://github.com/firebase/firebase-ios-sdk	Up to Next Major Version 9.0.0 < 10.0.0
Charts	https://github.com/danielgindi/Charts	Up to Next Major Version 3.0.0 < 4.0.0

Şekil 6.4.3: Ios Uygulama Kullanılan Kütüphane Paketleri Ekran Görüntüsü



Şekil 6.4.4: XCode Uygulama Arayüz Tasarım Özet Ekranı Şeması


```

1 //
2 // DataService.swift
3 // kossb
4 //
5 // Created by Eray ADIYOKLAR on 19.08.2022.
6 //
7
8 import Foundation
9 import Firebase
10
11 let BASE_URL = "https://fir-test-75f29-default-rtdb.firebaseio.com/"
12
13 class DataService {
14
15     let dataService = DataService()
16
17     private var _BASE_REF = Database.database().reference(fromURL: "\(BASE_URL)")
18     private var _ITEM_REF = Database.database().reference(fromURL: "\(BASE_URL)/items")
19
20     var BASE_REF: DatabaseReference {
21         return _BASE_REF
22     }
23
24     var ITEM_REF: DatabaseReference {
25         return _ITEM_REF
26     }
27
28 }

```

Şekil 6.4.5: Google Firebase Servis Veritabanı Servisi Ekran Görüntüsü

```

1 //
2 // Data.swift
3 // tubitak
4 //
5 // Created by Eray ADIYOKLAR on 18.10.2022.
6 //
7
8 import Foundation
9 import Firebase
10
11 class Data {
12
13     var datetime_start_string : String
14     var datetime_stop_string : String
15     var datetime_difference_int : Int
16     var flow_quantity : Double
17     var flow_quantity_formula : Double
18
19     init (datetime_start_string: String, datetime_stop_string: String, datetime_difference_int: Int, flow_quantity: Double, flow_quantity_formula: Double) {
20
21         self.datetime_start_string = datetime_start_string
22         self.datetime_stop_string = datetime_stop_string
23         self.datetime_difference_int = datetime_difference_int
24         self.flow_quantity = flow_quantity
25         self.flow_quantity_formula = flow_quantity_formula
26     }
27
28 }

```

Şekil 6.4.6: Google Firebase Servis Veritabanı Sınıfı Ekran Görüntüsü

```

@IBAction func switchSKYDidChange(_ sender: UISwitch) {
    if sender.isOn {
        lbl_sky.text = "Sesli komut devrededir!"
        webview.load(URLRequest(url: URL(string: "https://maker.ifttt.com/trigger/open_valve/json/with/key/UzNocAXnCFcmRL2Wmpfhm")))
    }
    else {
        lbl_sky.text = "Sesli komut devre dışıdır!"
        webview.load(URLRequest(url: URL(string: "https://maker.ifttt.com/trigger/close_valve/json/with/key/UzNocAXnCFcmRL2Wmpfhm")))
    }
}

@IBAction func switchBirBardakDidChange(_ sender: UISwitch) {
    if sender.isOn {
        lbl_birbardak.text = "1 Bardak su dolduruluyor..."
        Timer.scheduledTimer(withTimeInterval: 4, repeats: false) { (_) in
            self.switch_birbardak.setOn(false, animated: true);
            self.lbl_birbardak.text = "1 Bardak su için butona tıklayın!"
        }
        webview.load(URLRequest(url: URL(string: "https://maker.ifttt.com/trigger/1_glass_water/json/with/key/cPPTFhVhBYTxDyiw_CRDYm")))
    }
    else {
        lbl_birbardak.text = "1 Bardak su için butona tıklayın!"
    }
}

@IBAction func switchBirLitreDidChange(_ sender: UISwitch) {
    if sender.isOn {
        lbl_birlitre.text = "1 Litre su dolduruluyor..."
        Timer.scheduledTimer(withTimeInterval: 10, repeats: false) { (_) in
            self.switch_birlitre.setOn(false, animated: true);
            self.lbl_birlitre.text = "1 Litre su için butona tıklayın!"
        }
        webview.load(URLRequest(url: URL(string: "https://maker.ifttt.com/trigger/1lt_water/json/with/key/cPPTFhVhBYTxDyiw_CRDYm")))
    }
    else {
        lbl_birlitre.text = "1 Litre su için butona tıklayın!"
    }
}
}

```

Şekil 6.4.7: Sesli Komut ile Su Akışı Sağlayan Foksiyon Kodu Ekran Görüntüsü

```

alert.addAction(UIAlertAction(title: "Hayır", style: .default, handler: { action in
    switch action.style{
        case .default:
            arrayListSizinti.removeAll()
            print(arrayListSizinti.count)
            let alertNo = UIAlertController(title: "Uyarı", message: "Sızıntı olmasına rağmen isteğiniz doğrultusunda ana şebeke hattı çalışmaya devam edecektir!",
                preferredStyle: .alert)
            alertNo.addAction(UIAlertAction(title: "OK", style: .default, handler: nil))
            self.present(alertNo, animated: true, completion: nil)

        case .cancel:
            print("cancel")

        case .destructive:
            print("destructive")

        @unknown default:
            print("error")
    }
}))

```

Şekil 6.4.8: Arıza Bildirimi Sızıntı Tespit Edilen Fonksiyon Kodu Ekran Görüntüsü

Bölüm 7

Sonuç ve Bulgular

Bu rapor, akıllı ev sistemlerinde fotoselli armatürlerin entegrasyonunun sağladığı kapsamlı tasarım ve uygulama stratejilerini ortaya koymaktadır. Sistemin sunduğu temel avantajlar arasında enerji verimliliği, su tasarrufu ve artırılmış güvenlik bulunmaktadır. Detaylı bir şekilde ele alınan tasarım özellikleri ve işlevselliği, akıllı ev teknolojilerinin ev sahiplerine nasıl daha etkili hizmet verebileceğini göstermektedir.

7.1 Sistemin Sağladığı Avantajlar

Sistem, fotoselli armatürlerin kullanımı sayesinde enerji verimliliğini artırarak hem kullanıcılara maliyet tasarrufu sağlar hem de çevresel etkiyi minimize eder. Su tüketimini anlık olarak izleyerek, kullanıcıların su kullanım alışkanlıklarını görselleştirmesi ve optimize etmesi mümkün kılar.

Sistemin, fotoselli armatürlerde oluşabilecek su kaçağı durumlarını anlık olarak tespit edip kullanıcıya bildirim gönderme özelliği, ev sahiplerine potansiyel hasarları önleme ve hızlı müdahale imkânı sağlar.

Proaktif bakım süreçleri ile sistem, fotoselli armatürlerde oluşabilecek arızaları önceden tespit ederek, kullanıcılara sorunsuz bir kullanım deneyimi sunar ve cihaz ömrünü uzatır.

Sistem, IoT ve büyük veri kavramlarını kullanarak yaşlı bireylerin güvenliğini artırmayı hedefler. Bu doğrultuda banyoda entegre edilen sensörler sayesinde düşme

veya yaralanma riski taşıyan durumları algılayarak kullanıcılara anlık uyarılar gönderir.

7.2 Gelecek Çalışmalar ve Geliştirmeler

Sistemin gelecekteki güncellemeleri, yapay zekâ (YZ) ve makine öğrenimi (MO) tekniklerinin entegrasyonunu içerebilir. Bu sayede, sistemin kullanıcı alışkanlıklarını daha etkili bir şekilde öğrenmesi ve buna göre önerilerde bulunması mümkün olabilir.

Sistem, çeşitli cihazlar arasında daha kapsamlı bir entegrasyon sağlamak adına farklı akıllı ev teknolojileriyle uyumlu hale getirilebilir. Böylece, ev sahipleri tek bir platform üzerinden evlerini daha etkili bir şekilde yönetebilir. (Örneğin Google Asistan, Alexa)

Enerji verimliliğini artırmak için sistemde kullanılan sensör teknolojileri ve iletişim protokolleri sürekli olarak güncellenebilir. Bu sayede, sistem daha hızlı ve doğru veri toplayabilir, kullanıcıları daha etkili bir şekilde bilgilendirebilir.

Sistem, kullanıcı geri bildirimleri ve pazar talepleri doğrultusunda yeni özellikler ekleyerek ev sahiplerinin ihtiyaçlarına daha iyi yanıt verebilir. Kullanıcı arayüzü ve deneyimini sürekli olarak geliştirmek, sistemin kullanıcı dostu olmasını sağlayabilir.

Sonuç olarak, tasarlanan akıllı ev fotoselli armatür sistemi, kullanıcıların günlük yaşamlarını kolaylaştırmak ve çevresel etkileri azaltmak adına önemli bir adımdır. Gelecekteki çalışmalarla birlikte, sistem daha da geliştirilecek ve akıllı ev teknolojilerindeki yeniliklere uyum sağlayacaktır.

Kaynaklar

- [1] B.L. Risteska Stojkoska, K.V. Trivodaliev, A review of Internet of Things for smart home: challenges and solutions, *J. Clean. Prod.* 140 (2017) 1454–1464,
- [2] W.K. Edwards, R.E. Grinter, in: *At Home with Ubiquitous Computing: Seven Challenges*, Springer, Berlin, Heidelberg, 2001, pp. 256–272,
- [3] L. Douglas. (2001), 3D data management: Controlling data volume, velocity and variety. *Gartner*. Retrieved, 6, 2001.
- [4] IBM (2013), *What is big data? - Bringing big data to the enterprise*.
- [5] Dragos Mocrii, Yuxiang Chen, Petr Musilek, IoT-based smart homes: A review of system architecture, software, communications, privacy and security, *Internet of Things*, Volumes 1–2, 2018, Pages 81-98,
- [6] Paetz, C., 2018. *Z-Wave Essentials*. Christian Paetz. URL: <https://books.google.com/books?id=t80nDwAAQBAJ>.
- [7] Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B., Ideker, T., 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research* 13, 2498–2504.
- [8] ZigBee Alliance, 2008. Zigbee document 053474r17. *ZigBee Specification*, ZigBee Alliance.
- [9] Poulakis, Marios & Vassaki, Stavroula & Pitsiladis, Georgios & Kourogiorgas, Charilaos & Panagopoulos, Athanasios & Gardikis, Georgios & Costicoglou, Socrates. (2016). *Wireless Sensor Network Management Using Satellite Communication Technologies*. 10.1201/b20085-12.

- [10] O. Horyachyy, Comparison of wireless communication technologies used in a smart home: analysis of wireless sensor node based on Arduino in home automation scenario (2017).
- [11] M. Kuzlu, M. Pipattanasomporn, S. Rahman, Review of communication technologies for smart homes/building applications, in: Proceedings of the IEEE Innovative Smart Grid Technologies – Asia (ISGT ASIA), IEEE, 2015, pp. 1–6,
- [12] C. Withanage, R. Ashok, C. Yuen, K. Otto, A comparison of the popular home automation technologies, in: Proceedings of the IEEE Innovative Smart Grid Technologies – Asia (ISGT ASIA), IEEE, 2014, pp. 600–605,
- [13] O. Bello, S. Zeadally, Network layer inter-operation of Device-to-Device communication technologies in Internet of Things (IoT), *Ad Hoc Netw.* 57 (2017) 52–62,
- [14] <https://www.threadgroup.org/news-events/press-releases/ID/20/Introducing-Thread-A-New-Wireless-Networking-Protocol-for-the-Home>
- [15] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, C.-H. Lung, Smart home: integrating internet of things with web services and cloud computing, in: Proceedings of the IEEE Fifth International Conference on Cloud Computing Technology and Science, IEEE, 2013, pp. 317–320,
- [16] S. Guoqiang, C. Yanming, Z. Chao, Z. Yanxu, Design and Implementation of a Smart IoT Gateway, in: Proceedings of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, IEEE, 2013, pp. 720–723,
- [17] Amir Modarresi, John Symons, Modeling and Graph Analysis for Enhancing Resilience in Smart Homes, *Procedia Computer Science*, Volume 160, 2019, Pages 197-205,
- [18] Zhen-ya Liu, Hardware Design of Smart Home System based on zigBee Wireless Sensor Network, *AASRI Procedia*, Volume 8, 2014, Pages 75-81,

[19] Dan D. Koo, John J. Lee, Aleksei Sebastiani, Jonghoon Kim, An Internet-of-Things (IoT) System Development and Implementation for Bathroom Safety Enhancement, Procedia Engineering, Volume 145,2016, Pages 396-403,

[20] <https://developer.android.com/>, Android Resmi Developer Web Sitesi,

[21] <https://developer.apple.com/>, Ios Resmi Developer Web Sitesi,

[22] <https://github.com/>, üzerinde açık kaynaklı projeleri inceleyerek projeye katkı sağlamıştır,

[23] <https://stackoverflow.com/>, geliştiriciler arasında soru-cevap platformu olarak bilinir ve diğer geliştiricilerin çözümlerinden yardım alınmıştır,

[24] <https://medium.com/>, mobil uygulama geliştirme konusunda birçok yazı bulunmaktadır, deneyim paylaşımları ve öğretici makaleler kaynak alınmıştır,

[25] <https://android-developers.googleblog.com/>, Android geliştirme ekibi tarafından paylaşılan güncel bilgileri içerir,

[26] <https://console.firebase.google.com/>, Google'ın sunduğu bir araç setidir ve mobil uygulama geliştirme sürecinde kullanılmıştır,

[27] <https://cocoapods.org/>, Ios projeleri için bir bağımlılık yöneticisidir. 98 binden fazla kütüphaneye sahiptir ve 3 milyondan fazla uygulamada kullanılmaktadır. CocoaPods projelerinizi zarif bir şekilde ölçeklendirmenize yardımcı olabilir,

[28] <https://gradle.org/>, Gradle, mobil uygulamalardan mikro hizmetlere, küçük girişimlerden büyük işletmelere kadar ekiplerin daha iyi yazılımları daha hızlı oluşturmasına, otomatikleştirmesine ve sunmasına yardımcı olur,