

## 1. Giriş ve Proje Tanımı

**GreenTouch (Akıllı Bitki Bakım Sistemi)**; nesnelerin interneti (IoT), bulut bilişim ve mobil yazılım teknolojilerini entegre ederek geleneksel bitki yetiştirciliğini dijitalleştiriren kapsamlı bir biyo-teknoloji çözümüdür.

Günümüzde hızla artan kentleşme ve yoğun çalışma temposu, bireylerin ev içi bitki bakımı süreçlerini aksatmasına neden olmaktadır. "Unutkanlık" veya "yanlış sulama" gibi insan faktörlü hatalar, bitki kayıplarının ve su israfının temel sebebidir. Bu proje, söz konusu problemleri minimize etmek amacıyla; ortam verilerini (nem, sıcaklık, ışık, bitki boyu) analiz eden **ESP8266 tabanlı** bir donanım ünitesi ile kullanıcıya uzaktan kontrol imkânı sunan **Flutter tabanlı** bir mobil arayüzden oluşmaktadır.

Projenin temel hedefleri şunlardır:

- **Sürdürülebilirlik ve Verimlilik:** Sensör verilerine dayalı otonom sulama ile su israfını önlemek ve bitki yaşam döngüsünü uzatmak.
- **Uzaktan Erişim ve Güvenlik:** Firebase altyapısı üzerinden bitkiye dönyanın her yerinden erişim sağlarken, geliştirilen "Doğrulama Kodu" protokolü ile cihaz güvenliğini garanti altına almak.
- **Veriye Dayalı Gelişim:** Geçmişe dönük sıcaklık ve büyümeye verilerini raporlayarak kullanıcıya rehberlik etmek.

**Vizyon ve Hedef Kitle** Projenin vizyonu, evlerdeki bitkileri akıllı ev ekosisteminin "konuşan" bir parçası haline getirmektir. Gelecek projeksiyonunda, veritabanındaki altyapısı hazırlanan **Yapay Zeka (AI)** modülleriyle görüntü işleme tabanlı hastalık tespiti yapılması hedeflenmektedir. Sistem; sık seyahat eden profesyonellerden teknoloji meraklılarına ve profesyonel seralara kadar geniş bir kullanıcı kitlesine hitap etmektedir.

## 2. Destanlar (Epics) ve Hikâye Kartları

Proje backlog'u, karmaşıklığı yönetmek ve geliştirme sürecini modüler hale getirmek amacıyla 3 ana "Destan" (Epic) altında toplanmıştır. Bu destanlar; **(A) Kimlik ve Profil**, **(B) Donanım-Bulut İletişimi** ve **(C) Veri Analizi ve Bildirimler** süreçlerini kapsamaktadır.

## 2. DESTANLAR (EPİCS) VE HİKÂYE KARTLARI

Proje Backlog Panosu - Grup 6

### 2.1. DESTAN A: KİMLİK

#### Hikâye 1: Güvenli Oturum Açma

Rol: Yeni Üye

"Bir kullanıcı olarak, e-posta/şifre veya Google hesabımla sisteme güvenli giriş yapmak istiyorum, çünkü bitki verilerimin başlarını tarafından görülmemesini veya doğrulamasını istemiyorum."

##### Kabul Kriterleri:

- Firebase Authentication servisi entegre edilmeli.
- Hatalı şifre girişinde kullanıcıya "Kullanıcı adı veya şifre hatalı" uyarısı dönmelii.
- "Beni Hatırla" özelliği ile oturum düşmemeli.

Efor: 3

Öncelik: Yüksek

### 2.2. DESTAN B: DONANIM

#### Hikâye 4: Akıllı Sulama Koruması

Rol: Çevre Dostu Kullanıcı

"Bir kullanıcı olarak, sulama komutu verildikten sonra belirli bir süre ikinci bir sulamanın engellenmesini istiyorum, çünkü yanlışlıkla bitkiye çürütmek istemiyorum."

##### Kabul Kriterleri:

- Firebase üzerinde last\_watered\_time tutulmalı.
- Süre dolmadıysa sayaç (Geri sayım) gösterilmeli.
- Arduino'da millis() ile donanımsal kilit olmalı.

Efor: 5

Öncelik: Kritik

### 2.1. DESTAN A: KİMLİK

#### Hikâye 2: Bitki Profil Özelleştirmesi

Rol: Bitki Sahibi

"Bir kullanıcı olarak, takip ettiğim bitkiye bir isim vermek ve fotoğrafını yüklemek istiyorum, çünkü uygulamada hangi sensörün hangi saksaşa ait olduğunu karıştırmak istemiyorum."

##### Kabul Kriterleri:

- Kullanıcı galeriden/kameradan fotoğraf yükleyebilmeli.
- Fotoğraflar Firebase Storage üzerinde saklanmalı.
- Bitki adı veritabanında güncellenebilmesi.

Efor: 5

Öncelik: Orta

### 2.2. DESTAN B: DONANIM

#### Hikâye 3: Cihaz Sağlıklı Durumu

Rol: Teknik Kullanıcı

"Bir kullanıcı olarak, donanımın o an çevrimiçi olup olmadığını görmek istiyorum, çünkü çevrimdışı bir cihaza sulama komutu gönderip boş beklemek istemiyorum."

##### Kabul Kriterleri:

- Cihaz her 30 saniyede bir timestamp gönderebilme.
- Son sinyal eskiye "Cihaz Çevrimdışı" uyarısı vermelii.
- Cihaz çevrimdışıken "Sula" butonu pasif olmalı.

Efor: 8

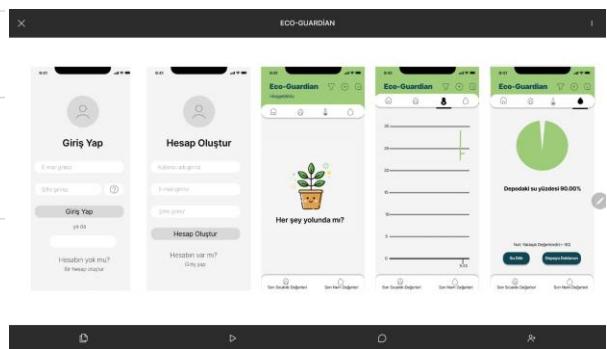
Öncelik: Yüksek

## 2. İterasyon (Sprint) Planı

Proje geliştirme süreci, Çevik (Agile) metodoloji prensiplerine uygun olarak **3 temel Sprint (Koşu)** halinde planlanmış ve uygulanmıştır. Her sprint, belirli bir fonksiyon seti üzerine odaklanmış, GitHub üzerinde ayrı branch'ler (dollar) veya commit grupları ile yönetilmiştir.

Sprint No	Sprint Hedefi	Hikâye ID	Hikâye Adı (User Story)	Görev Tanımı (Task)	Atanan Kişi	Tahmini Efor (SP)	Sprint No	Sprint Hedefi	Hikâye ID	Hikâye Adı (User Story)	Görev Tanımı (Task)	Atanan Kişi	Tahmini Efor (SP)	Sprint No	Sprint Hedefi	Hikâye ID	Görev Tanımı (Task)	Atanan Kişi	Tahmini Efor (SP)	
Sprint 1	Temel Altyapı & MVP	US-01	Güvenli Oturum Açıma	Firebase Projesi oluştur ve Auth servisini aktif et.	Backend Dev	3	Sprint 2	Kontrol & Profil	US-02	Bitki Profil Özelleştirme	"Bitki Ekle" ekranını tasarıla (Form validasyonları ile).	Mobile Dev	5	Sprint 3	Anlık Bildirim (Push)	US-06	Anlık Bildirim (Push)	Firebase Cloud Messaging (FCM) entegre edilmeli.	Backend Dev	8
	(Altyapı & Kimlik)	US-01		Login ve Register ekran tasarımlarını (Flutter UI) kodla.	Mobile Dev	-	(Etkileşim)		US-02		Kamera/Galeri izni al ve seçilen resmi Firebase Storage'a yükle.	Mobile Dev	-	(Akıllı Özellikleme)	US-06		Nem değeri <%20 olunca tetiklenen Cloud Function (veya app logic) yaz.	Mobile Dev	-	
		US-01		Giriş/Kayıt fonksiyonlarını "AuthService" sınıfına bağla.	Mobile Dev	-			US-02		Veritabanında (Realtime DB) yeni bitki düğümü (Node) oluşturulur.	Backend Dev	-		US-06		Bildirime tıklandığında ilgili detay sayfasına yönlendirme yap.	Mobile Dev	-	
		US-03	Cihaz Sağlıklı Durumu	ESP8266 Wi-Fi bağlantı kodlarını yaz ve Firebase'e bağla.	IoT Dev	8			US-04	Akıllı Sulama Koruması	Mobil uygulamaya "Sula" butonu ekle ve DB'de 'pump_status' değiştir.	Mobile Dev	5			US-05	Geçmiş Veri Grafiği	Yıllık kütüphanesini proje dahil et.	Mobile Dev	13
		US-03		Cihazdan her 30 sn'de bir "timestamp" gönderen döngüyü kur.	IoT Dev	-			US-04		Arduino koduna 'pump_status' dinleyici (Listener) ekle.	IoT Dev	-		US-05		Firebase'den son 24 saatin verilerini çekten sorguya yaz.	Backend Dev	-	
		US-03		Mobil arayüzde "Online/Offline" ikonunu duruma göre değiştir.	Mobile Dev	-			US-04		Arduino tarafına "millis()" ile Soğuma (Cooldown) koruması ekle.	IoT Dev	-		US-05		Verileri işleyip X/Y eksenine oturt ve grafiği çizdir.	Mobile Dev	-	
Sprint 1 Toplam						11 SP	Sprint 2 Toplam							10 SP	Sprint 3 Toplam					21 SP

Öncelik Sırası	ID	Hikaye Adı (User Story)	Açıklama / Kapsam	Tahmini Efor (SP)
1 (Çok Yüksek)	US-01	Güvenli Oturum Açıma	Kullanıcıların sisteme güvenli kayıt olması, giriş yapması ve kimlik doğrulama süreçleri (Firebase Auth).	3
2 (Yüksek)	US-03	Cihaz Sağlık Durumu	IoT cihazının Wi-Fi bağlantısını sağlama, online/offline durumunun mobil uygulamada görüntülenmesi.	8
3 (Orta)	US-02	Bitki Profil Özelleştirme	Kullanıcının yeni bitki ekleyebilmesi, bitkiye isim vermesi ve fotoğraf yükleyebilmesi (Storage işlemleri).	5
4 (Orta)	US-04	Akıllı Sulama Koruması	Uzaktan "Sula" butonu ile tetikleme yapılması ve motorun zarar görmemesi için soğuma (cooldown) koruması.	5
5 (Düşük)	US-06	Anlık Bildirim (Push)	Nem değeri kritik seviyeye (%20 altı) düşüğünde kullanıcıya bildirim gönderilmesi ve tıklayınca yönlendirme.	8
6 (Düşük)	US-05	Geçmiş Veri Grafiği	Cihazdan gelen son 24 saatlik verilerin çekilmesi ve anlamlı bir grafik (Chart) üzerinde kullanıcıya sunulması.	13



Sağ tarafta Backlog örneği bulunmaktadır. Üst kısmda bulunan UI tasarım örneği ise birinci sprint sonucu **FIGMA** ile elde edilmiştir.

Sağ kısmda bulunan **JIRA** görevlendirme yapısını görüyorsunuz. Kişiler görevlerini buradan kontrol edebiliyor ve takip edebiliyor. Alt kısmda bulunan örnek ise görevlerin tahmini teslim edilme zamanlarını ve süreci temsil etmektedir.

Öğeler	Zaman çizelgesi arayın	Bugün	Paylaş	Daha fazla
<input checked="" type="checkbox"/> YESIL-1 Veri tabanı araştırması +5 <input checked="" type="checkbox"/> YESIL-2 Veri analizleri nasıl yapılabilir? +5 <input checked="" type="checkbox"/> YESIL-3 Veri tabanı oluşturma +15 <input checked="" type="checkbox"/> YESIL-4 Veriler kullanıcıya nasıl yanıtlanmalı? +5 <input checked="" type="checkbox"/> YESIL-5 Ai eklenimi nasıl kullanılır? +10 <input checked="" type="checkbox"/> YESIL-7 UI için genel temel araştırması +10 <input checked="" type="checkbox"/> YESIL-8 UI için taslaik oluşturma +15 <input checked="" type="checkbox"/> YESIL-9 UI Home sayfası +5 <input checked="" type="checkbox"/> YESIL-10 Veri tasarımları yapıldıktan sonra ona uygun... 	<p>27 28 29 30 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 TEM</p> <p>Logo tasarımı +10</p>	<p>Logo tasarımı +10</p>	<p>Logo tasarımı +10</p>	<p>Daha fazla</p>

### 3. İterasyon (Sprint) Süreci

Proje geliştirme süreci 2 haftalık periyotlarla, toplam 3 Sprint olarak planlanmıştır.

#### Sprint 1: Temel Altyapı ve MVP

- **Hedef:** Donanımın internete bağlanması ve Firebase veri akışının sağlanması.
- **Gerçekleşen İşler:**
  - Firebase projesi oluşturuldu.
  - ESP8266 sensör verileri (nem, sıcaklık) veritabanına yazdırıldı.
  - Flutter'da Login ekranı kodlandı.
- **Retrospektif (Engel & Çözüm):** Windows 11 sürücü uyumsuzluğu nedeniyle karta kod atılamadı. CH340 sürücülerini manuel güncellenerek sorun çözüldü.

#### Sprint 2: Kontrol Mekanizması ve UI

- **Hedef:** Kullanıcının sisteme müdahale edebilmesi ve Bitki Ekleme.
- **Gerçekleşen İşler:**
  - "Bitki Ekle" sayfası ve Fotoğraf Yükleme (Storage) tamamlandı.
  - Uzaktan "Sula" butonu aktifleştirildi.
- **Retrospektif:** Pompa tetiklendikten sonra sonsuz döngüye girdi. Arduino koduna, işlem bitince su\_durum = 0 yapan satır eklenerek çözüldü.

#### Sprint 3: Analiz ve Final

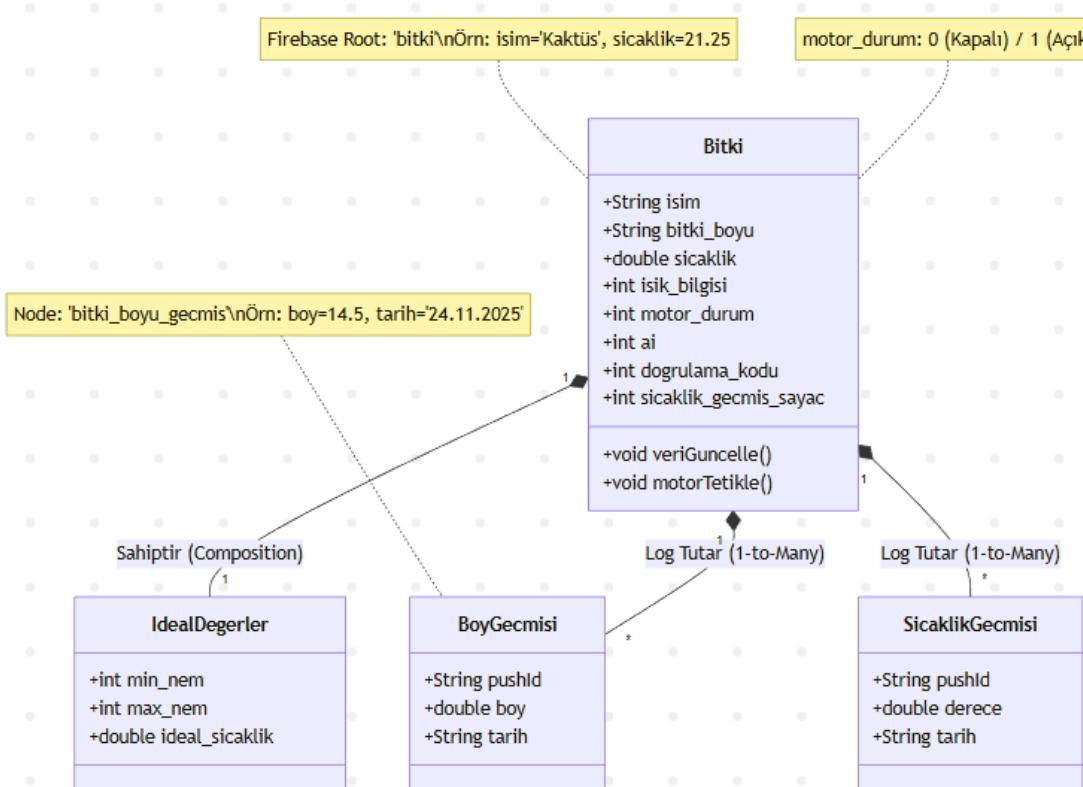
- **Hedef:** Grafikler, Bildirimler ve AI Hazırlığı.
- **Gerçekleşen İşler:**
  - Bildirim servisi kuruldu.
  - Grafik arayüzleri tamamlandı.
  - Kod temizliği (Refactoring) yapıldı.

Sürüm	Tarih	Kapsam & Hedeflenen İşlevler
v0.5	20.04.2023	Veri okuma, temel bağlantı testleri. (Tamamlandı)
v1.0	28.04.2023	Kullanıcı girişi, bitki ekleme, manuel sulama. (Tamamlandı)
v1.2 (şuan)	12.05.2023	Bildirim sistemi, grafikler, güvenlik yamaları. (Yayında)
v2.0 (Gelecek)	2026 Q1	Çoklu cihaz desteği ve Yapay zeka analizi

## 4. Modelleme Çalışmaları

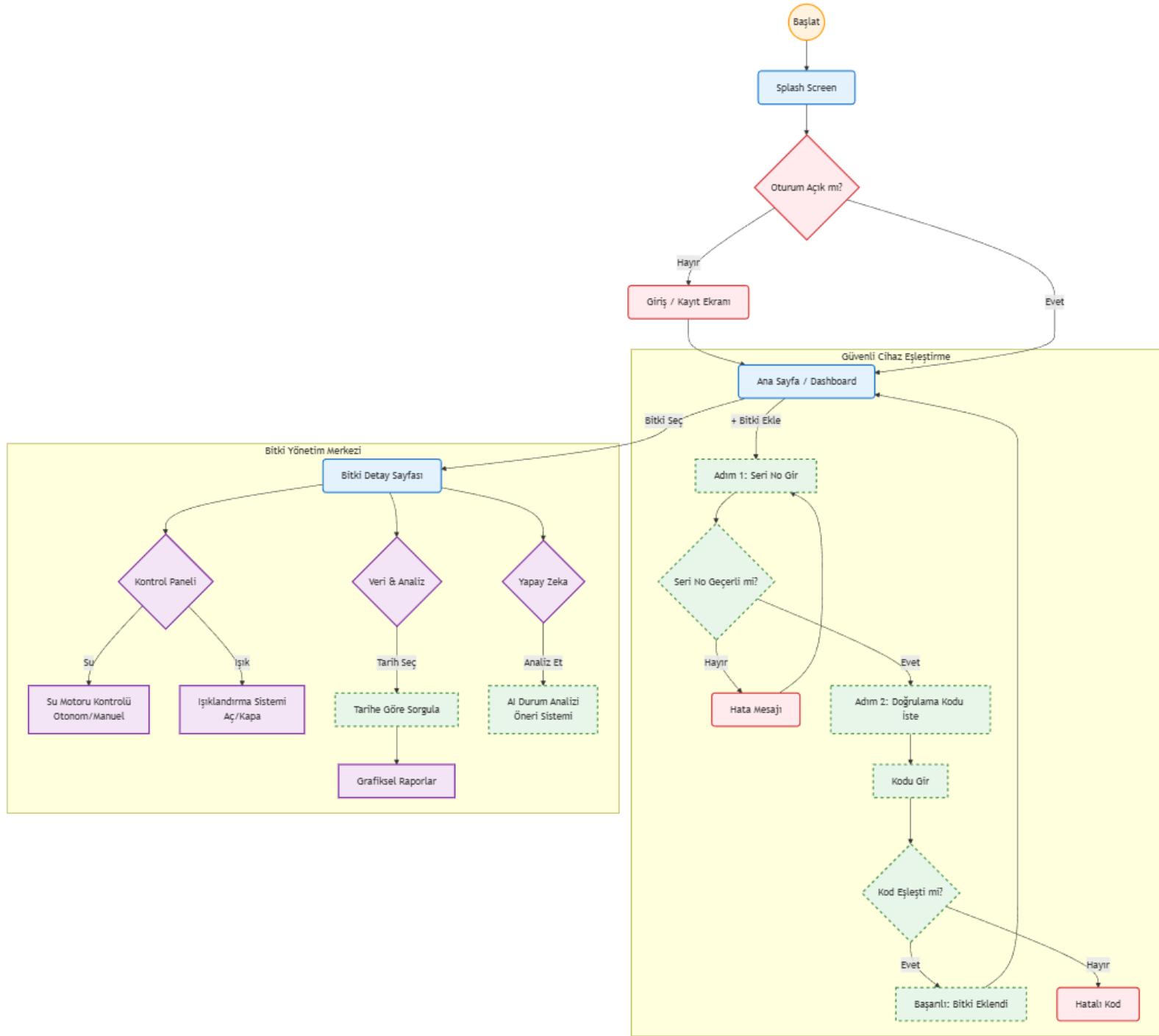
### 1. Domain (Veri ve Nesne) Modeli

Bu şema, veritabanında "Kullanıcı" ve "Bitki"nin birbirine nasıl bağlı olduğunu ve hangi verilerin tutulduğunu gösterir.



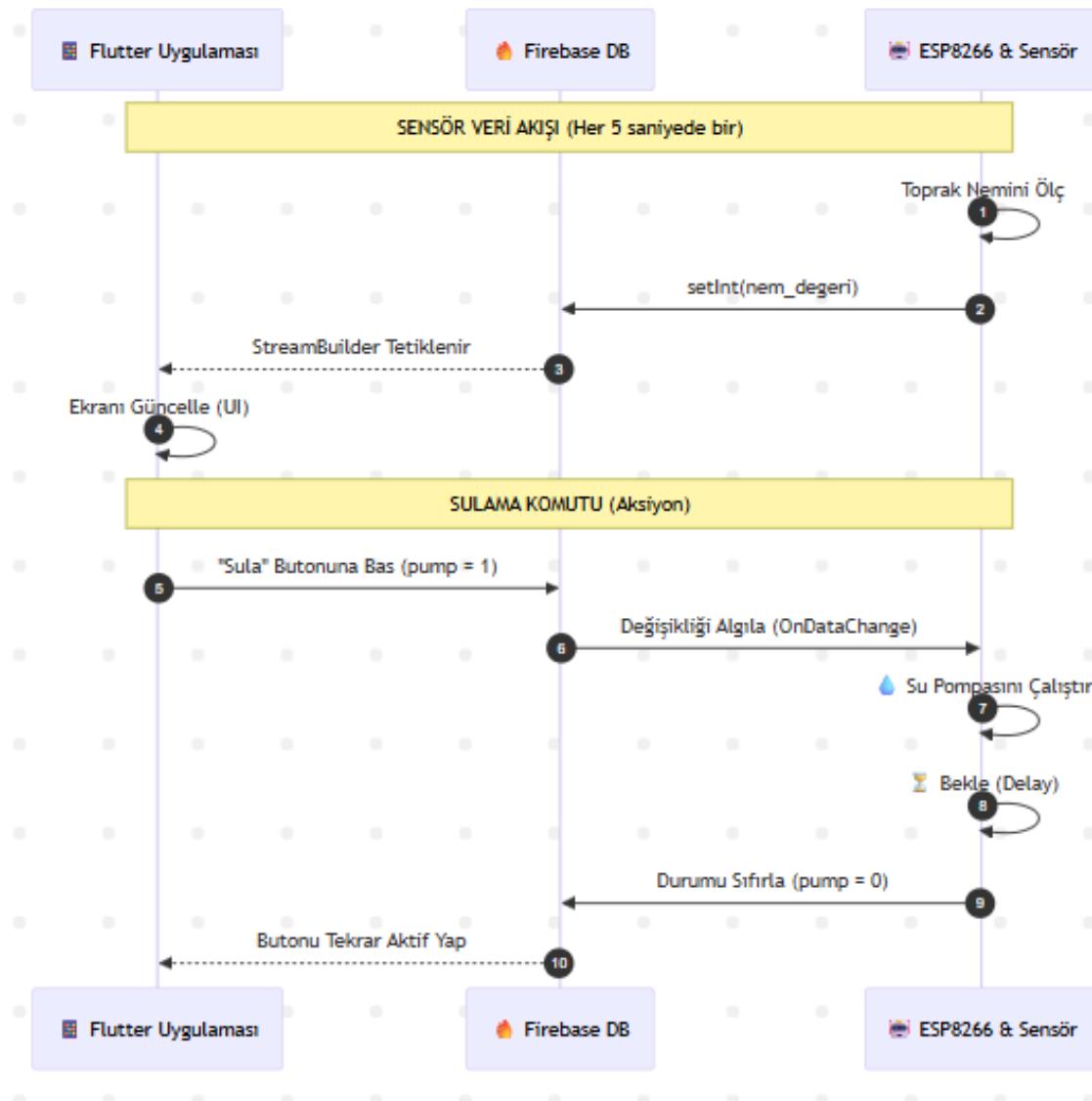
## 2. Sayfa Navigasyon (Page Flow) Diyagramı

Bu şema, kullanıcının uygulamayı açtığı andan itibaren hangi ekranlardan geçtiğini gösterir. (Kırmızı: Giriş İşlemleri, Mavi: Ana Uygulama, Yeşil: Aksiyonlar)



### 3. Sistem Mimarisi (IoT Çalışma Mantığı)

Bu, projenin en havalı diyagramıdır. Mobil uygulamanın, Firebase'in ve Arduino'nun (ESP8266) nasıl konuştuğunu gösterir



Not: Projede kullanılan metinlerin daha anlaşılır olması ve imla kurallarına tam uyum sağlama adına Google Gemini kullanılmıştır. Tahmini kullanım oranı **%13.8 - %27.4** arasındadır. (**Mesela bu cümleyi de ben yazdım ama Gemini'ye atıp imlasını düzelttirdim.**)