Tasarım Belgesi (Design Document)

Proje Adı: TerraSense

Oluşturanlar (List of Contributors):

- Ahmet Yusuf ŞAHİN
- Furkan YAHŞİ
- Mehmet Ali USLU
- Yusuf Can BAHADIRLIOĞLU

İçindekiler (Table of Contents)

Görev Matrisi (Task Matrix)

Bölüm	Sorumlu Kişi	Açıklama
Tasarım Belgesi Formatı	Furkan YAHŞİ	Genel sistem mimarisi ve özet tanım
Implementation Details	Ahmet Yusuf ŞAHİN	Önemli bileşenler, arayüzler ve kod yapısı
Use Case Support in Design	Ahmet Yusuf ŞAHİN	Use case seçimi, gereksinim eşleştirme ve tasarım
Design Decisions	Furkan YAHŞİ	Teknoloji karşılaştırmaları ve nihai seçim gerekçeleri
GitHub Commit Requirement	Mehmet Ali USLU, Yusuf Can BAHADIRLIOĞLU	Kod ve dokümantasyon yükleme, commit mesajları

1. Sistem Genel Bakış (System Overview):

1.1 Brief Project Description

TerraSense, iklim verilerini analiz ederek tarım uygulamalarını optimize etmeyi ve çiftçilere gerçek zamanlı, kişiselleştirilmiş öneriler sunmayı amaçlayan bir mobil uygulamadır. Bu sayede kullanıcılar, toprak analiz sonuçlarını sisteme girerek bitki yetiştirme koşullarını iyileştirebilir ve hava durumu gibi çevresel faktörlere göre uyumlu stratejiler geliştirebilir.

1.2 System Architecture (Layered Architecture)

- **Sunum Katmanı (Mobile App)**: Flutter tabanlı mobil uygulama.
- **İş Mantığı Katmanı (Backend Services)**: API ve yapay zekâ modülleri (Python).
- **Veri Erişim Katmanı (Database)**: Bulut tabanlı bir veritabanı (SQLAlchemy).

1.3 Technology Stack

- **Frontend (Mobil)**: Flutter
- **Backend**: Python (Flask)
- **Database**: SQLAlchemy (Veri saklama, kullanıcı kayıtları, geçmiş veriler)
- **AI/API**: OpenAI

2. Implementation Details

2.1 Codebase Structure

a. `mobile_app/`: Flutter projesi
b. `backend/`: Sunucu tarafı kodları (Python)
c. `ai_module/`: Yapay zekâ ve model dosyaları
d. `docs/`: Tasarım, QA planı ve benzeri dokümantasyon

2.2 Key Implementations

Toprak Analizi Modülü: Kullanıcıların girdiği toprak verilerini işleyen ve bitki türüne göre öneriler oluşturan bileşen.
Hava Durumu Servisi: Gerçek zamanlı API'lerden hava durumu verilerini çekerek, uygulama içinde uyarılar oluşturan modül.
Öneri Motoru: İklim ve toprak verilerini birleştirerek kullanıcıya "en uygun yetiştirme stratejisi" sunan yapay zekâ tabanlı servis.

2.3 Component Interfaces

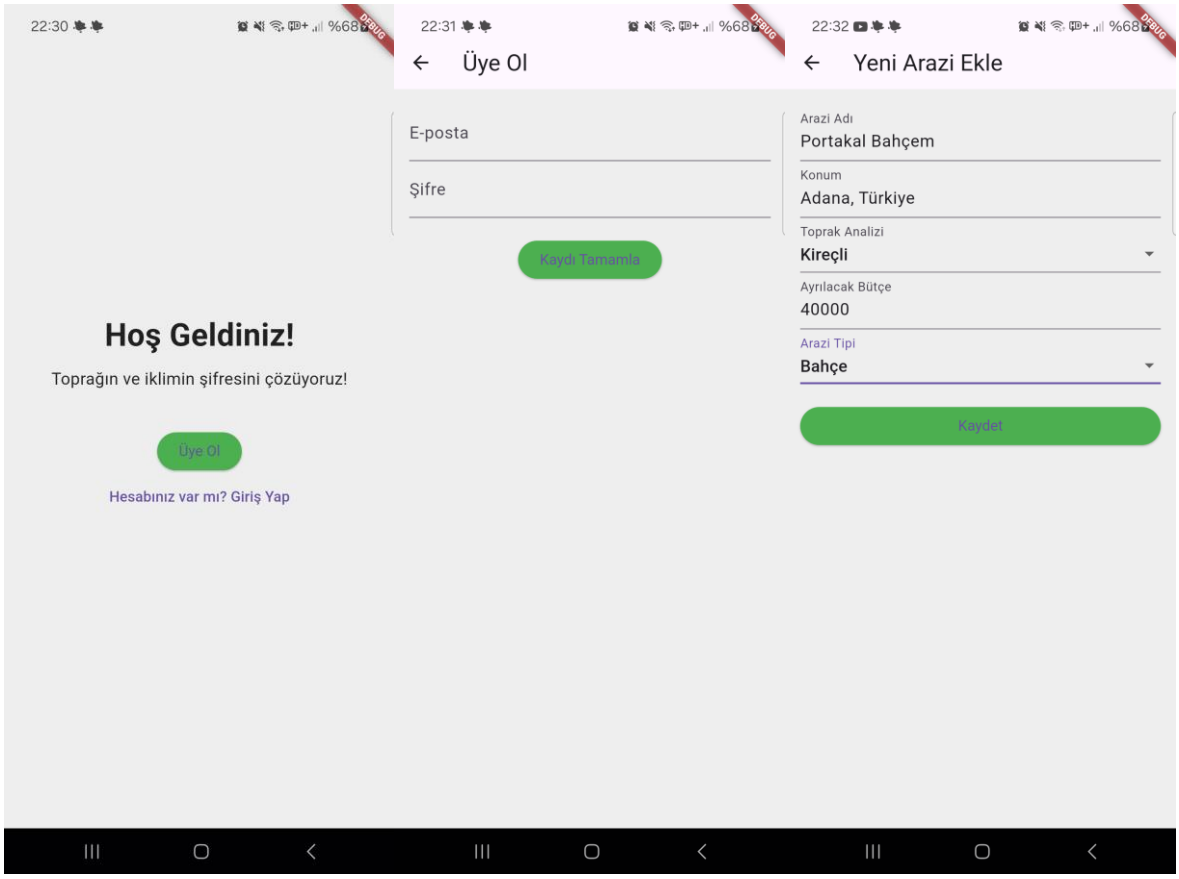
API Endpoint Örneği:

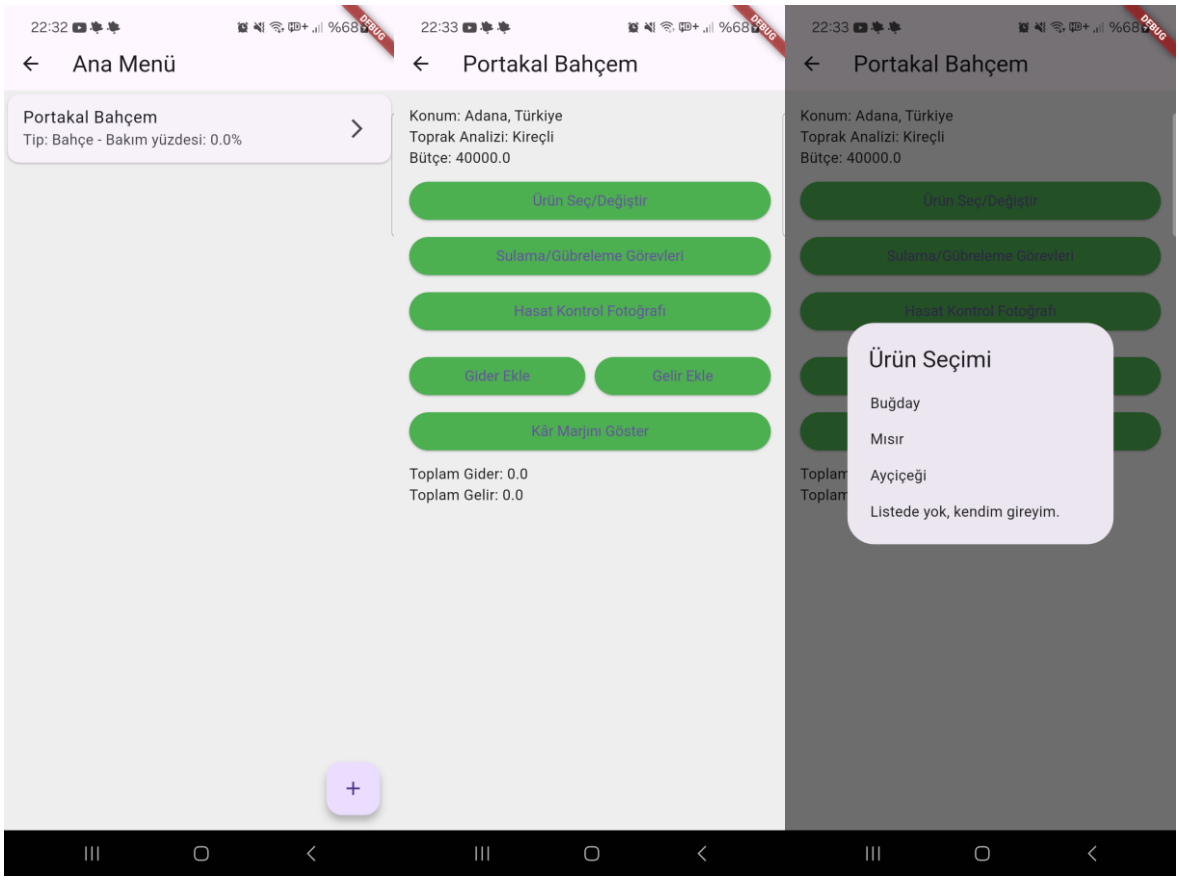
i. **get_recommendation(user_id: str) →** Kullanıcını verilerini yapay zeka sistemine gönderir ve öneriler alıp tekrar bu önerileri kullanıcıya gösterir.

2.4 Visual Interfaces

Mobil uygulama için Flutter ile oluşturulmuş basit ekran taslakları:

- **Ana Sayfa**: Hava durumu özeti, hızlı menü
- **Toprak Analizi Girişi**: pH, nem, mineral değerleri vb. alanlar
- **Öneriler Ekranı**: Hangi ürünün yetiştirilebileceği, sulama seviyesi, gübre miktarı vb.


Hoş Geldiniz!
Toprağın ve iklimin şifresini çözüyoruz!
Üye Ol
Hesabınız var mı? Giriş Yap
Üye Ol
E-posta
Şifre
Kaydı Tamamla
Yeni Arazi Ekle
Arazi Adı
Portakal Bahçem
Konum
Adana, Türkiye
Toprak Analizi
Kireçli
Ayrılacak Bütçe
40000
Arazi Tipi
Bahçe
Kaydet

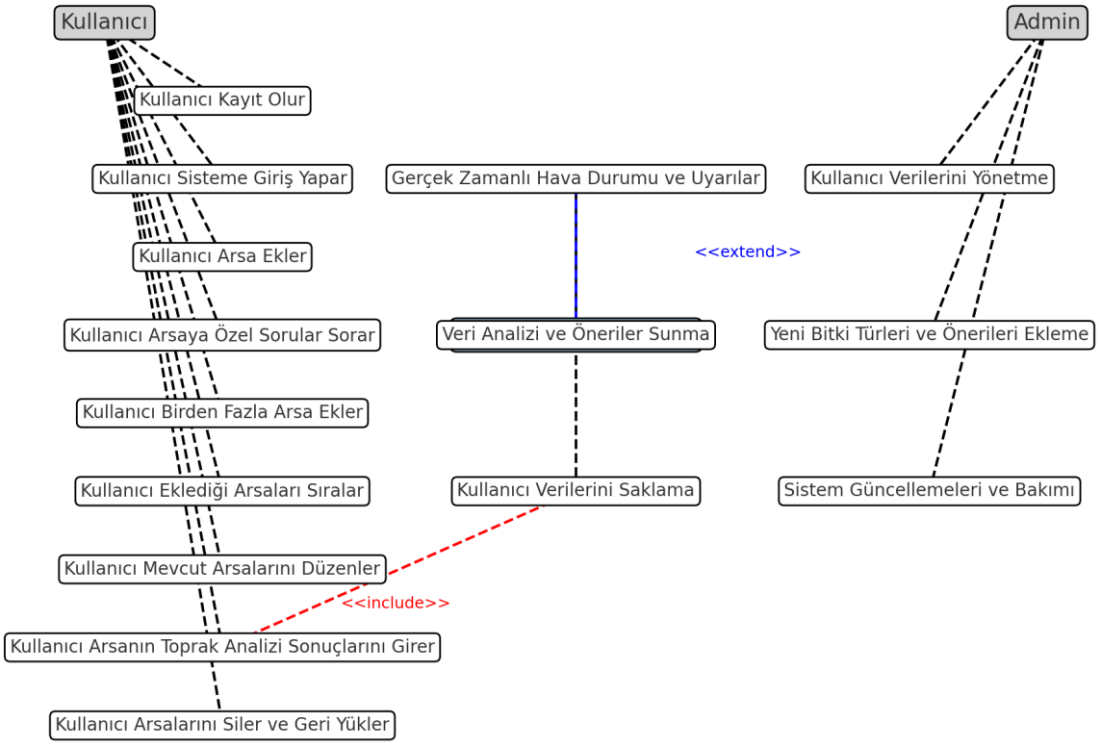

Ana Menü
Portakal Bahçem
Tip: Bahçe - Bakım yüzdesi: 0.0%
Portakal Bahçem
Konum: Adana, Türkiye
Toprak Analizi: Kireçli
Bütçe: 40000.0
Ürün Seç/Değiştir
Sulama/Gübreleme Görevleri
Hasat Kontrol Fotoğrafı
Gider Ekle
Gelir Ekle
Kâr Marjını Göster
Toplam Gider: 0.0
Toplam Gelir: 0.0
Portakal Bahçem
Konum: Adana, Türkiye
Toprak Analizi: Kireçli
Bütçe: 40000.0
Ürün Seç/Değiştir
Sulama/Gübreleme Görevleri
Hasat Kontrol Fotoğrafı
Ürün Seçimi
Buğday
Mısır
Ayçiçeği
Listede yok, kendim gireyim.

3. Use Case Support in Design

3.1 Use Case Selection:

Gereksinimler Belgesi'nden seçilen 4 ana kullanım durumu:

- Yapay zeka destekli öneriler sunma
- Kullanıcı sisteme giriş yapabilecek
- Kullanıcı arsa ekleyebilecek
- Kullanıcı eklediği arsaları istediği sırayla görüntüleyebilecek



3.2 Requirement Mapping:

- **Toprak Analizi ve Öneriler:** Kullanıcı, toprak verilerini girerek yapay zekâ destekli öneriler alır.
- **Hava Durumu Uyarıları:** Gerçek zamanlı veri akışı ile uyarılar sağlanır.
- **Mobil Erişim:** Uygulama, mobil cihazlarda sorunsuz çalışır.

- **Veri Saklama/Raporlama:** Kullanıcı verileri güvenli şekilde depolanır ve dönemsel raporlar oluşturulur.

3.3 Use Case Design:

- Her kullanım durumu için, sistem mimarisi uygun modüllere ayrılmıştır.
- Veri akış diyagramları ve durum geçişleri, kullanıcı etkileşimleri detaylandırılmıştır.

3.4 Demo Requirement:

- Belirlenen dört use case, final sunumunda canlı olarak gösterilecek ve işlevsellikleri detaylıca açıklanacaktır.

4. Design Decisions

4.1 Technology Comparisons:

- **Frontend:** Flutter vs. Native Android
  → Flutter, daha esnek bir yapıya sahip olduğundan ve geliştirme süresinde avantaj sağladığından tercih edilmiştir.
- **Backend:** Node.js vs. Python
  → Node.js yüksek performanslı I/O işlemleri sunarken, Python güçlü yapay zekâ entegrasyon kütüphaneleri sağlar.

4.2 Decision Justifications:

- **Flutter:** Mobil kullanıcı deneyimini optimize etmek ve geliştirme süresini kısaltmak için seçilmiştir.
- **Backend Tercihi:** Ekibimizdeki üyelerin de bu dile yatkın olması sebebiyle Python tercih edildi.
- **Veri Tabanı:** SQLAlchemy, ölçeklenebilir ve güvenli veri depolaması için en uygun seçenek olarak belirlenmiştir.

5. GitHub Commit Requirement

5.1 Code Implementations & Interfaces:

- Tüm kod, bileşen arayüzleri ve UI tasarım dosyaları GitHub repository'sinde saklanacaktır.
- Her commit, açık ve anlamlı commit mesajları ile yapılacak; takım üyelerinin katkıları net şekilde izlenecektir.

5.2 Technology Comparisons:

- Alternatif teknolojilerle yapılan karşılaştırmaların kod örnekleri de repository'de yer alacak; bu örnekler commit geçmişinde detaylandırılacaktır.