

Design Document

I. System Overview

Brief Project Description

FitOl, kullanıcıların vücut kitle indeksini (VKİ) hesaplamalarına olanak tanıyan ve sonuçlara göre beslenme ve egzersiz önerileri sunan bir web sitesidir. Sistem, kullanıcıdan boy ve kilo bilgilerini alarak VKİ hesaplar ve kullanıcıyı "Zayıf", "Normal Kilo", "Kilolu" veya "Obez" kategorilerine ayırır. Her kategoriye uygun öneriler sunarak sağlıklı yaşam sürecine katkı sağlar.

System Architecture

Sistem, istemci tarafında HTML, CSS ve JavaScript kullanarak geliştirilmiştir. Backend gereksinimi bulunmamaktadır, tüm işlemler frontend tarafında gerçekleşmektedir. Web sitesi, GitHub Pages üzerinde barındırılarak erişilebilir olacaktır.

Technology Stack

Frontend: HTML, CSS, JavaScript

Hosting: GitHub Pages

Testing: Selenium (UI testleri için)

Design Tools: Figma (UI prototipleme)

2. Implementation Details

Codebase Structure

index.html: Ana sayfa ve kullanıcı formu.

style.css: Web sitesinin stil dosyası.

script.js: VKİ hesaplama ve kategori belirleme işlemlerini içeren JavaScript fonksiyonları.

test_cases.js: Test senaryolarını içeren JavaScript dosyası.

Key Implementations

VKİ Hesaplama Algoritması: Kullanıcının girdiği boy ve kilo değerlerine göre VKİ hesaplanır.

Kategori Belirleme Modülü: Hesaplanan VKİ değerine göre kullanıcı uygun bir kategoriye atanır.

Öneri Sistemi: Kullanıcının kategorisine göre uygun diyet ve egzersiz önerileri gösterilir.

Visual Interfaces

Kullanıcı giriş formu (boy, kilo bilgisi alma)
VKİ sonucunun ve kategori bilgisinin gösterildiği ekran
Kategoriye uygun önerilerin gösterildiği arayüz

3. Use Case Support n Design

Use Case Selection

Kullanıcının boy ve kilo bilgilerini girerek VKİ hesaplaması
Kullanıcının VKİ sonucuna göre kategori belirlenmesi
Kullanıcının kategorisine özel diyet önerileri alması
Kullanıcının kategorisine özel egzersiz önerileri alması

Requirement Mapping

Her bir use case için fonksiyonel gereksinimler tanımlanmıştır.

Use Case Design

Kullanıcı formdan bilgilerini girer.
Sistem VKİ hesaplamasını yapar ve sonucu ekranda gösterir.
Kullanıcıya kategorisine uygun öneriler sunulur.

4. Design Decisions

Technology Comparisons

Teknoloji	Alternatif	Seçim Nedeni
GitHub Pages	Netlify, Vercel	Ücretsiz, kolay yönetilebilir
JavaScript	Python (Flask)	Sadece frontend yeterli
Selenium	Cypress	Selenium daha geniş destekli

Decisions Justifications

GitHub Pages, hosting maliyetlerini sıfıra indirerek projenin erişilebilir olmasını sağlar.

JavaScript, frontend tabanlı bir uygulama için en uygun seçimdir.

Selenium, UI testleri için yaygın olarak kullanılan bir araçtır.

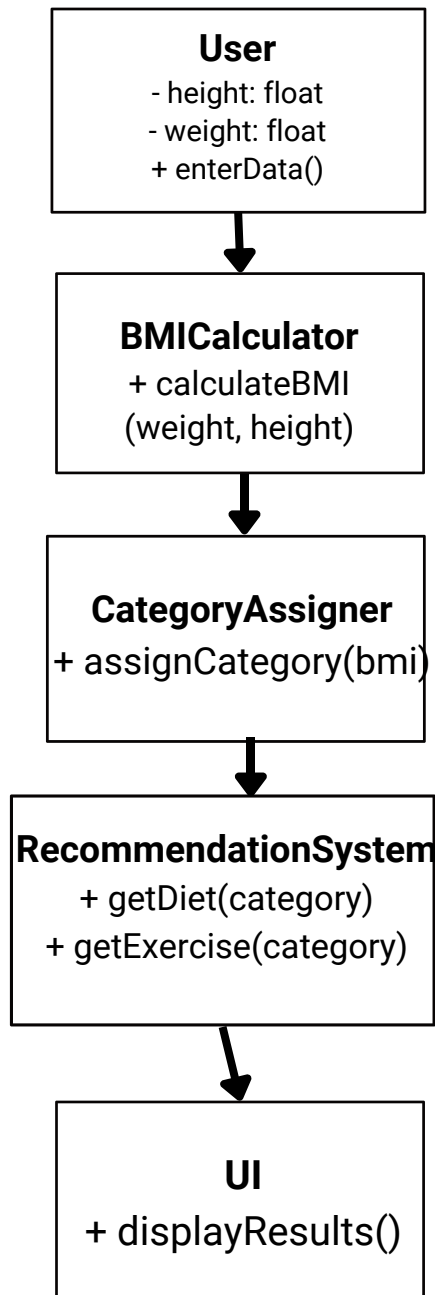
5. GitHub Commit Requirement

Code Implementations & Interfaces: Tüm kodlar GitHub'a yüklenecek.

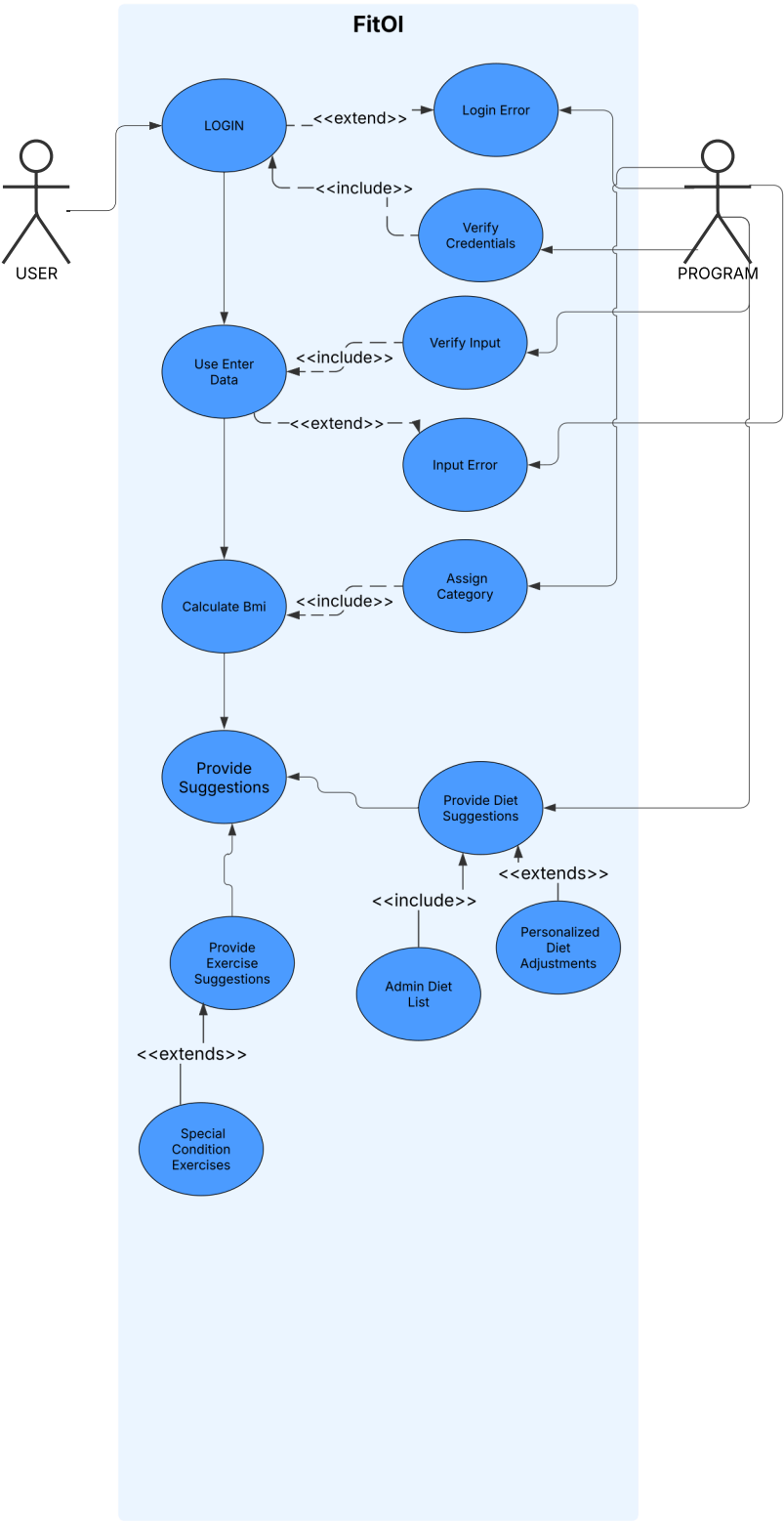
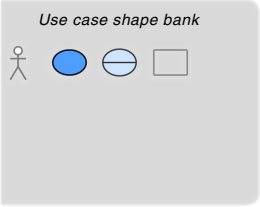
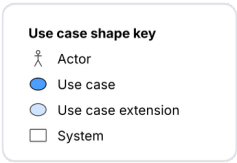
Technology Comparisons: Karşılaştırma yapılan teknolojilerin kod örnekleri GitHub'da tutulacaktır.

Commit Mesajları: Her commit mesajı, değişikliklerin ne olduğunu açıkça belirtecektir.

Class Diagram



Use Case Diagram



Quality Assurance (QA) Plan

1. Quality Assurance Strategy

Overview

FitOl projesinin yazılım kalitesini sağlamak için bir dizi test ve kalite güvence prosedürü uygulanacaktır. Bu süreç, fonksiyonel doğruluğu, performansı ve kullanılabilirliği kapsayacaktır.

Testing Methodologies

- Unit Testing: JavaScript fonksiyonlarının doğru çalıştığını doğrulamak için Jest veya Mocha kullanılarak yapılacaktır.
- Integration Testing: Sistem bileşenlerinin birlikte düzgün çalıştığını test etmek için yapılacaktır.
- UI/UX Testing: Selenium kullanılarak web arayüzünün test edilmesi sağlanacaktır.
- Usability Testing: Gerçek kullanıcı testleri yapılarak kullanılabilirlik değerlendirmesi gerçekleştirilecektir.

- Automated vs. Manual Testing**

- Otomatik Testler: VKİ hesaplama fonksiyonu, kategori belirleme ve öneri sisteminin doğruluğunu test eden otomatik testler.
- Manuel Testler: Kullanıcı deneyimi testleri ve görsel tasarım kontrolleri.

2. Quality Factors & Metrics

Quality Factor	Description	Measurement Metric
Performance	Sistem yanıt süresi	Ortalama yanıt süresi (ms)
Usability	Kullanıcı deneyimi kolaylığı	Kullanıcı anketi puanı
Maintainability	Kodun değiştirilme kolaylığı	Kod karmaşıklık skoru
Accuracy	VKİ hesaplamalarının doğruluk seviyesi	Hatalı hesaplama yüzdesi
Reliability	Sistem hata oranı	Çökme veya hata yüzdesi

3. Test Plan

Test Cases

En az beş test senaryosu oluşturulacaktır:

- 1.Kullanıcıdan geçerli boy ve kilo değerleri alındığında doğru VKİ hesaplanmalı.
- 2.Kullanıcı negatif veya sıfır değer girerse hata mesajı göstermeli.
- 3.Kullanıcı VKİ sonucuna göre doğru kategoriye atanmalı.
- 4.Kullanıcı kategoriye özel diyet önerilerini görebilmeli.
- 5.Kullanıcı kategoriye özel egzersiz önerilerini görebilmeli.
- 6.Kullanıcı web sitesindeki butonlar ve arayüz bileşenleri düzgün çalışmalı.

Test Case 1

Kullanıcıdan Geçerli Boy ve Kilo Değerleri Alındığında Doğru VKİ Hesaplanmalı

- Amaç: Kullanıcı tarafından girilen boy (m) ve kilo (kg) değerleriyle doğru Vücut Kitle İndeksi (VKİ) hesaplandığını doğrulamak.
- Giriş Verileri:
 - Kullanıcı boy olarak 1.75 m, kilo olarak 70 kg girer.
 - Kullanıcı boy olarak 1.60 m, kilo olarak 55 kg girer.
- Beklenen Sonuç:
 - VKİ Formülü: $VKİ = \text{kilo} / (\text{boy} * \text{boy})$ Test 1 Sonucu: $VKİ = 70 / (1.75 * 1.75) = 22.86$ (Normal Kilo) Test 2 Sonucu: $VKİ = 55 / (1.60 * 1.60) = 21.48$ (Normal Kilo) Ekranda hesaplanan VKİ'nin doğru şekilde gösterilmesi.
- Gerçekleşen Sonuç:
 - Eğer hesaplanan sonuç beklenenle aynıysa test başarılıdır.
 - Eğer yanlış hesaplanıyorsa hata kodu ve düzeltme süreci başlatılır.

Test Case 2

Kullanıcı Negatif veya Sıfır Değer Girerse Hata Mesajı Göstermeli

- Amaç: Kullanıcının yanlış veya geçersiz giriş yapması durumunda sistemin uygun hata mesajı gösterdiğini doğrulamak.
- Giriş Verileri:
 - Kullanıcı boy olarak 0 girer, kilo olarak 65 kg girer.
 - Kullanıcı boy olarak -1.80 m girer, kilo olarak 70 kg girer.
 - Kullanıcı boy olarak 1.75 m, kilo olarak 0 girer.
- Beklenen Sonuç:
 - Hata mesajı gösterilmeli: "Lütfen geçerli bir boy ve kilo değeri girin."
 - Kullanıcının işlemi tamamlamasına izin verilmemeli.
- Gerçekleşen Sonuç:
 - Hata mesajı çıkıyorsa test başarılıdır.
 - Hata mesajı çıkmıyorsa kod gözden geçirilerek düzeltme yapılır.

Test Case 3

Kullanıcı VKİ Sonucuna Göre Doğru Kategoriyeye Atanmalı

- Amaç: Kullanıcının hesaplanan VKİ değerine göre doğru sağlık kategorisine yerleştirildiğini doğrulamak.
- Giriş Verileri:
 - Kullanıcı boy: 1.75 m, kilo: 50 kg (VKİ = 16.33)
 - Kullanıcı boy: 1.75 m, kilo: 70 kg (VKİ = 22.86)
 - Kullanıcı boy: 1.75 m, kilo: 85 kg (VKİ = 27.76)
 - Kullanıcı boy: 1.75 m, kilo: 100 kg (VKİ = 32.65)
- Beklenen Sonuç:
 - $VKİ < 18.5 \rightarrow$ "Zayıf"
 - $VKİ 18.5 - 24.9 \rightarrow$ "Normal Kilo"
 - $VKİ 25 - 29.9 \rightarrow$ "Kilolu"
 - $VKİ \geq 30 \rightarrow$ "Obez"
- Gerçekleşen Sonuç:
 - Kullanıcının uygun kategoriye atanması beklenir.
 - Yanlış atanırsa sistem hatalıdır ve düzeltilmelidir.

Test Case 4

Kullanıcı Kategorisine Özel Diyet Önerilerini Görebilmeli

- Amaç: Kullanıcının VKİ kategorisine uygun diyet önerilerini almasını doğrulamak.
- Giriş Verileri:
 - Kullanıcı VKİ 16.5 (Zayıf) çıkarsa,
 - Kullanıcı VKİ 23 (Normal Kilo) çıkarsa,
 - Kullanıcı VKİ 27 (Kilolu) çıkarsa,
 - Kullanıcı VKİ 33 (Obez) çıkarsa.
- Beklenen Sonuç:
 - Zayıf kullanıcıya önerilenler: Kalori alımını artıran beslenme planı.
 - Normal kilolu kullanıcıya önerilenler: Dengeli diyet programı.
 - Kilolu kullanıcıya önerilenler: Daha düşük karbonhidrat ve yağ oranına sahip diyet.
 - Obez kullanıcıya önerilenler: Düşük kalorili ve diyetisyen önerisine dayalı plan.
- Gerçekleşen Sonuç:
 - Kullanıcılar kategorilerine uygun diyet önerilerini görmelidir.
 - Yanlış öneriler çıkarsa sistem düzeltilmelidir.

Test Case 5

Kullanıcı Kategorisine Özel Egzersiz Önerilerini Görebilmeli

- Amaç: Kullanıcının VKİ kategorisine uygun egzersiz planlarını görmesini sağlamak.
- Giriş Verileri:
 - Kullanıcı VKİ 16.5 (Zayıf) çıkarsa,
 - Kullanıcı VKİ 23 (Normal Kilo) çıkarsa,
 - Kullanıcı VKİ 27 (Kilolu) çıkarsa,
 - Kullanıcı VKİ 33 (Obez) çıkarsa.
- Beklenen Sonuç:
 - Zayıf kullanıcıya önerilenler: Kas kütlesini artırmaya yönelik egzersizler.
 - Normal kilolu kullanıcıya önerilenler: Genel fitness programı.
 - Kilolu kullanıcıya önerilenler: Yağ yakımına yönelik orta yoğunluklu egzersizler.
 - Obez kullanıcıya önerilenler: Kardiyo ve düşük yoğunluklu antrenmanlar.
- Gerçekleşen Sonuç:
 - Kullanıcının uygun egzersiz önerisini görmesi gerekir.
 - Yanlış veya uygun olmayan egzersiz önerisi çıkarsa düzeltme yapılır.

Test Case 6

- Amaç: Kullanıcının web sitesindeki butonları ve arayüz bileşenlerini doğru şekilde kullanabildiğini doğrulamak.
- Giriş Verileri:
 - Kullanıcı VKİ hesapla butonuna basar.
 - Kullanıcı Temizle butonuna basar ve giriş alanlarının temizlendiğini kontrol eder.
 - Kullanıcı Diyet Önerileri butonuna basarak önerileri görmelidir.
 - Kullanıcı Egzersiz Önerileri butonuna basarak önerileri görmelidir.
- Beklenen Sonuç:
 - Butonlar ve arayüz bileşenleri doğru şekilde çalışmalı ve ilgili işlemleri gerçekleştirmelidir.
 - Kullanıcı geri bildirimleri doğru bir şekilde sunulmalıdır.
 - Yanlış yönlendirme, butonların işlevsiz olması veya sayfa çökmesi gibi durumlar test edilerek raporlanmalıdır.
 - Mobil ve masaüstü uyumluluğu sağlanmalıdır.

Bug Tracking

- **Hata Yönetimi:** GitHub Issues kullanılarak tüm hatalar takip edilecek.
- **Hata Raporlama Süreci:**
 - Hata Bildirimi: Hata tespit edildiğinde, hata açıklaması, ekran görüntüsü ve tekrar üretme adımlarıyla birlikte GitHub Issues'da açılacaktır.
 - Önceliklendirme: Hata, etkisine göre sınıflandırılacaktır (Kritik, Yüksek, Orta, Düşük).
 - Atama: Hata, ilgili geliştiriciye veya test ekibine atanacaktır.
 - Çözüm ve Güncelleme: Hata düzeltildiğinde, ilgili commit mesajında açıklama yapılacaktır.
 - Doğrulama: Düzeltme sonrası test edilerek hatanın çözüldüğü onaylanacaktır.
 - Kapanış: Hata çözüldüğünde, ilgili GitHub Issue kapatılacaktır.
- **Hata Önceliklendirme:**
 - Kritik: VKİ hesaplama yanlış çıkıyorsa.
 - Yüksek: UI çalışmıyorsa.
 - Orta: Kullanıcı deneyimini etkileyen hatalar.
 - Düşük: Küçük stil bozulmaları.