

ENTEGRASYON RAPORU

Hazırlayan: İzzet Esener **210229048** –
Furkan Öztürk **230229083**

Grup No: 12

Grup Yöneticisi
İzzet Esener **210229048**

Grup Üyeleri
Salih Can Turan **210229040**
Kerem Kartal **210229019**
Furkan Öztürk **230229083**
Ders: Yazılım Proje Yönetimi

1. Giriş:

Bu projenin temel amacı, bireylerin sağlıklı beslenme alışkanlıkları kazanmalarını destekleyecek, kullanıcı dostu ve kapsamlı bir Online Diyet Takip Uygulaması geliştirmektir. Uygulama, kullanıcıların günlük besin tüketimlerini, ihtiyacına ve hedeflerine yönelik diyet planlarını güvenli bir şekilde yönetmelerine imkân tanımaktadır.

Proje, yazılım geliştirme yaşam döngüsünün tüm adımlarında proje yönetimi disiplinlerinin en iyi uygulamalarıyla yapılandırılmış, stratejik bir yaklaşımla ele alınmıştır. Her bir adım dikkatle planlanmış ve uygulamaya geçirilmiştir.

Bu entegrasyon raporu, uygulamanın farklı bileşenlerinin bir araya getirilmesi sürecini kapsamlı şekilde ortaya koyar. Her aşama — belirleme, planlama, uygulama ve değerlendirme — projenin başarısı açısından kritik öneme sahiptir. Rapor, tüm bileşenlerin entegre bir yapı içerisinde çalışmasını sağlayarak kullanıcıya işlevsel, güvenli ve kaliteli bir ürün sunulmasını amaçlamaktadır.

2. Entegrasyon Süreci

2.1. Gereksinimlerin Belirlenmesi

Hedef Kitlenin Analizi: Uygulamanın başarılı bir şekilde konumlanabilmesi için farklı kullanıcı profilleri (kilo vermek isteyenler, sporcular, sağlık problemi olan bireyler vb.) detaylı analiz edilmiştir. Diyetisyenler, sağlık danışmanları ve potansiyel kullanıcılarla yapılan anket ve görüşmeler sonucunda ihtiyaçlar kategorize edilmiştir.

Fonksiyonel Gereksinimlerin Tanımı: Besin günlüğü, kalori hesaplama, su tüketimi takibi, öğün planlama, kişisel hedef belirleme, raporlama, hatırlatıcı sistemleri gibi temel özellikler önceliklendirilmiştir. Her fonksiyonun kullanıcıya sağlayacağı fayda ve kullanım sıklığına göre önem derecesi belirlenmiştir.

Fonksiyonel Olmayan Gereksinimler: Sistem güvenliği, mobil performans, veri gizliliği, çoklu platform desteği, çevrimdışı kullanım ve erişilebilirlik gibi özellikler ayrı olarak ele alınmış ve bu alanlarda belirli kalite hedefleri belirlenmiştir.

Sürekli İyileştirme Yaklaşımı: Gereksinim analizinin proje başlangıcıyla sınırlı kalmaması; kullanıcı testlerinden gelen geri bildirimlerle bu gereksinimlerin revize edilmesi esas alınmıştır.

2.2 Analiz ve Tasarım

Modüler Sistem Mimarisi: Uygulama, farklı işlevsel modüllerin bağımsız şekilde geliştirilebileceği bir yapı üzerine kurulmuştur. Örneğin: Diyet programı modülü, beden ölçümü modülü, kullanıcı hedef yönetimi gibi parçalar birbirinden bağımsız olarak planlanmıştır.

Sistem Akışı ve Veri Diyagramları: Kullanıcı akış diyagramları, use case senaryoları ve veri akış şemaları ile sistemin nasıl çalışacağı detaylandırılmıştır. Bu, yazılım ekiplerinin aynı dili konuşmasını ve entegrasyon kolaylığı sağlamıştır.

Prototipleme Süreci: Wireframe, mockup ve düşük-yüksek çözünürlüklü prototiplerle

kullanıcı arayüzü önceden test edilmiş, bu sayede tasarıma başlamadan önce kullanıcı deneyimi test edilmiştir.

Erişilebilirlik ve Kullanılabilirlik Standartları: Görme bozukluğu, renk körlüğü gibi durumlar da düşünülerek tasarım sürecinde WAI-ARIA erişilebilirlik standartları dikkate alınmıştır.

2.3. Veritabanı Tasarımı

Veri Modelleme: Kullanıcılar, öğünler, besin değerleri, tüketim günlükleri, hedefler, geri bildirimler gibi tüm sistem varlıkları için normalleştirilmiş bir veritabanı yapısı oluşturulmuştur.

Gizlilik ve Güvenlik: GDPR ve KVKK gibi veri koruma mevzuatlarına uygunluk sağlanarak kullanıcıların kişisel verileri şifreleme, token bazlı doğrulama ve rollerle erişim kontrolü gibi yöntemlerle güvence altına alınmıştır.

Performans Optimizasyonu: Veri erişim hızlarını artırmak için indeksleme, sorgu optimizasyonu, yedekleme stratejileri ve cache mekanizmaları uygulanmıştır.

Esneklik ve Büyüme: Sistem, ileride yemek tarifi veritabanı, yapay zekâ destekli öneri motoru gibi genişlemeleri destekleyecek şekilde ölçeklenebilir biçimde tasarlanmıştır.

2.4. Kullanıcı Arayüzü Tasarımı

Mobil-Öncelikli Yaklaşım: Tüm arayüz tasarımları, mobil cihazlarda tek elle kullanım, ekran boyutlarına göre uyum ve hızlı erişilebilirlik hedeflenerek yapılandırılmıştır.

Renk Paleti ve Tipografi: Sağlıklı yaşam temasıyla uyumlu, dinlendirici ve motive edici renkler tercih edilmiştir. Yazı tipleri okunabilirliği artıracak şekilde seçilmiştir.

Kullanıcı Testlerine Dayalı İyileştirme: Tasarım süreçleri boyunca A/B testleri, anketler ve kullanılabilirlik testleriyle kullanıcı deneyimi sürekli ölçülmüş ve tasarım iteratif olarak güncellenmiştir.

Temalar ve Kişiselleştirme: Kullanıcıların ihtiyaçlarına göre arayüzün teması, yazı büyüklüğü, dil ayarları gibi görsel unsurları değiştirebilecekleri opsiyonlar sunulmuştur.

2.5. Yazılım Geliştirme

Katmanlı Yazılım Mimarisi: Backend (API) ve frontend süreçleri ayrı takımlarda geliştirilmiş; her biri bağımsız olarak test edilmiş ve sonradan birleştirilmiştir.

Kod Standartları ve Versiyonlama: Kodlama sürecinde clean code ilkeleri ve SOLID prensipleri benimsenmiş; Git üzerinden dallanma stratejileri ve versiyon kontrolü sağlanmıştır.

CI/CD Uygulamaları: Continuous Integration ve Continuous Deployment süreçleri ile her güncelleme otomatik olarak test ve dağıtım boru hattından geçmiştir.

Dokümantasyon: Tüm API uç noktaları, veri modelleri ve modül kullanımları detaylı biçimde Swagger ve benzeri araçlarla belgelenmiştir.

2.6. Test ve Doğrulama

Çok Katmanlı Test Süreci:

Birim Testleri: Her modülün işlevselliği izole edilerek kontrol edilmiştir.

Entegrasyon Testleri: Modüllerin birlikte çalışma durumları test edilmiştir.

Fonksiyonel Testler: Kullanıcının karşılaştığı tüm senaryolar canlandırılmıştır.

Kabul Testleri (UAT): Gerçek kullanıcılarla sistemin son hali test edilmiştir.

Performans ve Yük Testleri: Uygulamanın anlık 1000+ kullanıcıyı kaldırıp kaldırmadığı, veri sorgularının ne kadar sürede tamamlandığı gibi metrikler ölçülmüştür.

Hata Yönetimi: Crash takip sistemleri (Firebase, Sentry) ile hatalar anlık olarak toplanmış ve sınıflandırılarak çözüme kavuşturulmuştur.

2.7. Dağıtım ve Kurulum

Bulut Tabanlı Altyapı: Uygulama, güvenli ve ölçeklenebilir bir bulut altyapısına (AWS) kurulmuştur. Gerekli sunucu yapılandırmaları yapılmış, trafik yönlendirme ve yük dengeleme sistemleri devreye alınmıştır.

Web: Web sürümü tarayıcı üzerinden kullanılabilir.

Kurulum Kılavuzları ve Yardım Dökümanları: Hem son kullanıcılar hem de teknik ekipler için adım adım kurulum kılavuzları hazırlanmıştır.

2.8. Kullanıcı Eğitimi

Eğitim Materyalleri: Kısa videolar, infografikler, interaktif kılavuzlar ve adım adım uygulama kullanımı gösteren dokümanlar hazırlanmıştır.

İnteraktif Eğitim Oturumları: Test kullanıcılarıyla canlı eğitimler gerçekleştirilmiş, eğitim sırasında oluşan sorular kayıt altına alınarak SSS (Sıkça Sorulan Sorular) bölümü güncellenmiştir.

Çeşitli Kullanıcı Gruplarına Özel Eğitim: Yaşlı bireyler, çocuklar ve sağlık profesyonelleri için farklı anlatım dillerine sahip içerikler hazırlanmıştır.

2.9. Bakım ve Destek

7/24 Teknik Destek: Kullanıcılar uygulama içi mesajlaşma, e-posta ve yardım formu üzerinden destek alabilmektedir.

Güncellemeler ve Yama Yönetimi: Sistem, düzenli olarak versiyonlanmış güncellemelerle desteklenmiş; her yeni sürümde hata düzeltmeleri ve yeni özellikler sunulmuştur.

Sistem İzleme: Uptime, sunucu yanıt süreleri, hata oranları gibi metrikler sürekli izlenmiş ve gerektiğinde otomatik müdahaleler devreye sokulmuştur.

Kullanıcı Geribildirim Süreci: Kullanıcılar uygulama içinden öneri ve şikayet bildirebilmekte, bu bildirimler sistematik olarak analiz edilerek iyileştirme listelerine dahil edilmektedir.

3. Performans Hedefleri

3.1. Gereksinimlerin Belirlenmesi

- Projenin ilk adımında, uygulamanın hedef kitlesi olan bireylerin beslenme alışkanlıkları, yaş grubu, sağlık geçmişi ve hedefleri (kilo verme, alma, koruma) gibi kriterler göz önünde bulundurularak fonksiyonel ve fonksiyonel olmayan gereksinimler titizlikle analiz edilmelidir.
- Kullanıcı araştırmaları, anketler, persona çalışmaları ve diyetisyenlerle yapılan odak grup görüşmeleri aracılığıyla detaylı veri toplanmalıdır.
- Gereksinimler sadece ilk aşamada belirlenip bırakılmamalı, proje süresince değişen kullanıcı beklentileri doğrultusunda iteratif olarak güncellenmelidir.
- Gereksinimlerin netliği, sonradan oluşabilecek hataların ve revizyonların önüne geçer; bu nedenle tüm gereksinimler dokümanite edilmeli ve versiyonlanmalıdır.

3.2. Analiz ve Tasarım

- Sistemin mimarisi modüler olacak şekilde planlanmalı, her bileşen (gıda takibi, su tüketimi, egzersiz entegrasyonu, kullanıcı hedefleri vb.) ayrı ayrı analiz edilmelidir.

- UX (Kullanıcı Deneyimi) ve UI (Kullanıcı Arayüzü) tasarımları bilimsel prensiplere dayanarak kullanıcı profiline özel tasarlanmalı; yaşlı bireyler, diyet sürecinde olanlar veya sağlık sorunları olan kullanıcılar için erişilebilirlik ön planda tutulmalıdır.
- Diyet öneri algoritmalarının kişiselleştirilmesi amacıyla yapay zeka ya da makine öğrenmesi kullanılması hedeflenebilir (Sisteme daha sonrasında entegre edilebilir.).
- Veritabanı yapısının esnek ve yatay olarak ölçeklenebilir olması, ileriye dönük sistem performansı ve veri büyüklüğü açısından kritik önem taşır.

3.3 Veritabanı Tasarımı

- Veritabanı, kullanıcıların tüm geçmiş verilerini (Diyet programı, vücut ölçümleri vs.) güvenle depolayabilecek şekilde şifreleme ve yedekleme sistemleriyle donatılmalıdır.
- Besin veri setleri, doğru kalorik bilgiler ve makro-mikro besin değerleriyle desteklenmeli; dış veri kaynakları (ör. USDA API) ile senkronize olabilme yeteneğine sahip olmalıdır.
- Veritabanı performansını artırmak için uygun endeksleme stratejileri, sorgu optimizasyonları ve veri bütünlüğü kontrolleri uygulanmalıdır.
- GDPR ve KVKK gibi veri koruma düzenlemelerine uygunluk sağlanmalıdır.

3.4 Kullanıcı Arayüzü Tasarımı

- Arayüz, kullanıcıların günlük girişlerini en az tıklama ile yapabileceği şekilde sade, sezgisel ve rehberleyici olmalıdır.
- Görsel unsurlar (renkler, ikonlar, grafikler) kullanıcıyı motive edecek yapıda tasarlanmalı; örneğin günlük hedefini tamamladığında olumlu geri bildirimler sunulmalıdır.
- Mobil kullanım için optimize edilmiş responsive tasarımlar oluşturulmalı; farklı ekran çözünürlüklerinde aynı deneyimi sunmalıdır.
- Arayüz sürekli olarak A/B testleriyle değerlendirilerek kullanıcı geri bildirimleri ışığında geliştirilmeye açık olmalıdır.

3.5 Yazılım Geliştirme

- Geliştirme süreci Iterative (Yinelemeli) metodolojisine uygun şekilde sprintlerle yürütülmeli; kullanıcıdan alınan geri bildirimler her iterasyonda ürüne yansıtılmalıdır.

- Kod kalitesi, modülerlik, okunabilirlik ve sürdürülebilirlik esas alınarak geliştirilmeli, yazılımın ileride başka sistemlerle entegre olabilmesi için esnek API mimarisi benimsenmelidir.
- CI/CD (Continuous Integration/Deployment) süreçleri ile yazılım sürekli olarak test edilerek dağıtılmalıdır.

3.6. Test ve Doğrulama

- Uygulamanın her fonksiyonu, kullanıcıya sunulmadan önce detaylı şekilde manuel ve otomatik testlerden geçirilmelidir.
- Gıda takibi doğruluğu, hedef uyarı sistemi ve raporlama modülü gibi hayati bileşenler için edge-case (sınır) test senaryoları uygulanmalıdır.
- Performans testleri ile sistemin eş zamanlı kullanıcı yükü altındaki davranışı ölçülmelidir.
- Güvenlik testleri, özellikle kullanıcı verisinin gizliliğini tehdit eden açıkları tespit etmek için düzenli olarak yapılmalıdır.

3.7. Dağıtım ve Kurulum

- Sistemin dağıtımı, versiyon yönetimiyle yapılandırılmış şekilde güvenli sunuculara yapılmalı; devreye alma süreci kullanıcıyı etkilemeyecek şekilde planlanmalıdır.
- Geliştirme, test ve üretim ortamları ayrı tutulmalı; staging sunucuları üzerinden ön izleme yapılmalıdır.
- Kurulum sonrası kullanıcılar üzerinde pilot testler uygulanarak olası problemler erken tespit edilmelidir.

3.8. Kullanıcı Eğitimi

- Kullanıcı eğitimi yalnızca yazılı kılavuzlarla sınırlı olmamalı; görsel anlatımlar, rehber videolar ve interaktif yardım araçlarıyla desteklenmelidir.
- Farklı kullanıcı seviyeleri (başlangıç, orta, ileri) için ayrı eğitim içerikleri hazırlanmalıdır.
- Eğitim sonunda kullanıcıların bilgilerini ölçen kısa sınavlar veya görev bazlı pratiklerle kullanıcı adaptasyonu ölçülmelidir.

3.9. Bakım ve Destek

- Yazılım yayımlandıktan sonra düzenli bakım takvimi oluşturularak sistem güncellemeleri ve hata düzeltmeleri yapılmalıdır.
- Kullanıcılardan gelen geri bildirimlerin sınıflandırıldığı bir destek portalı kurulmalı ve her sorun için SLA süreleri tanımlanmalıdır.
- Performans metrikleri (yüklenme süresi, işlem gecikmesi, sunucu yanıt süresi) düzenli olarak izlenerek gerektiğinde önleyici müdahaleler yapılmalıdır.

4. Öneriler

4.1. Gereksinimlerin Belirlenmesi

- Kapsamlı Kullanıcı Profili Oluşturulmalı: Uygulamanın hedef kitlesi yalnızca kilo vermek isteyenler değil; kilo almak, mevcut kilosunu korumak, sağlıklı beslenmek veya hastalıklarına göre özel diyet takip etmek isteyen kullanıcıları da içerecek şekilde genişletilmeli.
- Diyetisyen ve Kullanıcı Geri Bildirimleri Sürece Dahil Edilmeli: Gereksinim belirleme aşamasında hem uzman görüşleri (diyetisyen, beslenme uzmanı) hem de hedef kullanıcıların deneyimlerine dayalı anketler ve ön prototip testleri kullanılmalı.
- Fonksiyonel Olmayan Gereksinimler Belirginleştirilmeli: Sistem güvenliği, erişilebilirlik, mobil performans, veri gizliliği gibi teknik gereksinimler yazılı olarak tanımlanmalı ve ölçülebilir metriklerle desteklenmeli.

4.2. Analiz ve Tasarım

- Mimari Tasarımın Genişletilebilirliği Sağlanmalı: Modüler mimari tasarım benimsenmeli. İleride yapılabilecek egzersiz modülü, psikolojik destek bölümü veya gıda alışverişi entegrasyonları kolaylıkla eklenebilecek şekilde altyapı planlanmalı.
- Kişiselleştirme Katmanları Eklenmeli: Tasarım sürecinde kullanıcıların sağlık geçmişine, günlük alışkanlıklarına ve diyet tercihlerine göre kişiselleştirilmiş öneriler sunabilecek yapılar (örneğin kullanıcı segmentasyonu) kurgulanmalı.
- Etki Diyagramlarıyla Süreçler Görselleştirilmeli: Kullanıcı akışları ve sistemin iç etkileşimleri etki diyagramları ve kullanıcı senaryoları (use case) ile desteklenerek ekip içi iletişim artırılmalı.

4.3. Veritabanı Tasarımı

- Harici Besin Veritabanlarıyla Entegrasyon Planlanmalı: USDA veya TÜBİTAK destekli besin veri kaynakları ile eşzamanlı çalışan bir yapı kurulmalı; böylece sürekli güncel ve doğru kalori/makro-mikro verileri kullanılmalı.
- Veri Giriş Doğrulama Mekanizmaları Eklenmeli: Kullanıcıların elle girdiği verilerde hata yapmasını önlemek için otomatik tamamlayıcı, öneri sistemi veya QR barkod tarama gibi özellikler eklenmeli.
- Veri Yedekleme ve Felaket Kurtarma Senaryoları Oluşturulmalı: Kullanıcı verilerinin güvenliği için günlük yedekleme, versiyonlama ve yedekten geri dönüş politikaları geliştirilmeli.

4.4 Kullanıcı Arayüzü Tasarımı

- Minimalist ve Etkileşimli Tasarım İlkeleri Benimsenmeli: Kullanıcılar için dikkat dağıtmayan, sade, fakat etkileşimli (animasyon, bildirim, grafik gösterimler) bir tasarım benimsenmeli.
- Özelleştirilebilir Arayüz Opsiyonları Sunulmalı: Kullanıcılar, temaları (gece/gündüz modu), yazı boyutlarını veya görünüm düzenini ihtiyaçlarına göre kişiselleştirebilmeli.
- Mobil-Öncelikli (Mobile-First) Tasarım Uygulanmalı: Arayüz, özellikle mobil cihazlarda kullanım kolaylığı sağlayacak biçimde optimize edilmeli. Tek elle kullanım, büyük butonlar, hızlı veri girişi öncelikli olmalı.

4.5. Yazılım Geliştirme

- Temiz Kod Standartları Belirlenmeli: Kod yazımında SOLID prensipleri, yorum satırları ve versiyon kontrolü kuralları standart hâline getirilmeli.
- API-First Yaklaşımı Uygulanmalı: Diğer sistemlerle entegrasyon için (fitness uygulamaları, e-sağlık sistemleri) önceden planlanmış ve açık dökümantasyonlu API'ler geliştirilmeli.
- Kod Test Kapsamı Artırılmalı: Geliştirilen her fonksiyon için birim test yazımı zorunlu hâle getirilmeli ve kod kapsama oranı (code coverage) %80'in üzerinde hedeflenmeli.

4.6. Test ve Doğrulama

- Gerçek Kullanıcılarla Pilot Test Uygulanmalı: Özellikle kullanıcı arayüzü ve kalori hesaplama fonksiyonları, gerçek kullanıcılara test ettirilmeli ve kullanıcı deneyimi ölçülmeli.

- Otomatik Test Altyapısı Kurulmalı: CI/CD boru hatları içine entegrasyon ve regresyon testleri yerleştirilmeli, hata olasılığı insan faktöründen bağımsız hâle getirilmeli.
- Kullanıcı Rol Bazlı Test Senaryoları Geliştirilmeli: Diyetisyen, bireysel kullanıcı, kurumsal kullanıcı gibi farklı senaryolara özel testler kurgulanmalı.

4.7. Dağıtım ve Kurulum

- Kademeli Dağıtım Stratejisi Benimsenmeli (Canary Deployment): Yeni özellikler önce küçük bir kullanıcı grubuna sunulmalı, başarılı olursa tüm sisteme yayılmalı.
- Dağıtım Sonrası Kabul Testleri Otomatize Edilmeli: Sistem güncellemeleri sonrası uygulamanın temel fonksiyonlarının çalıştığını otomatik testlerle kontrol eden yapı kurulmalı.
- Kapsamlı Kullanım Kılavuzları ve Video Eğitimler Sunulmalı: Kurulum sonrası, kullanıcıların sistemi kendi başlarına kullanabilecekleri net ve görsel materyaller sağlanmalı.

4.8. Kullanıcı Eğitimi

- Gamification (Oyunlaştırma) Öğeleriyle Eğitim Süreci Zenginleştirilmeli: Yeni başlayan kullanıcılar için rozet kazanma, günlük görevler tamamlama gibi teşvik edici mekanizmalarla öğrenme süreci desteklenmeli.
- Farklı Kullanıcı Tiplerine Uygun Eğitim Materyalleri Hazırlanmalı: Yaşlı kullanıcılar, gençler ve sağlık personeli için farklı anlatım düzeylerine sahip içerikler oluşturulmalı.
- Eğitim Sonrası Anket ve Ölçüm Süreci Eklenmeli: Kullanıcıların eğitimi ne kadar kavradığı ölçülerek gerekli güncellemeler sistemli şekilde yapılmalı.

4.9. Bakım ve Destek

- Proaktif Destek Mekanizması Kurulmalı: Kullanıcı sistemde hatayla karşılaşmadan önce sistemsel uyarılar, öneriler ve çözümler sunulmalı (örneğin: "Son 3 gündür su tüketimi girilmedi, hatırlatma yapılısın mı?").
- Kapsamlı Geri Bildirim Mekanizması Entegre Edilmeli: Kullanıcılar hem uygulama içi hem de web portalı üzerinden kolaylıkla şikayet, öneri veya istek bildirebilmeli.
- Sistem İzleme Panelleri (Monitoring Dashboard) Oluşturulmalı: Canlı sistem izlenmeli, performans düşüklüğü veya hatalarda otomatik bildirim sistemleri kurulmalı.

- Uzun Vadeli Güncelleme Takvimi Belirlenmeli: Belirli periyotlarla sistemin güvenlik, performans ve kullanıcı deneyimi açısından gözden geçirilmesi planlanmalı ve kamuya açık yol haritası yayımlanmalı.

5. Sonuç

Bu entegrasyon raporu, "Online Diyet Takip Uygulaması" projesinin yönetimini ve ilgili süreçleri detaylı bir şekilde ele almaktadır. Projenin her aşaması için belirlenen görevler, süreçlerin etkin bir şekilde yönetilmesini ve uygulamanın başarıyla hayata geçirilmesini sağlamak açısından kritik öneme sahiptir. Bu rapor, proje sürecinin başarıyla yürütülmesi ve hedeflenen işlevlerin eksiksiz şekilde entegre edilmesi için bir çerçeve sunmaktadır. Her aşamada tanımlanan görevler ve performans kriterleri, proje ekibinin hedefe odaklanmasına yardımcı olmuş ve uygulamanın başarılı bir şekilde tamamlanmasına katkı sağlamıştır.

Entegrasyon Yönetimi

