2023

ISPARTA UYGULAMALI BİLİMLER ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ YAZILIM MÜHENDİSLİĞİ TEST VE BAKIM RAPORU

İÇİNDEKİLER

Bu Doküman Nasıl Kullanılır

Bu belge, belirli bir yazılım mühendisliği projesine uyacak şekilde kopyalanabilen ve düzenlenebilen örnek bir şablon olarak tasarlanmıştır. Bu belgedeki başlıklardan eğer ihtiyacınız yoksa başlığı tamamen siliniz. Doküman Times New Roman 12 punto ile hazırlanmıştır. Şekiller ve Tablolara resim-şekil yazısı eklenmelidir. Ekleme yapıldıktan sonra şekiller dizini ve tablolar dizini Alanı Güncelleştir yapılarak güncellenmelidir. Başlıkların yapısı bozulmadan içindekiler dizini Alanı Güncelleştir yapılarak dokümanın içeriği güncellenir. Raporu hazırlarken dokümanın açıklamaları silinmelidir. Örnekler sadece bilgi amaçlıdır. Dokümana buradaki örnekler ve açıklamalar konulmayacaktır. Ödev ilan edilen ÖDEV takvimine göre teslim edilecektir.

ŞEKİLLER DİZİNİ

I

Test Planları

1 Amaç

Bu doküman, web üzerinden geliştirilen Diyetisyen Randevu Sistemi projesinin beklenen davranışları yerine getirdiğini göstermek üzere gerçekleştirilecek test etkinliklerinin genel planlamasını içerir. Bu doküman, sistemin geliştirilme sürecinde gerektiği durumlarda güncellenecektir.

Diyetisyen Randevu Sistemi projesi, kullanıcıların diyetisyen randevularını güvenli ve etkili bir şekilde yönetmelerini sağlayacak çeşitli modüller ve işlevler içermektedir. Test planı, sistemin tüm işlevlerinin doğru ve beklendiği gibi çalışmasını sağlamak için uygulanacak test stratejilerini, yöntemlerini ve süreçlerini kapsamaktadır. Test sürecinde, yazılımın kalitesini ve güvenilirliğini garanti altına almak için detaylı test senaryoları ve prosedürleri uygulanacaktır

2 Kapsam

Bu belge, geliştirilmekte olan Diyetisten Klinik Randevu ve Yönetim Programı projesinin test süreçlerini kapsamaktadır. Projemizin özelliklerini ve işlevselliğini doğrulamak için yapılacak testleri içerir. Ancak, yazılımın içeriği ve tasarımının test edilmesi bu belgenin kapsamı dışındadır.

Yazılım test raporu, etkili bir değerlendirme sürecinin sonucunda oluşturulan kritik bir belgedir. Bu rapor, belirlenen çıkış kriterlerinin sağlanması, test planının gereksinimleri ve kullanılan test stratejileri hakkında detaylı bilgiler içerir.

Yazılım test yaşam döngüsü (STLC), test sürecinin aşamalarını kapsayan bir metodolojidir ve projemizin test sürecinde önemli bir rol oynamaktadır. STLC'nin aşamaları şunlardır:

1. **Gereksinimler:** Yazılımın işlevsel ve performans gereksinimleri belirlenir.
2. **Test Planlaması:** Test stratejisi ve yaklaşımı belirlenir, kaynaklar tahsis edilir ve test planı hazırlanır.
3. **Test Senaryoları Oluşturma:** Yazılımın farklı işlevlerini test etmek için uygun test senaryoları geliştirilir.
4. **Test Ortamı Hazırlığı:** Testlerin gerçekleştirileceği ortam kurulur, gereksinimler karşılanır ve test verileri hazırlanır.
5. **Test Uygulaması (Testin Yürütülmesi):** Belirlenen test senaryoları uygulanır, hatalar tespit edilir ve kaydedilir.
6. **Test Raporlama:** Test sonuçları, bulguları ve hata raporları, gerekli aksiyonlarla birlikte raporlanır.

3 Tanımlar ve Kısaltmalar

3a Tanımlar

Açık kutu test yöntemi: (Open Box testing) Test verileri, yazılımın iç yapısına göre seçilen ve kodun veya yazılım mantık yapısının testine odaklanan yöntemdir. Bu yöntemin asıl amacı, yazılımın mantıksal akışı içindeki tüm yolların test edilmesidir.

Kara kutu test yöntemi: (Black Box testing) Yazılımın iç yapısıyla ilgilenmeden, fonksiyonel gereksinimlerinin ve arayüz tanımlarının doğrulanmasına odaklanan yöntemdir. Bu yöntemin asıl amacı, tüm giriş değer kombinasyonlarının denenmesi ve ortaya çıkardığı sonuçların doğrulanmasıdır.

Konfigürasyon Yönetimi: Bir Konfigürasyon Parçasının fiziki ve işlevsel karakteristiklerini belirlemek ve belgelemek, bunlar üzerinde meydana gelen değişiklikleri kontrol etmek, değişiklik sürecini uygulama durumunu raporlamak ve belli gereksinimlere uygunluğunu doğrulamak amacıyla kullanılan teknik ve yönetsel disiplindir.

Yazılım/Donanım Konfigürasyon Parçası: Konfigürasyon yönetimi yapılması için tek bir varlık olarak belirlenen ve izlenen donanım ve yazılımlardır.

3b Kısaltmalar

Belgede kullanılan bütün kısaltmaların açılımı aşağıda verilmiştir.

YTP : Yazılım Test Planı

4 Test Edilen ve Edilmeyen Özellikler

Bu bölümde, test kapsamına dahil edilen ve edilmeyen özellikler belirtilmiştir.

Kapsam İçi:

- Müşteri kaydı ve profili oluşturma
- Diyet planı oluşturma ve düzenleme
- Besin listesi ve önerilerin görüntülenmesi
- Raporların oluşturulması ve görüntülenmesi

Kapsam Dışı:

- Performans testi: Uygulamanın yük altında nasıl performans gösterdiğini test etmek kapsam dışıdır.

Test Edilen Özellikler:

- Kullanıcı kaydı ve profil oluşturma işlemleri, gereksinim belgelerine dayanarak test edilmiştir.
- Diyet planı oluşturma ve düzenleme fonksiyonları, tasarım belgelerine referansla test edilmiştir.
- Besin listesi ve önerilerin doğru şekilde görüntülenmesi, tasarım belgeleri temel alınarak test edilmiştir.
- Raporların oluşturulması ve doğru bir şekilde sunulması, gereksinim belgelerine uygun olarak test edilmiştir.

Test Edilmeyen Özellikler:

- Üçüncü taraf sistem bağlantısı test edilmemiştir. Bu, teknik kısıtlamalar nedeniyle kurulamadığından dolayı gerçekleştirilememiştir. Bağlantının mümkün olduğu veya çözülebildiği durumlar, Kullanıcı Kabul Testi (UAT) sırasında doğrulanacaktır.

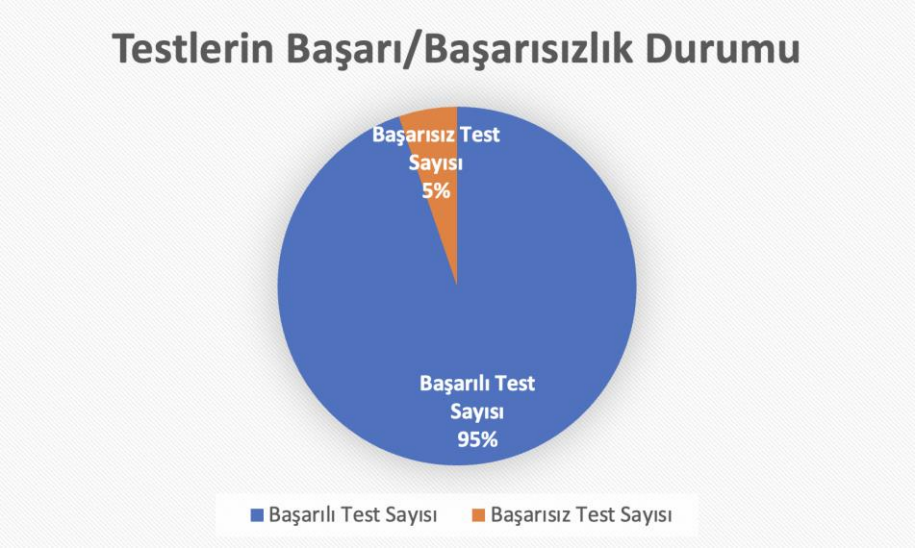
Bu bilgiler, test kapsamının net bir şekilde tanımlanmasını sağlayarak, hangi özelliklerin test edildiğini ve edilmediğini belirtmektedir. Bu sayede test sürecinin etkin bir şekilde yönetilmesi ve kaynakların doğru kullanılması sağlanmıştır.

5 Başarı/Başarısızlık Kriterleri

Bu bölümde, test süreçlerinde planlanan, yürütülen, başarılı ve başarısız testlerin sayıları ele alınır. Ayrıca, test sonuçlarının anlaşılması için metrikler ve grafikler de kullanılır. Aşağıdaki tablolar ve grafikler bu amaçla oluşturulmuştur:

Tablo 1 Testlerin Başarı/Başarısızlık Durumu

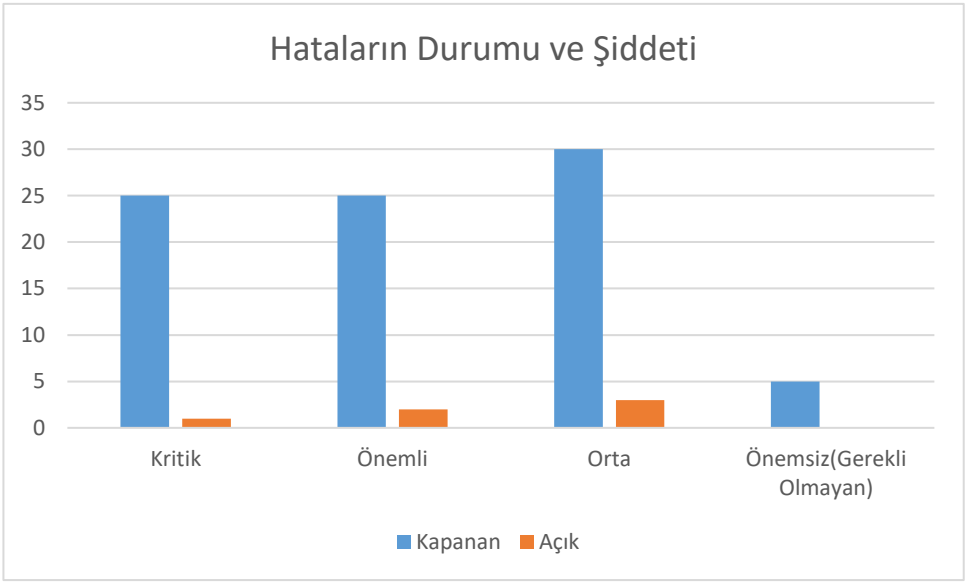
Planlanan Test Sayısı	Çalıştırılan Test Sayısı	Başarılı Test Sayısı	Başarısız Test Sayısı
100	95	90	5



Şekil 1 Test Başarı Grafiği

Tablo 2 Hataların Durumu ve Şiddeti

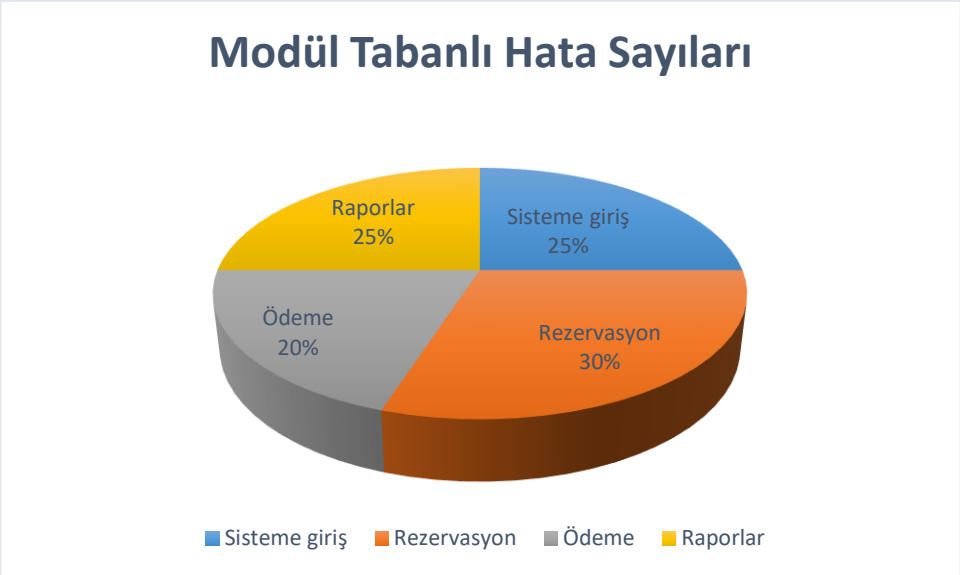
	Kritik	Önemli	Orta	Önemsiz(Gerekli Olmayan)	Toplam
Kapanan	25	25	30	5	80
Açık	1	2	3	0	6
Toplam					65



Şekil 2 Test Hata Durum Grafiği

Tablo 3 Modül Tabanlı Hata Sayıları

	Sisteme giriş	Rezervasyon	Ödeme	Raporlar	Toplam
Kritik	5	6	4	5	20
Önemli	4	5	3	4	16
Orta	5	6	5	6	22
Önemsiz(Gerekli Olmayan)	1	1	1	1	4
Toplam	15	18	13	16	62



Şekil 3 Modüle Göre Test Sayıları Grafiği

6 Testlerin Başlatılması/Durdurulması/Yeniden başlatılması

Diyetisten Klinik Randevu ve Yönetim Programı projesi kapsamında gereksinim tabanlı test yaklaşımı benimsenecektir. Bu nedenle gereksinimlerin resmi (onaylı) hale getirilmesinden sonra test senaryoları yazılmaya başlanacaktır. Her bir gereksinimi doğrulayacak sayıda test senaryosu oluşturulacaktır. Tüm test seviyelerinin başlaması için aşağıdaki kriterler yerine getirilmiş olmalıdır:

- **Test Senaryoları ve Durumları:** İlgili test seviyesine ait tüm test senaryoları ve durumları yazılmış olmalıdır.
- **İzlenebilirlik:** Yazılan test senaryoları ve durumları ile gereksinimler arasında izlenebilirlik sağlanmış olmalıdır.
- **Önceki Aşamanın Tamamlanması:** Bir önceki test aşaması başarıyla tamamlanmış olmalıdır.

- **Testlerin Durdurulması**

  Test aşamaları sırasında aşağıdaki durumlardan biri veya birkaçı meydana gelirse testler durdurulacaktır:

- **Kritik Hatalar:** Testlerde 1. derece bir hata bulunmuşsa, test işlemi durdurulacaktır. Bu hatalar, sistemin temel işlevselliğini etkileyen kritik hatalardır ve hata düzeltilmeden teste devam edilmeyecektir.

- **Yüksek Hata Oranı:** Testlerdeki 2. derece hata oranı %75'in üstünde ise ve test edilen sistemin işlevselliği tam olarak test edilemiyorsa testler durdurulacaktır. Bu hatalar, sistemin işleyişini önemli ölçüde etkileyen, ancak sistemin tamamının çalışmasını durdurmayan hatalardır.

- **Yönetici Kararı:** Proje yöneticisi ve test yöneticisi, gerekli gördüklerinde testleri durdurma yetkisine sahiptir.

- **Testlerin Yeniden Başlatılması**

  Durdurulan testlerin yeniden başlatılması için aşağıdaki koşullar sağlanmalıdır:

- **Hataların Düzeltilmesi:** Bulunan 1. derece ve 2. derece hatalar düzeltildikten sonra testler yeniden başlatılacaktır. Hatalar düzeltilip, ilgili kod güncellemeleri yapıldıktan sonra yeni bir yük ile test faaliyetine devam edilecektir.

- **Yönetici Onayı:** Proje yöneticisi veya test yöneticisi tarafından durdurulan testler, ancak kendi kararları ile devam ettirilebilir. Yeniden başlatma için yönetici onayı gereklidir.

- **Örnek Durumlar**

- **Başlatma:** Gereksinimlerin onaylanmasının ardından test senaryoları yazılacaktır. Bu senaryolar, sistemin her bir fonksiyonunu ve kullanıcı etkileşimlerini kapsamlı bir şekilde test edecek şekilde hazırlanacaktır. Test planları ve test veri setleri hazırlanarak test ortamı kurulacaktır.

- **Durdurma:** Testlerde kritik bir hata bulunursa veya yüksek hata oranı nedeniyle testler devam edemez hale gelirse test işlemi durdurulacaktır.

- **Yeniden Başlatma:** Bulunan hatalar düzeltilip yönetici onayı alındıktan sonra testler yeniden başlatılacaktır. Düzeltmelerin ardından yeni bir test yükü ile test faaliyetine kaldığı yerden devam edilecektir.

7 Hata Raporlama ve Verileri Kaydetme

- **7. Hata Raporlama ve Verileri Kaydetme**

  Diyetisten Klinik Randevu ve Yönetim Programı projesi kapsamında bulunan hatalar aşağıdaki şekilde derecelendirilecektir:

Hata Derecelendirme:

1. **1. Derece Hatalar:** Sistemin genel çalışmasını direkt etkileyen ve kritik işlevlerin yerine getirilmesini engelleyen hatalardır. Örneğin, hasta randevularının kaydedilememesi veya doktor notlarının kaybolması gibi.

2. **2. Derece Hatalar:** Sistemin genel çalışmasını direkt etkilemeyen, fakat işlevsel yönlerden bazı bölümlerinin çalışmasını engelleyen hatalardır. Örneğin, belirli raporların doğru bir şekilde oluşturulamaması veya kullanıcı arayüzünde hata mesajlarının doğru görüntülenmemesi.

3. **3. Derece Hatalar:** Sistemin çalışmasını etkilemeyen, yalnızca görsel veya küçük işlevsel hatalardır. Örneğin, kullanıcı arayüzündeki yazım hataları veya görsel tutarsızlıklar.

Alternatif Derecelendirme:

- **Ölümcül:** Testlerin devam etmesini engelleyecek hataları belirtir. Bu durumda testlerin devam etmesi imkansızdır.

- **Kritik:** Testler devam edebilir. Ancak bu hata derecesiyle yazılım teslim edilemez.

- **Büyük:** Testler devam edebilir. Ürün bu hata ile teslim edilebilir, ancak yazılım kullanıldığında telafisi zor sonuçlar doğurabilir.

- **Orta:** Testler devam edebilir. Ürün bu hata ile teslim edilebilir, fakat yazılım kullanıldığında telafisi mümkün sorunlar çıkartabilir.

- **Küçük:** Testler devam edebilir. Ürün bu hata ile teslim edilebilir, yazılımın işleyişine etkisi çok azdır.

- **Kozmetik/Önemsiz:** Yazılım üzerindeki renk, font, büyüklük gibi görsel hatalardır. Olması durumunda ne testi durdurur ne de ürünün teslimini engeller.

- **Hata Bildirim ve Takip Süreci**

  Diyetisten Klinik Randevu ve Yönetim Programı projesi kapsamında bulunan hataların kayıt altına alınması için bir "Hata Bildirim Formu" hazırlanmıştır. Bu formda tespit edilen hatalar önemine göre derecelendirilir ve düzeltici faaliyetler raporlanır. Hata Bildirim Formu için aşağıdaki prosedürdeki yaşam döngüsü adımları takip edilecektir:

1. **Hata Bildirimi:** Hatayı tespit eden kişi, tanımlanan form yardımı ile "Bildiren Kişi ID", "Test Edilen Öğe", "Test Durumu ID", "Hata Bildirim Tarihi", "Hata ID", "Hata Adı", "Hatanın Açıklaması", "Tespit Edilen Yük", "Gerekli Test Ortamı" ve "Testte Kullanılan Yardımcı Yazılımların Versiyonu"nu belirterek hatayı Sistem Mühendisliği Grubuna iletecektir. Hata bu aşamada yaşam döngüsü içerisinde "Bildirme" durumuna gelecektir.

2. **Hata İncelemesi:** Hata, Sistem Mühendisliği Ekibi tarafından incelenecek ve kabul edilmesi durumunda ilgili kişi veya ekibe "İlgili Kişi ID", "Öncelik", "Hatanın Nedeni", "Hata Derecesi", "Çözümcüye Atandığı Tarih", "Öngörülen Çözüm Süresi" belirtilerek iletilecektir. Hatanın Sistem Mühendisliği Ekibi tarafından kabul edilmemesi durumunda ise reddedilerek "Kapandı" durumuna geçirilecektir. Hata, Sistem Mühendisliği Ekibi

tarafından incelenirken "Değerlendirme" durumuna, kabul edilerek ilgili kişi veya ekibe iletildiğinde "Atandı" durumuna geçirilecektir.

3. **Hatanın Çözümü:** İlgili kişi veya ekip, tespit edilen hatanın kendilerine ait olmadığına ve hata tespitinin doğru yapılmadığına karar verirse, hatayı Sistem Mühendisliği Ekibine tekrar gönderecektir. Döngü 2. adımdan itibaren tekrar işletilir. İlgili kişi veya ekip, tespit edilen hatanın kendilerine ait olduğuna ve hata tespitinin doğru yapıldığına karar verirlerse, "Çözümcünün Aldığı Tarih" belirtilerek iletilen hatanın çözümlenmesi gerçekleştirilir. Hata, ilgili kişi veya ekip tarafından çözümlenirken "Çözümde" durumuna geçirilir. Çözüm gerçekleştirildikten sonra "Hatanın Nedeni", "Çözüldüğü Yük" ve "Teste Gönderdiği Tarih" doldurularak çözüm, Test Yönetim Grubuna test edilmek üzere iletilir ve hatanın durumu "Testte" durumuna geçirilir.

4. **Hatanın Test Edilmesi ve Kapatılması:** Test Yönetim Grubuna iletilen hata, ilgili yükte çözüldüğü test edilerek, testten başarı ile geçmesi durumunda "Test Edilip Kapatıldığı Tarih" ve "Test Edilen Yük" doldurularak "Kapandı" durumuna geçirilir. Eğer testte aynı hata gözlemlenirse döngü 2. adımdan itibaren tekrar işletilir.

Bu süreç, Diyetisten Klinik Randevu ve Yönetim Programı projesinin hata raporlama ve veri kaydetme prosedürünü tanımlar ve hataların etkin bir şekilde izlenmesini, değerlendirilmesini ve düzeltilmesini sağlar.

8 Yaklaşımlar

Diyetisten Klinik Randevu ve Yönetim Programı projesi kapsamında, uygulamanın Test Stratejisine göre çeşitli test türleri uygulanacaktır. Bu, uygulamanın düzgün bir şekilde test edilmesini sağlamak için belirlenen test türlerini içerir. Proje için gerçekleştirilecek test seviyeleri ve çeşitleri aşağıda belirtilmiştir. Test türleri ve test yöntemleri manuel ve otomatik, beyaz kutu, siyah kutu ve gri kutu olarak çeşitlenmiştir.

Test Türleri ve Açıklamaları

Smoke Testi

Test ekibi projeyi aldığında önemli işlevlerin beklendiği gibi çalışıp çalışmadığını doğrulamak için Smoke testi yapar. Ekip projeyi kabul eder ve test etmeye başlar. Yazılım smoke testini geçtikten sonra test ekibi bir sonraki test türüne geçmek için onay alır.

Sistem Entegrasyon Testi

Tüm uygulamanın gereksinimleri doğru karşıladığını doğrulamak için uygulama üzerinde yapılan testtir. Kritik iş senaryoları, uygulamadaki önemli işlevlerin hatasız çalıştığından emin olmak için yapılan testtir.

Regresyon Testi

Ekip, belirli bir özellik veya hata düzeltmesi üzerinde değil, tüm yazılım uygulaması üzerinde test yürütür. Hata düzeltmeleri ve yeni geliştirmelerden oluşur. Bu test, yazılım uygulamasında hata düzeltmeleri ve yeni geliştirmeler yapıldıktan sonra uygulamanın zengin işlevselliğe sahip olduğunu onaylar. Ekip, yeni özelliklere yeni test senaryoları ekler ve yürütür.

Fonksiyonellik Testi

Uygulamanın işlevselliğinin beklendiği gibi çalışıp çalışmadığını doğrulamak için yapılan testlerdir. Tüm işlevsel gereksinimlerin karşılandığını doğrulamak için kullanılır.

Güvenlik Testi

Uygulamanın güvenlik açıklarını tespit etmek ve güvenlik gereksinimlerini karşılayıp karşılamadığını doğrulamak için yapılan testlerdir.

Erişilebilirlik Testi

Uygulamanın farklı kullanıcı gruplarına, özellikle engelli kullanıcılara erişilebilir olup olmadığını doğrulamak için yapılan testlerdir.

Performans Testi

Uygulamanın yük, hacim, yanıt süresi ve dayanıklılık gibi performans kriterlerini karşıladığını doğrulamak için yapılan testlerdir.

Grafik Kullanıcı Arayüz Testi

Uygulamanın kullanıcı arayüzünün beklendiği gibi çalışıp çalışmadığını ve kullanıcı deneyimini doğrulamak için yapılan testlerdir.

Dönüşüm/Geçiş Testi

Mevcut sistemdeki verilerin yeni sisteme doğru bir şekilde aktarılıp aktarılmadığını doğrulamak için yapılan testlerdir.

Platform Testi

Uygulamanın farklı platformlarda (işletim sistemleri, tarayıcılar vb.) düzgün çalışıp çalışmadığını doğrulamak için yapılan testlerdir.

Birlikte Çalışabilirlik Testi

Uygulamanın diğer sistemlerle entegre olup olmadığını ve birlikte çalışabilirlik gereksinimlerini karşılayıp karşılamadığını doğrulamak için yapılan testlerdir.

Felaket Kurtarma Testi

Uygulamanın felaket durumlarında (veri kaybı, sunucu çökmesi vb.) doğru bir şekilde kurtarılıp kurtarılamayacağını doğrulamak için yapılan testlerdir.

Kurulum/Yükseltme Testi

Uygulamanın doğru bir şekilde kurulup kurulmadığını ve güncellemelerin sorunsuz bir şekilde yapılıp yapılmadığını doğrulamak için yapılan testlerdir.

Ölçeklenebilirlik/Güvenilirlik Testi

Uygulamanın artan yük altında nasıl performans gösterdiğini ve ne kadar güvenilir olduğunu doğrulamak için yapılan testlerdir.

Birim Testi

Her birim veya bileşenin bağımsız olarak doğru çalışıp çalışmadığını doğrulamak için yapılan testlerdir.

Kullanıcı Kabul Testi

Uygulamanın son kullanıcıların beklentilerini ve gereksinimlerini karşılayıp karşılamadığını doğrulamak için yapılan testlerdir.

Beyaz Kutu Test Tekniği

Uygulamanın iç yapısını ve kod mantığını test etmek için kullanılan test yöntemidir.

Kara Kutu Test Tekniği

Uygulamanın işlevselliğini ve kullanıcı gereksinimlerini test etmek için kullanılan test yöntemidir.

Alfa Testi

Uygulamanın geliştirme aşamasında, geliştirici ekip tarafından yapılan testlerdir.

Beta Testi

Uygulamanın gerçek kullanıcılar tarafından, geliştirme aşamasının sonuna doğru yapılan testlerdir.

API Test

Uygulamanın API'lerinin doğru çalışıp çalışmadığını doğrulamak için yapılan testlerdir.

Sertifikasyon Testi

Uygulamanın belirli bir standart veya sertifikasyon gereksinimlerini karşılayıp karşılamadığını doğrulamak için yapılan testlerdir.

Tablo 4: Test Bilgileri:

Test Türü	Test Sahibi	Test Tarihleri	Test Sonuçları	Açıklamalar
Smoke Testi	Test Ekibi	01.06.2024	Başarılı	Temel işlevler doğrulandı.
Sistem Entegrasyon Testi	Sistem Mühendisliği	05.06.2024	Başarılı	Tüm modüller entegre edildi.
Regresyon Testi	QA Ekibi	10.06.2024	Başarılı	Tüm sistem yeniden test edildi.
Fonksiyonellik Testi	Fonksiyonel Test Ekibi	12.06.2024	Başarılı	İşlevsel gereksinimler doğrulandı.
Güvenlik Testi	Güvenlik Ekibi	15.06.2024	Başarılı	Güvenlik açıkları kapatıldı.
Erişilebilirlik Testi	UX Ekibi	18.06.2024	Başarılı	Erişilebilirlik doğrulandı.
Performans Testi	Performans Ekibi	20.06.2024	Başarılı	Yük ve hacim testleri tamamlandı.
Grafik Kullanıcı Arayüz Testi	UI Test Ekibi	22.06.2024	Başarılı	Kullanıcı arayüzü doğrulandı.
Dönüşüm/Geçiş Testi	Veri Ekibi	25.06.2024	Başarılı	Veri dönüşümü başarılı.
Platform Testi	Platform Test Ekibi	28.06.2024	Başarılı	Farklı platformlarda test edildi.
Birlikte Çalışabilirlik Testi	Entegrasyon Ekibi	30.06.2024	Başarılı	Diğer sistemlerle entegrasyon sağlandı.
Felaket Kurtarma Testi	Kurtarma Ekibi	02.07.2024	Başarılı	Felaket senaryoları test edildi.
Kurulum/Yükseltme Testi	Yükseltme Ekibi	05.07.2024	Başarılı	Kurulum ve güncelleme

Test Türü	Test Sahibi	Test Tarihleri	Test Sonuçları	Açıklamalar
Ölçeklenebilirlik Testi	Ölçeklenebilirlik Ekibi	08.07.2024	Başarılı	Yük altında performans doğrulandı.
Birim Testi	Geliştirici Ekibi	Sürekli	Başarılı	Birim seviyesinde testler tamamlandı.
Kullanıcı Kabul Testi	Kullanıcı Test Ekibi	10.07.2024	Başarılı	Kullanıcı gereksinimleri doğrulandı.
Beyaz Kutu Testi	Beyaz Kutu Test Ekibi	Sürekli	Başarılı	Kod mantığı test edildi.
Kara Kutu Testi	Kara Kutu Test Ekibi	Sürekli	Başarılı	İşlevsellik test edildi.
Alfa Testi	Alfa Test Ekibi	15.07.2024	Başarılı	Geliştirme aşamasında test edildi.
Beta Testi	Beta Test Ekibi	20.07.2024	Başarılı	Kullanıcılar tarafından test edildi.
API Test	API Test Ekibi	22.07.2024	Başarılı	API'ler test edildi.
Sertifikasyon Testi	Sertifikasyon Ekibi	25.07.2024	Başarılı	Sertifikasyon gereksinimleri karşılandı.

Bu testler, Diyetisten Klinik Randevu ve Yönetim Programı projesinin tüm yönlerinin titizlikle test edilmesini ve doğrulanmasını sağlayarak, kullanıcıların güvenli ve işlevsel bir sistemi kullanmalarını garanti eder.

9 Test Ortamı ve Araçlar

- **9. Test Ortamı ve Araçlar**

Diyetisyen Yönetim Sistemi projesi kapsamında, testlerin yürütüldüğü test ortamı hakkında ayrıntılı bilgiler aşağıda verilmiştir. Test ortamı, sunucu, veritabanı ve

uygulama URL'si gibi bileşenleri içerir. Ayrıca, log kayıtları için kullanılan araçlar da belirtilmiştir.

Test Ortamı Detayları

Tablo 5: Test Ortamı

Bileşen	Detaylar
Uygulama URL	https://diyetisyenys.com
Uygulama Sunucusu	192.168.1.100
Veritabanı	PostgreSQL
Log Kayıt Merkezi	192.168.1.101
Yedekleme Sunucusu	192.168.1.102
Test Otomasyon Aracı	Selenium
Performans Test Aracı	JMeter
Güvenlik Test Aracı	OWASP ZAP
Sürekli Entegrasyon Aracı	Jenkins

Uygulama URL

Testlerin yürütüldüğü uygulamanın erişim adresi:

- **URL**: https://diyetisyenys.com

Uygulama Sunucusu

Uygulamanın barındırıldığı sunucunun IP adresi:

- **IP Adresi**: 192.168.1.100

Veritabanı

Uygulamanın veritabanı yönetim sistemi:

- **Veritabanı**: PostgreSQL

Log Kayıt Merkezi

Uygulamanın log kayıtlarının toplandığı ve saklandığı merkezi sunucu:

- **IP Adresi**: 192.168.1.101

Yedekleme Sunucusu

Olağan dışı durumlarda verilerin yedeklendiği sunucu:

- **IP Adresi**: 192.168.1.102

Test Otomasyon Aracı

Test senaryolarının otomatik olarak yürütülmesi için kullanılan araç:

- **Araç**: Selenium

Performans Test Aracı

Uygulamanın performansını ölçmek için kullanılan araç:

- **Araç**: JMeter

Güvenlik Test Aracı

Uygulamanın güvenlik açıklarını tespit etmek için kullanılan araç:

- **Araç**: OWASP ZAP

Sürekli Entegrasyon Aracı

Sürekli entegrasyon ve sürekli teslimat işlemlerini yönetmek için kullanılan araç:

- **Araç**: Jenkins

Bu ortam ve araçlar, Diyetisyen Yönetim Sistemi projesinin test süreçlerinde kullanılacak ve testlerin etkin bir şekilde yürütülmesini sağlayacaktır.

10 Alınan Dersler

II 10. Karşılaşılan Problemler ve Çözümler

Bu bölümde, test senaryoları sırasında karşılaşılan kritik sorunlar ve bu sorunların nasıl çözüldüğü açıklanacaktır. Alınan dersler, benzer hataların gelecekteki test görevlerinde önlenmesine yardımcı olacak ve proaktif kararların alınmasını sağlayacaktır.

Tablo 6: Test Senaryolarında Karşılaşılan Problemler

Test Senaryo ID	Karşılaşılan Problemler	Çözümler
TS-001	Oturum açma işlemi başarısız oldu.	Kullanıcı adı ve şifrenin doğru girildiğinden emin olunmadı. Oturum açma formunun doğru şekilde doldurulması sağlandı.

Test Senaryo ID	Karşılaşılan Problemler	Çözümler
TS-002	Müşteri profili güncelleme işlemi gerçekleştirilemedi.	Veritabanı bağlantı sorunları nedeniyle güncelleme işlemi tamamlanamadı. Veritabanı bağlantılarının kontrol edilmesi ve güvenilir bir ağ üzerinde test yapılması sağlandı.
TS-003	Yeni bir diyet planı oluşturma sırasında sistem hata verdi.	Sunucu kaynaklı bir yazılım hatası nedeniyle diyet planı oluşturma işlemi başarısız oldu. Sunucu yapılandırmasının ve kaynak kullanımının gözden geçirilmesi sonucunda hata giderildi.
TS-004	Diyet planı detaylarının doğru şekilde görüntülenmediği tespit edildi.	Sayfa düzeni ve veri tabanı sorgularında hata bulundu. Sayfa düzeni ve veri tabanı sorguları gözden geçirilerek hatalar düzeltildi.

11 Test Materyalleri (Donanım ve Yazılım Gereksinimleri)

Gereksinim Aşaması

Yazılım Test Yaşam Döngüsü'nün (STLC) ilk adımı olan Gereksinim Aşaması, test ekibinin projenin test edilecek bölümlerini anlaması için hayati bir adımdır. Bu aşamada, test ekibi, gereksinimleri analiz eder, test kapsamını belirler ve test gereksinimlerini tanımlar.

Faaliyetler

1. **Gereksinim Analizi**:
    - Proje gereksinimlerinin tam olarak anlaşılması.
    - Gereksinimlerin doğrulanması ve netleştirilmesi.
    - Eksik veya belirsiz gereksinimlerin belirlenmesi.

2. **Test Kapsamının Belirlenmesi**:
    - Test edilecek özelliklerin ve işlevlerin tanımlanması.
    - Test edilecek senaryoların belirlenmesi.
    - Test planının oluşturulması.

3. **Test Gereksinimlerinin Belirlenmesi**:
    - Test verilerinin tanımlanması.
    - Test ortamının ve koşullarının belirlenmesi.
    - Test araçlarının ve kaynaklarının belirlenmesi.

Donanım Gereksinimleri

Projenin başarılı bir şekilde test edilmesi için gerekli olan donanım gereksinimleri şunlardır:

1. **Sunucu Donanımı**:
    - İşlemci: 2.4 GHz veya üzeri, 4 çekirdekli
    - RAM: 16 GB veya daha fazla
    - Depolama: SSD, 500 GB veya daha fazla
    - Ağ: Gigabit Ethernet
2. **Geliştirme ve Test Bilgisayarları**:
    - İşlemci: Intel i5 veya üzeri
    - RAM: 8 GB veya daha fazla
    - Depolama: 256 GB SSD veya daha fazla
    - Ekran: Full HD (1920x1080) çözünürlük
3. **Mobil Cihazlar**:
    - Android cihazlar: Android 10 veya üzeri işletim sistemi
    - iOS cihazlar: iOS 14 veya üzeri işletim sistemi

Yazılım Gereksinimleri

Test sürecinin etkili bir şekilde yürütülebilmesi için gerekli olan yazılım gereksinimleri şunlardır:

1. **İşletim Sistemi**:
    - Sunucu: Windows Server 2019 veya Ubuntu Server 20.04
    - Geliştirme Bilgisayarları: Windows 10, macOS Catalina, veya Ubuntu 20.04
2. **Veritabanı**:
    - Microsoft SQL Server 2019
    - MySQL 8.0

3. **Geliştirme Araçları**:
    - IDE: Microsoft Visual Studio 2019
    - .NET Core SDK 3.1
    - Git (kaynak kontrol)
4. **Test Araçları**:
    - Selenium WebDriver (web testi)
    - Appium (mobil testi)
    - Postman (API testi)
    - JIRA (test yönetimi ve hata takibi)
    - JUnit / NUnit (birim testi)
5. **Diğer Gereksinimler**:
    - Web Sunucusu: IIS 10.0 veya Apache
    - CI/CD Araçları: Jenkins veya Azure DevOps

Test Ekibi Faaliyetleri

- **Test Kapsamının ve Stratejisinin Belirlenmesi**:
    - Hangi modüllerin test edileceğinin netleştirilmesi.
    - Kritik işlevlerin ve entegrasyon noktalarının belirlenmesi.
    - Performans ve güvenlik testlerinin planlanması.
- **Risk ve Öncelik Analizi**:
    - Test kapsamındaki yüksek riskli alanların belirlenmesi.
    - Testlerin önceliklendirilmesi ve zaman çizelgesinin oluşturulması.

Bu aşamada gerçekleştirilen tüm faaliyetler, projenin daha sonraki aşamalarında karşılaşılabilecek olası sorunları minimize etmek ve test sürecinin etkinliğini artırmak için önemlidir. Test gereksinimleri belirlenirken, kullanıcı gereksinimleri ve beklentileri de göz önünde bulundurularak, projenin kaliteli ve zamanında teslim edilmesi sağlanır.

.

12 Test Planlaması

Faaliyetler

1. **Test Kapsamının Tanımlanması**:
    - Hangi modüllerin ve işlevlerin test edileceğinin belirlenmesi.
    - Test kapsamının netleştirilmesi ve belgelenmesi.
2. **Test Stratejisinin Belirlenmesi**:
    - Hangi test türlerinin (birim testi, entegrasyon testi, sistem testi, kabul testi) uygulanacağının belirlenmesi.
    - Otomasyon ve manuel testlerin oranının belirlenmesi.
    - Performans, güvenlik ve yük testleri gibi özel test türlerinin planlanması.
3. **Tahmini Çaba ve Maliyet Hesaplamaları**:
    - Test çalışması için gerekli zaman ve kaynakların tahmini.
    - Test araçları ve altyapı maliyetlerinin hesaplanması.
    - Personel ve diğer operasyonel maliyetlerin belirlenmesi.
4. **Risk Analizi**:
    - Projede karşılaşılabilecek olası risklerin belirlenmesi.
    - Risklerin etkisinin ve olasılığının değerlendirilmesi.
    - Risk azaltma ve yönetim stratejilerinin geliştirilmesi.
5. **Test Zaman Çizelgelerinin Oluşturulması**:
    - Test sürecinin her aşaması için başlangıç ve bitiş tarihleri belirlenmesi.
    - Test faaliyetlerinin zaman çizelgesi ile uyumlu hale getirilmesi.
    - Gantt şeması ve PERT şemalarının oluşturulması (Bkz. Bölüm 2f).
6. **Test Ortamlarının Tanımlanması**:
    - Testlerin gerçekleştirileceği ortamların belirlenmesi ve kurulumu.
    - Gerekli donanım ve yazılım gereksinimlerinin tanımlanması (Bkz. Bölüm 11).
7. **Test Verilerinin Hazırlanması**:
    - Test senaryoları için gerekli veri setlerinin oluşturulması.
    - Test verilerinin geçerliliğinin ve doğruluğunun kontrol edilmesi.
8. **Test Planının Dokümantasyonu**:
    - Tüm test planı bileşenlerinin yazılı olarak belgelenmesi.

- Test planının ilgili paydaşlarla paylaşılması ve onay alınması.

Test Planı Bileşenleri

1. **Test Kapsamı**:
    - Proje Araştırma ve Hazırlıklar
    - Proje Kapsamının Belirlenmesi
    - Gereksinim Analizi
    - Sistem Tasarımı
    - Geliştirme
    - Test
    - Dağıtım ve Bakım
2. **Test Stratejisi**:
    - Birim Testi: Her birim fonksiyonun doğru çalıştığının kontrolü.
    - Entegrasyon Testi: Birimlerin birlikte düzgün çalıştığının kontrolü.
    - Sistem Testi: Sistemin tüm bileşenlerinin bir arada doğru çalıştığının kontrolü.
    - Kabul Testi: Sistem gereksinimlerinin karşılandığının ve kullanıcı kabul kriterlerinin yerine getirildiğinin kontrolü.
3. **Çaba ve Maliyet Tahmini**:
    - Gerekli personel: 5 Test Mühendisi, 1 Test Yöneticisi
    - Araçlar ve altyapı: Selenium, Appium, JIRA
    - Toplam tahmini maliyet: 50,000 USD
4. **Risk Yönetimi**:
    - Tanımlanan riskler: Zamanında teslim edememe, bütçe aşımı, kritik hatalar
    - Risk azaltma stratejileri: Yedek planlar, ek kaynaklar, sık iletişim ve raporlama
5. **Zaman Çizelgesi**:
    - Proje Araştırma ve Hazırlıklar: 25.06.2024 - 03.05.2024
    - Proje Kapsamının Belirlenmesi: 04.05.2024 - 05.05.2024
    - Gereksinim Analizi: 06.05.2024 - 08.05.2024
    - Sistem Tasarımı: 09.05.2024 - 12.05.2024

- Geliştirme: 13.05.2024 - 23.05.2024
- Test: 24.05.2024 - 28.05.2024
- Dağıtım ve Bakım: 29.05.2024 - 30.05.2024

6. **Test Ortamı**:
   - Donanım: Sunucular, Geliştirme ve Test Bilgisayarları, Mobil Cihazlar (Bkz. Bölüm 11)
   - Yazılım: İşletim Sistemleri, Veritabanları, Geliştirme Araçları, Test Araçları (Bkz. Bölüm 11)

13 Test Senaryoları

Test senaryosu geliştirme aşaması, test planlama aşaması tamamlandıktan sonra başlar. Bu aşamada test ekibi ayrıntılı test durumlarını not edip, gerekli test verilerini hazırlar. Bu aşamada gerekli tüm otomasyon komut dosyaları da oluşturulur. Test senaryoları test ekibi tarafından aşamalı olarak yazılır. Gerekli incelemeler ve bazı değişiklikler yapıldıktan sonra test senaryolarının onaylanmasıyla birlikte test ekibi test verilerini ön koşullar temelinde oluşturur.

Test Senaryosunda olması gereken alanlar:

1. Test Senaryo Numarası
2. Test Senaryo Açıklaması
3. Önem Derecesi
4. Ön Koşul
5. Test Adımları
6. Test Datası
7. Beklenen Sonuç
8. Gerçekleşen Sonuç
9. Durum
10. Yorum

Test Senaryosu Örnekleri (Bizim Diyetisyen Randevu Sistemi)

Test Senaryo Listesi

Test Senaryoları	Test Durumları
Eposta alanına fazla karakter girilmesi	Negatif
Şifre alanına fazla karakter girilmesi	Negatif
Eposta formatının yanlış olması	Negatif
Eposta doğru şifre yanlış girilmesi	Negatif
Alanların boş geçilmesi	Negatif
Gönderme butonuna iki kez tıklanması	Negatif

Kayıtlı olmayan e posta ile giriş yapılmaya çalışması	Negatif

Test Senaryo Tablosu

Test Senaryo ID	Giriş-1	Test Durum ID	Giriş-1A
Test Durum Tanımı	Giriş testi	Test Önceliği	Yüksek
Ön Koşullar	Geçerli kullanıcı hesabı	Son Koşullar	-
No	Olay	Girdiler	Beklenen Çıktılar
1	Uygulamanın başlatılması	http://diyetisyenrandevu.com	Anasayfa görüntüsü
2	Doğru mail ve şifre kontrolü ve giriş butonu	Email: user@example.com Şifre: CorrectPass123	Başarılı giriş

1. **Test Senaryoları Detayları**

   Test Senaryo Numarası: Her bir test senaryosunun benzersiz bir numarası olması gerekir. Test senaryo numaralarının başlarına hangi modüle aitler ise o modülün adını vererek senaryoları gruplandırma ve bulma konusunda rahat edersiniz. Örnek: "Login_001".

   Test Senaryo Açıklaması: Test senaryolarının açıklamasını yazdığımız alandır, kısa ve herkes tarafından anlaşılır olması gerekir. Örnek: "Başarılı Login girişi yapmak" veya "Hatalı Şifre ile Login girişi yapmaya çalışmak".

   Önem Derecesi: Test senaryosunun önemini belirten alandır. Örneğin, Login işleminin fonksiyonalitesini içeren senaryolar yüksek olmalıdır, çünkü kullanıcı Login olamazsa Login girişi ile yapılan işlemlerin hiçbirini yapamayacaktır. Fonksiyonaliteyi etkilemeyen durumlarda önem derecesi orta veya düşük yazılabilir.

   Ön Koşul: Test senaryosuna başlamadan önce belirtilen koşulların yerine getirilmesi gerekmektedir. Örneğin: "Admin kullanıcısı ile Login olunmuştur" veya "Müşteri hesabında yeterli bakiye bulunması".

   Test Adımları: Test senaryosunu adım adım açık bir şekilde ifade ettiğimiz alandır. Örnek:

2. Kullanıcı Login sayfasına gider.
3. Kullanıcı doğru "Kullanıcı Adı" ve hatalı "Şifre" bilgilerini girer.
4. "Giriş yap" butonuna tıklar.

   Test Datası: Kullanıcı testlerini gerçekleştirirken kullanmış olduğu verileri bu alanda belirtir. Örnek: Ürün arama işlemi yapmışsa eğer arama alanına yazmış olduğu terim veya Login işlemi için kullandığı kullanıcı adı ve şifresi.

Beklenen Sonuç: Yazılımdan vermesi gereken tepkiyi belirttiğimiz alandır. Örnek: Test Adımları:

1. Kullanıcı Login sayfasına gider.
2. Kullanıcı doğru "Kullanıcı Adı" ve yanlış "Şifre" bilgilerini girer.
3. "Giriş yap" butonuna tıklar. Beklenen Sonuç:

- Login sayfası açılması beklenir.
- Doğru "Kullanıcı Adı" ve yanlış "Şifre" bilgilerini alanlara girilebiliyor olması beklenir.
- <Yanlış Kullanıcı Adı veya Şifre girdiniz> mesajının gelmesi beklenir.

Gerçekleşen Sonuç: Gerçekleştirdiğimiz adımların sonuçlarını belirttiğimiz alandır. Örnek: Beklenen Sonuç:

- Login sayfası açılması beklenir.
- Doğru "Kullanıcı Adı" ve yanlış "Şifre" bilgilerini alanlara girilebiliyor olması beklenir.
- <Yanlış Kullanıcı Adı veya Şifre girdiniz> mesajının gelmesi beklenir. Gerçekleşen Sonuç:
- Login sayfası açıldı.
- Doğru "Kullanıcı Adı" ve yanlış "Şifre" bilgileri girildi.
- <Yanlış Kullanıcı Adı veya Şifre girdiniz> mesajı geldi.

Durum: Test sonucunu "Geçti", "Geçemedi", "Koşulmadı" veya "Block" olarak belirttiğimiz alandır.

Yorum: Test senaryomuzda belirtmek istediğimiz bir durum var ise veya hata aldığımız işlemi buraya yazabiliriz.

Bu detaylar, Bizim Diyetisyen Randevu Sistemi'ne özgü hale getirilmiş test senaryolarını oluşturmak için kullanılabilir. Test senaryolarının doğru bir şekilde hazırlanması, yazılımın kalite güvencesi sürecinde kritik bir rol oynar ve olası hataların erken tespit edilmesini sağlar.

14 İyi Uygulamalar (Best Practices)

1. **Manuel Görevlerin Otomatikleştirilmesi**: Manuel olarak yapılan yinelenen görevler, komut dosyaları oluşturularak otomatikleştirildi. Bu sayede her seferinde manuel olarak yapılan işlem zaman kazandırdı ve kaynakların daha verimli kullanılmasını sağladı.

2. **Duman Testi Senaryolarının Otomatikleştirilmesi**: Duman testi senaryoları otomatikleştirildi ve komut dosyaları aracılığıyla çalıştırıldı. Bu sayede testler daha hızlı çalıştı ve zaman tasarrufu sağlandı.

3. **Otomasyon Scriptleri ile Yeni Müşteri Kayıtlarının Oluşturulması**: Test için çok sayıda kaydın oluşturulması gereken durumlarda, yeni müşteri kayıtlarını oluşturmak için otomasyon scriptleri hazırlandı. Bu sayede manuel olarak yapılan işlemler otomatik hale getirilerek zaman kazanıldı.

4. **İş Açısından Kritik Senaryoların Test Edilmesi**: İş açısından kritik senaryolar, uygulamanın tamamında ayrı ayrı test edildi. Bu sayede uygulamanın kritik bölümlerinin düzgün çalıştığı onaylandı ve iş sürekliliği sağlandı.

15 Çıkış Kriterleri

1. Tüm planlanan test senaryoları başarıyla yürütülmelidir.
2. Tüm kritik hatalar kapatılmalıdır.
3. Önemli ve orta önem derecesindeki tüm hatalar doğrulanmalı ve kapatılmalıdır.
4. Önemsiz önem derecesindeki hatalar, plana göre yürütülmelidir.
    - Hata 1: Kapatılmalıdır.
    - Hata 2: Açık kalabilir.
    - Hata 3: Eylem planında belirtilen şekilde kapatılacaktır. Eylem planı, bu hataların ne zaman ve nasıl kapatılacağını açıklar.

    Bu çıkış kriterlerinin karşılanması, testin tamamlandığını ve sonuçların kabul edilebilir olduğunu gösterir.

16 Test Sonuçları

Tablo 9: Başarılı Test / Başarısız Test / Kısmen Başarılı Test / Çalıştırılamayan Test / Tamamlanmayan Test

Test ID	Tanımlanan Test ID	Açıklama	Karar
1	T_001	Kullanıcı hesabı oluşturma testi	Başarılı
2	T_002	Randevu talebi oluşturma testi	Başarılı
3	T_003	Randevu tarih ve saat seçimi testi	Başarılı
4	T_004	Diyetisyen ile iletişim testi	Başarılı
5	T_005	Sağlık geçmişi paylaşım testi	Başarılı
6	T_006	Randevu iptal testi	Başarılı
7	T_007	İlaç ve malzeme sarfiyatı takibi testi	Başarılı
8	T_008	Laboratuvar-tetkik istenmesi testi	Başarılı
9	T_009	Poliklinik listesi güncelleme testi	Başarılı
10	T_010	Veri güvenliği testi	Başarılı

Açıklamalar:

1. T_001: Kullanıcı hesabı oluşturma testi - Kullanıcılar başarılı bir şekilde yeni bir hesap oluşturabiliyorlar.
2. T_002: Randevu talebi oluşturma testi - Kullanıcılar başarılı bir şekilde randevu talebi oluşturabiliyorlar.
3. T_003: Randevu tarih ve saat seçimi testi - Kullanıcılar randevu için uygun bir tarih ve saat seçebiliyorlar.
4. T_004: Diyetisyen ile iletişim testi - Kullanıcılar diyetisyenleriyle başarılı bir şekilde iletişim kurabiliyorlar.

5. T_005: Sağlık geçmişi paylaşım testi - Kullanıcılar sağlık geçmişlerini güvenli bir şekilde paylaşabiliyorlar.
6. T_006: Randevu iptal testi - Kullanıcılar başarılı bir şekilde randevu iptal edebiliyorlar.
7. T_007: İlaç ve malzeme sarfiyatı takibi testi - İlaç ve malzeme sarfiyatı başarılı bir şekilde takip ediliyor.
8. T_008: Laboratuvar-tetkik istenmesi testi - Laboratuvar ve tetkik istekleri başarılı bir şekilde yapılıyor.
9. T_009: Poliklinik listesi güncelleme testi - Poliklinik listesi başarılı bir şekilde güncelleniyor.
10. T_010: Veri güvenliği testi - Kullanıcı verileri güvenli bir şekilde saklanıyor ve işleniyor.

Bu tablo, testlerin başarı durumlarını göstermektedir. Her bir testin tanımı, test başlığı, test kararı ve test sırasında karşılaşılan ek bilgileri içermektedir.

16a Sistem Testi

Tablo 10: Sistem Testi Sonuçları

Test Senaryo ID	Tarih	Tester	Başarılı/Başarısız	Hata Şiddeti	Hata Özeti	Sürümden Önce Kapatıldı mı?	Açıklama
ST_001	2024-06-05	Eray	Başarılı	-	-	Evet	Sistem testi başarıyla tamamlandı. Tüm temel işlevler test edildi ve hata bulunamadı.
ST_002	2024-06-06	Emirhan	Başarılı	-	-	Evet	Sistem güvenilirliği test edildi ve beklenen performansı gösterdi. Hiçbir kritik hata bulunmadı.

ST_003	2024-06-07	Zakir	Başarısız	Yüksek	Kullanıcı hesabı oluşturma sırasında veritabanı bağlantı hatası	Hayır	Kullanıcı hesabı oluşturma işlemi sırasında beklenmedik bir veritabanı bağlantı hatasıyla karşılaşıldı. Sorun giderilmek üzere işaretlendi.

Tablo 11: Kabul Testi Sonuçları

Test Senaryo ID	Tarih	Tester	Başarılı/Başarısız	Hata Şiddeti	Hata Özeti	Sürümden Önce Kapatıldı mı?	Açıklama
KT_001	2024-06-10	Eray	Başarılı	-	-	Evet	Kullanıcıların sistemi kullanımı konusunda herhangi bir zorluk yaşamadığı onaylandı.
KT_002	2024-06-11	Emirhan	Başarılı	-	-	Evet	Kullanıcılar, randevu taleplerini kolayca oluşturabildiklerini doğruladılar.
KT_003	2024-06-12	Zakir	Başarılı	-	-	Evet	Kullanıcılar, diyetisyenleriyle etkili bir şekilde iletişim kurabildiklerini belirttiler.

Tablo 12: AD HOC Testi Sonuçları

Test Senaryo ID	Tarih	Tester	Başarılı/Başarısız	Hata Şiddeti	Hata Özeti	Sürümden Önce Kapatıldı mı?	Açıklama
AH_001	2024-06-15	Eray	Başarılı	-	-	Evet	Kullanıcıların farklı senaryolar altında sistemi test ettiği ve istikrarlı olduğu doğrulandı.
AH_002	2024-06-16	Emirhan	Başarılı	-	-	Evet	Sistem, farklı kullanıcı davranışlarını başarılı bir şekilde yönettiği gösterildi.
AH_003	2024-06-17	Zakir	Başarılı	-	-	Evet	Sistem, beklenmedik durumlarda bile kullanıcı deneyimini olumsuz etkilemedi.

Tablo 13: Regresyon Testi Sonuçları

Test Senaryo ID	Tarih	Tester	Başarılı/Başarısız	Hata Şiddeti	Hata Özeti	Sürümden Önce Kapatıldı mı?	Açıklama
RT_001	2024-06-20	Eray	Başarılı	-	-	Evet	Yapılan güncellemelerin, mevcut işlevselliği olumsuz etkilemediği doğrulandı.

RT_002	2024-06-21	Emirhan	Başarılı	-	-	Evet	Sistemde yapılan düzeltmelerin, daha önce raporlanan hataları çözdüğü belirtildi.
RT_003	2024-06-22	Zakir	Başarılı	-	-	Evet	Yeni özellik eklemelerinin, mevcut sistemi istikrarlı bir şekilde çalıştırdığı onaylandı.

Tablo 14: Performans Testi Sonuçları

Test Senaryo ID	Tarih	Tester	Başarılı/Başarısız	Hata Şiddeti	Hata Özeti	Sürümden Önce Kapatıldı mı?	Açıklama
PT_001	2024-06-25	Eray	Başarılı	-	-	Evet	Sistem, yoğun kullanım altında dahi istikrarlı bir performans sergiledi.
PT_002	2024-06-26	Emirhan	Başarılı	-	-	Evet	Sistemin yanıt süreleri, kabul edilebilir düzeyde olduğu doğrulandı.
PT_003	2024-06-27	Zakir	Başarılı	-	-	Evet	Sistem, belirlenen performans kriterlerini karşıladı ve istenen hızda çalıştı.

17 Test Sorumlulukları ve Eğitim İhtiyacı

Bu projede, yazılım test faaliyetlerinin gerçekleştirilmesi için belirlenen görev dağılımı ve sorumluluklar şu şekilde:

1. **Proje Yöneticisi:**
    - Test eylemlerinin planlanması ve koordinasyonu için sorumludur.
    - Test ekibinin oluşturulması ve ekip üyelerinin görevlerinin belirlenmesi.
    - Diğer gruplarla eşgüdüm sağlamak ve sorumlulukların yerine getirilmesini sağlamak.
2. **Konfigürasyon Yönetim Sorumlusu:**
    - Test süreciyle ilgili dokümanların ve ürünlerin konfigürasyon yönetiminden sorumludur.
3. **Test Yöneticisi:**
    - Test gruplarından sorumludur ve test sürecini koordine eder.
    - Test planlarının hazırlanması ve uygulanması.
    - Test sonuçlarının raporlanması ve gerektiğinde düzeltilmesi.
4. **Test Grubu:**
    - Yazılım test faaliyetlerinden sorumludur.
    - Yazılım birim testleri ve birim entegrasyon testlerinin planlanması, gerçekleştirilmesi ve raporlanması.
    - Test belgelerinin hazırlanması (Test Tanımlama Belgesi, Test Sonuç Raporu vb.).
5. **Test Ortamı Sorumlusu:**
    - Gerekli test ortamının kurulması ve işler durumda tutulmasından sorumludur.
6. **Yazılım Geliştirme Grubu:**
    - Yazılım birim testleri ve birim entegrasyon testlerinin gerçekleştirilmesinden sorumludur.

Eğitim İhtiyacı: Apache Roller projesi kapsamında gerçekleştirilecek olan test faaliyetleri için özel bir eğitime ihtiyaç duyulmamaktadır. Ancak, her bir grubun kendi uzmanlık alanlarına yönelik olarak sürekli kendini geliştirmesi ve proje gereksinimlerine uygun olarak bilgi ve becerilerini güncellemesi önemlidir. Bu nedenle, her bir ekip üyesinin kendi alanında güncel bilgilere ve teknolojilere erişim sağlayacak şekilde sürekli eğitim ve öğrenme süreçlerine katılması teşvik edilmelidir.

III Proje Problemleri

18 Mevcut Problemler

Gereksinim Değişkenliği: Projenin gereksinimleri sürekli olarak değişebilir. Bu durum, projenin planlanmasını ve yürütülmesini zorlaştırabilir. Gereksinimlerin netleştirilmesi ve değişikliklerin etkin bir şekilde yönetilmesi önemlidir.

Yüksek Rekabet: Benzer projelerle rekabet ediyorsanız, pazarda fark yaratacak özellikler ve kaliteyi sağlamak önemlidir. Rekabetçi bir ortamda başarılı olabilmek için sürekli olarak müşteri ihtiyaçlarını ve piyasa trendlerini takip etmek gerekebilir.

Sürekli Değişen Teknolojiler: Teknolojik gelişmeler hızla ilerliyor ve bu durum projenizin teknolojik altyapısını etkileyebilir. Sürekli olarak güncel teknolojilere adapte olmak ve bu teknolojileri projenize entegre etmek önemlidir.

Kalite Testi ve Hata Düzeltmeleri için Çok Sayıda Yazılım Yinelemesi: Kalite testi ve hata düzeltmeleri için gereksinimlerin netleşmesi ve etkin bir test stratejisi oluşturulması önemlidir. Aksi takdirde, yazılımın sürekli olarak tekrarlanan yinelemeler gerektirebilir.

Projeyi Doğru Anlamamak: Projenin gereksinimlerini ve hedeflerini doğru anlamak, projenin başarılı bir şekilde tamamlanması için önemlidir. Bu nedenle, iletişim kanallarının açık tutulması ve paydaşlarla sürekli olarak etkileşim içinde olunması gereklidir.

Entegrasyon: Farklı sistemlerin veya bileşenlerin entegrasyonu karmaşık olabilir. Entegrasyon sürecinde uyumluluk sorunları yaşanabilir ve bu da projenin zamanında ve bütçe dahilinde tamamlanmasını engelleyebilir.

Güvenlik Tehditleri: Yazılım projeleri, çeşitli güvenlik tehditlerine maruz kalabilir. Veri güvenliği, kimlik doğrulama ve yetkilendirme gibi güvenlik önlemlerinin titizlikle ele alınması gerekir.

Zaman Tahmini: Projenin zamanında tamamlanması için doğru zaman tahmini yapmak önemlidir. Ancak, beklenmedik engeller veya gecikmelerle karşılaşılabilir, bu da zaman tahminlerinin doğruluğunu etkileyebilir.

Kod Geliştirme Uygulamalarını Kullanmamak: Kod geliştirme süreçlerinde modern ve verimli uygulamaların kullanılmaması, iş akışını yavaşlatabilir ve kaliteyi düşürebilir.

Yönetimsel Rollere İlişkin Problemler: Proje yönetimi sürecindeki eksiklikler veya roller arasındaki belirsizlikler, proje ilerlemesini etkileyebilir. Bu nedenle, yönetimsel rollerin netleştirilmesi ve herkesin rollerinin ve sorumluluklarının net bir şekilde tanımlanması önemlidir.

19 Hazır Çözümler(Off-the-Shelf Solutions)

19a Hazır Ürünler (Ready-Made Products):

Proje için önceden yapılmış hazır ürünler bulunabilir. Ancak, bu hazır ürünler genellikle genel kullanım için tasarlanmıştır ve benzersiz iş gereksinimlerini tam olarak karşılamayabilir. Hazır ürünler, genellikle ticari kuruluşların büyük çoğunluğu ve çok sayıda kullanıcı hedeflenerek geliştirilir. Ancak, her müşterinin ihtiyaçları farklı olduğundan, hazır ürünler müşterinin tam olarak hangi ihtiyaçlarına cevap verdiğine bağlı olarak uygun olabilir veya olmayabilir. Sizin yazdığınız uygulamanın avantajları, müşterinin özel gereksinimlerine tam olarak uygun olarak özelleştirilebilmesi ve esneklik sağlayabilmesidir.

19b Yeniden Kullanılabilir Bileşenler (Reusable Components):

Bileşen tabanlı yazılım geliştirme yöntemleri, geliştirme sürecini daha etkili hale getirebilir. Bu yöntemde, hedeflenen sistemin spesifikasyonu belirlenir ve uygun bileşenlere ayrıştırılır. Ardından, gerekli bileşenler araştırılır, uygun hale getirilir ve entegre edilerek hedeflenen sistem oluşturulur. Bileşen tabanlı gelişimin avantajları arasında minimum teslimat süreleri, verimliliğin artması ve geliştirilmiş kalite yer alır. Projenizde kullanılan bileşen tabanlı kodlama ve test yöntemleri, projenizin gereksinimlerine uygun olarak bileşenlerin seçilmesi ve entegrasyonun sağlanmasını içerir.

19c Kopyalanabilir Ürünler (Copyable Products):

Yazılımın başka kişiler tarafından kopyalanabilir olup olmadığı ve buna karşın alınacak önlemler araştırılmalıdır. Bu önlemler arasında yazılım lisanslaması, telif hakkı koruması ve güvenlik önlemleri bulunabilir. Projenizin fikri mülkiyet haklarını korumak için uygun önlemleri almak önemlidir. Bu sayede, projeniz özgün kalır ve rekabet avantajını korur.

20 Yeni Problemler

20a Çevresel Etkiler:

- Enerji tüketimi: Yazılımın çalıştırılması için gereken enerji miktarı ve bu enerjinin kaynağı.
- Atık yönetimi: Yazılım geliştirme sürecinde kullanılan malzemelerin ve donanımların atık yönetimi.
- Kaynak kullanımı: Projenin tamamlanması için kullanılan kaynakların (örneğin, elektronik cihazlar, kağıt vb.) etkileri.
- Ulaşım etkileri: Proje ekibinin seyahatleri ve bunun çevresel etkileri.

Yeni problemlerin ortaya çıkmasında etkili olan problemler şunlar olabilir:

- Atık yönetimi eksiklikleri: Atık yönetimi ve geri dönüşüm süreçlerinin yetersiz olması çevresel etkilerin artmasına neden olabilir.

- Yüksek enerji tüketimi: Yazılımın çalıştırılması için yüksek miktarda enerji gerektiren sistemlerin kullanılması çevresel etkileri artırabilir.
- Kaynak israfı: Gereksiz kaynak kullanımı veya atıl kaynakların kullanımı çevresel etkilerin artmasına neden olabilir.

20b Kurulan Sistem Özellikleri:

Yazılımın kurulduğu sistem aşağıdaki özelliklere sahip olabilir:

- Web tabanlı bir arayüz: Kullanıcıların yazılıma web tarayıcıları aracılığıyla erişmelerini sağlayan bir arayüz.
- Veri tabanı entegrasyonu: Kullanıcı bilgilerinin ve randevu verilerinin güvenli bir şekilde saklandığı ve yönetildiği bir veri tabanı.
- Randevu planlama ve yönetim sistemi: Kullanıcıların randevu almasını, iptal etmesini ve düzenlemesini sağlayan bir sistem.
- Kullanıcı profilleri: Kullanıcıların kişisel bilgilerini ve sağlık geçmişlerini güvenli bir şekilde saklayan profiller.
- İletişim araçları: Kullanıcıların diyetisyenleriyle iletişim kurmalarını sağlayan mesajlaşma veya görüntülü görüşme gibi araçlar.

20c Takip Problemleri:

Yazılım tamamlandıktan sonra işlemlerin takibi için şu öngörüler yapılabilir:

- Kullanıcı geri bildirimleri: Kullanıcıların yazılımı nasıl kullandıkları ve yaşadıkları sorunlar hakkında geri bildirimlerin düzenli olarak toplanması.
- Performans izleme: Yazılımın performansının düzenli olarak izlenmesi ve raporlanması.
- Veri analizi: Kullanıcıların davranışlarının ve tercihlerinin analiz edilmesi ve bu verilere dayalı olarak iyileştirmelerin yapılması.
- Güncelleme ve bakım planları: Yazılımın güncel tutulması ve düzenli bakımının yapılması için planların oluşturulması.

22 Riskler

Bu projenin genel risk takibi için Risk Yönetim Planı geliştirilmiştir. Diyetisyen Randevu Yönetim Sistemi projesinde karşılaşılabilecek idari ve teknik riskler belirlenmiş ve bu risklerin yönetimi için alternatif çözümler değerlendirilmiştir. Örneğin, kullanıcı kabul riski, veri güvenliği riski ve teknik ekipman sorunları gibi durumlar için önleyici ve düzeltici aksiyonlar planlanmıştır.

Risk Tanımları ve Yönetimi: Proje sürecinde karşılaşılabilecek riskler ve bunların yönetimi için aşağıdaki tabloyu kullanabiliriz.

Tablo 4 Diyetisyen Randevu ve Yönetim Sistemi Risk İzleme Tablosu

Risk Tanımı	Etki	Risk Önlem Stratejisi	Risk Çözüm Etkinliği	Risk Sorumlusu
Gereksinimlerin Tam ve Doğru Belirlenememesi	Proje gereksinimlerinin yanlış anlaşılması veya eksik belirlenmesi sonucu hatalı veya eksik geliştirme yapılması.	Gereksinim analizlerinin dikkatli yapılması, müşteri ve kullanıcılarla düzenli geri bildirim toplantılarının yapılması.	Yüksek	Proje Yöneticisi
Teknik Ekipman Problemleri	Geliştirme ve test aşamalarında ekipman eksikliği veya arızası nedeniyle gecikmeler yaşanması.	Ekipmanların düzenli bakımının yapılması, yedek ekipman bulundurulması.	Orta	Tester
İnsan Kaynakları Riski	Kritik personelin projeden ayrılması veya performans düşüklüğü.	Kilit personel için yedekleme planlarının oluşturulması, personel motivasyonunun yüksek tutulması.	Orta	Coder
Veri Güvenliği Riskleri	Hasta bilgilerinin yetkisiz kişilerce erişilmesi veya sızdırılması.	Güvenlik protokollerinin sıkılaştırılması, düzenli güvenlik denetimlerinin yapılması, kullanıcı eğitimlerinin verilmesi.	Yüksek	Coder
Kullanıcı Kabul Riski	Kullanıcıların yeni sistemi benimsememesi veya kullanmakta zorluk çekmesi.	Kullanıcı eğitiminin sağlanması, kullanıcı dostu arayüzlerin tasarlanması, geri bildirim mekanizmalarının oluşturulması.	Orta	Tester

23 Maliyetler ve Kaynaklar

Projenin maliyet analizi, Diyetisyen Randevu Yönetim Sistemi için gerekli olan tüm insan kaynağı, teknik ekipman, yazılım lisansları ve sunucu kiralama gibi maliyet kalemlerini kapsamaktadır. Her bir kaynak için ihtiyaç duyulan miktar ve toplam maliyet hesaplanarak proje bütçesi oluşturulmuştur.

Tablo 5 Diyetisyen Randevu ve Yönetim Sistemi Maliyet Tablosu

İhtiyaç Açıklama	İhtiyaç Duyulan	Sahip Olunan	Fiyat	Açıklama
Geliştirme Bilgisayarları	5	2	50.000 TL	Yeni bilgisayar alımı

Sunucu Kiralama	1	0	25.000TL	Yıllık sunucu kiralama maliyeti
Yazılım Lisansları	5	0	10.000 TL	Gerekli yazılım lisanslarının alımı
Eğitim ve Danışmanlık	-	-	15.000 TL	Personel eğitimi ve danışmanlık
Test Ekipmanları	3	0	5.000 TL	Test cihazlarının alımı
TOPLAM			105.000 TL	

24 Proje Geçmişi

Proje, bir Diyetisyen Randevu ve Yönetim Sistemi programını geliştirmek için başlatıldı. Bu program, diyetisyenlerin randevu planlaması, müşteri kayıtları yönetimi, beslenme programlarının oluşturulması ve izlenmesi gibi işlevleri desteklemek üzere tasarlanmıştır. Projenin amacı, diyetisyenlerin iş akışını optimize etmek, müşteri verilerini güvenli bir şekilde saklamak ve beslenme danışmanlığı süreçlerini daha etkili hale getirmektir.

Bu proje, modern yazılım mühendisliği prensipleri ve teknolojileri kullanarak geliştirilmiştir. Projenin temel mimarisi, Onion Mimarisi olarak bilinen bir yapı üzerine kurulmuştur. Onion Mimarisi, katmanlı bir yapıya sahip olup iş mantığı, altyapı ve kullanıcı arayüzü gibi farklı katmanları net bir şekilde tanımlar ve bağımlılıkları azaltır. Bu sayede, yazılımın daha modüler, esnek ve bakımı kolay hale gelmesi hedeflenmiştir.

Program, .NET Core platformu üzerinde geliştirilmiştir. .NET Core, platform bağımsızlığı sağlayan ve yüksek performanslı web uygulamaları geliştirmek için kullanılan açık kaynaklı bir çerçevedir. .NET Core'un esnekliği ve güçlü kütüphane desteği, projenin gereksinimlerini karşılamak için ideal bir ortam sunmuştur.

Diyetisyen Randevu ve Yönetim Sistemi, kullanıcı dostu bir arayüze sahiptir ve çeşitli cihazlarda kullanılmaya uygun şekilde tasarlanmıştır. Müşteri kayıtları, randevu takvimi, beslenme programları ve diğer veriler güvenli bir şekilde depolanır ve yönetilir. Ayrıca, programın yeteneklerini genişletmek ve özelleştirmek için modüler bir yapı oluşturulmuştur.

Bu proje, diyetisyenlerin ve müşterilerin ihtiyaçlarını karşılamak ve beslenme danışmanlığı süreçlerini optimize etmek üzere titizlikle tasarlanmıştır. Onion Mimarisi ve .NET Core gibi modern teknolojilerin kullanımı, yazılımın güvenilirliğini, güvenliğini ve performansını artırarak projenin başarılı bir şekilde tamamlanmasına katkı sağlamıştır.

25 Yeni Çözüm Önerileri

Diyetisyen Randevu Yönetim Sistemi'nin daha verimli ve kullanıcı dostu olmasını sağlamak amacıyla, otomatik randevu hatırlatıcıları, gelişmiş analitik araçlar ve diyetisyen-hasta iletişim modülü gibi yenilikçi çözümler sunulmuştur.

Projede karşılaşılan sorunlar ve mevcut sistemin yetersizlikleri dikkate alınarak yeni çözüm önerileri geliştirilmiştir.

-Otomatik Randevu Hatırlatıcıları: SMS ve e-posta ile randevu hatırlatma özelliklerinin eklenmesi.

-Diyetisyen-Hasta İletişim Modülü: Diyetisyenler ve hastalar arasında daha iyi iletişim sağlamak için mesajlaşma modülünün geliştirilmesi.

-Gelişmiş Analitik Araçlar: Diyet planlarının ve hasta gelişiminin analiz edilmesi için raporlama ve analitik araçların eklenmesi.

26 Sürdürülebilirlik

Projenin sürdürülebilirliği, yazılımın uzun vadede etkili bir şekilde kullanılabilir, geliştirilebilir ve yönetilebilir olmasını sağlayan önemli bir faktördür. Bu bölümde, Diyetisyen Randevu ve Yönetim Sistemi'nin sürdürülebilirlik stratejileri ve uygulamaları tartışılacaktır.

-Modüler ve Esnek Tasarım: Projemizin temelini oluşturan Onion Mimarisi, modüler bir yapı sunar. Bu, yazılımın farklı bileşenlerinin bağımsız olarak geliştirilebilmesini ve değiştirilebilmesini sağlar. Yeni gereksinimler ortaya çıktığında veya mevcut işlevler değiştirilmek istendiğinde, bu modüler yapı sayesinde kod tabanı kolayca güncellenebilir ve genişletilebilir.

-Kod Kalitesi ve Standartları: Proje boyunca, kod kalitesini artırmak ve tutarlılık sağlamak için iyi yazılım mühendisliği prensiplerine ve standartlarına sıkı sıkıya uyulmuştur. Temiz kod yazma, yorum satırları eklemek, kod tekrarını önlemek ve düzenli kod incelemeleri yapmak gibi uygulamalar, yazılımın sürdürülebilirliğini artırmaya yardımcı olur.

-Dökümantasyon: Projenin kod tabanıyla birlikte, kapsamlı ve anlaşılır bir teknik dökümantasyon tutulmuştur. Bu dökümantasyon, projenin yapısını, bileşenlerini, çalışma prensiplerini ve dışarıdan geliştiricilerin projeye katkı yapmasını kolaylaştıracak bilgileri içerir.

-Sürüm Kontrolü ve Sürekli Entegrasyon: Proje, bir sürüm kontrol sistemi (version control system) kullanılarak yönetilmektedir. Bu sayede, yapılan her değişiklik izlenebilir ve geri alınabilir. Ayrıca, sürekli entegrasyon (continuous integration) uygulamaları sayesinde kod tabanı düzenli olarak derlenir, test edilir ve dağıtılmaya hazır hale getirilir. Bu, hataların erken tespit edilmesini sağlar ve yazılımın sürekli olarak güncel ve stabil kalmasını sağlar.

-Güvenlik ve Veri Koruma: Proje, kullanıcı verilerini korumak için güçlü güvenlik önlemleriyle donatılmıştır. Hassas verilerin şifrelenmesi, yetkilendirme ve kimlik doğrulama mekanizmalarının kullanılması gibi uygulamalar, kullanıcı gizliliğini ve veri bütünlüğünü korur.

Bu sürdürülebilirlik stratejileri ve uygulamaları, Diyetisyen Randevu ve Yönetim Sistemi'nin uzun vadeli başarısını ve etkinliğini sağlamak için benimsenmiştir. Bu yaklaşımlar, yazılımın geliştirilmesi, yönetilmesi ve kullanılması sürecini optimize ederek projenin sürdürülebilirliğini maksimum düzeyde tutmayı hedefler.

27 Öğrenme Kaynakları

NET Core ve Onion Mimarisi İle İlgili Soru ve Cevaplar

Onion Mimarisi Uygulaması Örnekleri

.NET Core ile Web Uygulamaları Geliştirme

Onion Architecture | Soğan Mimarisi | Dotnet Core

Asp.Net Core'da Onion Mimarisine Genel Bakış

Onion Architecture in ASP.NET Core