

**KULLANICI DENEYİMİNİN YEŞİL YAZILIM
MÜHENDİSLİĞİ ÜZERİNE ETKİSİ**

Ashhan KILIÇ

Danışman: Prof. Dr. Yüksel GÖKTAŞ

Yüksek Lisans Tezi

Yazılım Mühendisliği Ana Bilim Dalı

2024

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YAZILIM MÜHENDİSLİĞİ ANA BİLİM DALI

KULLANICI DENEYİMİNİN YEŞİL YAZILIM MÜHENDİSLİĞİ ÜZERİNE ETKİSİ
(Impact Of User Experience On Green Software Engineering)

YÜKSEK LİSANS TEZİ

Aslıhan KILIÇ

Danışman: Prof. Dr. Yüksel GÖKTAŞ

Erzurum
Ocak, 2024

KABUL VE ONAY TUTANAĞI

Aslıhan KILIÇ tarafından hazırlanan “KULLANICI DENEYİMİNİN YEŞİL YAZILIM MÜHENDİSLİĞİ ÜZERİNE ETKİSİ” başlıklı çalışması 22 / 01 / 2024 tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Yazılım Mühendisliği Ana Bilim Dalı, Yazılım Mühendisliği Bilim Dalında, yüksek lisans tezi olarak kabul edilmiştir.

Jüri Başkanı: Doç. Dr. Mehmet BAYĞIN
Erzurum Teknik Üniversitesi

Aslı Islak İmzalıdır

Danışman: Prof. Dr. Yüksel GÖKTAŞ
Atatürk Üniversitesi

Aslı Islak İmzalıdır

Jüri Üyesi: Dr. Öğr. Üyesi Tolga AYDIN
Atatürk Üniversitesi

Aslı Islak İmzalıdır

Enstitü Yönetim Kurulunun
.../.../.... tarih ve
sayılı kararı.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Saltuk Buğrahan CEYHUN

Enstitü Müdürü

Aslı Islak İmzalıdır

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans Tezi olarak Prof. Dr. Yüksel GÖKTAŞ danışmanlığında sunulan “KULLANICI DENEYİMİNİN YEŞİL YAZILIM MÜHENDİSLİĞİ ÜZERİNE ETKİSİ” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	1	30
Kuramsal Temeller	1	30
Materyal ve Yöntem	0	35
Bulgular	5	20
Tartışma	0	20
Tezin Geneli	9	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Aslıhan KILIÇ	Prof. Dr. Yüksel GÖKTAŞ
22.1.2024	22.1.2024
İmza: Aslı Islak İmzalıdır	İmza: Aslı Islak İmzalıdır

* Tez ile ilgili YÖKTEZ’de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/.../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/.../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŞEKKÜR

Bu çalışmanın her aşamasında değerli bilgi, fikir ve önerileri ile bana daima yol gösteren, destek olan kıymetli danışmanım Sayın Prof. Dr. Yüksel GÖKTAŞ' a en içten duygularıyla teşekkürlerimi sunarım. Tez savunmama jüri olarak katkılar sağlayan sayın Doç. Dr. Mehmet BAYĞIN' a ve Sayın Dr. Öğr. Üyesi Tolga AYDIN' a teşekkürlerimi sunarım.

Hayatımın her anında bana inanan, güvenen, yanımda olan, destek ve yardımlarını daima hissettiğim, motivasyon kaynağım olan aileme, babam Halis KILIÇ, annem Çiğdem KILIÇ ve kardeşlerim Yasemin Buse KILIÇ, Meva Sena KILIÇ' a en derin duygularıyla teşekkür ederim.

Tez çalışma sürecimi içerisinde yürüttüğüm, çalıştığım ve imkanlarından faydalandığım Atatürk Üniversitesine, yine tez sürecine katkıda bulunup, kıymetli dönütler veren değerli mesai arkadaşlarım Arş. Gör. Osman DURDAĞ'a ve Öğr. Gör. İsmail KARA'ya teşekkür ederim.

Aslıhan KILIÇ

ÖZET

YÜKSEK LİSANS TEZİ

KULLANICI DENEYİMİNİN YEŞİL YAZILIM MÜHENDİSLİĞİ ÜZERİNE ETKİSİ

Aslıhan KILIÇ

Danışman: Prof. Dr. Yüksel GÖKTAŞ

Amaç: Bu çalışmada, yazılım uygulamalarının geliştirilme yaşam döngüsüne kullanılabilirlik testlerinin uygulanmasının, karbon salınımı ve enerji tüketimi bileşenleri aracılığıyla yeşil yazılım açısından incelenmesi amaçlanmaktadır. Böylece, insan bilgisayar etkileşimi ve yeşil yazılım alanları için ortak bir çalışma alanı oluşturacaktır.

Yöntem: Bu çalışmada, kullanıcıların hastane randevusu oluşturabilecekleri bir uygulama geliştirilmiştir. Uygulamanın ilk versiyonunun geliştirilme aşamalarında herhangi bir kullanılabilirlik testi yapılmamıştır. Uygulamanın kullanılabilirlik testi ile görüşleri dahil edilmemiş versiyonu son kullanıcılara sunularak uygulama üzerinde kendilerine verilen görevleri yerine getirmeleri istenmiştir. Kullanıcıların uygulamayı kullanımları esnasında yapılan karbon salınımı ve enerji tüketimi değerleri ölçülmüştür. Sonrasında kullanıcılar ile uygulama hakkında bir görüşme yapılarak kullanılabilirlik testi yapılmıştır ve geri dönütler elde edilmiştir. İkinci versiyon için kullanıcılardan elde edilen geri dönütler hastane randevu uygulamasına dahil edilmiştir. Böylece uygulamanın kullanıcı deneyimi dahil edilmiş hali olan ikinci versiyonu oluşturulmuştur. İkinci versiyonda son kullanıcılara sunularak, burada görevleri yerine getirmeleri istenmiştir. Bu esnada yine karbon salınımı ve enerji tüketimi değerleri ölçülmüştür. İlk aşamada 12 kişi ile kullanıcı testleri gerçekleştirilmiştir. İkinci aşamada ise ilk 6 kişi ilk aşamada test yapılan, diğer 6 kişi ise daha önce uygulamayı kullanmayan kişilerden seçilmiştir. Her iki versiyon içinde değerler kıyaslanmıştır.

Bulgular: Çalışmanın, ikinci aşamasında elde edilen uygulamanın kullanım süresi, karbon salınımı ve enerji tüketimi değerleri, ilk aşamasında elde edilen uygulamanın kullanım süresi, karbon salınımı ve enerji tüketimi değerlerinden daha düşük olduğu gözlemlenmiştir. Her iki versiyonu kullanan kullanıcıların, ilk versiyondaki kullanım süresi, karbon salınımı ve enerji tüketimi değerlerine kıyasla ikinci versiyondaki kullanım süresi, karbon salınımı ve enerji tüketimi değerlerinin düştüğü gözlemlenmiştir. Uygulamanın ilk versiyonunu ilk kez kullanan kullanıcıların ölçüm sonuçları ile ikinci aşamada uygulamanın ikinci versiyonunu ilk kez kullanan kullanıcıların ölçüm sonuçları karşılaştırılmıştır. İkinci versiyonu ilk kez kullanan kullanıcıların ortalama kullanım süresi, karbon salınımı ve enerji tüketimi değerleri, ilk versiyonu ilk kez kullanan kullanıcıların ortalama değerlerinden daha düşük olduğu gözlemlenmiştir.

Sonuç: Çalışma, yazılım uygulamalarının geliştirilme aşamalarına kullanılabilirlik testleri aracılığıyla kullanıcıların görüş ve önerilerinin dahil edilmesinin, uygulamaların kullanım sürelerini ve buna bağlı olarak kullanım esnasında yapılan karbon salınımı ile enerji tüketimi değerlerinin düşmesine katkı sağlayabileceği sonucuna varmıştır.

Anahtar Kelimeler: yeşil yazılım, insan bilgisayar etkileşimi, kullanılabilirlik, kod karbon emisyonu, kod enerji tüketimi

Ocak 2024, 88 sayfa

ABSTRACT

MASTER DISSERTATION

IMPACT OF USER EXPERIENCE ON GREEN SOFTWARE ENGINEERING

Aslıhan KILIÇ

Supervisor: Prof. Dr. Yüksel GÖKTAŞ

Purpose: This study aims to examine the application of usability tests to the development life cycle of software applications in terms of green software through carbon emission and energy consumption components. Thus, it will create a common workspace for human-computer interaction and green software fields.

Method: In this study, an application was developed where users can create hospital appointments. No usability tests were conducted during the development stages of the first version of the application. The usability test version of the application, which did not include their opinions, was presented to end users and they were asked to perform the tasks assigned to them on the application. Carbon emission and energy consumption values of users while using the application were measured. Afterwards, a usability test was conducted by interviewing the users about the application and feedback was obtained. For the second version, the feedback obtained from users was included in the hospital appointment application. Thus, the second version of the application with user experience included was created. In the second version, it was presented to end users and they were asked to perform tasks there. Meanwhile, carbon emission and energy consumption values were measured again. In the first stage, user tests were carried out with 12 people. In the second stage, the first 6 people were selected from those who were tested in the first stage, and the other 6 people were selected from those who had not used the application before. Values are compared in both versions.

Findings: It was observed that the usage time, carbon emission and energy consumption values of the application obtained in the second stage of the study were lower than the usage time, carbon emission and energy consumption values of the application obtained in the first stage. It has been observed that users using both versions have decreased the usage time, carbon emission and energy consumption values in the second version compared to the usage time, carbon emission and energy consumption values in the first version. The measurement results of the users who used the first version of the application for the first time were compared with the measurement results of the users who used the second version of the application for the first time in the second stage. It has been observed that the average usage time, carbon emission and energy consumption values of users who use the second version for the first time are lower than the average values of users who use the first version for the first time.

Results: The study concluded that including users' opinions and suggestions through usability tests in the development stages of software applications can contribute to reducing the usage period of the applications and, accordingly, the carbon emissions and energy consumption values during use.

Keywords: green software, human computer interaction, usability, code carbon emission, code energy consumption

January 2024, 88 pages

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI.....	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
TABLolar DİZİNİ.....	viii
ŞEKİLLER DİZİNİ	ix
KISALTMALAR VE SİMGELER DİZİNİ	x
GİRİŞ.....	1
KURAMSAL TEMELLER.....	6
Yeşil Yazılım	6
İnsan Bilgisayar Etkileşimi	12
Kullanılabilirlik	14
MATERYAL VE YÖNTEM	23
Yöntem.....	23
Visual Studio Code	25
Django Web Çatısı	25
Python Programlama Dili.....	25
Uygulamanın Geliştirilmesi	25
CodeCarbon Kütüphanesi	30
Net Karbon Yoğunluğu Nasıl Hesaplanır?	33
Enerji Tüketimi ve Hesaplanması	33
Kütüphane Nasıl Çalışır?	34
Kullanılabilirlik Ölçme Yöntemi	36
Deney Kullanıcı Profili	38
Test Protokolü	38
Gözlem Kâğıdı	39
Görev Listesi	39
Kullanıcı Testi Soruları	41
Uygulamanın Tekrar Revize Edilmesi	43

ARAŞTIRMA BULGULARI	46
İlk Versiyon CodeCarbon Verileri	46
Kullanıcı Testi Sonucunda Kullanıcılardan Toplanan Geri Dönütler.....	51
İkinci Versiyon CodeCarbon Verileri	53
TARTIŞMA.....	61
SONUÇ VE ÖNERİLER	65
KAYNAKÇA	68
EKLER	71
EK-1. Genel Soru Şablonu.....	71
EK-2. Anasayfa Soru Şablonu	71
EK-3. Bölümlerimiz Sayfası Soru Şablonu	72
EK-4. Doktorlarımız Sayfası Soru Şablonu	72
EK-5. Yeni Randevu Alma Sayfası Soru Şablonu.....	73
EK-6. Randevu Sorgu İptal Sayfası Soru Şablonu.....	73
EK-7. Müsait Randevu Durumu Sayfası Soru Şablonu	74
EK-8. Check-Up Randevu Sayfası Soru Şablonu	74
EK-9. Laboratuvar Sonuçları Sayfası Soru Şablonu.....	75
EK-10. Profil Sayfası Soru Şablonu.....	75
ÖZGEÇMİŞ.....	76

TABLÖLAR DİZİNİ

Tablo 1. Greenpeace Verileri.....	8
Tablo 2. Timestamp ve Run_id Değerleri.....	48
Tablo 3. Süre Değerleri.....	49
Tablo 4. Emisyon Değerleri.....	49
Tablo 5. Cpu_energy, Gpu_energy ve Ram_energy Değerleri.....	50
Tablo 6. Toplam Enerji Tüketimi Değerleri	50
Tablo 7. İkinci Versiyon Süre Değerleri.....	53
Tablo 8. İkinci Versiyon Emissions Değerleri.....	54
Tablo 9. İkinci Versiyon Cpu_energy, Gpu_energy ve Ram_energy Değerleri.....	55
Tablo 10. İkinci Versiyon Toplam Enerji Tüketimi Değerleri	56

ŞEKİLLER DİZİNİ

Şekil 1. Yeşil Yazılım Prensipleri	6
Şekil 2. Smart 2020 Verileri.....	7
Şekil 3. Üye Olma ve Giriş Yapma Sayfaları	26
Şekil 4. Ana Sayfa	27
Şekil 5. Bölümler Sayfası.....	27
Şekil 6. Doktorlar Sayfası	28
Şekil 7. Online Randevu Oluşturma Sayfaları	29
Şekil 8. Check-Up Randevu Sayfası	29
Şekil 9. Laboratuvar Sonuçları Sayfası	30
Şekil 10. Kullanıcı Profil Sayfası	30
Şekil 11. Türkiye Elektrik Yoğunluğu Değerleri	32
Şekil 12. Enerji Kaynakları Karbon Yoğunluğu	33
Şekil 13. CodeCarbon Entegrasyonu	34
Şekil 14. Kullanıcı Testi Soru Kalıpları	41
Şekil 15. Örnek Soru Formatı	42
Şekil 16. İkinci Versiyon Bölümler ve Doktorlar Sayfası	43
Şekil 17. İkinci Versiyon Online Randevu Sayfaları	44
Şekil 18. İkinci Versiyon Laboratuvar Sonuçları ve Randevu Sorgu/İptal Sayfaları	44
Şekil 19. İkinci Versiyon Check-Up Randevu Sayfası	45
Şekil 20. İkinci Versiyon Kullanıcı Profil Sayfası	45
Şekil 21. Her İki Versiyonu Kullanan Kullanıcılar için Yaklaşık Süre Değerleri	57
Şekil 22. Her İki Versiyonu Kullanan Kullanıcılar için Karbon Emisyonu Değerleri.....	57
Şekil 23. Her İki Versiyonu Kullanan Kullanıcılar için CPU Enerji Tüketim Değerleri.....	58
Şekil 24. Her İki Versiyonu Kullanan Kullanıcılar için GPU Enerji Tüketim Değerleri	58
Şekil 25. Her İki Versiyonu Kullanan Kullanıcılar için RAM Enerji Tüketim Değerleri	59
Şekil 26. Her İki Versiyonu Kullanan Kullanıcılar için Toplam Enerji Tüketim Değerleri	59
Şekil 27. İlk ve İkinci Aşama Değerlerinin Kıyas Grafiği	62
Şekil 28. Genel Kıyas Grafiği	62

KISALTMALAR VE SİMGELER DİZİNİ

BT	: Bilişim Teknolojileri
IoT	: Internet of Things
IPCC	: The Intergovernmental Panel on Climate Change
ISO	: International Organization for Standardization
İBE	: İnsan Bilgisayar Etkileşimi
MCC	: Mobile Cloud Computing
MEC	: Mobile Edge Computing
PoS	: Proof of Stake
PoW	: Proof of Work
PUE	: Power Usage Effectiveness
UNFCCC	: United Nations Framework Convention on Climate Change

GİRİŞ

Sürdürülebilirlik ve doğa dostu kavramları günümüzde birçok alan için söz edilen kavramlar haline gelmişlerdir. Genel anlamda “Bugünün gereksinimlerini, gelecek nesillerin ihtiyaçlarını karşılama durumunu engellemeden yerine getirebilmektir.” şeklinde ifade edilen sürdürülebilirlik kavramının, ekonomi, çevre ve toplum boyutlarında ele alınması gerektiğinden bahsedilmektedir (United Nations World Commission on Environment and Development, 1987).

İklim değişikliğinin günümüzün en önemli problemlerinden biri haline gelecek olması muhtemel bir durum olarak görülmekte ve gelişen, değişen dünya ile birlikte sera gazı konsantrasyonları ile küresel ısınma, geçmiş zamanlara göre daha hızlı bir şekilde artarak etkilerini iklim bozulması, insan sağlığı üzerinde bozulmalar, deniz seviyelerinin yükselmesi, küresel kaynakların azalması gibi sonuçları ile açıkça göstermektedir (Easterbrook, 2010).

Alanında çalışan birçok kurum, kuruluş ve şirket çevre dostu, sürdürülebilir olma hedeflerini çalışmalarına dahil etmektedirler. Özellikle geliştirilen ve kullanılan yazılım ürünlerinin de süreçlerinin ne kadarının çevre dostu olduğu da önemli faktörlerden biri haline gelmiştir (Calero vd., 2013).

Yeşil bilgi işlem ise üretim açısından enerji verimliliğini üst seviyelere çıkarmaya çalışarak, bilgisayar kullanımında enerji tüketimini düşürmeye yönelik sistemler ile algoritmaların tasarlanmasıdır ve enerji tasarrufu odaklı bilişsel çözümlerin üretilmesidir. Yazılım ürününün üretilmesinden kullanılmasına kadar birçok aşamasında, gerekli olan enerji tüketimi bileşeni doğrudan veya dolaylı yollardan etkili olmaktadır ve sürdürülebilir kalkınma üzerinde olumlu etkileri olan yazılımları sürdürülebilir yazılım olarak da adlandırabiliriz. Aynı zamanda yazılım ürünleri, bilgisayar donanımlarının satılmasında en önemli faktörlerden biri olması sebebiyle yeni donanımların benimsenmesi ve üretilmesiyle önemli ölçüde ilişkilidir. Bundan ötürü yazılım geliştiricilerin yeşil ve sürdürülebilir yazılım bilinci ile çevresel sorumluluğun farkında olarak yazılımlar geliştirmeleri sürdürülebilir ve yeşil yazılım açısından önemlidir (Ağaralı vd., 2023).

Yeşil ve sürdürülebilir yazılım, yazılım ürününün tasarlanma, geliştirilme, dağıtılma ve kullanılma gibi temel aşamalarında çevreye, insana ve topluma olan olumsuz etkileri en az seviyede olan ve sürdürülebilirlik açısından da optimize edilebilecek durumda olabilen yazılımları ifade ederken, yeşil ve sürdürülebilir yazılım mühendisliği ise geliştirilen yazılım

ürünün tüm aşamalarını sürdürülebilirlik açısından değerlendirebilen, tanımlayabilen ve buna göre geliştirebilen mühendislik alanıdır (Naumann vd., 2011).

Sürdürülebilirlik kavramı, yazılımların geliştirilmesi ve geliştirilme aşamalarının bir parçası haline getirilmesi amacıyla geliştirilen ürünler için bir kalite unsuru haline getirilebilir. Böylelikle sürdürülebilirlik, ürün geliştiricileri için bir gereksinim ve öncelik haline gelerek dikkate alınmaya başlanacaktır (Calero vd., 2013).

Yeşil bilişim ile ilgili olan farklı araştırma çalışmaları, genel olarak çevresel sürdürülebilirliğe bilgisayar donanımı açısından odaklanmışlardır. Ancak donanım bileşenleri gibi yazılım bileşenleri de CO₂ salınımlarında bulunurlar. Yazılım ürünleri temel anlamda donanım bileşenlerini yöneterek çalıştırmaktadır ve bundan dolayı donanım ürünlerinin sahip olduğu CO₂ salınımlarına da dolaylı yoldan etki etmektedirler (Mahmoud & Ahmad, 2013).

Yapılan başka bir çalışmada, bölgesel bazda sera gazı envanterleri üretmek amacıyla karbon ayak izi raporları oluşturulması hedeflenmiştir. Bu çalışma kapsamında bölgesel karbon ayak izi salınımlarının ölçülmesi ve rapor halinde sunulması amacıyla bir veri yönetim aracı oluşturulmuştur ve bu araç girilen verileri dinamik bir şekilde raporlayarak çıktılar üretmektedir (Heider vd., 2013).

Yazılım geliştirme süreçlerinde sürdürülebilirlik kavramı dört yönden ele alınabilir. Bunlar; sistemin kavramsal ve yapısal olarak tasarlandığı aşamaları için sürdürülebilirlik, yazılımın üretimi için kaynak kullanımı ve kullanılan donanım bileşenlerinin sürdürülebilirliği, yazılımın bakım süreçleri için sürdürülebilirlik ve ürünün kullanım süreçleri için sürdürülebilirliktir (Penzenstadler vd., 2012).

Kullanılabilirlik terimi ise birçok alan için, çıktı olarak üretilen herhangi bir ürünün müşteri tarafından kullanımı ve kabulü için ana kriterdir. Nihayetinde bir ürün son kullanıcı veya diğer bir ifadeyle müşteri tarafından kullanılmadığı zaman başarısız bir ürün olarak değerlendirilebilir. Tüm bunlardan dolayı bir yazılımı geliştiren kişilerin yazılımı geliştirirken en önemli odak noktalarından biri müşteri olmalıdır (Cheikhi vd., 2006).

Kullanılabilirlik, disiplinler arası çalışmaya uygun bir alan olduğundan dolayı neredeyse her alan ile uyulanabilmektedir. Bu alan, yazılım mühendisliği ile de birlikte çalışılmaktadır. Bununla ilgili yapılan bir alanyazın çalışmasında yazılım kullanılabilirliği ile ilgili 2011-2020 yılları arasındaki çalışmalar incelenmiş olup çalışmalarda kullanılabilirlik, etkililik ve öğrenilebilirlik ifadelerinin ortak değerlendirme ölçütleri olabileceği kanaatine varılmış fakat değerlendirme ölçütü ifadesinin her çalışma için standart bir yapısının olmadığı ve bunun standartlaştırılması gerektiği sonucuna da ulaşılmıştır (Almazroi, 2021).

Mazantas Üniversitesi Bilgisayar Mühendisliği bölümünde yazılım kullanılabilirliği üzerine yapılan bir çalışmada bilgisayar mühendisliği öğrencilerinin müfredatlarına seçmeli olarak yazılım geliştirilirken kullanılabilirlik kavramı ile bir arada ele alınması üzerine bir ödev konusu verilmiştir. Bu konuyu seçen ve çalışan öğrencilerin daha kullanışlı ve kaliteli prototipler geliştirdikleri gözlemlenmiştir (Segura, 2021).

Yazılım ürünleri için son kullanıcılara kullanılabilirlik testlerinin uygulanması geliştiricilere ve analistlere geri bildirim verilmesini sağlayabilmektedir. Kullanılabilirlik testlerini yazılım mühendisliği açısından inceleyen bir çalışmada, iki adet araştırma sorusu ve süre ile kullanıcı tarafından karşılaşılan sorunlar olmak üzere iki adet metrik belirlenmiştir. Çalışma sonucunda bu metriklere ait çıktılar üretilerek, kullanıcıların basit görevleri soru sormadan yaptıkları, kullanıcı sorularının çoğunun karmaşık görevler ile ilgili olduğu, kullanıcıların çoğunun sistemin işlevselliğini dikkate aldığı, ürün prototipi aracılığıyla kullanıcılar üzerinde test yapılmasının yazılım ürününü maliyet açısından düşürerek avantajlı hale getirdiği, yazılımın bakım aşamasını nispeten daha da kolaylaştırdığı sonuçlarına varılmıştır (Bandi & Heeler, 2013).

Alanyazında bulunan çalışmalarda yeşil yazılım kavramı ve kullanılabilirlik kavramının bir arada ele alındığı çalışmalar genellikle var olan sistemler üzerinde son kullanıcılardan ziyade kullanılabilirlik uzmanları ile yürütülen ve yalnızca enerji tüketimi üzerine odaklanılan çalışmalardan oluşmaktadırlar. Bu çalışmada ise var olan sistemler incelenerek yeni bir sistem oluşturulmuştur. Çalışmada odak nokta, uygulamayı kullanacak son kullanıcılardır ve çalışma verileri son kullanıcıların uygulamayı deneyimlemeleri yolu ile elde edilmiştir. Çalışmanın temel aşamalarını inceleyecek olursak; birinci adımda bir hastane randevu sistemi uygulaması geliştirilmiştir. Uygulama geliştirilirken herhangi bir kullanıcı görüşü dahil edilmemiştir. Uygulama, mevcutta var olan hastane randevu sistemleri incelenerek Python programlama dili ve Django web çatısı kullanılarak geliştirilmiştir.

İkinci adımda, geliştirilen uygulama üzerinde uygulamanın tüm bileşenlerinin kullanılabileceği şekilde bir görev listesi oluşturularak, son kullanıcılardan uygulamayı görev adımlarını yerine getirerek kullanmaları istenmiştir. Son kullanıcılar uygulama adımlarını tamamladıktan sonra kullanıcılar ile birebir açık uçlu sorulardan oluşan bir görüşme gerçekleştirilmiştir. Bu görüşmede, kullanıcıların uygulamayı kullanırken hangi alanlarda zorluklar yaşadıkları, hangi alanlardan memnun kaldıkları, uygulamayı daha kolay kullanabilmeleri için nelerin olması gerektiği gibi sorular kullanıcılara yöneltilerek kullanıcıların uygulama hakkında etkililik, verimlilik ve memnuniyet bileşenleri açısından görüş ve önerilerinin alınması hedeflenmiştir.

Aynı zamanda son kullanıcılar uygulamayı deneyimlerken, uygulamanın o anda ne kadar enerji tüketimi ve karbon salınımı yaptığı ölçümleri arka planda gerçekleştirilmiştir.

Üçüncü adımda, kullanıcılardan elde edilen görüş ve öneriler tek bir rapor altında toplanarak analiz edilmiştir. Yapılan analizler sonucunda, uygulamada bulunan her alan için kullanıcı görüşleri gruplandırılmış ve görüşlere göre uygulama parçalarının daha kolay kullanılabilmesi için nasıl tasarlanması gerektiği hakkında ifadeler sıralanmıştır.

Dördüncü aşamada, son kullanıcılardan alınan görüşlerin sonuçları ışığında uygulama hem tasarım olarak hemde uygulamanın fonksiyonelliği ve çalışma biçimi olarak revize edilerek son kullanıcı görüşleri dahilinde daha güncel bir uygulama elde edilmiştir.

Beşinci adımda ise güncel uygulama tekrar son kullanıcıların deneyimlerine sunulmuştur. Son kullanıcılar uygulamanın son halini kullanırken, ikinci aşamada arka planda gerçekleştirilen enerji tüketimi ve karbon emisyonu değerleri anlık olarak tekrar ölçümlenmiştir. Ölçümler uygulamanın son hali kullanılırken elde edildikten sonra kullanıcılardan tekrar uygulama ile ilgili geri dönüşler alınmıştır.

Kullanıcı görüşlerinin proje aşamalarına dahil edilmesinin birçok farklı faydası bulunmaktadır. Bunların başında zaman ve maliyet kavramları gelebilir. Uygulamaları son kullanıcılar kullanacakları için kullanıcıların dahil edilmediği bir proje tamamlandıktan sonra projenin kullanıcı açısından istenen sonuçları vermemesi ve buna bağlı olarak uygulamanın tekrardan geliştirilmesi veya düzeltilmesi durumları ortaya çıkabilmektedir. Bu durum ise geliştiriciler ve uygulama için hem zaman hemde maliyet kaybı anlamına gelebilir. Bu durumlar göz önüne alındığında çalışmanın amacı ve motivasyonu, yazılım projeleri geliştirilirken kullanıcı deneyimi ve görüşlerinin projeye dahil edilmesinin yeşil yazılım açısından ele alınarak, enerji tüketimi ve karbon emisyonu açısından nasıl bir etkisinin olduğunu deneysel olarak araştırmaktır.

Alanyazında bulunan birçok çalışmada da bahsedildiği gibi doğal kaynakların tükenmesi ve sera gazlarının salınımı gibi konular günümüz dünyasının ve geleceğin en büyük sorunlarından biri haline gelmiştir. Birçok çalışma dalı kendi alanlarında yeşil dönüşüme katkı sağlamaya çalışmaktadır ve çalışmalarını bu yönde geliştirmeyi hedeflemektedirler. Yazılım alanında ise yeşil dönüşüme evrilmek için yeşil yazılım ve yeşil yazılım mühendisliği kavramları kullanılmaktadır. Kullanılabilirlik ve kullanıcı deneyimi kavramları ise insan bileşeninin içerisinde olduğu her alan için geçerliliğini ve önemini koruyan kavramlardır. İlgili çalışmada günümüzün önemli ve popüler konularından olan yeşil yazılım ve kullanılabilirlik kavramları için ortak bir çalışma alanı oluşturacaktır. Kullanıcı deneyimlerinin yazılım projeleri aşamalarına dahil edilmesinin yeşil dönüşüme ve yeşil yazılıma olumlu veya olumsuz hangi

yönde etki edeceği araştırılarak, elde edilen sonuçların yazılım geliştiricilerine ve yazılım mühendislerine hem proje adımlarına kullanıcı deneyimi ve kullanılabilirlik kavramlarının dahil edilmesi için hem de bu şekilde enerji verimliliği açısından daha iyi yazılımlar üretmeleri için bir motivasyon kaynağı olacağı düşünülmektedir.

İlgili çalışmanın sonraki aşamaları aşağıdaki gibi organize edilmiştir. İkinci bölümde kuramsal çerçeve ve ilgili çalışmalar, kuramsal temeller başlığı altında anlatılmaktadır. Üçüncü bölümde materyal ve yöntem detayları ile ele alınmaktadır. Dördüncü bölümde deneysel çalışma sonucunda elde edilmiş araştırma bulguları sunulmaktadır. Beşinci bölümde tartışma yer almaktadır. Altıncı ve son bölümde ise sonuçlar ve öneriler yer almaktadır.

KURAMSAL TEMELLER

Yeşil Yazılım

Yeşil yazılım kavramı temelde karbon emisyonunun azaltılması yolu ile yazılım ürünlerinin en düşük seviyede karbon salınımı yapacak şekilde tasarlanması, geliştirilmesi ve üretilmesi şeklinde tanımlanabilir ve enerjinin verimli kullanılması, karbon verimliliği farkındalığının oluşturulması, verimli donanımların kullanılması durumları yazılımların karbon emisyonlarını azaltacak maddelerdir. Yeşil yazılım kavramında farkındalığın oluşturulması için geliştiriciler tarafından karbon verimliliği, enerji verimliliği, karbon bilinci, donanım verimliliği, ölçülendirme işlemi ve iklim taahhütleri maddelerinin iyice kavranması gerekmektedir (Anita Holczmann vd., 2023).

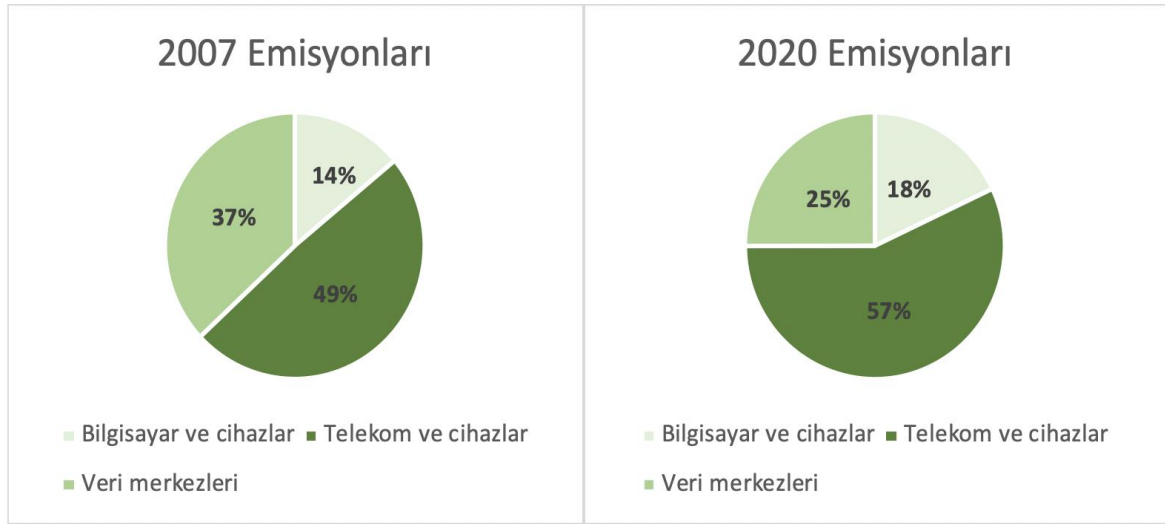


Şekil 1. Yeşil yazılım prensipleri

Sera gazları, atmosferde güneş tarafından gelen ısıyı hapsedirler ve bu yolla dünya üzerindeki sıcaklık seviyesini arttırırlar. Farklı türlerdeki sera gazlarının doğal çevre üzerinde yayılımına bağlı olarak dünyamızda meydana gelen küresel ısınma ve iklim değişiklikleri vb. durumlar sürdürülebilir dünya düzenine aykırı bir hal almaya başlamıştır ve bu sonuçların ortak etkeni insan faaliyetleridir. Buzulların erimesi, su döngülerinde meydana gelen değişiklikler, sıcaklık seviyelerinin olağan dışında artması ve okyanus akıntılarının tuz ile asidik oranlarında meydana gelen değişimler, kuraklık gibi durumlar iklim değişikliğinin sonuçları olarak gösterilebilirler. Karbon emisyonu etkisini ölçümlemek amacıyla “CO₂eq”, “Co₂-eq”, “CO₂e” ölçüm terimleri kullanılmaktadır. Bu duruma örnek vermek gerekirse, yaklaşık 1 tona yakın metan gazı 84 tona yakın karbondioksit ile atmosfer

üzerinde aynı seviyede sıcaklık etkisine sahip olmaktadır ve bu salınım 84 ton CO₂eq şeklinde de ifade edilebilmektedir (Anita Boltzmann vd., 2023).

Greenpeace tarafından sunulan “How dirty is your data?” adlı raporda, BT sektörü için karbon ayak izi tahmini doğru yapmanın, marka ve şirketlerin hem şeffaflık konusundaki eksikliklerinden hemde gizlilik ile ilgili endişeleri ve benzeri durumlardan kaynaklı olarak zor olduğundan bahsedilmiştir (Cook & Van Horn, 2011). Bununla ilgili yapılan en bilindik tahminlerden birinin Smart 2020 raporu olduğundan bahsedilmiştir. Bu raporda, 2007 ve 2020 yıllarına ait veri merkezleri, Telekom ve cihazlar, bilgisayar ve cihazlar olmak üzere sonuçlar 3 başlıkta verilmiş olup, raporun yayınlandığı son iki yıl içinde Telekom ve cihazların büyük oranda karbon ayak izine sahip olduğu ifade edilmiştir (The Climate Group, 2008).



Şekil 2. Smart 2020 verileri

Yine Greenpeace tarafından sunulan “Make IT Green Cloud Computing and its Contribution to Climate Change” adlı raporda SMART 2020 raporuna değinilerek kişisel bilgisayarların kullanımının 2007’den 2020’ye kadar yaklaşık 4 milyar cihaza çıkacağı ve dizüstü bilgisayar emisyonunun masaüstü bilgisayarları geride bırakacağı ifade edilmiştir. Aynı raporda kişisel cep telefonu kullanımının da neredeyse ikiye katlanacağından ve yine salınımı yapılan karbon miktarının tüm telekomünikasyon altyapısında iki katına kadar çıkacağı ifadeleri kullanılmıştır. Aynı çalışmada bir önceki paragrafta bahsedilen raporda bulunan rakamların yüzdelik halleri tablo haline getirilerek sunulmuştur (Greenpeace International, 2010).

Tablo 1. Greenpeace Verileri

	Emisyonlar 2007 (MtCO2e)	Yüzde 2007	Emisyonlar 2020 (MtCO2e)	Yüzde 2020
Dünya	830	%100	1430	%100
Sunucu/Veri Merkezleri	116	%14	257	%18
Telekom Altyapısı ve Cihazları	307	%37	358	%25
PC'ler ve Çevre	407	%49	815	%57

İklim değişikliğinin sonuçları ve bu sonuçlara bağlı olarak ortaya çıkan etkileri sürdürülebilir dünya hedeflerine büyük bir engel olduğundan dolayı dünya, ülkeler, şehirler, kurumlar ve kuruluşlar çapında bu konu ele alınmakta ve geçmiş zamandan beri bu konu üzerinde çalışmalar yapılmaktadır. Bunlara, 2015 yılında birçok ülke tarafından kabul gören “Paris İklim Anlaşması”, sera gazı salınımlarını önleme amacıyla çalışmalar yapmayı hedefleyen “Birleşmiş Milletler İklim Çerçeve Sözleşmesi (UNFCCC)”, hükümetlere iklim politikaları geliştirmeyi amaçlayan “Hükümetler arası İklim Değişikliği Paneli (IPCC)” gibi kuruluşlar örnek olarak verilebilir. Yaşam boyunca gerçekleştirdiğimiz her faaliyet atmosfere karbon saldıgı için bunu durdurmak mümkün gözükmesi de olabildiğince bu salınımı azaltma hedefleri edinmek ve uygulamak faydalı olabilecektir. Karbon salınımı konusunda bilinç kazanılması ve farkındalık oluşturulması toplumda tüm bu olumsuz durumların önlenmesi için önemli bir etkidir. Bu farkındalığın günlük hayatın her alanında edinilmesi gerekmektedir. Yazılım açısından ise bu farkındalık yeşil yazılım ile sağlanabilir. Hem geliştiricilerin hem de kullanıcıların bu konu üzerine çalışması ve bilgilendirilmesi yazılım alanı için yeşil dönüşüme önemli katkılar sağlayabilir. Örneğin bilgisayar üzerinde bir programı kullanırken bilgisayarın hangi kaynaklardan elektrik kullanıldığını bilmek kullanıcılar açısından pek mümkün değildir. Bilgisayarın kullandığı elektrik düşük ya da yüksek karbon salınımı yapan bir kaynaktan kullanılabilir ve karbon yoğunluğu bölgeden bölgeye değişiklik göstermektedir. Bundan ötürü toplumun her kesiminde genel olarak oluşacak karbon emisyonu bilinci her açıdan verim ve fayda sağlayabilmektedir (Asim Hussain vd., 2022).

Enerji tüketimi ise karbon salınımına etki eden en önemli faktörlerden bir diğeridir. Enerji, günümüzün teknoloji dünyasında hep kullandığımız cep telefonlarından zamanın popüler konularından olan yapay zekâ çalışmalarına kadar neredeyse her noktada bulunmaktadır. Aynı şekilde yazılım uygulamalarının geliştirilmesi ve çalıştırılabilmesi için

elektrik yani enerji kullanımı gerekmektedir. Yeşil yazılım ifadesi bu noktada işin içerisine dahil olmaktadır. Yazılım uygulamalarının her aşamasını tüketilen enerji açısından daha verimli hale getirmek yeşil yazılımın iş alanı içerisine girmektedir. Genel anlamda elektrik enerjisinin temiz bir enerji türü olduğu kanısı bulunmaktadır. Fakat elektrik fosil yakıtların yakılması yolu ile elde edilmektedir ve bu durumda elektrik tüketimi ile salınımı yapılan karbon tüketiminin aslında birbiri ile doğru orantılı olduğu anlamına gelmektedir. Enerji verimliliğini veri merkezleri bazında sağlayabilmek için ise güç kullanım etkinliği yani Power Usage Effectiveness (PUE) değeri dikkate alınmalıdır. PUE değeri temel anlamda bilgisayar ekipmanının ne kadar enerji kullandığını ifade etmektedir. Enerji tüketimi yalnızca yazılım uygulaması ile ilgili de değildir. Uygulamanın üzerinde çalıştırıldığı donanım bileşenleri elektrik kullanımına ek yük getirerek yazılım uygulamasının genel anlamda kullandığı enerji miktarını etkilemektedir. Örneğin gömülü karbon olarak ifade edilen kavram, bir donanım cihazının veya bileşeninin üretilme, kullanım süresi ve bitince imha edilme aşamalarında yapılan karbon salınımını ifade etmektedir (Sean Mcilroy vd., 2023).

Kullanılan teknolojik cihazların ve kullanım oranlarının giderek artması sebebiyle sera gazı emisyonlarının da arttığından bahsedilen bir çalışmada yazılım ürünlerinin işlevlerinin azaltılmadan enerji tüketimlerinin azaltılması gerektiğine değinilmektedir. Aynı çalışma, yazılım geliştirme süreçlerinin her aşamasında sürdürülebilirliği sağlamak için enerji tüketimine, CPU bileşeninin temelinde odaklanmaktadır. Çalışmada yazılımların güç tüketimlerinin tahmin edilmesini sağlamak amacıyla TEEC yani enerji tüketim tahmin aracı geliştirilmiştir. Bu araç aynı zamanda yazılımın kaynak kodunda fazla oranda enerji tüketen noktaları da dinamik olarak bulmaktadır. Yine aynı çalışmada yazılım güç tüketimini tahmin etmeyi sağlayan araçların oluşturulması aşaması için dört katmanlı bir tahmin metodolojisi de sunulmuştur (Acar, 2017).

Yeşil yazılımın kullanım alanları ve araçları üzerine 27 farklı yazılım dili için bellek kullanımına ve enerji tüketimine dair yapılan bir çalışmada, farklı programlama dilleri için yazılmış farklı problemler ve bunların çözümlerini içeren bir program seti kullanılmıştır. Odak noktaları ise enerji tüketimi, çalışma süresi ve bellek kullanımı ilişkileridir. Programlama seti ile hem çalışma süresi değerlerine hem de bellek kullanımına ilişkin değerlere ulaşılmıştır ve bu değerler başka ölçüm araçları ile hem donanım farklılıklarından oluşabilecek değişiklikleri değerlendirmek hem de doğrulama yapmak amacı ile tekrar ölçümlenmiştir. Yapılan ölçümlerden dört farklı değer elde edilmiştir. Bunlar, tüketilen enerji, çalışma süresi, enerji ve sürenin oranı, bellek kullanımıdır. Çalışma sonucunda üretilen çıktı tablolarında, çözümlenen problemlere göre dillerin enerji tüketimi ve bellek kullanım değerleri değişmektedir. Aynı

zamanda oran deęerleri incelendięinde, alıřmalarda enerji tketiminin zaman bileřeni ile her zaman doęru orantılı olmadığı, dil ve probleme gre farklılık gsterebileceęi belirtilmektedir. Tm deęerler ortak olarak normalize edilerek ıkartılan genel sonu tablosunda ise enerji ve zaman aısından C dili, bellek kullanımı aısından ise Pascal dili en verimli olarak gzkmektedir. Fakat bu sonular tm programlama dillerini kapsamadıęından dolayı net ifadelerde bulunmak doęru olmayacaktır (Pereira, 2018).

Yeřil veya dięer bir ifade ile enerji verimli veri yapılarının nasıl uygulanacağı zerine yapılan bařka bir alıřmada ise, zellikle pil mr olan cihazlar iin enerji verimlilięinin iyileřtirilmesi durumunun, cihazların kullanım srelerinin veya pil mrlerinin uzatılmasında etkili olduęundan bahsedilmiřtir. Bir cihazın kullanım sresinin ve pil mrnn uzatılması da srdrlebilirlięe fayda saęlayacaktır. Aynı zamanda alıřmanın bazı sonuları yazılımı daha hızlı alıřacak řekilde optimize etmenin, yazılımın her zaman daha az g tketeceęi anlamına gelmedięini kanıtlar niteliktedir. alıřmada, doęru iř yk iin doęru veri yapısının kullanılması metodolojisi zerinde durularak GreenC5 adında enerji verimlilięi durumuna gre dinamik olarak dnřmler saęlayan bir prototip geliřtirilmiřtir. Bu prototip makine ęrenimi veri yapısıdır. alıřmadaki veri yapısı orijinal C5 veri yapısından farklı olarak dinamik bir řekilde enerji farkındalıęına sahip olarak belirtilmiřtir. Yani bu veri yapısı enerjinin tasarruflu kullanımı metrięini dikkate alarak tahminde bulunma ve buna gre uyarlanabilme zellięine sahiptir. alıřmada karar verici bileřen olarak adlandırılan bileřen ile dnřm srecine karar verilir ve daha sonra ise yeni veri yapısı oluřturularak elde bulunan veriler yeni veri yapısına kopyalanıp eski veri yapısı atılır. alıřmada gerekleřtirilen prototipin amacı enerji verimini maksimize ederek fazla ykleri engellemektir (Michanan et vd., 2016).

Yeřil yazılıma ynelik prensipler, biliřim teknolojilerinin birok alanında uygulanmaya alıřılmaktadır ve bunlarla ilgili alıřmalarda yapılmaktadır. rneęin makine ęrenmesi algoritmalarının karbon lmlerinin yapılması zerine yapılan bir alıřmada, makine ęrenmesi algoritmalarında enerji tketimi durumunun, eęitim esnasında kullanılan sunucunun hangi konumda bulunduęu ile kullanılan enerjinin ne tr bir kaynaktan geldięi, eęitimin sre deęeri ve eęitimin zerinde gerekleřtirildięi donanım cihazları gibi bileřenlere baęlı olduęuna deęinilmiřtir. Aynı zamanda makine ęrenmesi algoritmalarının eęitimleri iin ne kadar karbon emisyonu yapıldıęı ile ilgili bir hesaplama aracı geliřtirmiřlerdir. alıřmada sunucu konumlanması ve eęitimin ne kadar srdę gibi bileřenler dikkate alınmıřtır nk bazı eęitimler haftalar hatta aylar kadar srebilmektedir. Bahsedilen bu lm aracı girdi olarak coęrafi blge, GPU tr ve eęitimin sresi deęerlerini alarak COeq deęerini ıktı olarak vermiřtir (Lacoste vd., 2019).

Blockchain mutabakat mekanizmasının enerji tüketimine ve karbon salınımına dikkat çeken bir çalışmada, PoW (Proof-of-work) ve PoS (Proof-of-stake) mutabakat mekanizmalarından hangisinin enerji açısından daha verimli olduğu üzerinde araştırmalar yapılmıştır. İşin kanıtı olarak ifade edilen PoW mekanizmasında doğrulama hesaplamaları ağır bir düzeyde yapılmaktadır. Burada zincire dahil olan noktalar, kendilerini doğrulamak için madencilik yaparak bir kriptografiyi çözmeye çalışırlar ve çözüme önce ulaşan madenci ödülü almaya hak kazanır. Bu işlemler ise çok miktarda enerji kullanımı anlamına gelmektedir. PoS mekanizmasında ise çözülmesi gereken herhangi bir kriptografi yoktur. Burada kullanıcının alacağı ödül kullanıcının zincirde olduğu zaman yani kullanıcının yaşı ile belirlenir. Çalışma kapsamında PoW tabanlı ethereum ve PoW tabanlı bitcoin' in karbon emisyonlarının dünyayı 2120 yılına kadar 3,93 °C kadar ısıtabileceğini göstermektedir. PoW' a kıyasla PoS mutabakat mekanizmasının 0.11 °C oranında bir artışa neden olacağı sonucuna varılmıştır. Çalışmada, blockchain mekanizmasının daha sürdürülebilir hale gelmesi için PoS mekanizmasının kullanılması gerektiği, bu mekanizmanın PoW mutabakat mekanizmasına göre karbon ayak izini %99 oranında azaltabileceği sonucuna varılmıştır (Shi et vd., 2021).

Nesnelerin İnterneti (Interent of Things - IoT) alanını yeşil bilişim ile birlikte inceleyen bir çalışmada, özellikle IoT kapsamında geliştirilen mobil uygulamalar dikkate alınmıştır. Çalışmada, nesnelerin interneti alanının kullanımının artması ile birlikte birbirine bağlantı halinde olan cihazların ve bunlarla birlikte mobil uygulamaların da hızlı bir biçimde artmasından dolayı kaynak kullanımı ve pil kapasitesi bileşenleri için sorunların ortaya çıktığından bahsedilmektedir. Mobil uygulamalarda hızın, IoT cihazlarında performansın üst seviyede tutulması için uygulama görevlerinin buluta aktarıldığından, mobil bulut bilişim (MCC) adı verilen yapının pil ömrünü uzatabildiği, ayrıca depolama alanı da sağladığı fakat maliyet ve hizmet gecikmesi açısından bu yapının olumsuz olduğundan bahsedilmiştir. Bu sınırlılıklara çözüm getirmek amacıyla çalışmada daha düşük miktarda enerji tüketimi ile daha az gecikme sağlama hedeflenen mobil kenar bilgi işlem (MEC) paradigması önerilmiştir. Çalışmada MEC paradigmasını enerji verimli hale getirmenin bir yolu olarak yeşil enerjiden faydalanılması önerilmiştir. Yani enerji kaynağı olarak güneş ve rüzgâr gibi yenilenebilir enerji kaynaklarının kullanılarak karbon emisyonunun düşürülmesi ve dış mekanlarda çalışmakta olan IoT cihazları içinde pil ömrünün uzatılabileceği önerilmiştir. Ancak yenilenebilir enerji kaynaklarının sürekli enerji sağlama durumu net olmadığından dolayı sürdürülebilir enerji kaynakları ve kömür gibi diğer enerji kaynaklarının hibrit olarak kullanıldığı bir tasarım sunmuşlardır (Xu vd., 2021).

İnsan Bilgisayar Etkileşimi

Günümüz dünyasında gelişen teknoloji yalnızca Bilgi Teknolojileri (BT) alanını ilgilendirmemektedir. Yapılan uygulamaların, geliştirilen ve üretilen yazılımların en önemli odak noktalarından biri olan kullanıcı yani insan faktörü de bu alanların gelişmesini çok büyük oranda etkilemekte ve bu alanlara paralel olarak kullanıcı faktörünün de ihtiyaç, istek ve beklentileri değişerek gelişmektedir.

İnsan bilgisayar etkileşimi ve kullanılabilirlik mühendisliğine dair yazılan “Teoriden Pratiğe İnsan-Bilgisayar Etkileşimi ve Kullanılabilirlik Mühendisliği” kitabında İnsan Bilgisayar Etkileşimi (İBE) kavramının birçok kişi tarafından farklı tanımlamalar ile tasvir edildiğinden ve bundan ötürü kavramı tek sabit bir tanım ile bağdaştırmak yerine alanın hangi tür problemler ile ilgilendiği, neleri ele aldığı ve olmazsa olmaz olarak önceliklendirdiği boyutlarını dikkate almanın daha anlaşılır olacağından bahsedilmiştir. Bu boyutlar ise İBE’ nin disiplinler arası olması, kullanılabilirlik kavramı, tasarım ve etki üzerine çalışması olmak üzere dört temel olarak ifade edilmiştir. İnsan bilgisayar etkileşiminin disiplinler arası olması aynı zamanda bu alanda çalışan uzman kişilerinde farklı görüş açılarına sahip olmaları gerekliliğini de yanında getirmektedir. Bunlara en temelde bilgisayar bilimi, psikoloji ve sosyoloji örnek verilmiştir. Özellikle 1990’lı yıllarda web teknolojilerinin de ilerleme ve gelişim göstermesi ile birlikte insan bilgisayar etkileşimi çalışmaları daha önemli ve dikkat çeker hale gelerek çalışmalarda teknik merkezi tasarımların yerine kullanıcı merkezli tasarımlar kullanılmaya başlanılmıştır. Aynı zamanda bilgisayarların ve teknoloji kavramının daha sıradan insanların gündelik hayatlarının bir parçası haline gelmesi ile birlikte kullanılabilirlik kavramı önem kazanmaya ve kullanılabilirlik problemleri de ortaya çıkmaya başlamıştır. Sürecin ilerlemesi ve teknolojik ürünlerin artık neredeyse her insan tarafından kullanılmasının sıradan hale gelmeye başlaması ve artması ile birlikte İBE kavramı daha da önem kazanarak ürünlerde kolay kullanılabilirlik özelliği, kullanıcılar için aranan bir özellik ve ürün adına bir avantaj haline gelmeye başlamıştır. İnsan bilgisayar etkileşimine dair çalışmalar, konunun yurtdışında öneminin anlaşılması ve birçok çalışmalar ile ilerlemelerin sağlanması sonrasında Türkiye’de çalışılmaya başlanmıştır. Türkiye’de 2002 yılında ilk insan bilgisayar etkileşimi dersi ODTÜ’de verilmiştir ve aynı zamanda yine ilk İBE laboratuvarı ODTÜ bünyesinde kurulmuştur. “İnsan bilgisayar Etkileşimi” alanı hem bilişim teknolojileri alanı için ve bu şekilde dolaylı olarak insan faktörü için, hemde çalıştığı diğer disiplinler için önem arz etmektedir. Çünkü gündelik hayatı artık bilgisayarlar ve bu teknolojilerden ayrı tutmak neredeyse imkânsız hale gelmiştir ve son kullanıcıların bilişim ürünlerinin tasarlanması, geliştirilmesi ve üretilmesi aşamalarından bağımsız olarak tutulması teknoloji için ciddi

problemlere ve kayıplara yol açmaktadır. Her alandan bağımsız olarak özellikle tıp gibi hata oranının en düşük seviyede tutulması gereken alanlarda teknolojilerin önemli düzeyde kullanılması, yazılım ve insan masraflarının artması, bilişim alanına yapılan yatırımlara oranla üretkenliğin daha düşük seviyede kalması ve insan davranışları ile bağlantılı olarak bilişsel yapıları anlama çabalarının artması insan bilgisayar etkileşimi alanının günümüz için ne derecede önemli olduğuna örnek olarak verilebilir (Çağıltay, 2011).

Bilgisayarların geçmişine kadar uzanan insan bilgisayar etkileşimi tarihinden bugüne kadar alan ile ilgili birçok çalışma yapılarak çalışmalarda birçok farklı açıdan tanımlamalar yapılmıştır. Bir çalışmada insan ve bilgisayar arasındaki bir ilişkide ilk amacın sistemin kullanıcının geri bildirimlerini algılaması ve buna göre uygun tepkiler vermesine odaklanması olduğundan, aynı zamanda insan bilgisayar etkileşiminin iki yönlü bir süreç olduğundan bahsedilmiştir (Issa & Isaias, 2015).

İnsan bilgisayar etkileşiminin tarihine dair yapılan bir çalışma kapsamında bilgisayar arayüzü tanımından genelleme yapılarak arayüzün insan ve bilgisayar arasında bir enerji ve etkileşim sistemi olduğundan bahsedilmektedir. Çalışmada insan ve bilgisayar arasında oluşan bu sistemin ilişkisini etkileyen birçok faktör olduğundan ve bu faktörlere, duyular, eller gibi insan bileşenleri ile kullanılan yazılım, giriş çıkış cihazları, ekranlar gibi örnekler verilmektedir. İnsan bilgisayar etkileşiminin tarihi ve arayüzler bilgisayar tarihi ve gelişimi ile doğru orantılı olarak büyük değişikliklere uğramıştır. Çalışmada bilgisayarlar ile etkileşimlerin birçok aşamada potansiyel olarak kullanıcılar için farkında olunmadan bir rutin haline geldiğinden ve bundan dolayı arayüzlerin fonksiyonellik ve tasarım olarak ön plana çıkarılmasının kullanıcıların bilgisayarlar ile etkileşimini daha somutlaştırılmış bir hale getirebileceği savunulmaktadır. Aynı zamanda insan bilgisayar etkileşiminin, bedensel etkileşim çeşitleri aracılığıyla insan vücudunun bilgisayar teknolojileri ile genişletilmesi yani giyilebilir teknolojiler ile entegrasyonların yapılması gibi bir çalışma alanı olarak da incelenebileceğinden bahsedilmiştir. Çalışmada İBE'nin genel anlamda sadece IT alanı ile birlikte ele alınmaması gerektiğinden, bilgi işlem tarihçesinden daha da fazla incelenmesi ve ele alınması gerektiğinden, İBE'nin yalnızca insan ve bilgisayar kavramlarının etkileşimi olarak değil de insan ve bilgisayar kavramlarının birbirleri ile iletişim kurarak birbirlerine yön verdiği, değiştirdiği bir kavram olarak incelenmesi gerektiğine dikkat çekilmiştir (Petrick, 2020).

İnsan bilgisayar etkileşimini bilişsel yük açısından inceleyen bir çalışmada iyi bir arayüz tasarımının insan kullanımını ve katılımını teşvik etmesi yönünden dolayı bilişsel yükün insan bilgisayar etkileşimi çalışmalarında bir ölçüm değeri olarak kullanıldığından bahsedilmektedir. Arayüzlerinin aşırı derecede karmaşık ve zor anlaşılır olması veya kullanıcının dikkatini

çekmeyecek kadar aşırı seviyede kolay olması bilişsel yük açısından olumsuz olacaktır. Çalışmada genel olarak yapılan insan bilgisayar etkileşimi kapsamında farklı arayüzlerinin karşılaştırılarak anket ve benzeri ölçüm araçları ile kullanıcıların bilişsel yüklerinin ölçüldüğünden bahsedilmektedir (Kosch vd., 2023).

İnsan bilgisayar etkileşimi kavramı en temelde makinelerin ortaya çıkması ile birlikte ortaya çıkmış bir kavramdır. Kavram, temel olarak sistemlerin işlevsellikleri ve kullanılabilirliklerini dikkate almaktadır. Buna göre de bir sistemin asıl olarak gerçekten etkin olup olmadığını görmek sistemdeki işlevsellik yani fonksiyonellik ve kullanılabilirlik arasındaki dengenin sağlanması ile mümkün olabilir (Sinha vd., 2010).

Türkiye’de, insan bilgisayar etkileşimi alanında yapılan uygulamalardan bahsedilen bir çalışma ODTÜ’de gerçekleştirilmiş olup, çalışmada özellikle kullanılabilirlik çalışmaları kapsamında test işlemlerinin yürütüldüğü ifade edilmektedir. Örneğin laboratuvar kapsamında yapılan bir çalışmada görme engelli öğrencilerin web sayfalarını kullandıklarını incelemek ve web sayfalarını bu öğrenciler için daha kullanılabilir hale getirerek standartlaştırmak amaçlanmıştır. Başka bir çalışmada ise iki farklı arayüz kullanılabilirlik açısından kıyaslanmıştır (Acartürk & Çağıltay, 2006).

Kullanılabilirlik

Kullanılabilirliği yalnızca insan bilgisayar etkileşimi veya bilişim teknolojileri ile kısıtlamak doğru olmayacaktır. Aslında kullanılabilirlik günlük hayatımızın büyük bir parçası halindedir. İnsanların günlük hayat rutinlerinde yaşadıkları binayı nasıl kullandıkları, binanın tasarımından kaynaklı bir problem yaşayıp yaşamadıkları, alışveriş yapmak için kredi kartlarını nasıl kullandıkları veya satın aldıkları herhangi bir ürünü nasıl kullanacaklarını ürün içeriğinden çıkan kullanım kılavuzlarından veya internet yardımı ile öğrenmeleri gibi örnekler kullanılabilirlik ile ilgilidir. Yani yaşanan alanların ve kullanılan ürünlerin ne kadar kullanılabilir olduğunun günlük yaşantıları doğrudan etkilediği göz önüne alındığında kullanılabilirlik ile insan yaşantısının birçok aşamasında karşılaşıldığı söylenebilir. Kullanılabilirlik, kullanıcıları gereksiz çaba, zaman ve kaynak sarf etmekten kurtarmayı, verimliliği sağlamayı ve arttırmayı, kullanıcı memnuniyeti ve rahatlığını sağlayarak kullanıcıyı mutlu etmeyi, geliştirme süreçleri için ekonomik olarak da verim ve fayda sağlamayı, geliştirme aşamasının sonunda son kullanıcılara daha eksiksiz bir ürün sunmayı, üründe sonradan çıkabilecek, ürünün içerik ve tasarımının değiştirilmesini gerektirecek problemlerin çıkma olasılığını en aza indirmeyi hedeflemektedir.

Kullanılabilirlik ifadesi dört ana bileşenden oluşmaktadır. Bunlar; uygulamayı veya ürünü kullanacak, hedef insan grubu olan “kullanıcı” bileşeni, ürün veya uygulama geliştiricileri tarafından ortaya konulan ve kullanıcı bileşeni kullanılarak etkileşim kuracağı “arayüz” bileşeni, kullanıcı ve arayüz bileşenleri için uygulamanın ortak bir noktası olarak görülebilecek ve kullanıcıdan arayüzü kullanırken yapması beklenen “görev” bileşeni ve tüm bu bileşenleri içinde barındıran “bağlam” bileşenidir. İnsan bilgisayar etkileşimi, kullanılabilirliği kişi ve bilgisayar arasındaki anlaşılabilirlik, doğru ve verimli kullanım açısından ele almaktadır. İnsan bilgisayar etkileşimi, kullanılabilirliği temel olarak 3 farklı tanım yapısında incelenmektedir. Bunlardan ilki, kişiye, konuma, ölçütlere ve göreve bağlı olarak değişiklik gösterebilen kullanılabilirliğin sabit olarak ifade edilemediği anlamsal tanımdır. İkincisi bir arayüzün hangi özellikleri barındırdığına odaklanılarak kullanılabilirliği tanımlayan, arayüz özellikleri tanım yaklaşımıdır. Ancak bu tanım yaklaşımı da kullanılabilirlik kavramı için tam olarak doğru tanımlı sunmamaktadır. Çünkü bu tanım şeklinde, kullanıcı kavramını işin içerisine katmadan yalnızca ekranda var olan özelliklere bakan bir tanım söz konusudur. Üçüncüsü ise kullanılabilirliğin işlevsel tanımıdır. Bu tanım yapısında alanda çalışma yapan bazı araştırmacıların tanımlamalarından bahsedilmiştir. Örneğin Jacob Nielsen kullanılabilirliği beş temel madde ile tanımlamıştır. Bunlar verimlilik, memnuniyet, öğrenilebilirlik, hatırlanabilirlik ve hatalar şeklindedir (Nielsen, 1993). Ancak bu tanım şeklinde de uzmanların tanımlarındaki çeşitliliklerden dolayı yine ortak kabul görmüş bir tanım ifadesine varılamamıştır. Temel, sabit ve herkes tarafından kabul görececek bir kullanılabilirlik tanımı için Uluslararası Standartlar Enstitüsü (ISO) “Kullanılabilirlik belirli bir kullanıcı grubunun, belirli görevleri, belirli bir bağlamda etkili, verimli ve memnuniyet ile yerine getirmeleridir.” şeklinde tanımlamıştır. İnsan Bilgisayar Etkileşimi (İBE) kapsamında ise kullanılabilirlik, kullanıcıların uygulamayı ne derecede verimli ve etkili kullandıkları ile yine uygulamadan ne derecede memnun kaldıkları ile ölçümlenebilmektedir. Etkililik ilkesi, kullanıcı ve kullanıcının yaptığı görev ile ilgilidir. Bu ilke kullanıcıdan kendisinden yapılması istenen görev adımlarını ne derecede tamamlayabildiği ve ne kadar doğru yapabildiği ölçümlenmektedir. Verimlilik ilkesi ise kullanıcının yapması gereken görevi ne kadar sürede yapabildiği, görevi yaparken ne kadar çaba sarf ettiği (yani kolay bir şekilde mi yapabildi mi yoksa görevi yaparken zorluk yaşadı mı) ve görevi yapmak için maliyet açısından ne gibi bir yol izlediği gibi maddeler göz önüne alınarak uygulamanın ne kadar verimli olduğu ölçümlenebilmektedir. Memnuniyet ilkesi ise kullanıcı uygulamayı deneyimlerken kullanıcıda oluşan duygu durumunu yani uygulamayı kullanırken kullanıcının uygulamadan ne kadar memnun kalıp kalmadığını ifade etmektedir. Bu üç ilke ayrı ayrı ele alındığında

kullanılabilirlik, değerlendirme açısından yetersiz kalabilmektedir ancak üç ilkenin de kullanılması kullanılabilirliği ölçmek için yeterli olabilmektedir (Çağıltay, 2011).

Kullanılabilirlik, kullanıcının performansının ne derecede kabul edilebilir olduğu ve kullanıcı memnuniyeti ile değerlendirilebilir. Sistem ve sistemi kullanan kişilerde, yapılan görevlerde veya sistem ortamında meydana gelebilecek herhangi bir değişiklik sistemin kullanılabilirliğini etkiler. Ayrıca kullanılan yazılımın ne kadar verimli ve kaliteli olduğu kullanılabilirliği etkileyen özellikler olarak değerlendirilebilir. Yani aslında bir yazılım ürünü için kullanılabilirlik kavramını kullanan kişilerin yazılımı ne derecede kaliteli ve kolay kullanılabilir gördüğü ile değerlendirilebilir (Bevan et vd., 1991).

Çin, Danimarka ve Hindistan ülkelerinden 24 adet kullanılabilirlik uzmanı ile yapılan bir çalışmada, uzman kişilerin kullanılabilirlik ile ilgili; ISO kullanılabilirlik tanımıyla uzmanlarca kabul gören kullanılabilirlik uygulamalarının tam olarak tutarlı olmadığı, uzmanların kullanılabilirliği çevre ve organizasyon sistemleri uyumundan ziyade kullanıcı düzeyinde dikkate aldığı ve kullanılabilirlik uzmanlarının kullanılabilirlik için bireysel kullanıma odaklanmak yerine bilişsel yapıları kullanmaya odaklandıkları sonuçlarına varılmıştır. Aynı zamanda 3 farklı milletten uzmanlar ile çalışılmasının bir sonucu olarak farklı milletlerin uzmanları için kullanılabilirlik kavramının benzer şekilde ifadelere sahip olduğu ve uygulama geliştiricileri ile kullanılabilirlik uzmanları arasındaki iletişimin sağlıklı olmamasının, kullanılabilirlik ile ilgili farklı yorumlanan yapıların belirlenmemesinin hatalı tasarımlar ve kullanıcı memnuniyetsizliğine yol açabileceği yorumları da yapılmıştır (Hertzum, 2012).

Kullanılabilirlik kavramını, yazılım yaşam döngüsü ile birlikte ele alan bir çalışmada kullanılabilirliğin yazılım yaşam döngüsü aşamalarına entegre edilmesinin projelerde daha yüksek başarı oranı sağlayabileceği ifade edilmiştir. Çalışmada, kapsam alanı Türkiye olarak belirlenmiş ve Türkiye’de genel olarak kullanılan yazılım yaşam döngülerini belirlemek, projelerde kullanılabilirlik kavramının nasıl ele alındığını belirlemek, yine uygulama geliştiricilerin kullanılabilirliği ne kadar dikkate aldıklarını ve ne kadar farkındalığa sahip olduklarını belirlemek amacıyla bir anket çalışması yapılmıştır. Anket içeriğine katılımcı bilgileri, kullanılabilirlik kavramının ne şekilde anlaşıldığı, kullanılabilirlik açısından yapılan değerlendirmelerin nasıl yapıldığı ve uzman uygulama geliştiricilerinin kullanılabilirlik algılarına yönelik veri toplamayı amaçlayan sorular yerleştirilmiştir. Çalışma sonucunda anket katılımcılarından toplanan veriler ışığında yazılım yaşam döngüsü aşamalarının hangi adımlarında kullanılabilirlik yöntemlerinin kullanıldığına dair bir tablo oluşturulmuştur. Tablo verilerine göre ise kullanılabilirlik ile ilgili yöntem ve testlerin genel olarak yazılım yaşam

döngüsünün son adımlarında uygulandığı sonucuna varılmıştır. Aynı zamanda çalışmada, uygulama yapılan organizasyonların genel anlamda kullanılabilirlik yöntemlerine hâkim olmadıkları, organizasyonların kullanılabilirliğin projeler için önem taşıdığı farkındalığına sahip oldukları ancak bu yöntem ve testleri uygulamanın ek zaman ve maliyet gerektirdiğini düşündükleri sonuçlarına da varılmıştır (Coşkan & Durdu, 2014).

Web sayfalarının tasarım problemleri üzerine yapılan bir çalışmada, web kullanıcılarının uygulamaları kullanırken görevleri daha doğru ve ihtiyaçları kapsamında kullanabilmeleri için bazı yöntemlerden bahsedilmiştir. Çalışmada insan bilgisayar etkileşimi kapsamında oluşturulan standartlardan bahsedilerek ISO ile oluşturulan standartların teknolojinin ilerlemesine bağlı olarak net bir şartname olmadığından fakat yazılım erişilebilirliği ve insan bilgisayar etkileşimi ile ilgili olan ISO 9241-11’ den bahsedilmiştir. Bu standardın, ürün kullanıcılarının ürünleri kullanım amaçlarına olan ulaşmaları ve memnuniyetleri ile ilgili olduğundan bahsedilmiştir (Hustak & Krejcar, 2016).

Kullanıcı deneyimi ile kullanılabilirlik kavramlarının İBE açısından incelendiği bir çalışmada kullanıcı deneyimi için memnuniyet durumunu değerlendirmek amacı ile bir model geliştirilmiştir. Geliştirilen bu model, İBE’ nin kullanılabilirlik üzerine bulunan katkılarını, iyi bir kullanıcı deneyimi için önemli faktörler ve bu iki durumun nasıl ilişkili olduğu sorularını cevaplamak amacı ile oluşturulmuştur. Bu model, kullanıcılar için iyi olarak nitelendirilebilecek veya tatmin edici olarak nitelendirilebilecek bir kullanıcı deneyimi sürecinin sadece kullanılan uygulamanın arayüzünün kullanılabilirliği ile ilgili olmadığını, aynı zamanda kullanım sırasında değerlendirilen kalite kavramının da kullanıcı deneyimi için önemli bir etken olduğunu önermektedir. Araştırma modelinde işlevsellik, öğrenilebilirlik, kullanıcı arayüzü estetiği ve uygunluk tanınabilirliği faktörleri önerilmiştir ve bu faktörlerin kullanılabilirlik üzerinde etkili olduğu, İBE’ nin memnuniyet ve kullanılabilirlik ile iç içe olduğu ve kullanılabilirliğin kullanışlılık, konfor ve güvene bağlı olarak memnuniyeti etkilediği sonucuna varılmıştır. Çalışmada anket çalışmaları kapsamında dört temel sonuca ulaşılmıştır. Bunlardan birincisi İBE kapsamında kalite kavramı için daha açıklayıcı yapıların olması gerektiğidir ve model çalışması bunun için gerçekleştirilmiştir. İkincisi ise model kapsamında seçilen faktörlerin kullanılabilirlik için doğru etki faktörleri olduğu, en önemli faktörün ise çalıştırılabilirlik olduğudur ve ürün kalitesi açısından kullanılabilirlik önemli bir noktadır. Üçüncüsü, tasarımcı kişilerin İBE kapsamında arayüz içeriklerini kullanıcı memnuniyetinin ana sebeplerini baz alarak oluşturmaları gerektiğidir. Dördüncüsü ise araştırma modeli kapsamında bir web uygulamasının sahip olduğu arayüzün iyi ve kötü yanlarının değerlendirilerek bir başka web uygulaması ile kıyaslanmasına olanak sağlamasıdır (Kivijärvi & Pärnänen, 2023).

İlk kez 1980’li yılların başında kullanılmaya başlanan ve uzmanlar tarafından genel çerçevede insan bilgisayar etkileşimi için önemli bir yeri olan “Kullanılabilirlik Mühendisliği” kavramı, uygulamalarda veya projelerde kullanılabilirlik ile ilgili planlamaların yapılması, gerçekleştirilmesi ve doğrulanması, kullanılabilirlik ile ilgili çalışacak kişilerin pozisyonlarının belirlenmesi, daha kullanılabilir sistemlerin geliştirilmesi için farklı araç ve yöntemler geliştirip kullanma gibi sorumlulukları içerir. Bu belirtilen adımlar öncelerde yazılım mühendisi veya geliştirici kişilerin yapması gereken alt görevler olarak ifade edilse bile günümüzde artık farklı bir mühendislik alanına ait sorumluluklar olarak ifade edilmektedir. Kullanılabilirlik mühendisleri, proje geliştirme ve bilişim sistemlerinin aşamalarının birçoğunda aktif olarak rol oynamaktadırlar. Kullanılabilirlik uygulamaları, ürünler, geliştiriciler ve kullanıcılar için bu denli önemli bir nokta olduğundan dolayı kullanılabilirliği ölçmek amacı ile kullanıcı testleri uygulanmaktadır. Kullanılabilirlik testleri için çeşitlilik, iki grupta sınıflandırılmaktadır. Bunlar tür ve yaklaşımdır. Test türleri, süreç içi ve süreç sonu olmak üzere iki farklı seçeneğe sahiptir. Süreç içi test türleri, uygulamaların son kullanıcılara sunulmadan, geliştirilme aşamalarının başından sonuna kadar gerekli sıklıklar ile yapılan ve uygulamaların son kullanıcıya gidene kadar yani piyasaya sürülene kadar oluşabilecek sorunları veya yanlışlıkları olabildiğince engellemek ve en az seviyeye indirmek amacı ile gerçekleştirilmektedirler. Bu testlerin uygulanmasının zaman, gereklilik, imkân ve olanakları göz önüne alınarak olabildiğince fazla sıklıkla yapılması uygulama için avantajlı ve olumlu olacaktır. Süreç sonu testleri ise uygulama veya ürünün geliştirilme süreci bittikten sonra yani uygulama artık son kullanıcıların kullanabileceği bir hal aldıktan sonra piyasaya sürülecek ürün veya uygulamanın geçerlilik durumunu değerlendirmek ve doğrulamak için yapılmaktadır. Süreç içi testlerinin doğru bir şekilde gerçekleştirilmemesi veya yeterli sıklıkta yapılmaması uygulama veya ürünün geliştirilme aşamalarında ortaya çıkabilecek problemlerin, sürecin son aşamasında yani süreç sonu testlerinde ortaya çıkması anlamına gelecektir. Böyle bir durumun gerçekleşmesi ise ürün veya uygulamanın piyasaya sürülme zamanını geciktirebilir. Bunun sonucunda üretici veya geliştiricilerin hem maddi açıdan hemde zaman ve sarf edilen çaba gibi yönlerin boşuna harcanması ile manevi açıdan zarara uğrayabilmeleri, kullanıcılar açısından ise ürün veya uygulamaya karşı oluşabilecek bir memnun olmama durumunun meydana gelmesi söz konusu olabilmektedir (Çağıltay, 2011).

Kullanıcı testleri, uygulamaları ve ürünleri değerlendirmek amacıyla kullanılmaktadırlar. Bu testlerin kullanılması üreticiler tarafından ürünlerinin geliştirilme ve kullanılma aşamaları için uygulamanın tasarım problemlerinin son kullanıcıya sunulmadan önce tespit edilmesini ve böylece ürünün geliştirilme aşamasında düzenleme ve değişikliklerin yapılması ile son kullanıcıya daha amacına uygun ve kolay kullanılacak bir ürün sunulmasını

sağlayabilmektedir. Bu testler geliştirilirken ve uygulanırken; uygulamanın amacına uygun bir şekilde araştırma ve test sorularının geliştirilmesi, son kullanıcı tarafında görüşlerin alınması, test uygulanırken gerçek bir çalışma ortamının sağlanması, test aşamalarında son kullanıcıların gözlemlenmesi, uygulayıcı tarafından test kullanıcıları ile görüşmelerin gerçekleştirilmesi, hem nicelik hem niteliksel olarak verilerin toplanması, kullanıcılar tarafından toplanan geri dönüşlerin uzmanlarca değerlendirilip uygulanması gerekmektedir. Kullanılabilirlik testleri, uygulamalar henüz tamamlanmadan yani tasarım adımlarında uygulanmaktadırlar. Yapılan bir çalışmada bu testler ürün yaşam döngüsü için üç adımda oluşturulmuştur. Bunlar, keşif testleri, değerlendirme testleri ve doğrulama testleridir. Keşif testlerinin temel amacı uygulama veya ürünün için ön tasarımlarının incelenmesidir. Arayüz tasarımı yapan kişilerin kullanıcılar için uygulamanın işlev ve arayüzlerinin tam olarak ne şekilde düşünüldüğünün kavranmasını sağlayabilmektedir. Değerlendirme testlerinde amaç, uygulama veya ürünün temel çalışma gibi fonksiyonel işlemleri ve kullanılabilirliğini incelemektir ve bu testler keşif testi sonucunda elde edilen verilerin daha detaylandırılmasını sağlamaktadır. Bu sayede uygulamanın eksiklikleri ve gerçekleştirilebilirliği gözlenebilmektedir. Doğrulama testi ise ürün veya uygulamanın son kullanıcılara sunulmadan önce performans standartları gibi standartları karşılayıp karşılamadığını değerlendirmeyi ve eğer karşılamıyor ise neden karşılamadığını çözümlemeyi amaçlamaktadır. Burada bahsedilen standartlar ise geliştirilme planlamasının yapıldığı aşamalarda ürün veya uygulama tarafından beklenen çıktılar ve ana amaç ile ilgilidir (Rubin vb., 2008).

Kullanılabilirlik test yaklaşımları, dört farklı başlık altında kendi içinde sınıflandırılmaktadır. Bunlar “Deneyisel Yaklaşım”, “Sezgiseller”, “Tasarım Rehberleri” ve “Model Temelli Yaklaşım” şeklindedir. Tasarım rehberleri yaklaşımı, arayüz özellikleri ve tasarımı ile ilgilenen parçadır. Tasarım rehberleri kavramı ürün veya uygulama sahibi kişi veya şirketlerce geliştirilerek ürün ve uygulamaların kendileri ile ve tasarımların genel çerçevesi ile uyum sağlamak amacıyla kullanılmaktadır. Fakat tasarım rehberlerinin sadece tasarımcı kişiler ile değil aynı zamanda kullanılabilirlik uzmanları ile birlikte hazırlanması daha doğru olacaktır. Sezgiseller olarak ifade edilen yaklaşım bir diğer adıyla uzman değerlendirmeleridir. Bu yaklaşım, arayüzlerin uzman ve yetkili kişilerce gözlenip değerlendirilmesi olarak ifade edilebilir. Bu değerlendirme yöntemi ürün veya uygulamalar için kullanılabilirlik etkenini iyileştirmeye yöneliktir. Sezgisel testlerini tek bir uzmanın yapmasından ziyade görüş ve öneri çeşitliliğini sağlamak amacı ile ve ortaya çıkabilecek çelişkileri önlemek amacı ile çok sayıda uzmanın gerçekleştirmesi tercih edilmelidir. Uzman kişiler belirlenen ölçüm bileşenleri kapsamında gözlemlene ve değerlendirme işlemlerini yaparak bir sonuç raporu oluşturur. Sezgisel kılavuzlar arayüzlerin gözlemlenmesi ve değerlendirilmesi adımlarının uygulanma

kısmı için çok net ifadeler barındırmamaktadır. Yani kılavuzlara göre değerlendirmelerde test kapsamında adımların ne şekilde yürütüleceği kılavuzu uygulayan uzman kişilerce farklı farklı yorumlanarak uygulanabilir. Bu yüzden sezgisel değerlendirmeler sonucunda uzmanlarca oluşturulan raporlar birbirinden ayrışabilmektedir. Günümüz için geçerliliğini koruyan sezgisellerden biri Nielsen'in 10 kullanılabilirlik sezgiselidir (Çağıltay, 2011). Nielsen'in 10 kullanılabilirlik sezgiseli, kullanılabilirlik sorunlarının deneysel bir şekilde incelenmesi için 9 adet sezgisel içeren bir yapı olarak 1990'lı yıllarda Nielsen ve Molich tarafından oluşturulmuştur. Daha sonra 1994 yılında Nielsen bu yapıyı 10 adet olacak şekilde tamamlamıştır. Bu 10 sezgisel ise şu şekildedir;

1. Sistem durumunun görünürlüğü
2. Sistem ile gerçek dünya arasındaki uyum
3. Kullanıcı kontrolü ve özgürlüğü
4. Tutarlılık ve standartlar
5. Hata önleme
6. Hatırlamak yerine tanıma
7. Esneklik ve verimli kullanım
8. Estetik ve minimalist tasarım
9. Kolay ayırt edilebilen iyi yapılandırılmış özellikler
10. Yardım ve Dokümantasyon

10 sezgiselden oluşan bu model uzmanların arayüzleri kullanılabilirlik bileşeni açısından değerlendirebilmeleri ve gözlemlmeleri için bir araç, yöntem ve rehber niteliğinde olmasına rağmen yine de tam anlamıyla tek başına yeterli sayılmayabilir ve bundan dolayı ek olarak değerlendirme yöntemlerine de başvurulabilir. Bu yöntem aynı zamanda öğrenme ortamları çalışmaları, sanal ortam çalışmaları ve web siteleri gibi birçok farklı çalışma ortamı için kullanılmış olup hala kullanılmaktadır. Fakat yapılan bazı çalışmalara göre bu sezgiseller bütününün günümüzün modern sistemleri için uyumlu olup olmadığı hakkında net kararlar bulunmadığı görülmektedir (Gonzalez-Holland vd., 2017).

Nielsen sezgisellerinin temel amaçlarından kısaca bahsedilecek olursa, sistem durumunun görünürlüğü, sistemin çalışma durumunun yeterli zaman aralıkları ile geri bildirimler şeklinde kullanıcı ile paylaşılması olarak açıklanabilmektedir. Sistem ile gerçek dünya arasındaki uyum, sistemin kullanıcının anlayacağı dilden konuşması anlamına gelmektedir yani sistem bazlı terimlerin kullanılması yerine kullanıcı dilinin tercih edilmesidir. Kullanıcı kontrolü ve özgürlüğü, kullanıcıların uygulama üzerinde çalışmaları esnasında yapmak istedikleri şeyin dışına çıkarak hata yapma durumlarına karşı, hataların içerisinde

kaybolmadan kısa bir şekilde geri gidebilecekleri alanların olması anlamına gelmektedir. Tutarlılık ve standartlar, uygulamada birbirleri ile çelişen fonksiyonel eylemlerin olmaması durumunu ifade etmektedir. Hata önleme sezgiseli, ilk temel amaç olarak kullanıcıların yaptıkları hataları doğru bir şekilde anlamaları ve düzeltmeleri üzerine çalışılması yerine daha iyi bir tasarım ve fonksiyonellik ile kullanıcının hata yapmasını önlemek üzerine gerçekleştirilen çalışmalardır. Hatırlamak yerine tanıma sezgiseli ise kullanıcının uygulama içinde gezindiği veya işlerini yürüttüğü esnada ilerlerken nereden geldiğini unutmaması veya bu konuda kafasının karışmamasının sağlanmasıdır yani sistem kullanımı ile ilgili yönergeler açık ve net olmalıdır. Esneklik ve verimli kullanım, uygulamanın kullanımı esnasında kullanıcıların kendi yetenek ve anlama durumları kapsamında daha kolay ve hızlı kullanabilmeleri açısından uygulama bileşenlerinin kullanım şekillerini düzenleyip değiştirebilme yetkisinin verilmesi anlamına gelmektedir. Böylece kullanıcı uygulama kapsamında daha sıklıkla kullandığı bileşen veya durumları kendisi için daha görünür hale getirebilir ve bu da kullanıcı memnuniyetini arttırıcı yönde bir etki sağlayabilecektir. Estetik ve minimalist tasarım, arayüzde karmaşıklığı ve buna bağlı olarak kullanıcılarda meydana gelebilecek kafa karışıklığı durumlarını engellemek amacı ile tasarımı ve uygulama kapsamında kullanıcılara sunulan ifadeleri olabildiğince sade tutulması gerektiği anlamına gelmektedir. Kolay ayırt edilebilen iyi yapılandırılmış olması ise kullanıcıların süreç içinde karşılaştıkları hatalar için kullanıcıya doğru dönütler verilerek hata mesajlarının açık bir şekilde ifade edilmesi üzerine çalışılması anlamına gelmektedir. Yardım ve dokümantasyon ise kullanıcıların sistemi tanımaları için gerekli olan maddedir (Nielsen, 2006).

Kullanılabilirlik test yöntemlerinden bir diğeri olarak deneysel yaklaşım ise uygulamaların geliştirilme aşamalarında tamamlanan arayüzün doğru ve gerçek kullanıcı hedef kitlesine sunulurken, deney kullanıcılarına uygulamaya ait temel ve önemli işlevleri barındıran bir görev listesi verilerek ve uygulama için doğru bir çalışma ortamı oluşturularak, uygulamanın deney kullanıcılarına görev listesi ve çalışma ortamı içerisinde kullandırılması ile gerçekleştirilmektedir. Uygulamada gerçekleştirilen kullanılabilirlik testleri bu kapsamda belirlenen kullanılabilirlik ölçütlerine göre değerlendirilerek ölçüm sonuçları ve kullanıcılardan toplanan geri dönüşler uygulamanın tasarımı için değerlendirilerek tasarım ve fonksiyonellik üzerinde tekrardan düzenlemeler ve değişiklikler yapılarak tasarım iyileştirilebilmektedir. Deney kullanıcıları, uygulamada kendilerine verilen görevleri yerine getirirken yani uygulamayı deneyimlerken, bu esnada kişinin verdiği fiziksel ve duygusal tepkiler gözlemlenerek veya kullanıcıdan deneyimleri esnasında sesli düşünerek duygu ve ifade durumunu belirtmesi istenerek test esnasında veriler toplanabilir. Test, gerçek kullanıcılar ile gerçek ortamlarda ve gerçek arayüzler üzerinde uygulandığından dolayı diğer yöntemlere

kıyasla kullanılabilirlik ile ilgili doğru sonuçların ve verilerin toplanabileceği bir test çeşidi olarak ele alınmaktadır. Fakat yine diğer yöntemlere kıyasla da daha zahmetli ve daha maliyetlidir (Çağıltay, 2011).

Bu kullanılabilirlik testleri için gerekli olan deney kullanıcı sayısı, bazı alan uzmanlarının görüşleri için farklılık göstermektedir. Nielsen “Usability Engineering” isimli kitabında doğru ve amacına uygun yapılandırılmış testler ile gerçekleştirilecek bir kullanıcı testi için beş deney kullanıcısının uygulamadaki arayüz sorunlarının yaklaşık %75’ ini gösterebileceğini ifade etmektedir (Nielsen, 1993).

Bu alanda araştırma yapan bir başka araştırmacı grubu ise beş deney kullanıcısı ile yapılacak kullanıcı testlerinin arayüz sorunlarını belirlemede yetersiz olacağı ve bu sayının en minimum seviyede 12 kişi olması gerektiğini savunmuşlardır (Albert & Tullis, 2008).

Deney kullanıcı sayısı üzerine net bir kanı olmamasına karşın bu testlerde özellikle deney kullanıcısı olarak seçilen kişilerin uygulama amacına uygun ve doğru seçilmeleri, testin gerçekleştirileceği ortam, testin nasıl gerçekleştirileceği, kullanıcılara nasıl sorular yöneltilip, nasıl gözlem yapılacağı ve aynı zamanda test esnasında kullanıcılara verilen görevlerin gerçekten uygulama için önemli noktalara değinebilecek ve kullanıcının uygulamayı doğru bir şekilde deneyimleyip değerlendirmesini sağlayabilecek şekilde tasarlanması önemli noktalar. Test gerçekleştirildikten sonra kullanıcılar ile yapılan görüşmelerde anketler, değerlendirme ve görüş almak amacıyla hazırlanan sorular kullanılabilir. Bu çalışmalar tasarlanırken ve yürütülürken süreçlerin tamamı için süreç içinde dikkat edilmesi gereken bazı hususlar vardır. Örneğin ilk önce uygulamanın kimlerin hangi ihtiyaçlarını karşılamak amacıyla geliştirileceği doğru yöntem ve teknikler ile belirlenmelidir. Uygulamayı kullanacak olan hedef insan topluluğunun uygulama için gerekli gördüğü beklenti ve isteklerinin ne olduğu yine doğru bir şekilde belirlenip geliştirici kişilere de doğru bir şekilde ifade edilmelidir. Yine testlerin doğru deney kişileri ile gerçekleştirilmesi amacı ve uygulama gereksinim ile hedeflerinin belirlenmesi için kullanıcı profilleri çıkarılmalıdır. Bu birçok açıdan uygulama gelişim süreci için faydalı olacaktır. Bir diğer kullanılabilirlik test yöntemi ise model temelli yaklaşımdır. Bu yöntem diğer yöntemlere istinaden hedef kullanıcıların hem fiziksel olarak hemde bilişsel olarak ifade etmeyi amaçlamaktadır (Çağıltay, 2011).

MATERYAL VE YÖNTEM

Literatür çalışmaları incelendiğinde, yeşil yazılım ve insan bilgisayar etkileşiminde kullanılabilirlik bileşenleri için ortak bir çalışmanın yapılmasının, yazılım uygulamaları geliştiricileri açısından uygulamalarının geliştirilme aşamalarına veya yaşam döngülerine kullanıcı testlerini dahil etme ve etmeme durumlarının, uygulamalarının yaptığı karbon salınımı ve enerji tüketimi değerlerini nasıl etkileyeceği üzerine bir fikir verebileceği ve geliştiricileri bu yönde bilinçlendirebileceği düşünülmektedir. Bu bölümde öncelikle gerçekleştirilen çalışmasının yürütülmesi sürecinde izlenen yol ve yöntemden bahsedilecektir. Bölümün bir sonraki adımlarında ise bu sürecin adımları detaylandırılacaktır.

Yöntem

Tez kapsamında çalışma birkaç adımda gerçekleştirilmiştir. Bu adımlardan bu bölümde detaylıca bahsedilecektir. Öncelikle çalışmada genel anlamda, sıradan bir şekilde kullanıcı testleri ve uygulamaları dahil edilmeden geliştirilmiş bir hastane randevu alma uygulaması ile aynı uygulamanın kullanıcı testleri dahil edilmesi ile geliştirilmiş versiyonunun uygulama kod karbon salınımı ve uygulama enerji tüketimi değerlerinin kıyaslanması amaçlanmıştır. Bu kıyas ile yazılım projelerinin geliştirilme aşamalarına kullanılabilirlik bileşeninin dahil edilmesinin yeşil yazılımı yani kod karbon salınımını ve enerji tüketimini nasıl etkileyebileceği üzerine deneysel veri sonuçları ortaya koymak amaçlanmaktadır.

İlk aşamada Django web çatısı, Python programlama dili ve Visual Studio Code platformu kullanılarak hastane randevu alma uygulaması geliştirilmiştir. Geliştirilen bu uygulama kullanıcıların uygulama üzerinde hastane randevusu oluşturmalarını amaçlayan bir sistemdir. Uygulama olarak hastane randevu sisteminin seçilmesinin sebebi, hem genel anlamda her yaş ve seviyeden insanın karşılaşılabileceği ve aşina olduğu bir alan olmasını sağlamak hemde kullanılabilirlik testlerinde bulunan görev listesi için daha fazla temel görev çeşitliliğini sağlayabilmektir. Hastane randevu alma uygulaması temelde 10 ana alandan oluşmaktadır. Bu uygulama geliştirilirken ilk aşamada herhangi bir kullanılabilirlik çalışması veya testi yapılmamış, kullanıcı görüşlerine başvurulmamıştır. Yalnızca türev uygulamalar incelenerek uygulama içeriği için fikir edinilmiştir.

Geliştirilme aşaması tamamlandıktan sonra, uygulamanın çalışması esnasında yapacağı kod karbon salınımını ve uygulama enerji tüketimini hesaplamak amacıyla uygulamaya CodeCarbon kütüphanesi entegre edilmiştir.

Bir sonraki aşamada web uygulamasında kullanıcıların gerçekleştirmesi amacıyla 10 adımdan oluşan ve uygulamanın temel işlevlerini kullanıcının test etmesini sağlamayı hedefleyen bir görev listesi oluşturulmuştur. Kullanılabilirlik ve test ölçümleri kapsamında 12 adet deney kullanıcısı seçilerek kişilere, hastane randevu uygulaması ve uygulama için oluşturulan görev listesi verilmiştir. Kullanıcılardan kendilerine verilen görev listesindeki adımları teker teker uygulama üzerinde yerine getirmeleri ve tamamlamaları istenmiştir. Kullanıcılar uygulamayı deneyimlerken de çalışma esnasında arka planda hem karbon salınımı hemde enerji tüketimine dair ölçümler yapılmıştır. Deney kullanıcıları uygulamadaki görevleri tamamladıktan sonra ise kullanıcılar ile birebir görüşmeler gerçekleştirilmiştir. Gerçekleştirilen görüşmelerde kullanıcıların uygulama hakkındaki hem tasarımsal, arayüz ile ilgili hem de fonksiyonellik ile ilgili memnuniyet, görüş ve önerilerini ölçmek amacıyla uygulamanın 10 ayrı sayfası için toplamda 60 adet sorudan oluşan bir test listesi kullanılmıştır. Test soruları kullanıcı görüşmeleri esnasında kullanıcılara yöneltilerek uygulama hakkındaki görüş ve öneriler toplanmıştır. Daha sonra değerlendirme raporlarının oluşturulması amacıyla, görüşmeler ses kaydına alınmıştır.

İlk kullanıcı testi gerçekleştirildikten sonra CodeCarbon kütüphanesi aracılığıyla çıktı olarak alınan veriler incelenerek düzenlenmiştir. Ayrıca kullanıcılar ile yapılan görüşmeler kapsamında toplanan görüş verilerinin incelenmesi ve düzenlenmesi amacıyla, alınan ses kayıtları Word dosyalarına dökülerek kullanıcı görüşleri, sayfalar bazında 10 grup altında sınıflandırılmıştır. Bir sonraki adımda, toplanan ve değerlendirilen kullanıcı görüşleri ilk geliştirilen uygulama üzerinde uygulanarak hastane randevu alma uygulamasının ikinci versiyonu yani kullanılabilirlik testinin ve kullanıcı görüşlerinin dahil edildiği hali geliştirilmiştir.

Uygulamanın bu hali deney kullanıcılarına tekrar sunularak görev listesi verilmiştir ve kullanıcıların tekrar görev adımlarını sırayla uygulama üzerinde yerine getirmeleri istenmiştir. Kullanıcılar görevleri ikinci versiyon üzerinde yerine getirirken arka planda yine uygulamanın yaptığı kod karbon salınımı ve kod enerji tüketimine dair değerler hesaplanarak uygulama sonunda kullanıcılar ile tekrar birebir görüşmeler gerçekleştirilmiş, bu versiyon üzerinde görüş ve öneriler toplanmıştır.

Son olarak uygulamanın hem ilk versiyonu hem de ikinci versiyonunun kullanımları sonucunda uygulamanın kullanımı esnasında CodeCarbon kütüphanesi aracılığıyla elde

edilen, kod karbon salınımı ve enerji tüketimi değerleri kıyaslanmıştır. Bu şekilde yazılım uygulamalarının geliştirilme yaşam döngülerine İBE kapsamında yapılan kullanıcı testlerinin dahil edilip edilmeme durumlarının yeşil yazılıma etkisinin ölçülmesi ve yazılım geliştiricilerine, şirketlere ve bu işi yapan kurumlara da kullanılabilirlik çalışmalarının yapılmasının önem ve sonuçları, yeşil dönüşüm ve yeşil yazılım açısından gösterilerek bu alan için bir teşvik ve örnek olunması sağlanabilecektir.

Visual Studio Code

Visual Studio Code çalışma ortamı, yaygın olarak kullanılan işletim sistemlerinde çalışabilen çok sayıda uzantı ve yazılım dili desteği sunan ücretsiz kaynak kodu editörüdür. Tez uygulamaları kapsamında kaynak kod editörü ve uygulama geliştirme platformu olarak Visual Studio Code platformu kullanılmıştır (Taniguchi & Nakasho, 2021).

Django Web Çatısı

Django, dinamik olarak kullanıcı arayüzüne sahip olan Python programlama dili ile çalışılan uygulamalar için tercih edilen bir web çatısıdır. Dinamik backend arayüzlerine sahip olmasından dolayı geliştiriciler için büyük oranda zamandan tasarruf ve kolaylık sağlamaktadır, aynı zamanda da ücretsiz bir yazılımdır. Tez uygulamaları kapsamında da hastane randevu alma uygulamasının geliştirilmesi için Django web çatısı kullanılmıştır (Ortiz, 2012).

Python Programlama Dili

Python, nesneye dayalı programlama yapılabilen yüksek seviyeli programlama dilidir. Daha basit syntax'ı ile birçok temel platform ve işletim sistemlerinde çalışabilmesi ile yazılım geliştirme alanında, web programlama, sistem programlama, ağ programlama vb. geniş ölçekte kullanılabilmesi ile günümüzün popüler ve sıkça kullanılan programlama dilleri arasında sayılmaktadır (Misra & Cafer, 2010). Tez uygulamaları kapsamında da hastane randevu alma uygulamasının geliştirilme aşamasında Python programlama dili kullanılmıştır.

Uygulamanın Geliştirilmesi

Tez çalışmasında karbon salınımı ve enerji tüketimi ölçümlerinin yapılması amacıyla, kullanılabilirlik deney ve testlerinin üzerinde gerçekleştirilmesi işlemleri için deney kullanıcılarının sisteme kayıt olduktan sonra görev adımlarını izleyerek sistem içinde hastane randevusu oluşturabilecekleri bir hastane randevu alma uygulaması geliştirilmiştir. Geliştirilme aşamalarının hiçbirinde kullanılabilirlik çalışması yapılmamış, herhangi bir

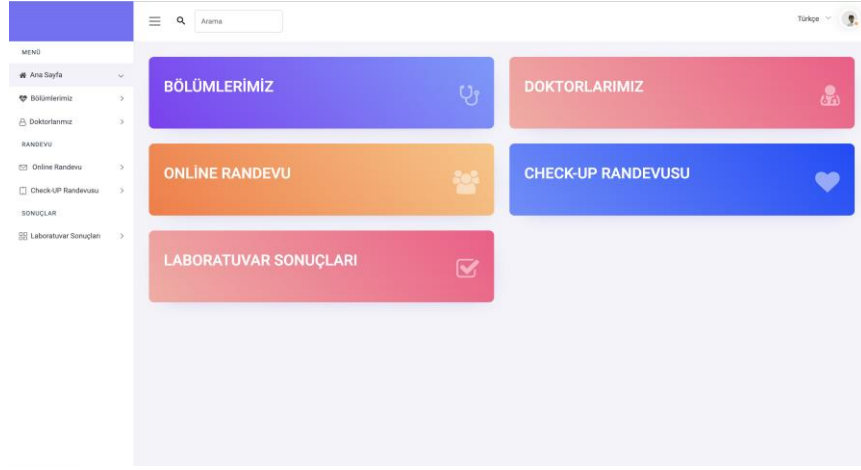
kullanıcı görüşüne başvurulmamıştır. Uygulama arayüzü için kullanılan şablon ise hem renk hemde şemaları açısından olabildiğince sade ve kafa karıştırıcılıktan uzak bir şablon olarak seçilmeye özen gösterilmiştir. Yalnızca hastane randevu alma uygulama benzeri bazı uygulamalar randevu sisteminin içeriği ve adımlarına örnek olması açısından incelenmiştir. Uygulama temel olarak 10 ana sayfadan oluşmaktadır.

Öncelikle diğer uygulamalarda yapılan gözlemlerde, uygulamaya kullanıcının kayıt olması ve giriş yapması gereken alanlar olduğu görülmüştür. Bundan dolayı hastane randevu web uygulamasına kullanıcıların kayıt olabilecekleri ve daha sonrasında da kayıt bilgileri ile istedikleri zaman giriş yapabilecekleri bir giriş alanı oluşturulmuştur.

The image displays two user interface forms side-by-side. The left form, titled 'ÜYE OL' (Sign Up), includes input fields for 'AD' (First Name), 'SOYAD' (Last Name), 'KULLANICI ADI' (Username), 'E-POSTA' (Email), 'ŞİFRE' (Password), and 'TEKRAR ŞİFRE' (Repeat Password). It features a blue 'ÜYE OL' button and a link 'Bir Hesaba Sahipseniz giriş yapın' (If you have an account, log in). The right form, titled 'GİRİŞ YAP' (Log In), includes input fields for 'KULLANICI ADI' (Username) and 'ŞİFRE' (Password), a blue 'GİRİŞ' button, and a link 'Bir Hesaba Sahip Değilseniz üye olun' (If you don't have an account, sign up).

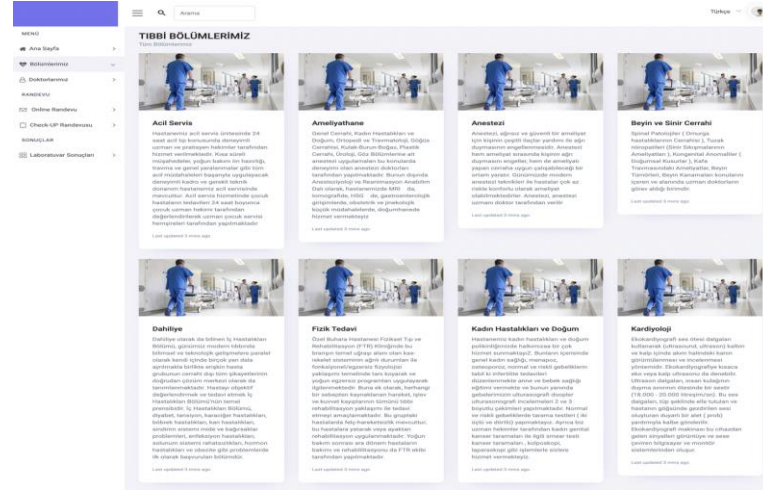
Şekil 3. Üye olma ve giriş yapma sayfaları

Kullanıcıların web uygulamasına kayıt olduktan ve giriş yaptıktan sonra karşılaştıkları sayfa ise uygulamanın ana sayfasıdır. Ana sayfa temel olarak kullanıcıların uygulamanın içerisinde bulunan genel sayfaları ve fonksiyonları basit bir şekilde görebilecekleri ve istedikleri işlem ne ise o işleme yönlendirilecekleri bir alan olarak düşünülüp tasarlanmıştır. Bu sayfa içerisinde kullanılan bileşenlerin yine genel şablon ile uyumlu olarak karmaşık olmayacak, temel ve basit bir şekilde ifade edilmesine dikkat edilmiştir.



Şekil 4. Ana sayfa

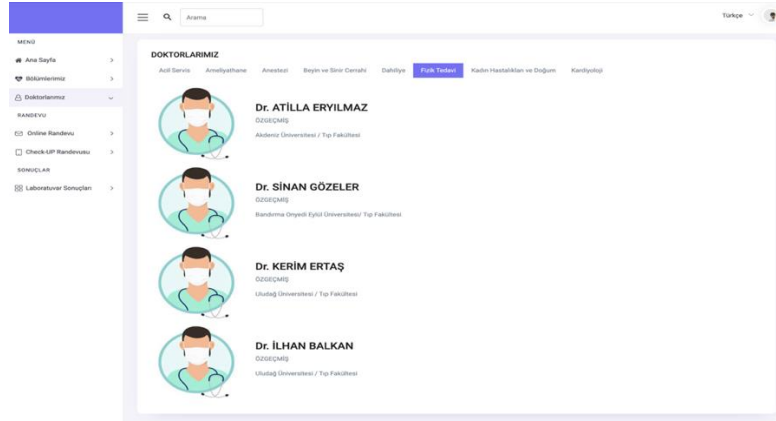
Ana sayfayı ziyaret eden kullanıcılar uygulamanın diğer bileşenleri olan bölümlerimiz, doktorlarımız, online randevu, check-up randevu, laboratuvar sonuçları bağlantıları ile karşılaşır. Kullanıcılar bu bağlantılar ile ilgili sayfalara geçiş yapabilmektedirler. Bölümlerimiz sayfasında randevunun alınacağı hastanede bulunan bölümlerden ve bu bölümlerin içeriklerinden bahsedilmektedir. Bu sayfanın uygulama içerisinde kullanıcıların randevu almak istedikleri bölümün uygulama içerisinde bulunup bulunmadığını kontrol etmeleri ve uygulama üzerinde gidecekleri bölüm hakkında fikir edinip karar vermelerine yardımcı olmak amacıyla uygulamaya eklenmiştir.



Şekil 5. Bölümler sayfası

Bir sonraki sayfa olan doktorlarımız sayfasında ise hastane randevu alma uygulamasında bulunan tüm bölümlere ait doktorların listesi sıralanmaktadır. İlgili sayfada gösterilen doktorlar, gerçekte var olan kişileri temsil etmemektedir. Sayfadaki bilgiler tamamen rastgele oluşturulmuştur. İlgili sayfada bölümlere ait bir filtre bulunmaktadır ve kullanıcılar bu filtreden bölümü seçerek yalnızca o bölüme ait doktorların listesini

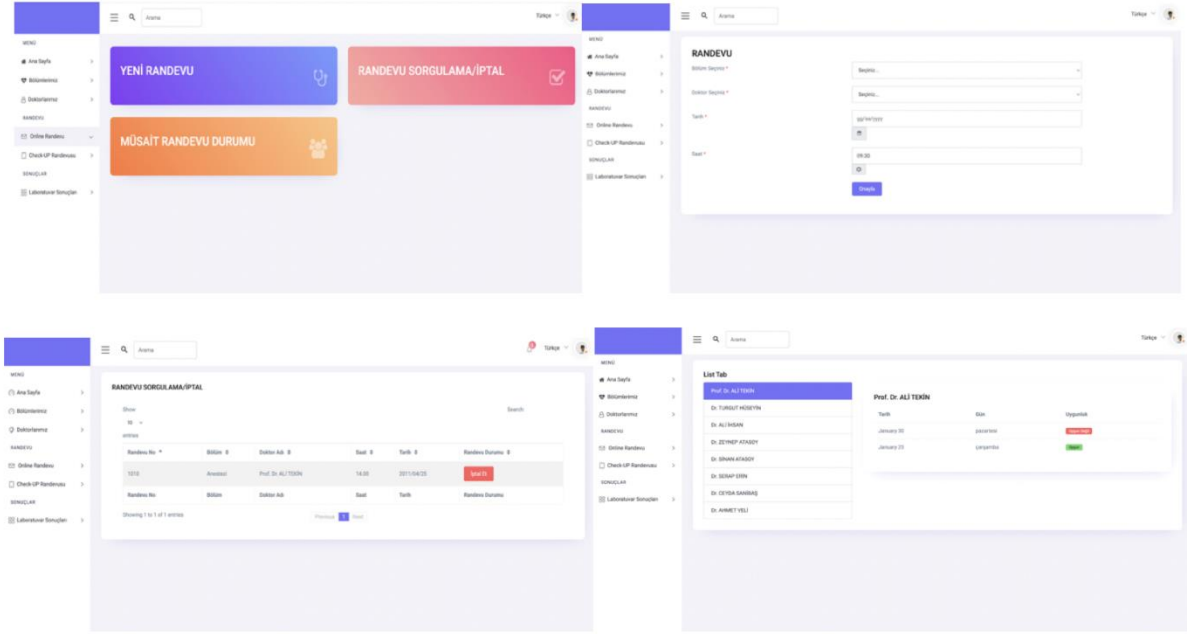
görebilmektedirler. Yani bu sayfa kullanıcıların, hangi doktorların hastalara hizmet verdiklerini görebilecekleri, şekilde tasarlanmıştır.



Şekil 6. Doktorlar sayfası

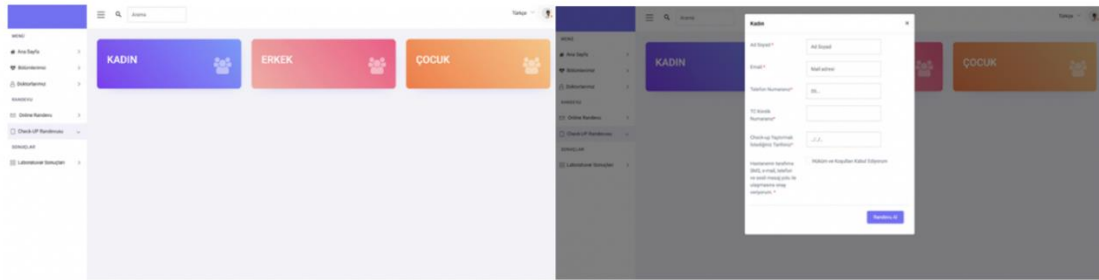
Randevu alınacak bölüm ve doktor kararı kullanıcı tarafından verildikten sonra kullanıcılar uygulamanın ana sayfasında bulunan bir diğer bileşen olan online randevu alanına tıklayarak online randevu oluşturabileceği alana ulaşabilmektedirler. Ana sayfada online randevu bağlantısına tıklayınca, kullanıcı online randevu işlemlerinin bulunduğu başka bir sayfaya yönlendirilecektir ve bu sayfada yeni randevu, randevu sorgulama/iptal ve müsait randevu durumu seçenekleri bulunmaktadır.

Bu alanda bulunan müsait randevu durumu sayfası, uygulamada bulunan doktorların hangi gün ve saatlerde randevu durumlarının uygun olup olmadığını kullanıcıların görmelerini sağlamaktadır. Böylece kullanıcılar, randevularını oluşturacakları gün ve saati belirleyebileceklerdir. Yeni randevu sayfası, online randevu bilgilerini sisteme girerek randevu kayıtlarını oluşturmalarını sağlayacaktır. Randevu sorgu/iptal sayfası ise kullanıcıların uygulamada oluşturdukları randevularının tamamını bir liste halinde görmelerini sağlayan bir alandan oluşmaktadır. Kullanıcılar oluşturdukları randevunun sisteme doğru olarak işlenip işlenmediğini bu sayfadan kontrol edebilirler.



Şekil 7. Online randevu oluşturma sayfaları

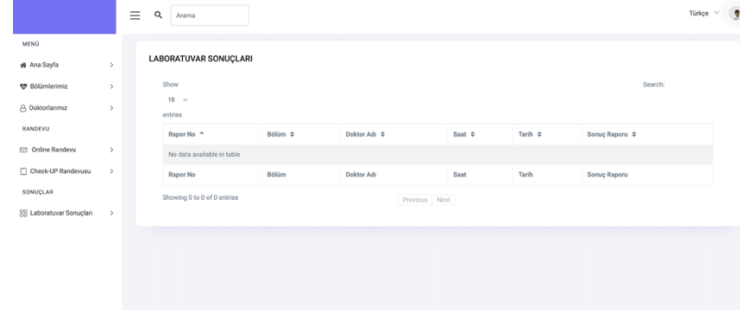
Uygulamasında hem kullanıcıların sistem üzerinde yapmaları gereken görevleri çeşitlendirmek amacıyla hem de benzer web uygulamalarının içerisinde bulunan bir alan olduğundan dolayı, kullanıcıların hastaneden online olarak check-up randevusu da oluşturabilecekleri bir sayfa uygulamaya dahil edilmiştir. Dahil edilen bu alan aracılığıyla kullanıcılar uygulamada gerekli bilgi alanlarını doldurarak hem kendileri hem de yakınları için uygulamada check-up randevu talebi oluşturabileceklerdir. Bu sayfada kullanıcıları ilk olarak randevunun alınacağı cinsiyet kategorilerini belirlemeleri gereken alan karşılamaktadır. Kullanıcı randevu talebinde bulunacağı kişinin cinsiyetinin bulunduğu model alanı üzerine tıkladığında ise check-up randevu bilgilerinin girileceği açılır pencere açılmaktadır. Bu pencerede içerik olarak kişinin Ad-Soyad, telefon numarası, mail adresi, check-up yapılması talep edilen tarih alanları bulunmaktadır. Kullanıcılar bu alanları doldurarak check-up randevu taleplerini oluşturabilmektedirler.



Şekil 8. Check-Up randevu sayfası

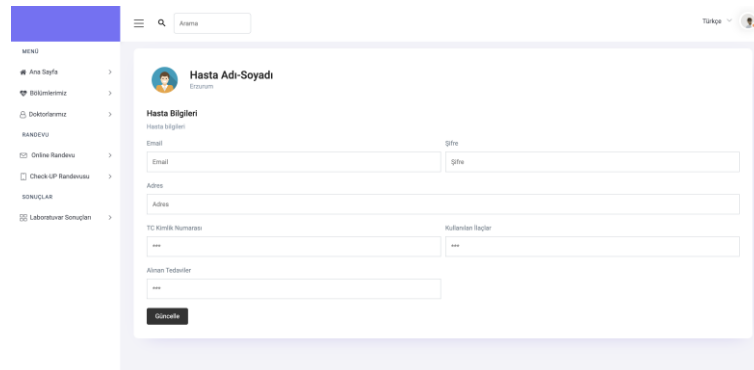
Uygulama içerisinde bulunan bir diğer bileşen ise yine görev çeşitliliğini sağlamak amacıyla laboratuvar sonuçları sayfasıdır. Bu sayfada içerik olarak kullanıcıların hastane

randevuları esnasında doktorlar tarafından talep edilen tahlil vb. sonuçlarının rapor numaraları, bu raporların hangi bölümden talep edildiği, hangi doktor tarafından talep edildiği ve görüleceği, tahlilin çıktığı tarih ve saat değerleri ile kullanıcıların sonuç raporları çıkmış ise bu raporları görebilecekleri sütunlarından oluşan bir tablo bulunmaktadır.



Şekil 9. Laboratuvar sonuçları sayfası

Hastane randevu alma uygulamasında son olarak ise kullanıcılara ait bir profil sayfası bulunmaktadır. Profil sayfası kullanıcıların kayıt bilgileri ile hasta bilgilerini sisteme kayıt edebilecekleri veya güncelleyebilecekleri bir alan sağlamaktadır. Bu sayfada kullanıcı adı-soyadı, mail, şifre, adres, TC kimlik numarası, aktif olarak kullandığı ilaçlar ve yine aktif olarak aldığı tedaviler alanları bulunmaktadır.



Şekil 10. Kullanıcı profil sayfası

Hastane randevu alma uygulaması yukarıda bahsedilen alanlardan oluşmaktadır ve kullanıcı testi içerisinde oluşturulan adımlarda bu alanlar üzerinde şekillenmektedir. Uygulamanın bu ilk versiyonunda herhangi bir kullanılabilirlik çalışması yapılmamıştır.

CodeCarbon Kütüphanesi

CodeCarbon kütüphanesi Mila, BCG GAMMA, HAVERFORD COLLEGE, comet kuruluşlarının ortak yapılanması ile geliştirilmiş ve ücretsiz topluluk kullanımına sunulmuştur. En basit ve temel manada bu kütüphane, üzerinde çalışılan bilgisayarda kullanılan uygulamaya veya içeriklere bağlı olarak yapılan karbon emisyonlarını tahmin etme, ölçümleme, gözlemleme ve inceleme eylemlerini gerçekleştirmektedir. Kütüphanenin

geliştirilmesindeki motivasyonlar ise aşağıdaki gibi açıklanabilir. Günümüzün gelişen teknolojisi ile bilgisayar bilimleri artık çok popülerdir ve hayatın neredeyse tüm alanları ile içi içe olarak çalışmaktadır. Bunun kapsamında makine öğrenmesi, yapay zekâ, nesnelerin interneti vb. bilişim alanlarının birçoğunda büyük tutarda veri kullanılmaktadır. Teknolojinin ve bilgisayarların her alanda çok yoğun bir şekilde kullanılması gibi durumların sonuçlarından biri de önemli ölçülerde yapılan enerji tüketimleridir ve karbon emisyonlarıdır. Yine günümüzün en önemli problemlerinden biri de yapılan sera gazı salınımları ve bunlara bağlı olarak dünyada meydana gelen, küresel ısınma, kıtlık, canlı nesillerinin tükenmesi vb. olaylardır. Bu problemlerin sonuçlarını olabildiğince azaltmak, gelecek nesillere daha yaşanabilir bir dünya bırakmak için bu konuda çalışmalar yapmak önemli bir motivasyon kaynağıdır ve gereklidir.

Kütüphane, CO₂ ölçümlerini CO₂-equivalents türünden ifade etmektedir. CO₂-equivalents karbondioksitin kilogram cinsinden eşdeğeri şeklinde yani denk küresel ısınma etkisine sahip CO₂ miktarı olarak ifade edilmektedir ve CO₂-eq ile gösterilmektedir. Bu salınımların üzerinde tüketilen elektrik miktarı önemlidir ve kütüphane ölçümlerini kilowatt-saat birimi üzerine düşen CO₂ eşdeğeri cinsinden ölçüm yapmaktadır.

CO₂-eq değeri iki temel etkenin sonucu olarak ortaya çıkmaktadır. Bunlar C ve E değerleridir. C değeri uygulama tarafından harcanan elektriğin karbon yoğunluğunu ifade eder ve bu değer yukarıda da belirtildiği gibi kullanılan elektriğin kilowatt-saat başına salınımı ile yapılan CO₂ değeridir. E değeri ise kilowatt-saat cinsinden tüketilen enerji ölçümü değerini ifade etmektedir. Kütüphane çalışılan uygulamanın karbon salınımı değerini de C*E şeklinde yani C ve E değerlerinin çarpımı olarak ölçümler.

$$\text{Karbon Salınımı} = C(\text{Elektrik Karbon Yoğunluğu}) \times E(\text{Tüketilen Enerji})$$

Kütüphanede karbon yoğunluğu, elektriğin oluşumu için kullanılan çeşitli enerji kaynaklarının karbon yoğunluklarının ortalaması alınarak hesaplanmaktadır. Bahsedilen çeşitli enerji kaynakları ise elektriğin yenilenebilir enerji kaynakları veya fosil yakıtlar gibi hangi tür kaynaktan oluşturulduğu anlamına gelmektedir. Fosil yakıtlar için doğal gaz, petrol, kömür vb. kaynaklar, yenilenebilir enerji kaynakları için ise hidroelektrik, jeotermal, güneş enerjisi gibi enerji kaynakları örnek verilebilir. Yenilenebilir enerji kaynaklarının daha az karbona sahip olmaları, bu kaynakları daha çevre dostu hale getirmektedir. Kütüphane, bilgisayarın kullanıldığı bölgedeki lokal şebekenin fosil yakıt ve yenilenebilir enerji kaynak türlerinden oluşan karışımını kullanarak kullanılan elektriğin karbon yoğunluğu değerini ölçümler.

CodeCarbon kütüphanesi Türkiye’deki global elektrik kaynak türlerinin karışımı için aşağıda belirtilen yapıyı kullanmaktadır;

```
"TUR": {  
  "biofuel_TWh": 6.45,  
  "carbon_intensity": 430.401,  
  "coal_TWh": 101.61,  
  "country_name": "Turkey",  
  "fossil_TWh": 214.86,  
  "gas_TWh": 111.18,  
  "hydroelectricity_TWh": 55.93,  
  "iso_code": "TUR",  
  "low_carbon_TWh": 118.55,  
  "nuclear_TWh": 0.0,  
  "oil_TWh": 2.07,  
  "other_renewable_TWh": 17.24,  
  "other_renewable_exc_biofuel_TWh": 10.79,  
  "per_capita_Wh": 3932.862,  
  "renewables_TWh": 118.55,  
  "solar_TWh": 13.94,  
  "total_TWh": 333.41,  
  "wind_TWh": 31.44,  
  "year": 2021  
},
```

Şekil 11. Türkiye elektrik karbon yoğunluğu değerleri

Yukarıda Türkiye için belirtilen elektrik karbon yoğunluğu verileri Oxford Üniversitesi ve Global Change Data Lab ortaklığında “Our Word in Data” verilerinden oluşmaktadır ve ilgili verilerde diğer ülkelerin bilgileri de bulunmaktadır. Bu verilere göre Türkiye 2000 yılından kilowatt-saat elektrik başına gram cinsinden 488 gCO₂e karbon salınımı yapıyorken 2022 yılında bu salınım miktarı 414 gCO₂e şeklinde hesaplanmıştır. Yani 22 yıllık bir süre zarfında, ölçülen karbon salınımında yaklaşık olarak %15 oranında bir azalma görülmektedir (Hannah Ritchie vd., 2023).

Kütüphane bilgi işlem alt yapısının çalıştığı bölge veya ülkenin elektrik enerji karışımı ile enerji kaynaklarının karbon yoğunluklarını kullanarak, kullanılan elektriğin karbon yoğunluğu değerini üretmektedir. Fosil enerji kaynaklarının karbon yoğunluğu değerleri “U.S Energy Information Administration Data” verilerinden elde edilmiştir. Yenilenebilir enerji kaynaklarına ait karbon yoğunluğu değerleri ise “Word Nuclear Association” tarafından yayınlanan “Comparison of Lifecycle Greenhouse Gas Emissions of Various Electricity Generation Sources” yani “Çeşitli Elektrik Üretim Kaynaklarının Yaşam Döngüsü Sera Gazı Emisyonlarının Karşılaştırılması” raporunda sunulan verilerden elde edilmiştir (Word Nuclear Association, 2011). Kütüphanenin net karbon yoğunluğunu hesaplamak için kullandığı fosil ve yenilenebilir kaynakların karbon yoğunluğu değerleri aşağıda bulunan tablodaki gibidir.

Enerji kaynağı	Karbon Yoğunluğu (kg/MWh)
Kömür	995
Petrol	816
Doğal gaz	743
Jeotermal	38
Hidroelektrik	26
Nükleer	29
Güneş	48
Rüzgâr	26

Şekil 12. Enerji kaynakları karbon yoğunluğu

Net Karbon Yoğunluğu Nasıl Hesaplanır?

Kütüphane, tüketilen net karbon yoğunluğunu, yukarıda da bahsedildiği gibi bilişim alt yapısının çalıştığı ülkenin elektrik enerjisi karışımının, karışımda kullanılan enerji türlerinin karbon yoğunluğu ile çarparak ve sonrasında tüm değerleri toplayarak hesaplamaktadır. Yukarıdaki tabloda belirtilen uluslararası enerji kaynaklarının karbon karışımı değerleri ise ABD Enerji Bilgi İdaresinin 2016 verilerinden elde edilmiştir. Bunu bir örnek ile açıklayacak olursak altyapının çalıştığı ülkenin elektrik enerjisi %20 kömür, %35 petrol, %30 jeotermal ve %15 güneş enerjisi kullanılarak sağlanıyor ise bu ülke için net karbon yoğunluğu;

$$\begin{aligned}
 \text{Net Karbon Yoğunluğu} &= 0.20 \times 995 + 0.35 \times 816 + 0.30 \times 38 + 0.15 \times 48 \\
 &= 503,2 \text{ kgCO}_2 / \text{kWh}
 \end{aligned}$$

şeklinde hesaplanacaktır.

Kütüphane, eğer bir ülkenin karbon elektrik enerjisi karışımına ve karbon yoğunluğu değerlerine ulaşamaz ise Uluslararası Enerji Ajansını kaynak olarak kullanarak dünya ortalaması olan 475 g CO₂.eq/KWh değerini kullanmaktadır.

Enerji Tüketimi ve Hesaplanması

CodeCarbon kütüphanesinde bilişim altyapısının veya uygulama için kullanılan bilgisayarın güç tüketimi değerini hesaplamak amacıyla temel donanım kaynağının güç tüketimi varsayılan şekilde 15 saniyede bir izlenerek takip edilmektedir ve bunu ifade eden parametre “measure_power_secs” parametresidir.

Çalışılan bilgisayarın güç tüketimlerinin izlendiği temel donanımlar ise GPU, CPU ve RAM bileşenleridir. Kütüphane bu donanımların enerji tüketimlerini hem ayrı ayrı ölçümlemektedir hemde bunların toplam ölçümlerini çıktı olarak üretmektedir. Kütüphanede, GPU güç tüketimini izlemek için GPU izleme ve yönetme işlemleri için kullanılan “pynvml” kitaplığı kullanılmaktadır. RAM bileşeni için güç tüketimi sistemin türü, bellek türü ve uygulamaların farklılık göstermesine bağlı olarak değişebilmektedir ve genelleme yapılarak 8 GB DDR3 ve DDR4 RAM’ ler için 3 watt şeklinde ifade edilmektedir. CPU bileşeni için ise Windows ve Mac işletim sistemine sahip bilgisayarlar için Intel Power Gadget güç yönetim aracı kullanılarak enerji tüketimi izlenmektedir.

Sonuç olarak kütüphanede net enerji tüketimi değerinin matematiksel formülü aşağıdaki denklem verilmiştir;

$$Net\ Enerji\ Tüketimi = Güç \times Kullanım\ Süresi$$

Kütüphane Nasıl Çalışır?

Kütüphane öncelikle çalışılan bilgisayarda terminal açılarak “pip install codecarbon” komutu çalıştırılarak bilgisayara yüklenmelidir. Paketin yüklenmesi ile birlikte “arrow, pandas, pynvml, requests, psutil, py-cpuinfo, fuzzywuzzy, click, prometheus_client” paketleri de yüklenmektedir. Daha sonra karbon emisyonu ve enerji tüketimi ölçülmek istenen uygulamaya kütüphane eklenmelidir. Son olarak kütüphanenin ölçümleri başlatmayı ve durdurmayı sağlayan start() ve stop() fonksiyonları, ölçümlerin yapılacağı kod alanını sarmalayacak şekilde uygulamaya yerleştirilmelidir.

Algoritma: CodeCarbon Entegrasyonu

```
1  From codecarbon import EmissionsTracker
2  tracker = EmissionsTracker()
3  tracker.start()
4  try:
5      #Kod İçeriği
6  finally:
7      tracker.stop()
```

Şekil 13. CodeCarbon entegrasyonu

Uygulamanın çalıştırılması kullanılması ve daha sonra tamamlanarak çalışmanın durdurulması ile birlikte kütüphane, elde edilen sonuçları .csv uzantılı bir dosya şeklinde üretmektedir. Csv uzantılı dosyalar excel programı aracılığıyla açılabilir ve verilen değerler virgül ile ayrılarak dosya içerisinde sıralanmaktadır.

Sonuç olarak kütüphanede verilen .csv uzantılı dosya içerisinde, kütüphanede tanımlı olan toplamda 25 farklı değerde çıktı üretilmektedir. Bu çıktılar incelenecek olursa;

- a. timestamp: Zaman damgası değeri uygulama veya bilgisayar üzerinde kütüphanenin çalıştırılmaya başladığı zamanı ifade eder. Değer yy-aa-ggTsa:dk:sn formatında ifade edilir. Zamanı yıl, ay, gün, saat, dakika ve saniye cinsinden göstermektedir.
- b. project_name: Bu değer projenin adını ifade etmektedir ve oluşturulan rapora varsayılan değer olarak “codecarbon” şeklinde atanmaktadır.
- c. run_id: Bu değer çalışmayı, diğer çalışmalardan tamamen ayırmayı sağlayan çalışma kimliğidir.
- d. duration: Bu değer uygulama veya bilgisayar üzerinde kütüphanenin start() ve stop() fonksiyonları arasında geçen süreyi ifade eder. Yani aslında bir nevi uygulamanın kullanım süresi olarak da ifade edilebilir. Kütüphane bu değeri saniye cinsinden üreterek çıktı vermektedir.
- e. emissions: Bu değer kütüphanenin içerisinde çalıştığı uygulama ile bilgisayarın çalışma esnasında salınımı yapılan CO₂ değerini göstermektedir. CO₂ kg türünden ifade edildiği için CO₂eq yani karbon eşdeğeri şeklinde belirtilmektedir.
- f. cpu_energy: Bu değer uygulama ve kütüphanenin çalışması esnasında CPU bileşeni başına kullanılan enerji değerini göstermektedir. Değer kilowatt saat (kWh) türünden ifade edilmektedir.
- g. gpu_energy: Bu değer uygulama ve kütüphanenin çalışması esnasında GPU bileşeni başına kullanılan enerji değerini göstermektedir. Değer kilowatt saat (kWh) türünden ifade edilmektedir.
- h. ram_energy: Bu değer uygulama ve kütüphanenin çalışması esnasında RAM bileşeni başına kullanılan enerji değerini göstermektedir. Değer kilowatt saat (kWh) türünden ifade edilmektedir.
- i. energy_consumed: Bu değer kilowatt saat (kWh) türünden toplamda tüketilen enerji miktarını ifade etmektedir. Toplam enerji tüketimi değeri hesaplanırken, öncesinde hesaplanan cpu_energy, gpu_energy ve ram_energy bileşenlerinin tamamının toplanması ile ortaya çıkmaktadır.
- j. country_name: Bu değer uygulama, bilgisayar ile kütüphanenin çalıştırıldığı ülkenin yani altyapının bulunduğu ülkenin adını ifade etmektedir.
- k. country_iso_code: Değer altyapının çalıştığı ülkenin ISO kodunu belirtmektedir ve üç harften oluşmaktadır.
- l. region: Bu değer bilgisayar, uygulama ve kütüphanenin çalıştığı yani bir nevi altyapının çalıştığı ülkedeki şehir, il veya eyaleti göstermektedir.

- m. `on_cloud`: Bu değer kütüphane ile birlikte çalıştırılan uygulamanın altyapının bulut üzerinde mi yoksa özel veya yerel bir altyapıda mı çalıştığını ifade etmektedir. Eğer çalışma altyapı bulut üzerinde ise Y değeri, özel bir altyapıda ise N değeri üretilmektedir.
- n. `cloud_provider`: Bu değer uygulamanın bulut sağlayıcısı üzerinde çalıştırılması durumunda hangi bulut altyapısı üzerinde (aws, azure, gcp) çalıştırıldığını ifade etmektedir. Eğer uygulama bulut sağlayıcısında çalıştırılmıyorsa bu değer boş olarak üretilmektedir.
- o. `cloud_region`: Bu değer eğer bulut sağlayıcısı kullanılıyorsa bulut sağlayıcısının coğrafi bölgesinin ifade edilmesi için kullanılmaktadır.
- p. `os`: Bu değer kütüphane ve uygulamanın çalıştırıldığı cihazın işletim sistemini ifade etmektedir.
- q. `python_version`: Bu değer çalışma içerisinde hangi Python versiyonunun kullanıldığı bilgisini vermektedir.
- r. `codecarbon_version`: Bu değer çalışma içerisinde CodeCarbon kütüphanesinin hangi versiyonunun kullanıldığı bilgisini vermektedir.
- s. `cpu_count`: Bu değer çalışmadaki CPU sayısını ifade etmektedir.
- t. `cpu_model`: Bu değer çalışmada kullanılan CPU bileşeninin modelini ifade etmek için kullanılmaktadır.
- u. `gpu_count`: Bu değer ise çalışmadaki GPU sayısını ifade etmektedir.
- v. `gpu_model`: Bu değer çalışmada kullanılan GPU bileşeninin modelini ifade etmek için kullanılmaktadır.
- w. `longitude`: Boylam değeri, çalışmanın yapıldığı alanın 11.1 km / 123 km² 'lik alana boylam dağılımını ifade etmektedir.
- x. `latitude`: Enlem değeri, yine çalışmanın yapıldığı alanın 11.1 km / 123 km² 'lik alana enlem dağılımını ifade etmektedir.
- y. `ram_total_size`: Bu değer toplam kullanılabilir ram alanını ifade etmektedir. Değişkenleri yukarıdaki şekilde açıklanabilmektedir.

Kullanılabilirlik Ölçme Yöntemi

Tez çalışması kapsamında gerçekleştirilen uygulamaya kullanılabilirlik testleri uygulanmıştır. Test için kullanılabilirlik test türlerinden süreç sonu testleri yöntemi kullanılmıştır. Yani yazılım geliştirme sürecinin tamamlanmasından sonra uygulama son kullanıcıya sunulmadan önce yapılan test türleridir. Yazılım geliştirme süreçleri içerisinde

herhangi bir uzman görüşüne veya hedef kullanıcı kitlelerine uygulama denettirilmemiş, görüş ve geri dönütler alınmamıştır.

Çalışmada kullanılabilirlik test yöntemi olarak ise deneysel yaklaşım kullanılmıştır. Yani uygulama gerçek kullanıcılara gerçek test ortamında denettirilmiştir. Öncelikle test ortamı ve hazırlıklar yapılmıştır. Tercih edilen uygulama hastane randevu alma uygulamasıdır. Test ortamında, ihtiyaç olarak uygulamanın çalıştırılacağı ve işlemlerin yapılacağı bir bilgisayar gerekmektedir. Bundan dolayı test ortamı olarak Atatürk Üniversitesi Yazılım Mühendisliğine ait Yazılım Geliştirme ve Test Laboratuvarı kullanılmıştır. Laboratuvarda testlerin yapılacağı bilgisayara testlerden önce format atılmıştır ve bilgisayara yalnızca uygulamada kullanılacak bileşenler kurulmuştur. Test öncesinde kullanıcılara test protokolü ve görev listesi verilmiştir. Daha sonra kullanıcılara test protokolü ile test hakkında bir bilgilendirme yapılmıştır. Aynı zamanda test esnasında kullanıcıyı gözlemlemek amacıyla bir gözlem kâğıdı da edinilmiştir.

Testin uygulanacağı kişi test ortamına getirildikten sonra kişiye test protokolü verilerek bunu okuması istenir. Kişi protokolü okuduktan sonra kişinin herhangi bir sorusu var ise bunlar cevaplanır ve test esnasında kullanacağı uygulamadan kısaca bahsedilir. Sonrasında kişiye test esnasında yerine getirmesi gereken adımları içeren görev listesi verilerek kişiden bu listeyi de gözden geçirmesi ve eğer bir sorusu var ise bunu sorması istenir. Bu aşama tamamlandıktan sonra kişiye test esnasında yaptığı işlemlerin video kaydına alınacağı ve aynı zamanda uygulama kullanımı tamamlandıktan sonra da kişi ile karşılıklı gerçekleştirilecek görüşmede ses kaydı alınacağı üzerine bir bilgilendirme yapılmaktadır. Kişinin bu konulardaki rızası alındıktan sonra ise test aşamasına geçilmektedir.

Testin gerçekleştirileceği kişi bilgisayar başına alındıktan sonra uygulama açılarak kullanıcının görev listesindeki adımları yerine getirmesi beklenir. Kişi görev adımlarını yerine getirirken uygulamaya başlama zamanı, uygulamayı tamamlama zamanı, görevleri yerine getirirken uygulamanın hangi sayfasında ne kadar takıldığı hangi bileşenleri bulmakta sorun yaşadığı, hangi görevlerde sorun yaşadığı gibi gözlemler kaydedilir. Aynı zamanda kişinin herhangi bir görevi yerine getiremediği ve ilerleme sağlayamadığı durumlar için kişi yardım isterse kişinin test sürecine müdahale edilebilmektedir.

Testin uygulandığı kişi test adımlarını başarılı bir şekilde sırayla tamamladıktan sonra uygulamanın kullanım aşaması tamamlanmış olmaktadır. Kullanım aşaması sonrasında kullanıcı ile bir görüşme gerçekleştirilmektedir. Görüşmede kullanıcıya testten önce hazırlanan kullanıcıların uygulama hakkındaki etkililik, verimlilik ve memnuniyet

kavramlarını ölçmeye dayalı içeriklere sahip olan 60 adet soru kullanıcılara sorularak kişilerden bu sorulara kısa ve fikirlerini açıkça ifade eden cevaplar vermeleri beklenmektedir. Bu görüşme sonrasında hiçbir noktanın kaçırılmadan değerlendirilmesi için başta izin alındığı üzere görüşme ses kaydı ile kayıt altına alınmaktadır. Görüşme tamamlandıktan sonra da ilk test aşaması tamamlanmış olmaktadır.

Deney Kullanıcı Profili

Hastane randevu alma uygulaması için birçok yaş, cinsiyet, eğitim farklı statü ve durumdan kitle hedef kullanıcı kitlesi olarak değerlendirilebilir. Bunun sebebi günümüzün yaygın teknolojik imkanları sayesinde insanların randevu alma gibi işlemleri hastanelere giderek yüz yüze yapmak yerine, internet ve telefon gibi imkanları kullanarak gerçekleştirmeleridir. Telefon seçeneğinin yanında internet üzerinden randevu almak da oldukça sık ve bu imkanlara sahip olan her kitleden insanlar tarafından tercih edilen bir yöntemdir.

Tez çalışması kapsamında bu çeşitlilik durumunu bir oranda sınırlandırmak amacıyla deney kullanıcısı olarak seçilen kişiler 25-35 yaş aralığında, kadın ve erkek, lisans ve üzeri eğitim düzeyine sahip, çalışma durumu değişebilen ve eğitim düzeylerinden dolayı bilgisayar kullanma konusunda deneyimleri olan kişiler seçilmiştir. Kişi sayısı ise bu alan ile ilgili yapılan ve kuramsal temellerde değinilen 2 çalışma dikkate alınarak her aşamada 12 kişi olacak şekilde seçilmiştir.

Test Protokolü

Kullanıcılara testten önce verilen test protokolü metni aşağıdaki gibidir;

“Bu oturuma katıldığınız için öncelikle teşekkürler.

Hosapp uygulaması neredeyse her insanın günlük yaşantısında hastaneden randevu almak için kullandığı bir randevu sistemidir. Kullanıcılar sisteme kaydolup giriş yaparak sistem üzerinden hastane randevusu ve Checkup randevusu alabilmektedirler. Bu oturumun amacı sistemi deneyimleyerek sistem hakkında geri dönüşler vermek ve bunlar ile sistemi kullanılabilirlik açısından daha iyi hale getirmektir.

Siz sistemi kullanırken aynı zamanda enerji tüketim ve kod karbon emisyonu ölçümleri de yapılarak elde edilen ölçümlerden çıkarımlar yapılacaktır. Bu amaçlar kapsamında uygulamada bazı problemler bulunmaktadır fakat uygulamayı kullanırken karşılaştığınız herhangi bir problem uygulamanın problemidir, sizin değil.

Uygulama, siz görevleri tamamlayınca sonlanacaktır. Sizden uygulama ile ilgili belli görevleri yapmanızı beklenecek ve bu esnada gözlem yapılacaktır. Siz görevleri yerine getirirken uygulama kullanımına dair bir video kaydı alınacaktır. Bu sadece uygulamadaki işlemlerin nasıl ilerlediğinin sonrasında hatırlanabilmesi için yapılmaktadır.

Uygulama kapsamında size verilen görevleri olabildiğince kendi başınıza deneyip yapmanız beklenmektedir. Görevleri yaparken herhangi durumda bir yerde takılırsanız ve ilerleyemeyeceğinizi düşünürseniz size yardımcı olunacaktır fakat yine de bu durumları kendi başınıza çözmeniz beklenmektedir. Süreç doğrultusunda gerekli görmeniz doğrultusunda görevlerden çıkabilir, başka bir göreve başlayabilir veya uygulamayı sonlandırabilirsiniz.”

Bu protokolde testin gerçekleştirileceği kişileri test hakkında bilgilendirmek amaçlanmıştır. Protokolde sırasıyla deney kullanıcısına hitap olarak teşekkür etme, uygulamaya dair genel bir bilgilendirme, uygulanacak talimatlardan yüzeysel olarak bahsetme ve uygulamanın başlama, devam süreci ile tamamlanması hakkından bilgiler verilmiştir.

Gözlem Kâğıdı

Gözlem kâğıdı ise deney kullanıcısının uygulamayı gerçekleştirdiği esnada, testi yürüten kişi tarafından deney kullanıcısının gözlemlemesi ve gerekli notları tutabilmesi amacıyla kullanılmaktadır. Bu kâğıdın içeriği yapılacak uygulamanın içeriğine göre düzenlenmelidir. Bu içerikte genel olarak uygulamanın başlama zamanı, bitiş zamanı, uygulamadaki her bir sayfanın adı ve sayfa adının karşı kısmında uygulama esnasında deney kullanıcısının ilgili sayfa üzerinde çalışırken yapılan gözlemlerin not edileceği bir alan bulunmaktadır. Tez çalışması kapsamında uygulamada 10 ayrı sayfa değerlendirildiği için bu formda çalışma için 10 alan bulunmaktadır.

Görev Listesi

Tez çalışmasının üzerinde yürütüleceği uygulama olan hastane randevu alma uygulamasında deneysel yaklaşımlı kullanılabilirlik testinin yürütülebilmesi için deney kullanıcılarına bir görev listesi hazırlanmıştır. Hazırlanan bu listede uygulamanın 10 farklı sayfasında gerçekleştirilecek görev adımları yer almaktadır. Listedeki adımlar deney kullanıcılarının en kolay şekilde anlayabilmeleri ve kafalarının karışmaması için olabildiğince açık ve anlaşılır şekilde ifade edilmeye çalışılmıştır. Görevler oluşturulurken uygulamadaki temel işlevlerin tamamının deney kullanıcısı tarafından deneyimlenmesine dikkat edilmiştir. Görev listesi aşağıdaki gibidir;

- a. Adım 1: Şu anda uygulama giriş sayfası ile karşı karşıyasınız. Uygulamaya giriş yapınız. Eğer kayıtlı değilseniz önce kayıt olunuz.
- b. Adım 2: Giriş yaptığınız için artık karşınızda uygulamanın ana sayfası bulunmaktadır. Buradan “bölümlerimiz” sayfasına gidiniz ve hastanede hangi bölümlerin mevcut olduğunu görerek randevu oluşturmak istediğiniz bölüme karar veriniz.
- c. Adım 3: Bölüm seçiminizi yaptıktan sonra uygulamada bulunan “doktorlarımız” sayfasına gidiniz ve belirlediğiniz bölüme ait doktorları görerek randevunuzu hangi doktordan oluşturmak istediğinize karar veriniz.
- d. Adım 4: Randevunuzu oluşturmak istediğiniz bölüm ve doktora karar verdikten sonra Seçtiğiniz doktorun hangi gün ve saatlerde müsait olduğunu öğrenmek için “müsait randevu durumu” sayfasına gidiniz ve randevu oluşturacağının gün ve saati belirleyiniz.
- e. Adım 5: Bu adımda randevu oluşturma işlemini tamamlamak için “yeni randevu” sayfasına gidiniz ve öncesinde belirlediğiniz bilgiler ile gerekli alanları doldurarak randevunuzu onaylayınız.
- f. Adım 6: Oluşturduğunuz randevunuzun bilgilerinin doğruluğunu teyit etmek için “randevu sorgu/iptal” sayfasına gidiniz. Randevunuzun bilgilerini ve sisteme doğru işlenip işlenmediğini kontrol ediniz.
- g. Adım 7: Bir sonraki işlem olarak check-up randevusu oluşturmanız beklenmektedir. Bu yüzden “check-up randevu” sayfasına giderek bu sayfadaki alandaki gerekli alanları doldurunuz ve check-up randevunuzu oluşturunuz.
- h. Adım 8: Bu adımda daha öncesinde bir tedavi sürecinizin olduğunu varsayın ve bu tedaviniz esnasında “Beyin ve Sinir Cerrahi” bölümünden “Dr. Ahmet Veli” doktoruna ait 12-06-2023 16.00 tarihli ve saatli bir laboratuvar sonucunuzun olduğunuz farz edin. Uygulamadaki “laboratuvar sonuçları” sayfasına giderek bu sonucunuzun ne durumda olduğunu kontrol ediniz.
- i. Adım 9: Bir önceki adımı tamamladıktan sonra uygulamada bulunan kullanıcı profil sayfasına gidiniz ve bu alanda bulunan bilgileri güncelleyiniz.
- j. Adım 10: Son olarak uygulama adımlarını tamamlamak adına çıkış yaparak uygulamayı kapatınız.

Yukarıda belirtilen görev listesi, gözüktüğü gibi deney kullanıcısının uygulama içerisindeki tüm sayfaları 10 adımda görmesini ve deneyimlemesini hedeflemektedir. Test esnasında bu liste deney kullanıcılarına verilmiştir.

Kullanıcı Testi Soruları

Deney kullanıcısı uygulamada görev adımlarını tamamladıktan sonra uygulamanın deneyimlenme kısmı tamamlanmış olacaktır. Sonrasında deney kullanıcıları ile görüşme aşamasına geçilecektir. Bu aşamada uygulamanın kullanılabilirliğini ölçebilmek amacıyla uygulamadaki her sayfa için ayrı ayrı memnuniyet, verimlilik ve etkililiğe yönelik görüşme soruları oluşturulmuştur. Sorular deney kullanıcısına yöneltilerek deney kullanıcısının, uygulamanın alanları hakkındaki görüş ve geri dönütleri toplanmıştır. Görüşme sözlü bir şekilde gerçekleştiğinden dolayı detayları kaçırmamak adına kullanıcılardan onay alınarak ses kayıtları alınmıştır. Sorulan soru kökleri temel olarak aşağıdaki gibidir;

1. Sistemin..... sayfasında bulunan bileşenlerin yerleşimi uygun mu? Değil ise nasıl olması gerektiğini düşünüyorsunuz?
Cevap:
2. Sistemin..... sayfasında bulunan bileşenlerin fonksiyonelliği (çalışma kapsamı) yeterli mi? Değil ise nasıl geliştirilebileceği hakkında bir fikriniz var mı?
Cevap:
3. Sistemin..... sayfasını daha kolay kullanabilmek için ne önerirsiniz?
Cevap:
4. Size verilen görev adımları kapsamında sistemin..... sayfasındaki görevleri yaparken harcadığınız zaman uygun mu? (az-çok-yeterli)
Cevap:
5. Size verilen görev adımları kapsamında sistemin..... sayfasındaki görevleri yaparken harcadığınız çaba seviyesi nedir? (kolay-orta-zor)
Cevap:
6. İlgili sayfayı kullanırken genel olarak sistemden memnun kaldınız mı?
Cevap:

Şekil 14. Kullanıcı testi soru kalıpları

Bu temel soru kalıpları tüm sayfalar için uyarlanmıştır ve kullanıcıya bu şekilde sorulmuştur. Soruların sayfalara nasıl uyarlandığına örnek olarak, aşağıda yeni randevu oluşturma sayfası için hazırlanan ve sorulan soruları gösterilmektedir.

1. Sistemin yeni randevu alma sayfasında bulunan bileşenlerin yerleşimi uygun mu? Değil ise nasıl olması gerektiğini düşünüyorsunuz?
Cevap:
2. Sistemin yeni randevu alma sayfasında bulunan bileşenlerin fonksiyonelliği (çalışma kapsamı) yeterli mi? Değil ise nasıl geliştirilebileceği hakkında bir fikriniz var mı?
Cevap:
3. Sistemde bulunan yeni randevu alma sayfasını daha kolay kullanabilmek için ne önerirsiniz?
Cevap:
4. Size verilen görev adımları kapsamında sistemin yeni randevu alma sayfasındaki görevleri yaparken harcadığınız zaman uygun mu? (az-çok-yeterli)
Cevap:
5. Size verilen görev adımları kapsamında sistemin yeni randevu alma sayfasındaki görevleri yaparken harcadığınız çabanın seviyesi nedir? (kolay-orta-zor)
Cevap:
6. İlgili sayfayı kullanırken genel olarak sistemden memnun kaldınız mı?
Cevap:

Şekil 15. Örnek soru formatı

Kullanıcılar ile yapılan görüşmeler tamamlandıktan sonra, alınan ses kayıtları Microsoft Word bileşeni kullanılarak yazılı hale getirilmiştir. Öncelikle her kullanıcının dosyaları ayrı ayrı olacak şekilde incelenerek analiz edilmiştir ve daha sonra ortak bir dosya oluşturulmuştur. Ortak dosya içerisinde, uygulamanın sayfaları olan; ana sayfa, bölümlerimiz sayfası, doktorlarımız sayfası, müsait randevu durumu sayfası, yeni randevu alma sayfası, randevu sorgu/iptal sayfası, check-up randevu sayfası, laboratuvar sonuçları sayfası, profil sayfası, giriş yapma ve kayıt olma sayfaları için 10 ayrı başlık altında olmak üzere, tüm deney kullanıcılarından bu sayfalar ile ilgili gelen geri dönütler toplanmıştır.

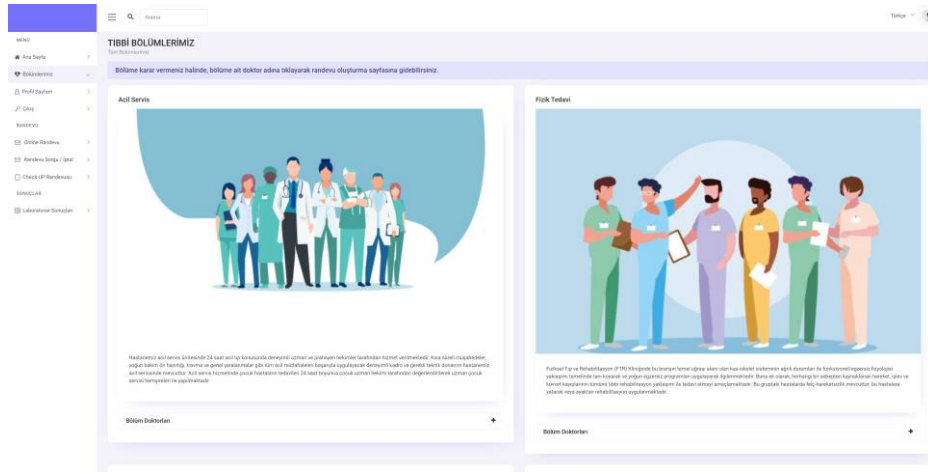
Her deney kullanıcısı için test gerçekleştirildikten sonra, uygulama çalışırken arka planında CodeCarbon kütüphanesi tarafından üretilen ve içerisinde öncesinde bahsedilen 25 farklı çıktı değerini barındıran .csv uzantılı dosyalar elde edilmiştir. Kullanıcıların uygulama deneyimleri sonucunda elde edilen bu çıktılar öncelikle .xlsx formatına geçirilmiştir ve daha sonra ise tek bir dosya altında toplanmıştır. Çıktı olarak verilen değerlerin bazılarında dönüşüm, bazılarında ise doğrulama hesapları yapılmıştır.

Tüm bu adımların gerçekleştirilmesiyle tez çalışmasının ilk aşaması tamamlanmaktadır. Sonrasında uygulamanın ikinci aşamasına geçilmiştir.

Uygulamanın Tekrar Revize Edilmesi

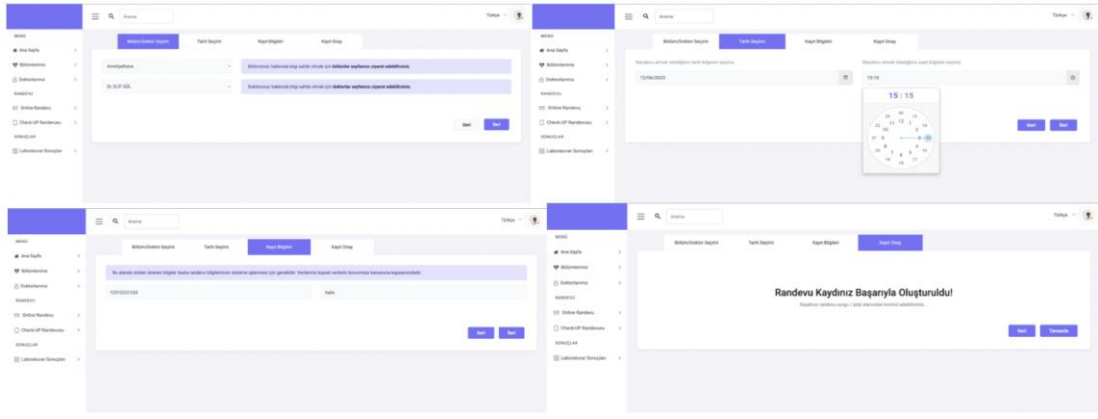
Tez çalışmasının ilk aşamasında, kullanıcı görüşlerinden toplanan geri dönütler analiz edilmiştir ve hem arayüzün tasarımı açısından hem de uygulama bileşenlerinin fonksiyonelliği açısından değerlendirmeler yapılmıştır. Yapılan değerlendirmeler sonucunda, hem arayüz tasarımı açısından hem de çalışma ve fonksiyonellik açısından uygulamada ne tür değişikliklerin yapılması gerektiğine dair bir rapor oluşturulmuştur. Oluşturulan bu rapor kapsamında ise uygulama sayfaları tekrardan düzenlenmiştir.

Hastane randevu uygulamasının ikinci versiyonunda ana sayfa, uygulamaya giriş sayfası ve uygulama kayıt sayfalarında hem arayüz olarak, hem de fonksiyonellik açısından bazı düzenlemeler yapılmıştır. Bölümümüz ve doktorlarımız sayfalarında ise arayüz tasarımı yönünden bazı değişiklikler yapılmıştır ve doktor içerikleri, bölümlerimiz sayfasına dahil edilmiştir.



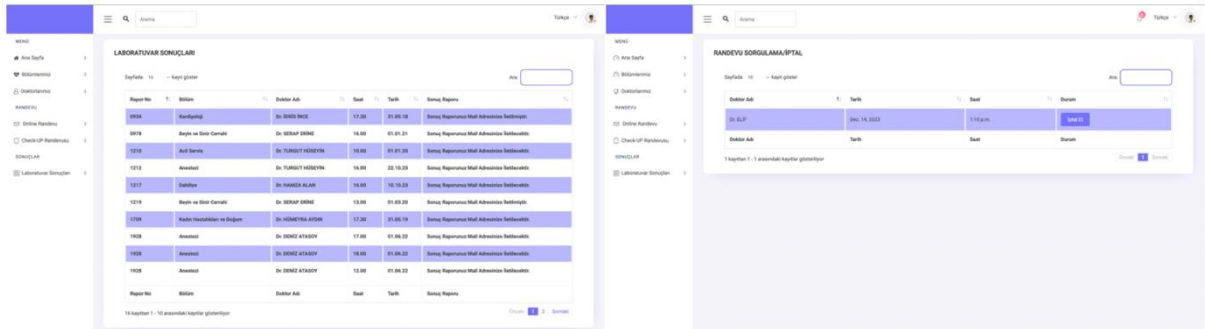
Şekil 16. İkinci versiyon bölümler ve doktorlar sayfası

Müsait randevu durumu sayfası ise deney kullanıcılarının geri dönütleri doğrultusunda, yeni randevu oluşturma sayfasında bulunan randevu oluşturma adımlarının içeriğine dahil edilmiştir. Yeni randevu alma sayfası ise hem arayüz tasarımı açısından hem de sayfanın işlevsellik durumu açısından güncellenmiştir. Kullanıcı görüşleri doğrultusunda, kullanıcının uygun görmesi durumunda tekrar bölümler veya doktorlar sayfasını kolaylıkla ziyaret edebileceği alanlar bu sayfaya eklenmiştir. Sayfada bulunan form geliştirilerek step form haline getirilmiştir. Aynı zamanda, kullanıcıları yaptıkları işlemlerin durumları hakkında bilgilendirmek amacıyla uygulama bildirimleri de eklenmiştir.



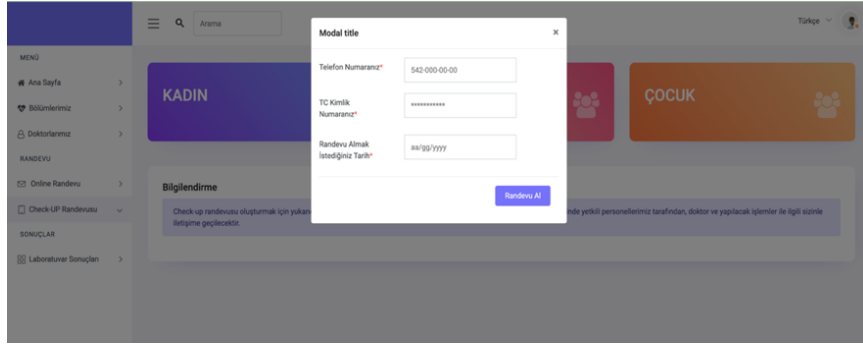
Şekil 17. İkinci versiyon online randevu sayfaları

Randevu sorgu/iptal ve laboratuvar sonuçları sayfalarının formatları birbirine benzediğinden dolayı, iki sayfa için deney kullanıcılarından gelen geri dönütler, ortak noktaların güncellenmesi gerektiğini gösteriyordu. Deney kullanıcılarının geri dönütleri kapsamında hem arayüz tasarımı olarak hem de sayfanın çalışma fonksiyonları üzerinde bu sayfalarda güncellemeler yapılmıştır. Örneğin, bir önceki versiyonda kullanıcılar randevu sayfalarında önceden alınan randevuları, görmelerinin kafa karıştırıcı olduğundan ve bu sayfada yalnızca o gün aldıkları veya gelecekte gerçekleştirilecek randevularını görmelerinin daha iyi olabileceğinden hakkında dönüt verdikleri için ikinci versiyonda bu alan güncellenmiştir.



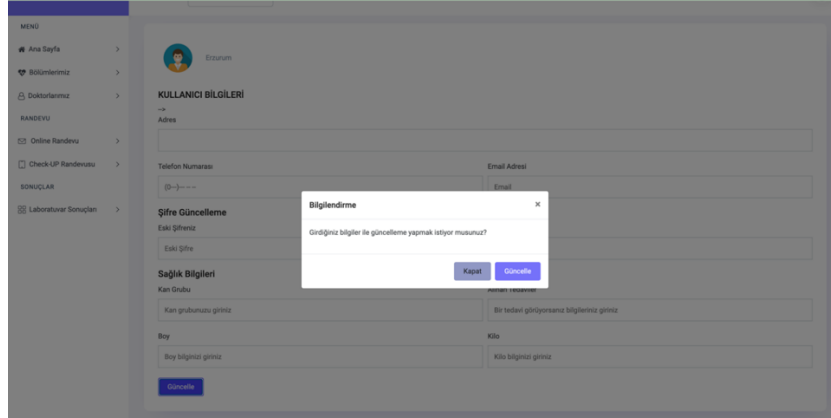
Şekil 18. İkinci versiyon laboratuvar sonuçları ve randevu sorgu/iptal sayfaları

Check-up randevu alma sayfası üzerinde ise hem arayüz tasarımı açısından hem de sayfanın çalışma fonksiyonları açısından güncelleme yapılmıştır. Sayfa içeriğine, deney kullanıcılarından elde edilen geri dönütler kapsamında bilgilendirme alanları eklenerek, check-up randevusu oluşturma formunda doldurulması gereken alanlarda değişiklikler yapılmıştır.



Şekil 19. İkinci versiyon check-up randevu sayfası

Kullanıcı profil sayfası ise yine deney kullanıcılarından elde edilen geri dönütler dahilinde hem arayüz tasarımı olarak hemde sayfanın çalışma fonksiyonları açısından tekrar düzenlenmiştir. Sayfada genel olarak kullanıcı bilgilerinin bulunduğu alanlar güncellenmiş ve kullanıcıların uygulama ile etkileşim sağlayabilmesi için uygulamaya durum bildirimleri eklenmiştir.



Şekil 20. İkinci versiyon kullanıcı profil sayfası

Yukarıda bahsedildiği gibi uygulamanın ilk versiyonu üzerinde gerçekleştirilen testler sonucunda toplanan geri dönütler doğrultusunda uygulamanın ikinci versiyonu oluşturulmuştur. Tez çalışmasının ikinci aşamasında, ilk aşamada uygulama üzerinde yapılan kullanıcı testleri tekrar uygulanmıştır.

Kullanıcı testinin, uygulamanın ikinci versiyonunun üzerinde gerçekleştirilmesi ile hem kullanıcılardan ikinci versiyon hakkında geri dönütler toplanmış, hem de ilk aşamada CodeCarbon kütüphanesi ile toplanan karbon salınımı ve enerji tüketimi verileri ikinci versiyon için de elde edilmiştir. Bu şekilde uygulamanın hem kullanıcı görüşleri olmayan versiyonu için hem de kullanıcı görüşlerinin uygulamaya dahil edildiği versiyonu için kod karbon salınımı ve enerji tüketimi verileri elde edilmiştir. Elde edilen bu değerlerin her iki versiyon için kıyaslanması yapılarak kullanıcı deneyiminin yeşil yazılım üzerine etkisi araştırılmıştır.

ARAŞTIRMA BULGULARI

Tez çalışmasının bu aşamasında tez uygulamasının hem ilk versiyonunda hem de ikinci versiyonunda CodeCarbon kütüphanesinden elde edilen çıktılar ve kullanıcı testlerinden toplanan kullanıcı görüşleri verileri ifade edilecektir.

İlk Versiyon CodeCarbon Verileri

Uygulamanın ilk versiyonunda gerçekleştirilen kullanıcı testleri esnasında 12 deney kullanıcısından, deney sonuçlarını içeren 12 adet .csv uzantılı dosya elde edilmiştir. Bu dosyalarda bulunan veriler düzenlenerek tek bir dosya içerisinde toplanmıştır. Bu dosya içerisinde toplamda 25 sütundan ve 12 satırdan oluşan toplamda 300 adet veri bulunmaktadır.

Yazılım ürünlerinin enerji tüketimleri ve buna bağlı olarak karbon salınımı değerlerini yalnızca kullanılan yazılım etkilememektedir. Yazılımın üzerinde çalıştığı bilgisayar veya cihazın donanım bileşenlerinin, uygulamaların tükettiği enerji miktarına ve karbon emisyonuna büyük oranda etkisi bulunmaktadır. Tez çalışması kapsamında yalnızca yazılım uygulamasının kullanılabilirliği üzerine karbon salınımı ile enerji tüketimi değerlerine odaklanıldığı için donanım bileşenlerinin, deney çalışmasının her kullanıcısı için uygulamanın hem ilk versiyonunda hem de ikinci versiyonunda sabit tutulması hedeflenmiştir. Bundan dolayı deney kullanıcılarının tamamı için kullanıcı testleri aynı bilgisayar üzerinde gerçekleştirilmiştir. CodeCarbon kütüphanesinin, uygulamanın donanım bileşenlerini ifade ettiği bazı çıktıları da her kullanıcı için aynı değeri üreterek bu durumu teyit etmektedir. Bu sabitler aşağıdaki gibidir.

İlk değer *project_name* değeridir. Bu değer, kütüphane tarafından her çalışmada varsayılan değer olarak “codecarbon” değerini üretmektedir ve tüm deney kullanıcılarında bu değer aynı olacak şekilde üretilmiştir. Bazı diğer değerler *country_name*, *country_iso_code* ve *region* değerleridir. İlk değer çalışılan ülkenin hangi ülke olduğunu, ikinci değer çalışılan ülkenin ISO kodunu, üçüncü değer çalışılan cihazın ülkenin hangi bölgesinde olduğunu ifade etmektedir ve çalışmada bu değerler her kullanıcı için “Türkiye”, “TUR” ve “Erzurum” şeklinde üretilmiştir. Diğer değerlerden *on_cloud*, *cloud_provider* ve *cloud_region* değerleridir. İlk değer çalışanın bir bulut sağlayıcısı üzerinde mi yoksa özel bir altyapıda mı çalıştığını, ikinci değer eğer bulut sağlayıcısı üzerinde çalışılıyorsa hangi sağlayıcı üzerinde çalışıldığı, üçüncüsü ise bulut sağlayıcısı kullanılıyorsa coğrafi bölgesini belirtmektedir. Tez

alışmasında yerel bir bölgede lokal bir cihaz kullanıldığı için bu değ erler sırasıyla “N”, “boş”, “boş” olacak şekilde  retilmiřtir.

Diğ er değ erler ise *os*, *python_version*, *codecarbon_version*, *cpu_count*, *cpu_model*, *gpu_count*, *gpu_model*, *longitude*, *latitude*, *ram_total_size* değ erleridir. İlk değ er olan *os* alışılan cihazın kullandığı iřletim sistemini ifade etmektedir. alışmada bu değ er her kullanıcı için “Windows-10-10.0.19045-SP0” şeklinde  retilmiřtir. İkinci ve     nc  değ erler olan *python_version* ve *codecarbon_version* değ erleri uygulama gerekleřtirilirken python ve CodeCarbon k t phanesinin versiyon değ erleridir. alışmada bu değ erler her kullanıcı için “3.10.6” ve “2.2.3” şeklinde  retilmiřtir. D  rd nc  ve beřinci değ erler olan *cpu_count* ve *cpu_model* değ erleri,  zerinde alışılan donanımın iřlemci bileřeninin cpu sayı ve model değ erlerini ifade etmektedir. alışmada bu değ erler her kullanıcıda aynı bilgisayar kullanıldığı için sabit olarak “16” ve “Intel(R) Core (TM) i7-10700F CPU @ 2.90GHz” şeklinde  retilmiřtir. Altıncı ve yedinci değ er olarak  retilen *gpu_count* ve *gpu_model* değ erleri alışılan donanımın grafik iřlem biriminin sayısı ve model değ erlerini ifade etmektedir ve alışma için her kullanıcıda aynı bilgisayar kullanıldığı için yine sabit olarak “1” ve “1 x NVIDIA GeForce RTX 3050” değ erleri  retilmiřtir. Sekizinci ve dokunc  değ erler ise *longitude* ve *latitude* değ erleridir. Bu değ erler boylam ve enlem değ erleridir ve alışmanın yapıldığı alanın 11.1 km / 123 km coğrafi koordinatlarını vermektedir. alışmada bu koordinatlar her kullanıcı için sırasıyla “41.1149” ve “40.24” şeklinde  retilmiřtir. Son değ er olan *ram_total_size* değ eri ise kullanılabilir toplam ram alanını ifade etmektedir ve bu değ erlerde her kullanıcı için aynı sonuları vermiřtir.

Bunlar dıřında kalan diğ er bileřenler kullanıcıların uygulamayı kullanma durumlarına baėlı olarak farklılık g stermektedir. Bunlardan ilki *timestamp* değ eridir. Bu değ er k t phanenin hangi yıl, ay, g n, saat, dakika ve saniye anından alışmaya bařladığı, yani testin hangi zaman diliminde gerekleřtiğini ifade etmektedir. Bir diğ er değ er ise *run_id* değ eridir ve bu değ erde deney kullanıcıları için her alışmada birbirinden benzersiz  retilen alışma kimliği olarak ifade edilmektedir.

Tablo 2. Timestamp ve run_id Değerleri

Kullanıcı	timestamp	run_id
Kullanıcı 1	2023-06-13T11:18:20	856a6165-9d98-46b8-bdb4-c4bf42434d4d
Kullanıcı 2	2023-06-14T10:45:04	925e0aac-553d-4b6a-aedb-69d8e3b3c1ef
Kullanıcı 3	2023-06-22T12:05:42	89d739da-b005-4860-b211-06657d821817
Kullanıcı 4	2023-06-22T10:57:30	923ba3b4-fd22-404a-a433-50b47f4e4975
Kullanıcı 5	2023-06-13T15:52:29	cece0e80-3f9f-4718-9e8a-877837f9471f
Kullanıcı 6	2023-06-21T16:22:54	f11f1f8c-ef78-4749-9787-ce935e430f19
Kullanıcı 7	2023-06-15T11:14:50	3f4126c9-b599-4c1b-9930-a601788c7869
Kullanıcı 8	2023-06-21T14:48:40	ccd4f2b9-d7b8-4430-8548-071f3973afbe
Kullanıcı 9	2023-07-07T14:53:36	f81f12f5-ba1a-4653-b1b4-58f906b48c4d
Kullanıcı 10	2023-07-19T13:41:36	806464bd-3951-4821-b654-61010d587ebc
Kullanıcı 11	2023-07-22T18:06:41	14913853-d872-4c62-8107-cc3ce59de226
Kullanıcı 12	2023-07-23T13:04:51	1a9a7bb7-2ff6-40ad-8ed8-ceb24fd2b5fb

Yukarıda verilen tablo, ilk testlerden elde edilen timestamp ve run_id değerlerini içermektedir. Fakat bu değerler çalışma kapsamında, bir anlam ifade etmediği için yalnızca örnek olması amacıyla gösterilmiştir.

Sonraki değer ise uygulamanın çalışma süresini belirten duration değeridir. Bu değer her deney kullanıcısının uygulama içerisinde ne kadar vakit geçirdiğini yani uygulamadaki görevleri ne kadar sürede yaptığını ifade etmektedir. Kütüphanenin ürettiği .csv uzantılı dosya içerisinde duration değeri saniye cinsinden verilmektedir. Çalışma kapsamından zaman değerlerini daha kolay ifade edilebilir ve analiz edilebilir hale getirilmesi amacıyla saniye türü dakikaya türüne dönüştürülmüştür. Aşağıdaki tabloda, hem kütüphanenin saniye cinsinden çıktı olarak verdiği değerler hem de bu değerlerin yaklaşık olarak dakika cinsine dönüştürülmüş hali verilmiştir.

Tablo 3. Süre Değerleri

Kullanıcı	duration (Saniye)	duration (dakika)
Kullanıcı 1	759,1077857017517	~12
Kullanıcı 2	463,0734486579895	~7
Kullanıcı 3	372,5622820854187	~6
Kullanıcı 4	332,2844922542572	~5
Kullanıcı 5	460,5994517803192	~7
Kullanıcı 6	497,9537744522095	~8
Kullanıcı 7	366,2096016407013	~6
Kullanıcı 8	415,2919387817383	~7
Kullanıcı 9	482,0758965015411	~8
Kullanıcı 10	387,3624060153961	~6
Kullanıcı 11	421,7966470718384	~7
Kullanıcı 12	311,8026537895202	~5

Kullanım süresi değerlerinden sonra çıktı olarak üretilen diğer iki değer, emisyon salınımını karbon kilogram eşdeğeri cinsinden ifade eden *emissions* değeridir.

Tablo 4. Emisyon Değerleri

Kullanıcı	emissions ([CO ₂ eq])
Kullanıcı 1	0,00500215958938071
Kullanıcı 2	0,00305914533088215
Kullanıcı 3	0,00255536423502023
Kullanıcı 4	0,00232740560636082
Kullanıcı 5	0,00310923282678348
Kullanıcı 6	0,00332926723915766
Kullanıcı 7	0,00247278709205788
Kullanıcı 8	0,00272117529917976
Kullanıcı 9	0,00325085603015094
Kullanıcı 10	0,00257965630415376
Kullanıcı 11	0,00282449943516659
Kullanıcı 12	0,00210941819313291

Her kullanıcı için farklı üretilen değerlerden diğer üçü ise CPU, GPU ve RAM bileşenlerinin uygulamanın kullanımı esnasında yaptıkları enerji tüketimlerini ifade eden *cpu_energy*, *gpu_energy* ve *ram_energy* değerleridir. Bu değerler kWh cinsinden üretilmektedir ve çalışmada elde edilen sonuçlar aşağıdaki tablodaki gibidir.

Tablo 5. cpu_energy, gpu_energy ve ram_energy Değerleri

Kullanıcı	cpu_energy	gpu_energy	ram_energy
Kullanıcı 1	0,00685279370596011	0,00343528732085936	0,00125555953336086
Kullanıcı 2	0,00418039791285991	0,00211346516074942	0,00076582254618303
Kullanıcı 3	0,00336331732786364	0,00191752379485295	0,00061625306277016
Kullanıcı 4	0,00299969898164272	0,00182170111451387	0,00054962681282376
Kullanıcı 5	0,00415805821882353	0,00225541693793998	0,00076179894814474
Kullanıcı 6	0,00449522906707392	0,00236436345930178	0,00082346189260121
Kullanıcı 7	0,00330594218820333	0,00179494595760666	0,00060564017711123
Kullanıcı 8	0,00374903977182176	0,00184388899186458	0,00068681281466288
Kullanıcı 9	0,00435183063033554	0,00235328242310121	0,00079698935941416
Kullanıcı 10	0,00349684707820414	0,00181537146727831	0,00064093520978212
Kullanıcı 11	0,00380771648751364	0,00201256692879882	0,00069790251106460
Kullanıcı 12	0,00281475815971692	0,00153726935216851	0,00051594300715312

Her kullanıcı için farklı olarak üretilen ve tez çalışması kapsamında anlam ifade eden son değer ise yapılan toplam enerji tüketimini ifade eden energy_consumed değeridir. Bu değer CPU, GPU ve RAM bileşenlerinin ayrı ayrı tükettiği enerji değerlerinin toplamı olacak şekilde üretilmektedir. Değer, toplamı oluşturan diğer bileşenler gibi kWh türünden değerler üretmektedir.

Tablo 6. Toplam Enerji Tüketimi Değerleri

Kullanıcı	energy_consumed
Kullanıcı 1	0,01154364056018035
Kullanıcı 2	0,00705968561979237
Kullanıcı 3	0,00589709418548676
Kullanıcı 4	0,00537102690898035
Kullanıcı 5	0,00717527410490827
Kullanıcı 6	0,00768305441897693
Kullanıcı 7	0,00570652832292043
Kullanıcı 8	0,00627974157834924
Kullanıcı 9	0,00750210241285070
Kullanıcı 10	0,00595315375526456
Kullanıcı 11	0,00651818592737707
Kullanıcı 12	0,00486797051903855

Çalışmanın ilk aşamasında CodeCarbon kütüphanesinden çıktı olarak alınan değerler yukarıda belirtildiği gibidir.

Kullanıcı Testi Sonucunda Kullanıcılardan Toplanan Geri Dönütler

Kullanıcı testlerinde yapılan görüşmelerde deney kullanıcılarına test soruları yöneltilerek cevapları alınmıştır ve uygulamanın ikinci versiyonu oluşturulmuştur. Bu aşamada yapılan görüşmelerde kişilerin kabul beyanı dikkate alınarak ses kaydı alınmıştır ve alınan ses kayıtları daha sonra Word belgesi formatında yazılı hale getirilmiştir. Her kullanıcının belgesi analiz edildikten sonra ortak bir dosya altında görüşler toplanmıştır.

Uygulamanın giriş, kayıt olma ve ana sayfaları için kullanıcılardan genel anlamda ortak olarak; giriş sayfasında bulunan ve kayıt ol sayfasına yönlendiren üye ol bağlantısının belirgin olmadığı, menü alanlarında, yalnızca yazı alanlarına link verildiği için tıklarken sorun yaşadıkları yönünde görüşler gelmiştir:

“Gözle görülemeyecek kadar küçük ya da bütün sayfanın ebatına göre çok küçük bir üye değilseniz üye olun yazısı var.” (Kullanıcı 1)

“Fonksiyonel olarak mesela menülere tıklamak istediğimde düz resim olarak zannettim ilk çünkü yüzde onluk kısmı sadece klik için ayrılmış tıklama için sadece yazı kısmı” (Kullanıcı 3)

Bu sayfalarda kullanıcıların görüşler dahilinde düzenlemeler yapılmıştır ve aynı zamanda bu sayfalar için kullanıcıların harcadıkları zaman ve çaba bileşenleri yeterli seviyede, arayüz tasarımı ise anlaşılır şekilde bulmuşlardır.

Deney kullanıcıları genel olarak bölümlerimiz, doktorlarımız, müsait randevu durumu ve online randevu sayfalarının birbirlerine bağlanmaları ile ilgili geri dönüşlerde bulunmuşlardır. Bölümlerimiz sayfasını ziyaret eden kullanıcıların bu sayfadan doktorlar sayfasına yönlendirilmesi, doktorlar sayfasından ise randevu oluşturma sayfasına yönlendirilmesi yönünde fikirler belirtilmiştir:

“Bölümlerimiz sayfasına geldiğimde yazılara bütüncül tıkladığımda acil servisle ilgili bir şeye gitmeli ya da randevu vermeliydi bana bence ya da bir sayfaya gitmeliydi.” (Kullanıcı 2)

“Doktorlar sayfasında da direkt doktorlara burada erişimi hangi doktorun olduğunu falan ayrı ayrı bölümler için görebiliyorum. Fakat belki şu olabilirdi, benim tavsiyem mesela acil servise geldiğimde herhangi bir doktora tıkladığım zaman arka planda randevu alma

sistemine yönlendirilebilirdim. Buraya bağlamak benim için daha kullanışlı ve kolay olabilirdi.” (Kullanıcı 6)

“Randevu alma işlemimi tamamladıktan sonra uygulama randevunuz oluşturuldu ve benzeri bir geri dönüş vermedi bana. Bu durum oluşturduğum randevunun sisteme doğru bir şekilde kaydedilip kaydedilmediği konusunda beni şüpheye düşürdü.” (Kullanıcı 7)

“Müsait randevu durum sayfası içerisindeki müsait gün ve saatlerin belirtildiği alanlar benim için kafa karıştırıcı geldi. Doktorların hepsi için bu alanın var olması karışık duruyor. Bu alan bence yeni randevu alma sayfasının içerisindeki adımlara dahil edilebilir. Bu benim için daha az kafa karıştırıcı olurdu.” (Kullanıcı 8)

“Yeni randevu oluşturma sayfasında doktor seçerken tüm doktorların isimlerini görmek kafamı karıştırdı. Bu alanda yalnızca seçtiğim bölüme ait doktorları görmek isterim. Bu yaptığım işlemi benim için daha kolaylaştıracaktır.” (Kullanıcı 2)

“Randevu oluşturma sayfası için bölüm, doktor, tarih seçimi alanlarının tasarım olarak bir akış şeması gibi adım adım ilerlemesi tasarımsal olarak beni daha mutlu ederdi. Bu sayfada seçim alanları birazda küçük geldi bana sayfaya oranla ondan dolayı alanlar biraz daha büyütülebilir.” (Kullanıcı 11).

“Bu sayfalarda giriş yapacağımız metin kutularının içerisinde içeriğin formatına örnekler yerleştirilebilir bu açıklayıcı olur. Mesela tarih girilecek alanlar için tarih formatının ne olduğuna dair örnekler verilebilir.” (Kullanıcı 9)

Elde edilen görüşler doğrultusunda bu alanlar için tekrar düzenleme işlemi yapılmıştır. Ayrıca deney kullanıcıları genel olarak bölümlerimiz, doktorlarımız ve müsait randevu durumu sayfaları için harcadıkları çabayı az düzey olarak, zamanı ise orta düzey olarak ifade etmişlerdir.

Check-up randevu alma sayfası için ise kullanıcılar, cinsiyet seçme işleminin anlaşılır olmadığı, bu sayfada yapılan işlemler ile ilgili uygulamadan herhangi bir bilgilendirme yapmaması durumunun kafa karıştırıcı olabileceği yönünde geri dönüşler yapmışlardır.

Ayrıca kullanıcı profili sayfası ile ilgili ise bilgi girişi alanlarında gruplandırma yapılması, gereksiz olarak görülen bazı veri girişi alanlarının kaldırılması, bazı veri girişi alanlarının ise eklenmesi gerektiğine ve profil sayfasına gitmek için sol menüye de bir alan eklenmesi gerektiğine dair geri dönüşler toplanmıştır.

İkinci Versiyon CodeCarbon Verileri

Uygulamanın ikinci versiyonu olarak bahsedilen bu sistem, kullanıcılar ile yapılan testler ve görüşmeler doğrultusunda uygulamanın tekrardan revize edilmesi ile elde edilen uygulamadır. İkinci aşamada, yeni versiyonlanan bu uygulama için de kullanıcılara test uygulanmıştır ve karbon salınımı ile enerji tüketimi ölçümleri yapılmıştır. İkinci aşamada yapılan kullanıcı testlerinde, deney kullanıcılarından 6 tanesi önceki aşamada da deney yapılan kişiler ve diğer 6 tanesi ise uygulamayı ilk kez kullanacak kişiler olacak şekilde seçilerek toplamda 12 kullanıcı testi gerçekleştirilmiştir. İkinci aşamada elde edilen süre değerleri aşağıdaki tablodaki gibidir.

Tablo 7. İkinci versiyon süre değerleri

Kullanıcı	duration (Saniye)	duration (dakika)
Eski Kullanıcı 4	254,5471339225769	~4
Eski Kullanıcı 7	283,0919458866119	~4
Eski Kullanıcı 8	282,7436430454254	~4
Eski Kullanıcı 3	327,5248432159424	~5
Eski Kullanıcı 11	370,1151063442232	~6
Eski Kullanıcı 12	275,7316491603851	~4
Yeni Kullanıcı 1	305,1301069259643	~5
Yeni Kullanıcı 2	303,7293140888214	~5
Yeni Kullanıcı 3	294,7672357559204	~4
Yeni Kullanıcı 4	300,4650838375092	~5
Yeni Kullanıcı 5	311,6463627815246	~5
Yeni Kullanıcı 6	290,7318954467773	~5

Tablo 3’de gösterildiği gibi ilk aşama kullanıcılarının kullanım sürelerinin ortalaması 7 dakika, ikinci aşama kullanıcılarının ise Tablo 7’de gösterildiği gibi kullanım sürelerinin ortalaması 4,6 dakikadır. Hem ilk hem de ikinci aşamada teste tabi tutulan kullanıcılarının ikinci aşama kullanım süresi ortalamaları 4,5 iken, yalnızca ikinci aşamada teste tabi tutulan kullanıcıların kullanım süresi ortalamaları 4,8 dir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcılarının ilk aşama kullanım süresi ortalamaları ise 6 dakikadır. Karbon emisyonu değeri ise ikinci versiyon için aşağıdaki tablodaki gibidir.

Tablo 8. İkinci versiyon emissions değerleri

Kullanıcı	emissions ([CO₂eq])
Eski Kullanıcı 4	0,0016886992681877
Eski Kullanıcı 7	0,0019217767663885
Eski Kullanıcı 8	0,0019729901267435
Eski Kullanıcı 3	0,0022676371713559
Eski Kullanıcı 11	0,0024732395746525
Eski Kullanıcı 12	0,0018235870058119
Yeni Kullanıcı 1	0,0020849329776995
Yeni Kullanıcı 2	0,0020819149340261
Yeni Kullanıcı 3	0,0020653933703622
Yeni Kullanıcı 4	0,0020742086052557
Yeni Kullanıcı 5	0,0021837007637311
Yeni Kullanıcı 6	0,0020725304355850

Tablo 4’e verilen ilk aşama kullanıcılarının karbon salınımı ortalaması 0,002945080598452 CO₂eq, Tablo 8’de verilen ikinci aşama karbon salınımı ortalaması ise 0,002059217583317 CO₂eq şeklindedir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcıların ikinci aşama karbon salınım ortalamaları 0,002024654985523 CO₂eq iken, yalnızca ikinci aşamada teste tabi tutulan kullanıcıların karbon salınım ortalamaları 0,00209378018111 CO₂eq dir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcılarının ilk aşama karbon salınımı ortalamaları ise 0,00250177497682 CO₂eq şeklindedir. Yine ikinci aşamada elde edilen ve uygulama için diğer dikkate değer değerler olan CPU enerji, GPU enerji ve RAM enerji değerleri de aşağıdaki gibidir.

Tablo 9. İkinci versiyon cpu_energy, gpu_energy ve ram_energy değerleri

Kullanıcı	cpu_energy	gpu_energy	ram_energy
Eski Kullanıcı 4	0,0022979135405686	0,0011779470521535	0,0004212036687985
Eski Kullanıcı 7	0,0025556190106603	0,0014108650089834	0,0004684604941606
Eski Kullanıcı 8	0,0025524569539566	0,0015328303714162	0,0004678438662561
Eski Kullanıcı 3	0,0029567219596775	0,0017344663928651	0,0005419090645432
Eski Kullanıcı 11	0,0033411804599894	0,0017540113342032	0,0006123807370021
Eski Kullanıcı 12	0,0024891899115509	0,0012629391882485	0,0004562198241329
Yeni Kullanıcı 1	0,0027545235936840	0,0015520689094056	0,0005048727163976
Yeni Kullanıcı 2	0,0027418597706491	0,0015600460129975	0,0005025946018048
Yeni Kullanıcı 3	0,0026609746136599	0,0016177078049293	0,0004876906362521
Yeni Kullanıcı 4	0,0027124248781137	0,0015772174784552	0,0004970738922961
Yeni Kullanıcı 5	0,0028134013512068	0,0017102880428456	0,0005157053427764
Yeni Kullanıcı 6	0,0026245302415556	0,0016772226879762	0,0004810905545540

Tablo 5’de verilen ilk aşama kullanıcılarının CPU enerji tüketimi ortalaması 0,003964635794168 kWh, Tablo 9’da verilen ikinci aşama kullanıcılarının CPU enerji tüketimi ortalaması ise 0,002708399690439 kWh şeklindedir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcıların ikinci aşama CPU enerji tüketimi ortalaması 0,002698846972734 kWh, yalnızca ikinci aşamada teste tabi tutulan kullanıcıların CPU enerji tüketimi ortalaması 0,002717952408145 kWh dir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcılarının ilk aşama CPU enerji tüketimi ortalaması ise 0,00334007881946 kWh şeklindedir.

Tablo 5’de verilen ilk aşama kullanıcılarının GPU enerji tüketimi ortalaması 0,002105423575753 kWh, Tablo 9’da verilen ikinci aşama kullanıcılarının GPU enerji tüketimi ortalaması ise 0,00154730085704 kWh şeklindedir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcıların ikinci aşama GPU enerji tüketimi ortalaması 0,002698846972734 kWh, yalnızca ikinci aşamada teste tabi tutulan kullanıcıların GPU enerji tüketimi ortalaması 0,001615758489435 kWh dir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcılarının ilk aşama GPU enerji tüketimi ortalaması ise 0,00334007881946 kWh şeklindedir.

Yine Tablo 5’de verilen ilk aşama kullanıcılarının RAM enerji tüketimi ortalaması 0,000726395489589 kWh, Tablo 9’da verilen ikinci aşama kullanıcılarının RAM enerji tüketimi ortalaması ise 0,000496420449915 kWh şeklindedir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcıların ikinci aşama RAM enerji tüketimi ortalaması 0,000494669609149 kWh, yalnızca ikinci aşamada teste tabi tutulan kullanıcıların RAM enerji

tüketimi ortalaması 0,00049817129068 kWh dir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcılarının ilk aşama RAM enerji tüketimi ortalaması ise 0,000612029730931 kWh şeklindedir.

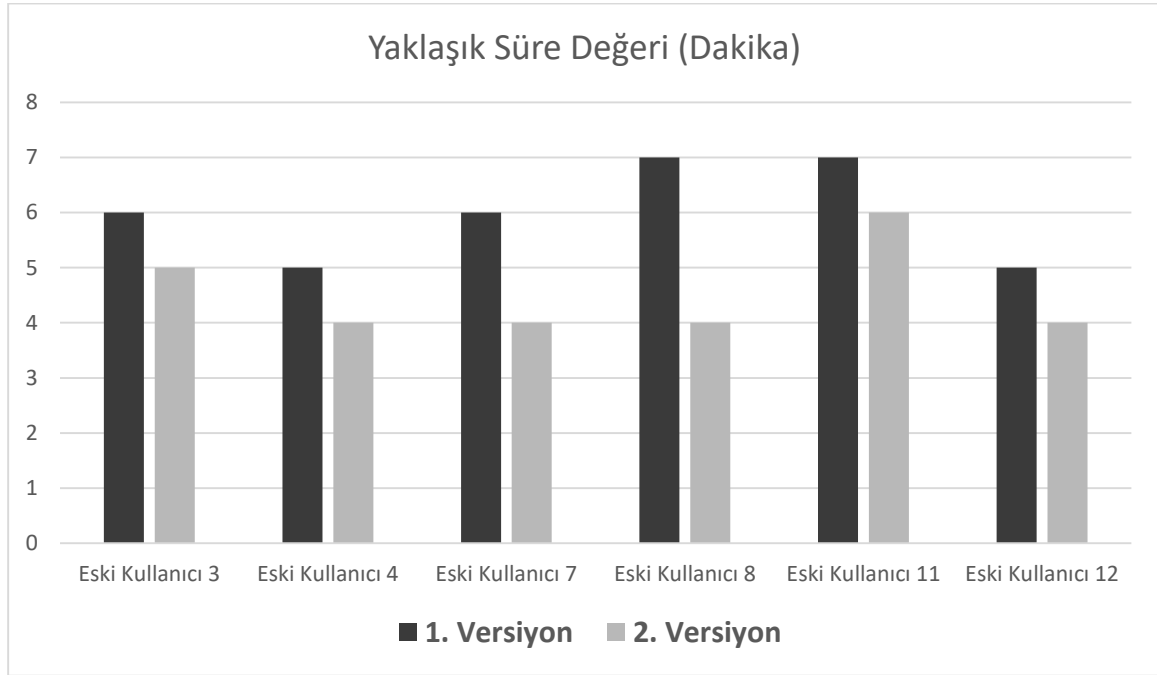
Her iki aşama için de önemli olan bir diğer değer, toplam enerji tüketimi değerini ifade eden energy_consumed değeridir ve bu değer ikinci aşamada aşağıdaki tablo gibidir.

Tablo 10. İkinci versiyon toplam enerji tüketimi değerleri

Kullanıcı	energy_consumed
Eski Kullanıcı 4	0,0038970642615207
Eski Kullanıcı 7	0,0044349445138045
Eski Kullanıcı 8	0,0045531311916283
Eski Kullanıcı 3	0,0052330974170853
Eski Kullanıcı 11	0,0057075725311948
Eski Kullanıcı 12	0,0042083489239324
Yeni Kullanıcı 1	0,0048114652194873
Yeni Kullanıcı 2	0,0048045003854514
Yeni Kullanıcı 3	0,0047663730548413
Yeni Kullanıcı 4	0,0047867162488651
Yeni Kullanıcı 5	0,0050393947368288
Yeni Kullanıcı 6	0,0047828434840859

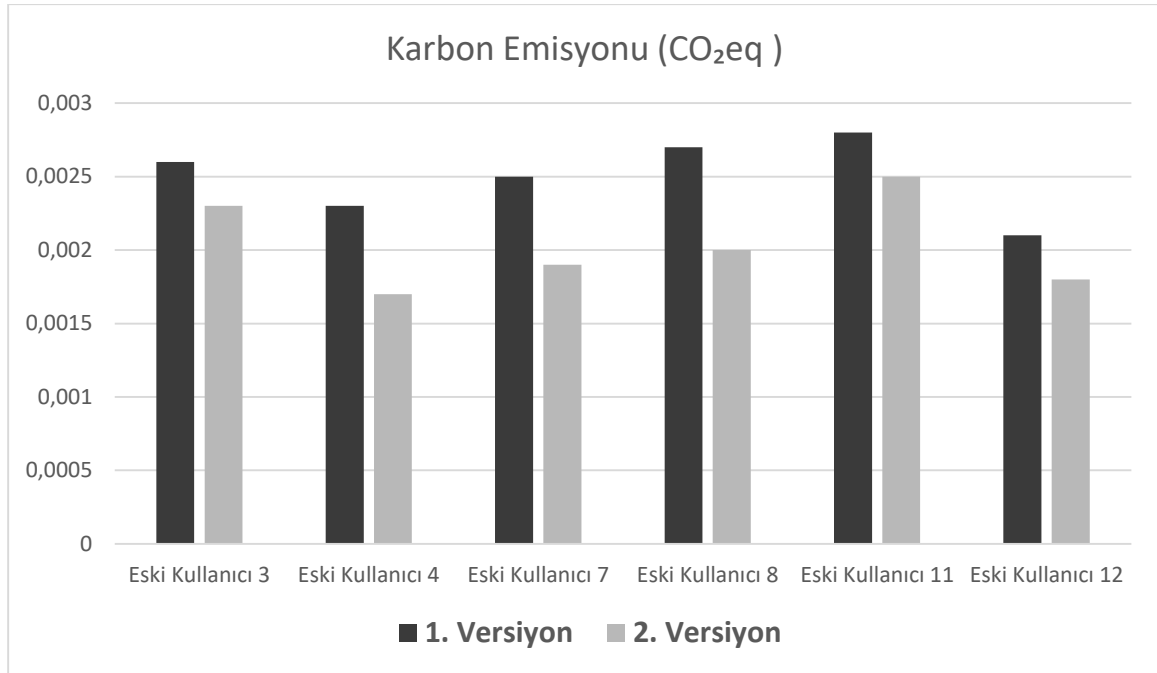
Tablo 6 da verilen ilk aşama kullanıcılarının toplam enerji tüketimi ortalaması 0,00679645485951 kWh, Tablo 10 da verilen ikinci aşama kullanıcılarının toplam enerji tüketimi ortalaması ise 0,004752120997394 kWh şeklindedir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcıların ikinci aşama toplam enerji tüketimi ortalaması 0,004672359806528 kWh, yalnızca ikinci aşamada teste tabi tutulan kullanıcıların toplam enerji tüketimi ortalaması 0,00483188218826 kWh dir. Hem ilk hem ikinci aşamada teste tabi tutulan kullanıcılarının ilk aşama toplam enerji tüketimi ortalaması ise 0,005773424573692 kWh şeklindedir.

Uygulamanın hem ilk versiyonu hem de ikinci versiyonu için kullanıcı testi uygulanan kullanıcılar; kullanıcı 3, kullanıcı 4, kullanıcı 7, kullanıcı 8, kullanıcı 11 ve kullanıcı 12’dir. Bu kullanıcıların her iki aşama için de süre, karbon emisyonu, cpu_energy, gpu_energy, ram_energy ve toplam tüketilen enerji miktarını ifade eden energy_consumed değerlerinin kıyaslandığı grafikler aşağıdaki gibidir;



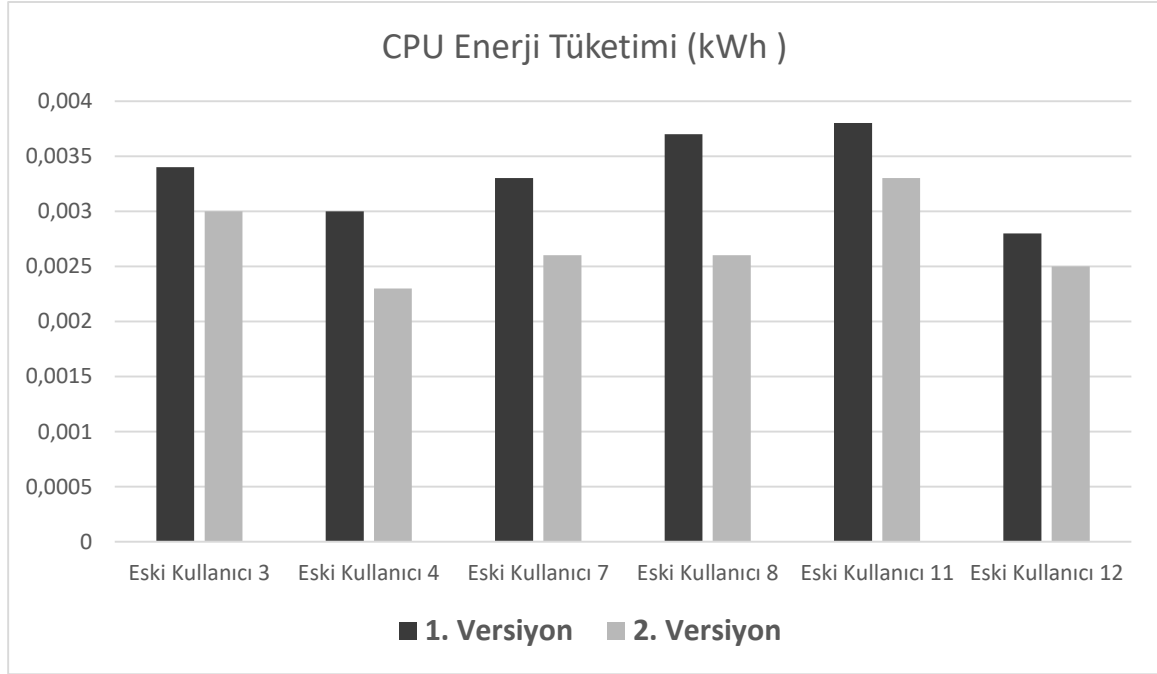
Şekil 21. Her iki versiyonu kullanan kullanıcılar için yaklaşık süre değerleri

Şekil 21’de bulunan grafikte her iki aşama içinde teste tabi tutulan kullanıcılar için süre bileşeninin ortalama değerleri yaklaşık olarak dakika cinsinden verilmiştir. İlk aşama için ortalama değer dakika cinsinden yaklaşık olarak 6 dakikadır. Bu değer ikinci aşama için ortalaması ise yaklaşık olarak 4,5 dakikadır.



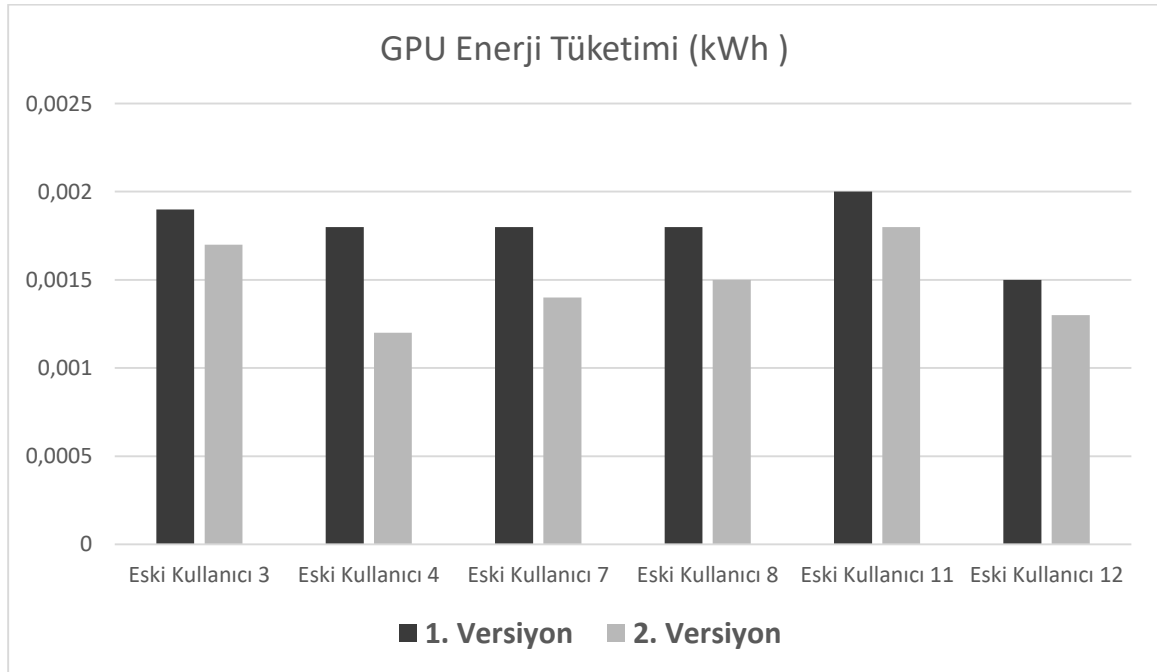
Şekil 22. Her İki versiyonu kullanan kullanıcılar için karbon emisyonu değerleri

Şekil 22’de bulunan grafikte her iki aşama içinde teste tabi tutulan kullanıcılar için karbon emisyonu bileşeninin ortalama değerleri verilmiştir. Aynı kullanıcılar için, yapılan karbon salınımı bileşeninin ilk aşama ortalaması 0,0025 CO₂eq, ikinci aşama ortalaması ise 0,0020 CO₂eq şeklindedir.



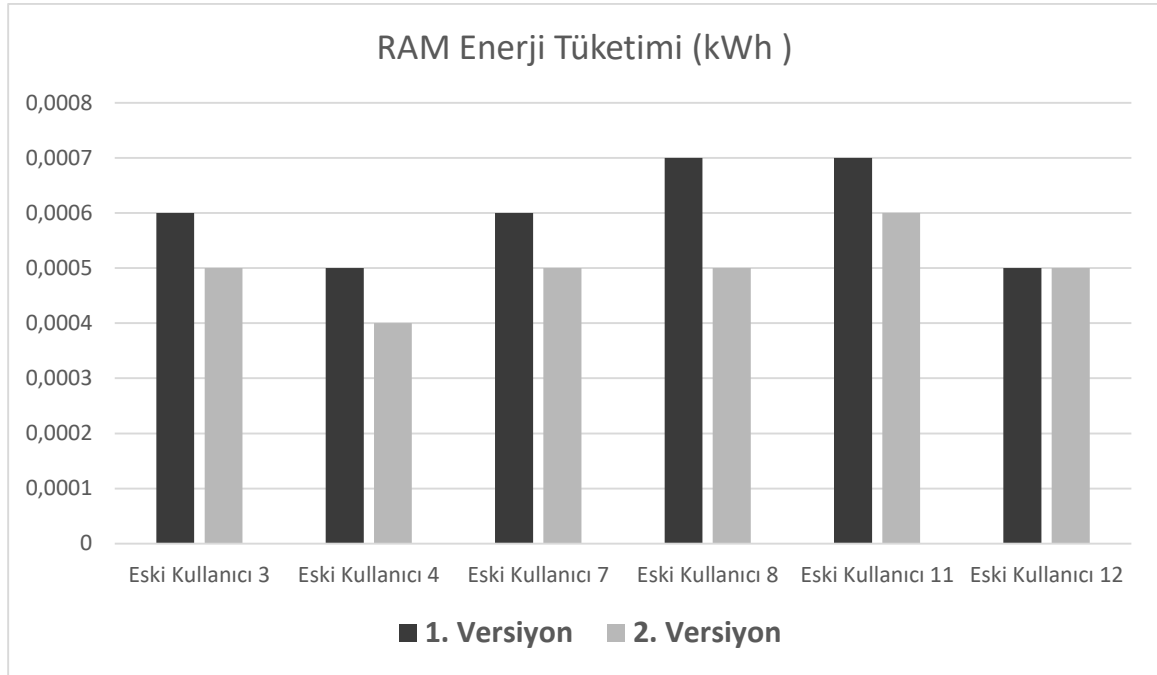
Şekil 23. Her İki versiyonu kullanan kullanıcılar için cpu enerji tüketim değerleri

Şekil 23’de bulunan grafikte her iki aşama içinde teste tabi tutulan kullanıcılar için CPU enerji tüketimi bileşeninin ortalama değerleri verilmiştir. Aynı kullanıcılar için ilk aşama ortalaması 0,0033 kWh, ikinci aşama ortalaması ise 0,0027 kWh şeklindedir.



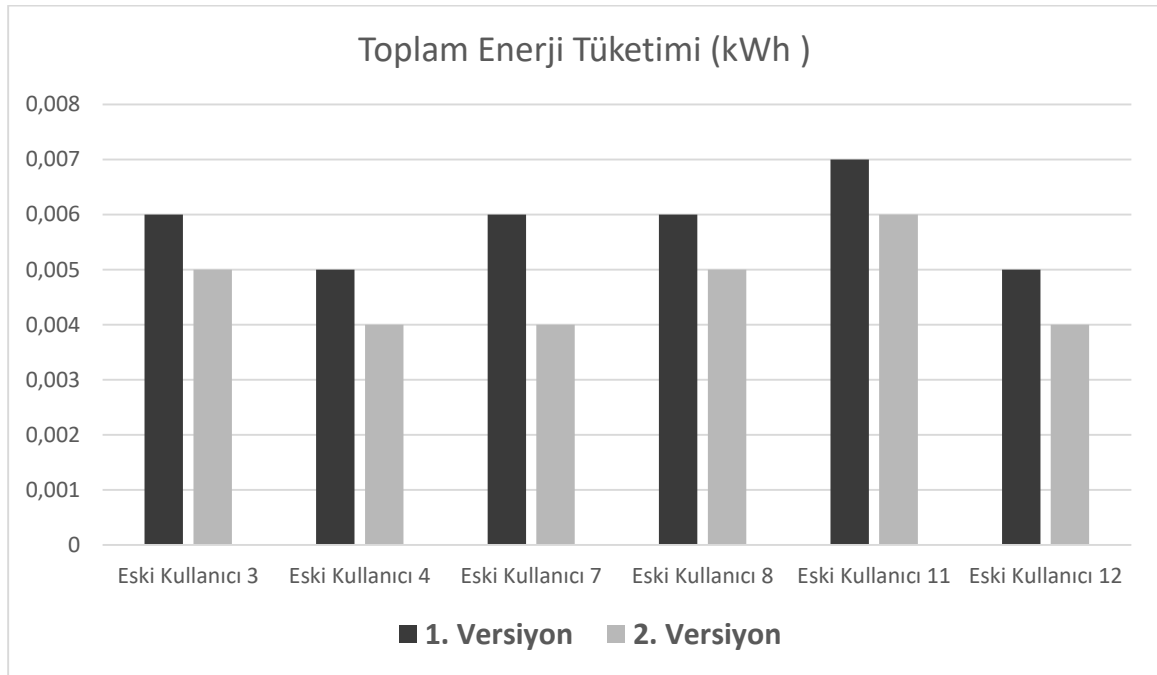
Şekil 24. Her İki versiyonu kullanan kullanıcılar için gpu enerji tüketim değerleri

Şekil 24’de bulunan grafikte her iki aşama içinde teste tabi tutulan kullanıcılar için GPU enerji tüketimi bileşeninin ortalama değerleri verilmiştir. Aynı kullanıcılar için, ilk aşama ortalaması 0,0018 kWh, ikinci aşama ortalaması ise 0,0015 kWh şeklindedir.



Şekil 25. Her İki versiyonu kullanan kullanıcılar için ram enerji tüketim değerleri

Şekil 25’de bulunan grafikte her iki aşama içinde teste tabi tutulan kullanıcılar için RAM enerji tüketimi bileşeninin ortalama değerleri verilmiştir. Aynı kullanıcılar için, ilk aşama ortalaması 0,0006 kWh, ikinci aşama ortalaması ise 0,0005 kWh şeklindedir.



Şekil 26. Her iki versiyonu kullanan kullanıcılar için toplam enerji tüketim değerleri

Şekil 26'da bulunan grafikte her iki aşama içinde teste tabi tutulan kullanıcılar için toplam enerji tüketimi bileşeninin ortalama değerleri verilmiştir. Aynı kullanıcılar için, ilk aşama ortalaması 0,006 kWh, ikinci aşama ortalaması ise 0,005 kWh şeklindedir.

Aynı zamanda çalışma verilerinin temelini oluşturan karbon salınımı ve süre değerleri için her iki grupta Mann Whitey U testi uygulanmıştır. Bu test t testinin parametrik olmayan karşılığı şeklindedir. Çalışma kapsamında grupların dağılımı normal olmadığından ve örneklem sayısının az olmasından dolayı Mann Whitey U testi uygulanmıştır. Test sonuçlarında ise kullanıcı deneyimi dahil edilmeyen grup ile kullanıcı deneyiminin dahil edildiği grubun değerleri arasındaki fark anlamlı bulunmuştur. Yine test sonuçlarında süre ve salınımı yapılan emisyon değerleri için kullanıcı deneyimi dahil edilmemiş uygulama için olan grubun ortalamaları, kullanıcı deneyiminin dahil edildiği grubun ortalamalarından daha fazla bulunmuştur.

Her iki aşamada da kullanıcı testine tabi tutulan kişilerin ölçümleri incelendiğinde, süre, kod karbon salınımı, cpu enerji, gpu enerji, ram enerji ve toplam enerji tüketimi değerlerinde ilk aşama testlerine göre ikinci aşama testlerindeki değerlerde düşüş gözlenmektedir.

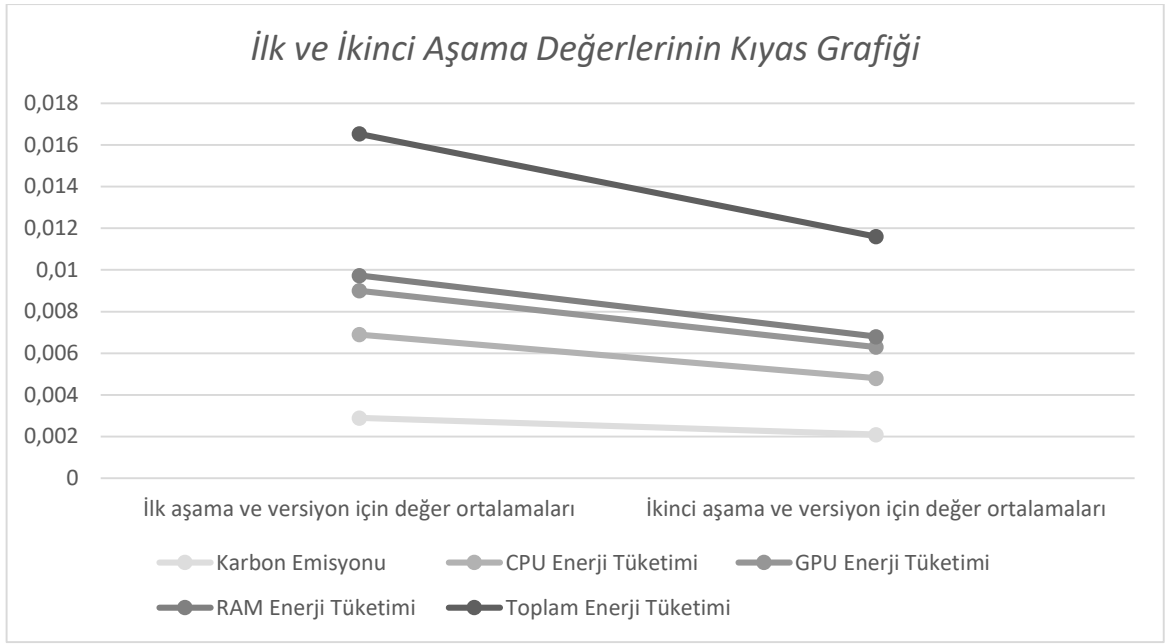
TARTIŞMA

Tez çalışması kapsamında, aynı uygulamanın kullanıcı görüşlerinin dahil edildiği ve edilmediği versiyonları üzerinde karbon salınımına ve enerji tüketimine ait bazı ölçümler yapılmıştır. Yapılan bu ölçümler kapsamında birinci aşamada ilk grup olarak tayin edilen 12 kullanıcı uygulamanın birinci versiyonunu (kullanıcı görüşlerinin dahil edilmediği hali) kullanmıştır. Sonraki adımda ise ikinci grup olarak belirlenen kullanıcılar uygulamanın ikinci versiyonunu (kullanıcı görüşlerinin dahil edildiği hali) kullanmıştır. İlk gruptaki kullanıcıların tamamı uygulamayı ilk kez kullanmışlardır. İkinci grup kullanıcılardan, 6 kişi uygulamanın ikinci versiyonunu ilk kez ve diğer 6 kişi ise uygulamanın ilk versiyonunu da kullanarak geri dönüşler veren kullanıcılardan olacak şekilde seçilmişlerdir.

CodeCarbon kütüphanesi ile arka planda yapılan ölçümler incelendiğinde, yapılan uygulamanın kullanım süresini ifade eden (görevlerin tamamlanma süresi) değerlerinin ortalaması birinci aşamada 7 dakikadır. İkinci aşamada ise bu değerlerin ortalaması 4,6 dakikaya düşmüştür.

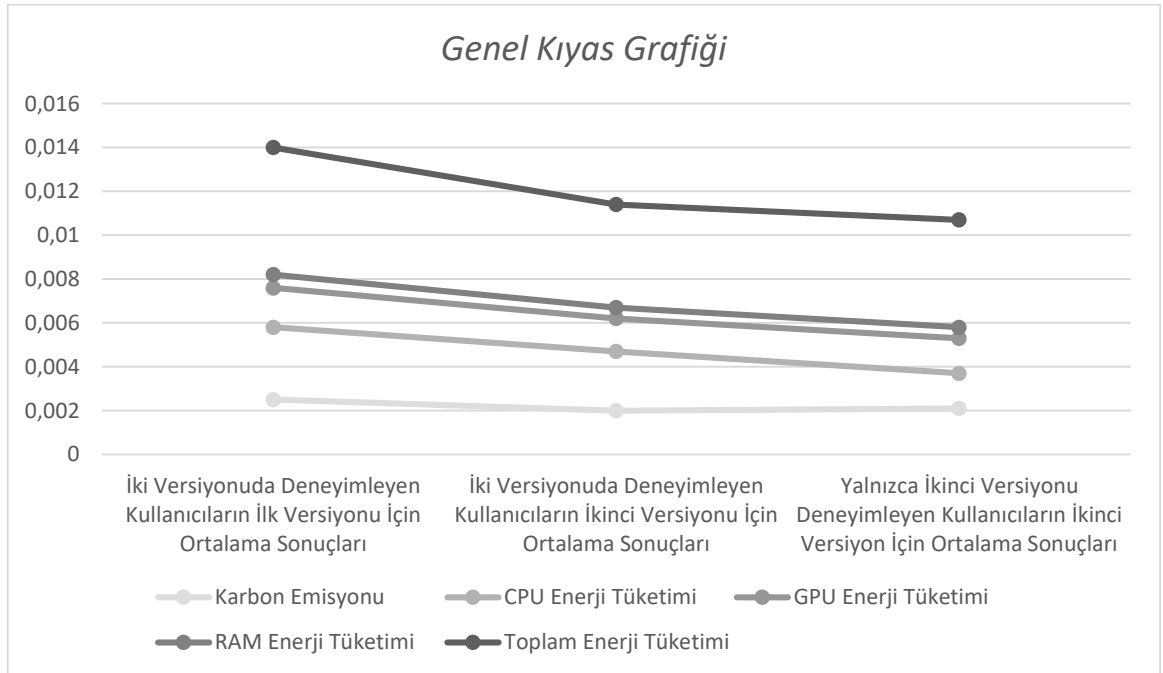
Her iki versiyonu da kullanan kişiler için birinci aşamada ortalama kullanım süresi 6 dakika iken ikinci aşamada bu süre 4,5 dakikaya düşmüştür. Uygulamanın ikinci versiyonunu ilk kez ikinci aşamada kullanan kişilerin ortalama kullanım süreleri ise 4,8 dakikadır ve bu değer, uygulamanın birinci versiyonunu ilk kez kullanan kişilerin kullanım süresi olan 6 dakikadan daha az olduğu görülmektedir. Sonuçlar incelendiğinde, uygulamanın kullanıcı deneyimi dahil edilmiş versiyonunun, uygulamanın kullanım süresini düşürebildiği gözlemlenmiştir. Diğer önemli değerler için her iki aşamadaki kullanıcı gruplarının ortalama değerleri şekil 27'deki grafikteki gibidir.

Şekil 27 de görüldüğü gibi uygulamanın kullanıcı görüşlerinin dahil edildiği ikinci versiyonuna ait ölçümü yapılan karbon emisyonu, cpu enerji tüketimi, gpu enerji tüketimi, ram enerji tüketimi ve toplam enerji tüketimi değerleri, kullanıcı görüşlerinin dahil edilmediği ilk versiyonun değerlerine kıyasla zaman bileşeninde olduğu gibi daha düşüktür.



Şekil 27. İlk ve ikinci aşama değerlerinin kıyas grafiği

Şekil 28’de bulunan grafikte görüldüğü gibi uygulamanın her iki versiyonunu da (hem kullanıcı deneyimi dahil edilmemiş hali hemde kullanıcı deneyimi dahil edilmiş hali) kullanan kişiler için ikinci versiyondaki ortalama karbon emisyonu, cpu enerji tüketimi, gpu enerji tüketimi, ram enerji tüketimi ve toplam enerji tüketimi değerleri, ilk versiyondaki değerlerin ortalamalarına oranla azalmıştır. Yani uygulamanın kullanıcı deneyimi dahil edilmiş ikinci versiyonunun kullanımı için karbon salınımı ve enerji tüketimi değerleri düşmüştür.



Şekil 28. Genel kıyas grafiği

Yine Şekil 28’de görüldüğü gibi uygulamayı birinci aşamada (kullanıcı deneyimsiz hali) ilk kez kullanan kullanıcıların ortalama karbon emisyonu, cpu enerji tüketimi, gpu enerji tüketimi, ram enerji tüketimi ve toplam enerji tüketimi değerleri, uygulamayı ikinci aşamada (kullanıcı deneyimi dahil edilmiş hali) ilk kez kullanan kişilerin değerlerine kıyasla fazladır. Yani bu noktada da uygulamanın kullanıcı deneyiminin dahil edildiği ikinci versiyonunda ilgili değerlerin düştüğü görülmektedir.

Son olarak uygulamanın yalnızca ikinci versiyonunu kullanan kişiler için ikinci aşamadaki karbon emisyonu, gpu enerji tüketimi ve toplam enerji tüketimi değerlerinin ortalamaları, uygulamanın her iki versiyonunu da kullanan kişilerin ikinci aşamadaki karbon emisyonu, gpu enerji tüketimi ve toplam enerji tüketimi değerlerinin ortalamalarına kıyasla daha fazladır. Uygulamanın yalnızca ikinci versiyonunu kullanan kişiler için ikinci aşamadaki cpu enerji tüketimi ortalama değeri, her iki versiyonu da kullanan kişilerin ikinci aşamadaki cpu enerji tüketimi ortalama değerinden daha az, ram enerji tüketimi ortalama değerinin ise bu iki grup için aynı kaldığı görülmektedir.

Donanım bileşenleri için enerji tüketim değerlerine bakıldığında her iki aşama içinde ise cpu, gpu ve ram bileşenlerinden en çok enerji tüketimi değeri cpu bileşenine ait olduğu görülmektedir.

Yeşil yazılım ve kullanılabilirlik kavramları ile ilgili olarak, literatürde ayrı ayrı yapılan birçok çalışma bulunmaktadır. Bu çalışmalar arasında enerji tüketiminin ve kullanılabilirliğin ortak olarak ele alındığı çalışmalar da bulunmaktadır.

Örneğin web sunucuları ile ilgili yapılan bir çalışmada, web uygulamalarında farklı web sunucularının kullanılması durumunun enerji tüketimleri üzerinde nasıl bir etkiye sahip olduğu incelenmiştir. Çalışma kapsamında bağımlı değişken olarak enerji kullanımı, bağımsız değişken olarak web sunucuları ele alınmıştır. Çalışma sonucunda enerji tüketimlerinin farklı web sunucularına bağlı olarak değişebileceği ve web sunucuları seçilirken web uygulamasının ihtiyaçlarını karşılayacak şekilde seçilmesinin de enerji tüketimine katkı sağlayacağı sonucuna varılmıştır. Aynı zamanda çalışmada, incelemeye alınan web sunucularının çalışma durumlarına göre web sunucularının yüksek seviyede etkililik ve cevap verebilirliğini arttırmak için yapılan tasarım tercihlerinin enerji verimliliğini korumak için yapılan tasarım tercihleriyle de farklı değerlere sebebiyet verebileceği sonucuna varılmıştır (Manotas et al., 2013).

Başka bir çalışmada PHR kayıt tutma işlemlerinin kullanılabilirliğinin enerji verimliliği üzerindeki etkisinin araştırıldığı bir çalışmadır. PHR, hastalara ait sağlık bilgilerinin hastalarca yönetilebildiği internet tabanlı araç dizileridir. Çalışma, farklı kategorilerde 19 farklı PHR grubuna uzmanlar tarafından uygulanmıştır ve GUI’ ler aracılığıyla monitör ve güç kaynağının

güç tüketimi ilişkileri incelenmiştir. Çalışmada PHR ler için oluşturulan görev listesi her PHR yapısına uygulanmış ve bu esnada güç kaynağının enerji ölçümleri anlık olarak alınmıştır. Çalışma sonucunda bazı durumlarda GUI kullanılabilirliğinin daha yüksek olmasının daha fazla güç azaltımı ile ilişkilendirilebileceği sonucuna varılmıştır (García-Berná vd., 2021).

Tez çalışması, genel kapsamda ele alındığında, doğrudan kullanıcı testleri ve kullanıcı deneyimi ile karbon salınımı ve enerji tüketimi ölçümlerine dair, literatüre farklı bir bakış açısı kazandıracak bir çalışma olacağı düşünülmektedir.

SONUÇ VE ÖNERİLER

Tez çalışması kapsamında Django web çatısı ve Python programlama dili kullanılarak, kullanıcıların uygulamaya girerek hastane randevusu alabilecekleri bir uygulama geliştirilmiştir. İlk aşama olarak ifade ettiğimiz aşamada uygulama geliştirilirken herhangi bir kullanıcı görüşü alınmamış ve uygulamaya dahil edilmemiştir. Bu şekilde uygulamanın ilk versiyonu oluşturulmuştur ve uygulamanın çalışma esnasında yaptığı karbon salınımı ve enerji tüketimi değerlerini hesaplayan CodeCarbon kütüphanesi entegre edilmiştir.

Daha sonra oluşturulan bu ilk versiyon ve 12 adet kullanıcı ile kullanıcı testi gerçekleştirilmiştir. Kullanıcı testinde, uygulamaya ait bir görev listesi kullanıcılara verilerek görevleri uygulama üzerinde yerine getirmeleri istenmiştir. Kullanıcıların uygulamayı kullandıkları esnada ise CodeCarbon kütüphanesi ile karbon salınımı ve enerji tüketimine dair ölçümler yapılmıştır. Kullanıcılar görevleri tamamladıktan sonra ise uygulamanın her sayfası için 6 adet olmak üzere uygulamanın etkililik, verimlilik ve memnuniyet bileşenlerine dair sorular kullanıcılara yönlendirilerek bir görüşme gerçekleştirilmiştir.

Gerçekleştirilen bu görüşmeler kapsamında 12 kullanıcıdan uygulamanın hem arayüzü hemde fonksiyonelliği ile ilgili geri dönütler elde edilmiştir. Bu dönütlere göre ikinci aşama olarak bahsedilen uygulamanın ikinci versiyonu, yani uygulamanın kullanıcı deneyimi ve görüşlerinin dahil edildiği hali geliştirilmiştir. Yine ikinci aşama kapsamında 12 adet kullanıcıya uygulamanın ikinci versiyonu için kullanıcı testleri gerçekleştirilmiştir. Test yapılacak kişilerin 6 adeti ilk uygulama içinde teste tabi tutulan kişilerden, diğer 6 adeti ise daha önce uygulamayı hiç kullanmamış kişilerden seçilmiştir. Bu 12 kişiye uygulamanın ikinci versiyonu için kullanıcı testi gerçekleştirilmiştir. Kullanıcılar uygulamayı kullanırken bu versiyon için de yine arka planda karbon salınımı ve enerji tüketimi değerler ölçülmüştür.

Yapılan ölçümler sonucunda ise uygulamanın kullanım süresinin artma ve azalma durumlarına bağlı olarak karbon salınımı ve toplam enerji tüketimi değerlerinin de artıp azaldığı yani yapılan enerji tüketimi ve karbon salınımı değerlerinin süre bileşeni ile doğru orantılı olduğu görülmüştür. Süre bileşeninin, ikinci aşama olan kullanıcı görüşlerinin dahil edildiği uygulama kullanımı için olan ortalamasının birinci aşama olan kullanıcı görüşlerinin dahil edilmediği uygulama ortalamasına kıyasla daha düşük olduğu görülmektedir.

Bu duruma baęlı olarak ikinci ařamada elde edilen karbon salınımı ve toplam enerji tüketimi deęerlerinin ortalaması ilk ařama deęerlerinin ortalamasından daha düşük olduęu gözlemlenmiřtir.

Aynı řekilde uygulamayı hem ilk ařamada hem de ikinci ařamada kullanan kullanıcıların deęerlerine karřılařtırma grafiklerinde (řekil 21, 22, 23, 24, 25,26) bakıldığında uygulamanın ikinci versiyonunda elde edilen, kullanım süresi, yapılan karbon salınımı, enerji tüketimi deęerlerinin uygulamanın ilk versiyonunda elde edilen deęerlerden daha düşük olduęu görölmektedir.

Her iki ařamaya dahil edilen kullanıcılar için uygulamaya dair bir önceki ařamadan deneyimleri olmasının ikinci ařama verilerinin üzerinde etkili olması durumuna karřı alıřmada birinci ařamada uygulamanın ilk versiyonunu ilk kez kullanan kullanıcıların verileri ile ikinci ařamada uygulamanın ikinci versiyonunu ilk kez kullanan kullanıcıların ölçüm sonuçları karřılařtırılmıřtır. řekil 28 de göröldüęü gibi ikinci versiyonu ilk kez kullanan kullanıcıların ortalama kullanım süresi, karbon salınımı ve enerji tüketimi deęerleri, ilk versiyonu ilk kez kullanan kullanıcıların ortalama deęerlerinden daha düşüktür. Bu durumda ilk ařamada gerekleřtirilen kullanıcı testlerinden elde edilen kullanıcı görüş ve önerilerinin ikinci versiyon üzerinde uygulanmasının uygulamanın kullanım süresi, karbon salınımı ve enerji tüketimi deęerlerini düşürebileceęi anlamına gelmektedir.

Kullanıcılar ile uygulamanın test edilmesinden sonra yapılan görüşmelerde, her kullanıcıdan hem uygulamanın farklı noktaları için farklı görüş ve öneriler hem de aynı noktalar için ortak görüş ve öneriler toplanmıřtır. Geliřtirilen uygulamanın kullanıcı gözünden incelenmesi ve deęerlendirilmesi, hem arayüz tasarımı hem de fonksiyonellik aısından uygulamanın ikinci versiyonunu daha iyi bir hale getirmiřtir. Kullanıcıların uygulama hakkında belirttikleri görüş ve öneriler, uygulamanın daha kolay bir řekilde anlařılabilmesini, daha kolay kullanılır hale gelmesini saęlamaya yöneliktir. Tez alıřması kapsamında geliřtirilme ařamalarına kullanıcı testlerinin ve görüşlerinin dahil edilmedięi uygulamanın kullanım süresi, salınımı yapılan karbon miktarı ve tüketilen enerji deęerlerinin, geliřtirilme ařamalarına kullanıcı test ve görüşlerinin dahil edildięi uygulamaya kıyasla daha fazla olması kullanıcı görüşlerinin uygulamanın geliřtirilme ařamalarına dahil edilmesinin, uygulamanın kullanım süresi, yapılan karbon salınımı ve enerji tüketimi deęerleri üzerinde de olumlu yönde etkili olabileceęi görölmektedir.

Günümüzün teknolojisinde internet ve uygulamaları günümüzün vazgeilmez birer parası halindedir. İnternet üzerinde her gün birok uygulamanın sürekli ve birok kiři

tarafından kullanıldığı göz önüne alındığında, son kullanıcıların uygulamaları daha kolay anlayabilmeleri ve daha kolay kullanabilmeleri için uygulamaların geliştirilme aşamalarına kullanıcı testlerinin dahil edilmesi önemli bir etkidir. Aynı zamanda kullanıcı deneyimlerinin uygulamaların kullanım sürelerini, buna bağlı olarak yapılan karbon salınımı ve enerji tüketimini düşürerek yeşil yazılım açısından da fayda sağlayabileceği düşünülmektedir.

KAYNAKÇA

- Abdulwahab Ali Almazroi. (2021). A Systematic Mapping Study of Software Usability Studies. *International Journal of Advanced Computer Science and Applications*, 12(9).
- Acar, H. (2017). *Software development methodology in a Green IT environment*.
- Agarwal, S., Nath, A., & Chowdhury, D. (2023). *Sustainable Approaches and Good Practices in Green Software Engineering*.
- Anita Holczmann, Bill Johnson, & Sarah Hsu. (2023). *What is green software?*
- Asim Hussain, Adam Jackson, Jenya Saprykina, Namrata Narayan, Osama Jandali, Russell Trow, Sean Mcilroy, Stephanie Stocker, Urs Villiger, & Taichi Imura. (2022). *Green Software Foundation*.
- Bandi, A., & Heeler, P. (2013). Usability testing: A software engineering perspective. *2013 International Conference on Human Computer Interactions (ICHCI)*, 1–8. <https://doi.org/10.1109/ICHCI-IEEE.2013.6887809>
- Bevan, N., Kirakowski, J., & Maissel, J. (1991). *What is Usability?*
- Calero, C., Bertoa, M., & Moraga, M. (2013). A systematic literature review for software sustainability measures. *2013 2nd International Workshop on Green and Sustainable Software, GREENS 2013 - Proceedings*, 46–53. <https://doi.org/10.1109/GREENS.2013.6606421>
- Cengiz Acartürk, & Kürşat Çağıltay. (2006). *İnsan Bilgisayar Etkileşimi ve ODTÜ’de Yürütülen Çalışmalar*.
- Cheikhi, L., Abran, A., & Suryn, W. (2006). Harmonization of usability measurements in ISO9126 software engineering standards. 3246–3251. <https://doi.org/10.1109/ISIE.2006.296137>
- Coşkan, D., & Durdu, P. O. (2014). Kullanılabilirlik ve yazılım yaşam döngüsü: Türkiye’deki Yazılım Organizasyonlarındaki Durum. *CEUR Workshop Proceedings*, 1221, 55–66.
- Easterbrook, S. (2010). Climate Change: A Grand Software Challenge. *Proceedings of the FSE/SDP Workshop on the Future of Software Engineering Research, FoSER 2010*, 99–104. <https://doi.org/10.1145/1882362.1882383>
- García-Berná, J. A., Ouhbi, S., Fernández-Alemán, J. L., de Gea, J. M. C., & Nicolás, J. (2021). Investigating the Impact of Usability on Energy Efficiency of Web-based Personal Health Records. *Journal of Medical Systems*, 45(6), 65. <https://doi.org/10.1007/s10916-021-01725-8>
- Gary Cook, & Jodie Van Horn. (2011). *How dirty is your data?*
- Gonzalez-Holland, E., Whitmer, D., Moralez, L., & Mouloua, M. (2017). Examination of the Use of Nielsen’s 10 Usability Heuristics & Outlooks for the Future. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 61(1), 1472–1475. <https://doi.org/10.1177/1541931213601853>
- Greenpeace International. (2010). *Make IT Green: Cloud computing and its contribution to climate change*.

- Hannah Ritchie, Pablo Rosado, & Max Roser. (2023). *Carbon intensity of electricity per kilowatt-hour*. Our Word in Data.
- Heider, J., Tasche, D., Lässig, J., & Will, M. (2013). A greenhouse gas accounting tool for regional and municipal climate change management. *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*, 1–3. <https://doi.org/10.1109/SustainIT.2013.6685204>
- Hertzum, M. (2012). How do usability professionals construe usability? *Int. J. Hum.-Comput. Stud.*, 70, 26–42. <https://doi.org/10.1016/j.ijhcs.2011.08.001>
- Hustak, T., & Krejcar, O. (2016). Principles of usability in Human-Computer interaction. *Lecture Notes in Electrical Engineering*, 354, 51–57. https://doi.org/10.1007/978-3-662-47895-0_7
- Issa, T., & Isaias, P. (2015). *Usability and Human Computer Interaction (HCI)* (pp. 19–36). https://doi.org/10.1007/978-1-4471-6753-2_2
- Jakob Nielsen. (1993). *Usability Engineering*.
- Jeffrey Rubin, Dana Chisnell, & Jared Spool. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests* (2nd Edition).
- Kivijärvi, H., & Pärnänen, K. (2023). Instrumental Usability and Effective User Experience: Interwoven Drivers and Outcomes of Human-Computer Interaction. *International Journal of Human-Computer Interaction*, 39(1), 34–51. <https://doi.org/10.1080/10447318.2021.2016236>
- Kosch, T., Karolus, J., Zagermann, J., Reiterer, H., Schmidt, A., & Woźniak, P. (2023). A Survey on Measuring Cognitive Workload in Human-Computer Interaction. *ACM Computing Surveys*, 55. <https://doi.org/10.1145/3582272>
- Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). *Quantifying the Carbon Emissions of Machine Learning*.
- Mahmoud, S. S., & Ahmad, I. (2013). A green model for sustainable software engineering. *International Journal of Software Engineering and Its Applications*, 7, 55–74.
- Manotas, I., Sahin, C., Clause, J., Pollock, L., & Winbladh, K. (2013). Investigating the impacts of web servers on web application energy usage. *2013 2nd International Workshop on Green and Sustainable Software (GREENS)*, 16–23. <https://doi.org/10.1109/GREENS.2013.6606417>
- Michanan, J., Dewri, R., & Rutherford, M. (2016). GreenC5: An adaptive, energy-aware collection for green software development. *Sustainable Computing: Informatics and Systems*, 13. <https://doi.org/10.1016/j.suscom.2016.11.004>
- Misra, S., & Cafer, F. (2010). *A Software Metric for Python Language*. 6017, 301–313. https://doi.org/10.1007/978-3-642-12165-4_24
- Naumann, S., Dick, M., Kern, E., & Johann, T. (2011). The GREENSOFT Model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems*, 1(4), 294–304. <https://doi.org/https://doi.org/10.1016/j.suscom.2011.06.004>
- Nielsen, J. (2006). *Ten Usability Heuristics*. <https://api.semanticscholar.org/CorpusID:59788005>
- Ortiz, A. (2012). *Web development with python and django (abstract only)*. <https://doi.org/10.1145/2157136.2157353>
- Penzenstadler, B., Bauer, V., Calero, C., & Franch, X. (2012). Sustainability in Software Engineering: A Systematic Literature Review. *IET Seminar Digest*, 2012. <https://doi.org/10.1049/ic.2012.0004>

- Pereira, R. A. A. (2018). *Energyware Engineering: Techniques and Tools for Green Software Development*. Universidade do Minho Escola de Engenharia.
- Petrick, E. (2020). A Historiography of Human-Computer Interaction. *IEEE Annals of the History of Computing*, PP, 1. <https://doi.org/10.1109/MAHC.2020.3009080>
- Prof. Dr. Kürşat Çağıltay. (2011). *Teoriden Pratiğe İnsan-Bilgisayar Etkileşimi ve Kullanılabilirlik Mühendisliği* (Vol. 2).
- Sean Mcilroy, Henry Richardson, & Abhishek Gupta. (2023). *Standards / SCI*. <https://learn.greensoftware.foundation/measurement>
- Segura, J. (2021). The Teaching of Usability in Software Development: Case Study in the Computer Engineering Career at the University of Matanzas. *International Journal of Engineering Pedagogy (IJEP)*, 11, 4. <https://doi.org/10.3991/ijep.v11i1.14837>
- Shi, X., Xiao, H., Weifeng, L., Chen, X., Lackner, K., Buterin, V., & Stocker, T. (2021). *Confronting the Carbon-footprint Challenge of Blockchain*.
- Sinha, G., Shahi, R., & Shankar, M. (2010). *Human Computer Interaction*. 1–4. <https://doi.org/10.1109/ICETET.2010.85>
- Taniguchi, H., & Nakasho, K. (2021). Visual Studio Code Extension and Auto-completion for Mizar Language. *2021 Ninth International Symposium on Computing and Networking (CANDAR)*, 182–188. <https://doi.org/10.1109/CANDAR53791.2021.00033>
- The Climate Group. (2008). *SMART 2020: Enabling the low carbon economy in the information age*.
- United Nations World Commission on Environment and Development. (1987). *Report of the World Commission on Environment and Development: Our Common Future*.
- William (Bill) Albert, & Thomas S. (Tom) Tullis. (2008). *Measuring the User Experience*. Elsevier. <https://doi.org/10.1016/C2018-0-00693-3>
- Word Nuclear Association. (2011). *Comparison of Lifecycle Greenhouse Gas Emissions of Various Electricity Generation Sources*.
- Xu, M., Gao, C., Ilager, S., Wu, H., Xu, C.-Z., & Buyya, R. (2021). *Green-Aware Mobile Edge Computing for IoT: Challenges, Solutions and Future Directions* (pp. 145–164). https://doi.org/10.1007/978-3-030-69893-5_7

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Aslıhan KILIÇ
Biyografi	
Deneyim	
Araştırma Görevlisi:	Sakarya Uygulamalı Bilimler Üniversitesi Bilgisayar Mühendisliği Atatürk Üniversitesi Yazılım Mühendisliği
Eğitim	
Yüksek Lisans:	Atatürk Üniversitesi Yazılım Mühendisliği
Lisans:	Atatürk Üniversitesi Bilgisayar Mühendisliği
Tezden Üretilmiş Yayınlar	
1. Kullanıcı Deneyiminin Yeşil Yazılım Mühendisliğine Etkisinin Araştırılması Üzerine Deneysel Bir Çalışma (Bildiri)	