T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

ÇOK BOYUTLU MORTALİTE İNDEKSİ KULLANILARAK YAŞLI BİREYLERDE MORTALİTE ÖNGÖRÜSÜ İÇİN MOBİL HESAPLAYICI GELİŞTİRİLMESİ

İHSAN DOĞAN
ORCID: 0000-0003-0934-1947

MEDİKAL İNFORMATİK ANABİLİM DALI

Tıbbi Bilişim Yüksek Lisans Programı

YÜKSEK LİSANS TEZİ

İZMİR
EYLÜL 2023

TEZ KODU: DEU.HSI.MSc-2019970069

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

ÇOK BOYUTLU MORTALİTE İNDEKSİ KULLANILARAK YAŞLI BİREYLERDE MORTALİTE ÖNGÖRÜSÜ İÇİN MOBİL HESAPLAYICI GELİŞTİRİLMESİ

İHSAN DOĞAN
ORCID: 0000-0003-0934-1947

MEDİKAL İNFORMATİK ANABİLİM DALI

Tıbbi Bilişim Yüksek Lisans Programı

YÜKSEK LİSANS TEZİ

Tez Danışmanı: Prof. Dr. Pembe KESKİNOĞLU
ORCID: 0000-0002-3459-1828

İZMİR
EYLÜL 2023

T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE
ETİK BEYANI

Dokuz Eylül Üniversitesi Sağlık Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırlayıp sunduğum "Çok Boyutlu Mortalite İndeksi Kullanılarak Yaşlı Bireylerde Mortalite Öngörüsü için Mobil Hesaplayıcı Geliştirilmesi" başlıklı Yüksek Lisans tezim içinde elde ettiğim verileri, bilgileri, belgeleri akademik ve etik kurallar çerçevesinde elde ettiğimi, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tezimde yararlandığım eserlere bilimsel kurallara uygun atıfta bulunarak kaynak gösterdiğimi, tezimin özgün olduğunu, tezimin çalışma ve yazımında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

İmza
Öğrencinin Adı Soyadı
Tarih

TEŞEKKÜR

Tez çalışmam süresince benden fikirlerini, önerilerini esirgemeyen, en zor anlarında bile bizi yalnız bırakmayan, her türlü yardımı ve desteğiyle bana her zaman yol göstererek akademik gelişimime büyük katkısı olan güler yüzünü ve samimiyetini benden esirgemeyen çok değerli hocam, tez danışmanım Prof. Dr. Pembe KESKİNOĞLU'na, kendisine ihtiyacım olduğunda kıymetli zamanından feragat edip bana yardımcı olabilmek için elinden geleni yapan, yol gösteren Prof. Dr. Oğuz Dicle' ye ve Prof. Dr. Mustafa Alper Selver'e teşekkürlerimi sunarım.

Bütün süreç boyunca desteğini esirgemeyen en zor anlarımda sabır, hoş görü ve sevgiyle yanımda olan, bana güç veren, güvenen biricik eşime, canım kızıma ve yine desteklerini esirgemeyen çok değerli iş arkadaşlarıma sonsuz teşekkürler.

İÇİNDEKİLER

TABLOLAR DİZİNİ.....i
ŞEKİLLER DİZİNİ.....ii
SİMGELER VE KISALTMALAR.....iv
ÖZET.....v
ABSTRACT.....vi
1. GİRİŞ VE AMAÇ.....1
1.1 Problemin Tanımı ve Önemi.....1
1.2 Araştırmanın Amacı.....1
1.3 Araştırmanın Hipotezleri.....2
1.4 Araştırmanın Sınırlılıkları.....2
2. GENEL BİLGİLER.....3
2.1 Mobil Bulut Bilişim.....3
2.2 Sanallaştırma Mimarileri.....4
2.2.1 Geleneksel Sanallaştırma Mimarisi.....4
2.2.2 Konteyner Sanallaştırma Mimarisi.....5
2.3 Teknolojik Gelişimin Sağlık Alanındaki Etkileri.....5
2.4 Mobil Programlama Dilleri.....6
2.5 ÇBPİ Hakkında.....9
2.6 Sağlıkta Mobil Uygulamalar.....11
2.7 Mobil Uygulamalarda Güvenlik.....12
3. GEREÇ VE YÖNTEM.....14
3.1 Araştırmanın Tipi.....14
3.2 Araştırmanın Yeri.....14
3.3 Araştırmanın Evreni ve Örneklemi.....14
3.4 Araştırmanın Değişkenleri.....14
3.5 Araştırmanın Planı.....14
3.6 Veri Toplama Araçları.....15
3.7 Çalışma Materyali.....16
3.7.1 Donanımsal Materyaller.....16
3.7.2 Yazılımsal Materyaller.....16

3.8 İş Akışı Tasarımı....17
3.8.1 Uygulamanın Sunucu ile Etkileşimi....17
3.8.2 Kullanıcının Uygulama ile Olan Etkileşimi....19
3.9 Verilerin Değerlendirilmesi....20
3.10 Araştırmanın Sınırlılıkları....21
3.11 Etik Kurul Onayı....21
4. BULGULAR....22
4.1 Veri tabanı Tasarımı....22
4.2 Sunucu Uygulaması Tasarımı....23
4.3 Kullanıcı Arayüzü Tasarımı....28
4.4 Sunucu Sistemin Hazırlanması....39
4.5 Play Store İşlemleri....42
4.5.1 Uygulama Performans Değerlendirmesi....43
4.5.2 Google Play Analizleri....45
4.5.3 Anket Sonuçları....48
5. TARTIŞMA....55
6. SONUÇ VE ÖNERİLER....61
7. KAYNAKLAR....63
8. EKLER....68
8.1 Etik Kurul onay formu....68
9. ÖZGEÇMİŞ....71
9.1 Özgeçmiş....71

TABLOLAR DİZİNİ

Tablo 1: Mobil programlama dilleri karşılaştırması (18).....8
Tablo 2: Araştırma planı.....14
Tablo 3: Memnuniyet anketi soruları ve yanıtları.....15
Tablo 4: ÇBPİ hesaplama tablosu.....22
Tablo 5: Uygulamanın kolaylığı ile ilgili yanıtların dağılımı.....48
Tablo 6: Uygulamayı önerme ile ilgili yanıtların dağılımı.....49
Tablo 7: Uygulamanın karmaşıklığı ile ilgili yanıtların dağılımı.....50
Tablo 8: Uygulama güvenliği ile ilgili yanıtların dağılımı.....50
Tablo 9: Uygulamanın gereksizliği ile ilgili yanıtların dağılımı.....51
Tablo 10: Uygulamanın kullanımında destek ihtiyacı hakkındaki yanıtların dağılımı.....51
Tablo 11: Uygulamayı kullanırken kolay ilerleme hakkındaki yanıtların dağılımı.....52
Tablo 12: Uygulamanın sık kullanımı hakkındaki yanıtların dağılımı.....52
Tablo 13: Uygulamada tutarsızlık hakkındaki yanıtların dağılımı.....53
Tablo 14: Uygulamanın süreci kolaylaştırması hakkındaki yanıtların dağılımı.....53

ŞEKİLLER DİZİNİ

Şekil 1: Uygulama mantığı.....18
Şekil 2: İş akış diyagramı.....20
Şekil 3: Veri tabanı diyagramı.....23
Şekil 4: Ayarlar dosyası örneği.....24
Şekil 5: Örnek yönlendirme tablosu.....25
Şekil 6: Hasta kayıtlarını sağlayan kontrolcü.....25
Şekil 7: DRF kütüphanesi ile oluşturulan kontrolcü.....26
Şekil 8: Serileştirici örneği.....27
Şekil 9: Serileştirilmiş anonim veri örneği.....28
Şekil 10: Parametre tablosu için model örneği.....28
Şekil 11: Cubit durum yöneticisi.....29
Şekil 12: Kullanıcı giriş ekranı.....30
Şekil 13: Şifremi unuttum ekranı.....31
Şekil 14: Şifremi Unuttum tıklandığında görülen uyarı.....31
Şekil 15: Şifre yenileme formu.....32
Şekil 16: Kullanıcı karşılama ekranı.....33
Şekil 17: Yeni hasta kayıt ekranı.....34
Şekil 18: Kayıtlı hasta işlemleri.....35
Şekil 19: Hasta parametreleri.....36
Şekil 20: Parametre üzerindeki işlemler.....37
Şekil 21: Parametre değerlendirmesi.....38
Şekil 22: Dockerfile dosyası.....39
Şekil 23: docker-compose.yml dosyası.....40
Şekil 24: Çalışan konteynerler.....41
Şekil 25: Nginx ayar dosyası.....41
Şekil 26: Performans analizi.....43
Şekil 27: Hafıza kullanımı.....44
Şekil 28: Ağ analizi.....45
Şekil 29: Uygulamanın test edildiği cihazlar.....45

Şekil 30: Google kararlılık testi.....46
Şekil 31: Google performans testi.....46
Şekil 32: Google erişilebilirlik testi.....47
Şekil 33: Google güvenlik testi.....48
Şekil 34: ‘Mpi portable’ uygulamasında faklı dillerde hata.....56
Şekil 35: 'Mpi portable' uygulamasında veri kontrolü.....57
Şekil 36: Özgeçmiş sayfa 1.....65
Şekil 37: Özgeçmiş sayfa 2.....66

SİMGELER VE KISALTMALAR

AAS	Alan Adı Sistemi
AGYA	Aletli Günlük Yaşam Aktiviteleri
ÇBPİ	Çok Boyutlu Prognostik İndeksi
ESÖ	Exton-Smith Ölçeği
FRS	Framingham Risk Score
GVKY	Genel Veri Koruma Yönetmeliği
GYA	Günlük Yaşam Aktiviteleri
GYK	Güvenli Yuva Katmanı
IOS	IPhone Operating System (İphone İşletim Sistemi)
KA	Kullanıcı Arayüzü
KHDÖ	Kümülatif Hastalık Değerlendirme Ölçeği
KTMDA	Kısa Taşınabilir Mental Durum Anketi
MBD	Mini beslenme değerlendirmesi
UPA	Uygulama Programlama Arayüzü
YGK	Yazılım Geliştirme Kiti

ÇOK BOYUTLU MORTALİTE İNDEKSİ KULLANILARAK YAŞLI BİREYLERDE MORTALİTE ÖNGÖRÜSÜ İÇİN MOBİL HESAPLAYICI GELİŞTİRİLMESİ

Yüksek Lisans Tezi

İhsan DOĞAN

DOKUZ EYLÜL ÜNİVERSİTESİ SAĞLIK BİLİMLERİ ENSTİTÜSÜ

Medikal İnformatik Anabilim Dalı

ÖZET

Son zamanlarda teknolojik gelişmelerle birlikte hemen hemen hayatımızın her alanında büyük değişimler yaşanmaktadır. Bunlardan en önemlilerinden biri de sağlık alanıdır. Bu teknolojik gelişmelerin sağlık alanına uygulanması hem sağlık sorunu yaşayanlar hem sağlık hizmeti verenler hem de sağlık sisteminde uygulanan programlar açısından büyük geliştirmeler sunmaktadır.

Amacımız yapacağımız mobil uygulama ile sağlık hizmetlerinde kullanılan çok boyutlu mortalite indeksinin uygulanabilirliğini kolaylaştırmak, hızlandırmak, daha anlaşılır ve yorumlanabilir hale getirerek verilen sağlık hizmetlerinin ve karar aşamalarının kalitesinin artırılmasına yardımcı olmaktır.

Uygulamamızın Android, IOS tabanlı mobil cihazlarda çalışabilmesi için Dart ile geliştirilmiş olan Flutter kütüphanesi kullanıldı. Uygulama verilerinin kaydı için ilişkisel veri tabanı oluşturuldu. Sunucu hizmeti sağlaması için Linux sistem üzerinde sanallaştırma teknolojileri kullanılarak tasarlanan yapı ile birlikte Python dilinde sunucu uygulaması oluşturuldu. Mobil uygulamanın sunucu ile etkileşimi sağlandı. Uygulama içerisinde, çevrim içi olarak anket yolu ile memnuniyet düzeyi araştırıldı.

Geliştirilen uygulama ile kullanıcı giriş sistemi, hasta kaydı, hastanın prognostik olarak değerlendirilmesi başarıyla gerçekleştirilmiştir. Anket yoluyla uygulanan beşli likert ölçeği ile uygulamanın bir gereksinim olduğu ve bu gereksinimi karşıladığı kullanıcıların memnuniyeti ile birlikte ortaya çıkmıştır.

Anahtar Sözcükler: indeks, flutter, mobil, ölüm, yaşlı
Tezin sayfa adedi: 72
Danışman: Prof. Dr. Pembe Keskinoğlu

DEVELOPMENT OF MOBILE CALCULATOR FOR PREDICTION OF MORTALITY IN ELDERLY INDIVIDUALS USING THE MULTIDIMENSIONAL MORTALITY INDEX

Master's Thesis

İhsan DOĞAN

DOKUZ EYLÜL UNIVERSITY INSTITUTE OF HEALTH SCIENCES

Department of Medical Informatics

ABSTRACT

Recently, with the technological developments, great changes have been experienced in almost every aspect of our lives. One of the most important of these is the field of health. The application of these technological developments to the field of health offers great improvements in terms of both those who have health problems, health care providers and the programs implemented in the health system.

Our aim is to facilitate and accelerate the applicability of the multidimensional mortality index used in health services, to make it more understandable and interpretable, and to help increase the quality of health services and decision stages with the mobile application we will make.

Flutter library developed with Dart was used for our application to run on Android and IOS based mobile devices. A relational database was created for recording application data. In order to provide server service, a server application in Python language was created with the structure designed using virtualization technologies on the Linux system. The interaction of the mobile application with the server is provided. Within the application, the level of satisfaction was investigated by means of an online questionnaire.

With the developed application, the user login system, patient registration, and prognostic evaluation of the patient were successfully carried out. With the five-point Likert scale applied through the questionnaire, it has emerged that the application is a requirement and that it meets this need with the satisfaction of the users.

Keywords: index, flutter, mobile, mortality, older
Page number: 72
Advisor: Prof. Dr. Pembe Keskinoğlu

1. GİRİŞ VE AMAÇ

1.1 Problemin Tanımı ve Önemi

Bilindiği gibi yaşlı bireyler genç yetişkinlere oranla daha çok sağlık problemleri yaşamaktadır. Birçok hastalığın yaşlılarda daha yüksek görülme sıklığına ve ölüm oranlarına sahip olması ve sağlık sistemlerinde de hastane koşullarının, yoğun bakım, yatak gibi kaynakların sınırlı olması, yaşlı insanlarda geriatrik çok boyutlu bir değerlendirme ile en iyi şekilde alınabilecek önlemlerin ve kararların önemini ortaya koymaktadır.

Aynı zamanda yapılacak bu değerlendirmelerin kolay ve hızlı yapılabilmesi, her zaman her yerden yapılabilir olması, gerekli gözden geçirmelere ve veri analizine imkan vermesi ile birlikte hasta ve sağlık sistemi açısından hizmetin hızlı ve daha kaliteli olmasına büyük katkı sağlayacağı düşünülmektedir.

Bu kriterler göz önüne alındığında sağlıkta doğru öngörülerin hızlı, kolay ve güvenilir bir şekilde sağlanabilmesi büyük önem taşımaktadır. Bu işlemler uygulamanın zorluğu, zaman ve diğer maliyetler açısından bakıldığında hemen yapılamayabilmektedir.

Çok Boyutlu Mortalite İndeksi (Çok Boyutlu Prognostik İndeks) gibi bir geriatrik değerlendirmenin geleneksel yöntemle yapılması, parametrelerin fazlalığı sebebiyle karıştırılabilir olması, zaman maliyeti, verimliliğin azalması gibi sorunlara yol açmaktadır. Ayrıca bu aracın daha kolay bir şekilde uygulanabilmesini sağlayacak çalışmanın mobil uygulamalar yönünden yapılmamış olması buradaki eksikliğin devam ettiğini göstermektedir.

1.2 Araştırmanın Amacı

Bu araştırmanın amacı; detaylı ve etkinliğini kanıtlamış, güvenilir bir tarama indeksi olan ÇBPİ (Çok Boyutlu Prognostik İndeks) kullanımını kolaylaştırmak, hızlandırmak ve her zaman her yerden ulaşımının sağlanacağı bir mobil uygulama geliştirmek ve bu uygulama ile tıp bilişimi alanında bir gereksinimi karşılamaktır. Bu sayede verilecek tedavi, bakım hizmetlerinin daha hızlı, kaliteli ve doğru bir şekilde

yapılabilmesine ve maliyet tasarrufuna katkıda bulunabilmektir. Aynı zamanda uygulamanın hedef kitlesine yönelik sınırlı sayıda görüşme ile kullanılabilirlik hakkında önsel bilgi edinmek amaçlanmıştır.

1.3 Araştırmanın Hipotezleri

Araştırma sorumuz: Kolay uygulanabilir, Android, IOS tabanlı mobil cihazlarda çalışan, uygulayıcıya değerlendirme kolaylığı sağlayan bir mobil uygulama geliştirilmesidir. Sınırlı sayıda kullanıcıda sistemin uygulanılabilirliği hakkında bilgi edinmektir. Yapılacak uygulama ile yaşlı bireylerde ÇBPİ' nin daha etkin bir şekilde kullanımına fayda ve memnuniyet sağlamaktır. Tanımlayıcı ve uygulama geliştirmeye yönelik amaçlar nedeni ile karşılaştırma içeren hipotezimiz bulunmamaktadır.

1.4 Araştırmanın Sınırlılıkları

Araştırma yalnızca ÇBPİ'nin kullanılabildiği geriatri gibi sağlık alanlarını kapsamaktadır. Bu nedenle sınırlı sayıda birim üzerinde sınanabilmektedir. Uygulama Android ve IOS uyumlu olarak planlanmıştır fakat IOS geliştirme için daha yüksek maliyetli donanım gerekmekte olduğundan kararlılık testleri ve Apple Store yüklemesi yapılamamaktadır.

2. GENEL BİLGİLER

Bu tez çalışmasında, uygulamanın hazırlanmasında mobil uygulama, servis uygulaması ve sunucu hizmeti olmak üzere 3 ayrı modül ile oluşturulmuştur. Bu modüllerde mobil bulut bilişim ve sanallaştırma mimarileri gibi teknolojiler kullanılmıştır. Uygulamanın geliştirilmesinde Google tarafından Dart programlama dili ile geliştirilmiş olan Flutter kütüphanesi kullanılmıştır.

2.1 Mobil Bulut Bilişim

Mobil bulut bilişim (Mobile Cloud Computing), mobil cihazlar ile bulut bilişim arasında bir entegrasyon sağlayan bir bilişim modelidir (1). Mobil bulut bilişimde mobil cihazlar, sınırlı işlem gücü, depolama alanı ve pil ömrü gibi kaynak sınırlamalarına sahiptir. Bulut bilişim hizmetleri, bu sınırlamaları aşmak için daha fazla kaynak sunar ve kullanıcılara daha güçlü ve işlevsel uygulamalar kullanma imkanı verir. Mobil cihazlarda depolama alanı sınırlı olabilir. Mobil bulut bilişim, kullanıcılara verilerini güvenli bir şekilde bulutta depolama ve yedekleme imkanı sunar. Böylece veriler güvende tutulur ve cihazlardaki depolama alanı serbest bırakılır (2).

Bulut bilişim hizmetleri, güçlü sunucular üzerinde işlemleri gerçekleştirerek mobil cihazlardaki işlem gücünü artırır. Bu sayede daha karmaşık ve yoğun işlemler, bulut üzerinde gerçekleştirilir ve mobil cihazın performansı artar (3). Farklı cihazlar arasında uygulama ve veri senkronizasyonunu kolaylaştırır. Kullanıcılar, farklı cihazlarında aynı uygulamaları ve verileri kullanabilir ve güncellemeleri anında senkronize edebilir. Bulut tabanlı hizmetlere daha kolay erişim sağlar. Uygulamalar, sunucular üzerinden çalıştığı için kullanıcılar istedikleri hizmetlere internet bağlantısı olan herhangi bir cihazdan erişebilir (4).

Bulut bilişim hizmetleri veren kurumlar, genellikle büyük veri merkezleri olarak adlandırılan sunucu alt yapılarına sahiptirler. Bu sunucular, yüksek işlem gücü, depolama kapasitesi ve ağ hızına sahiptir. Bu hizmetleri verirken bir fiziksel sunucuda daha fazla hizmet barındırmayı sağlayan Sanallaştırma Teknolojileri

kullanırlar.

Bulut bilişim hizmetlerinin sunulmasında bazı modeller kullanılmaktadır.

Altyapı olarak Hizmet (Infrastructure as a Service - IaaS): Bu modelde, kullanıcılara sanal sunucular, depolama alanı, ağ bileşenleri gibi temel altyapı kaynakları sağlanır. Kullanıcılar, bu kaynakları yönetme ve istedikleri uygulamaları çalıştırma özgürlüğüne sahiptir (5).

Platform olarak Hizmet (Platform as a Service - PaaS): Bu modelde, kullanıcılara uygulama geliştirme ve dağıtımı için bir platform sağlanır. Kullanıcılar, kendi uygulamalarını oluşturmak için sunulan araçları ve hizmetleri kullanabilirler. Altta yatan altyapıyı yönetmekle ilgilenmezler (6).

Yazılım olarak Hizmet (Software as a Service - SaaS): Bu modelde, kullanıcılara kullanıma hazır yazılımlar sağlanır. Kullanıcılar, bu yazılımları internet üzerinden erişebilir ve kullanabilirler. Yazılımın bakımı, güncellemeleri ve altyapısı hizmet sağlayıcı tarafından yönetilir (7).

2.2 Sanallaştırma Mimarileri

Sunucu altyapısı, sanallaştırma teknolojisi kullanılarak oluşturulan geleneksel sanal makineler (virtual machines - VM) veya konteynerler ile bölgelere veya bölgelerin içindeki veri merkezlerine bölünebilir. Bu, kaynakları daha verimli kullanmayı ve birden çok kullanıcıya hizmet sunmayı mümkün kılar.

2.2.1 Geleneksel Sanallaştırma Mimarisi

Bu mimaride, fiziksel sunucular üzerinde çalışan işletim sistemleri, sanal makinelere bölünür. Her bir sanal makine, kendi işletim sistemine ve kaynaklara sahiptir. Bu yöntem, kaynakların daha iyi kullanılmasını sağlar ve birden çok işletim sisteminin aynı fiziksel sunucu üzerinde çalışmasına olanak tanır. Her sanal makine, sanallaştırma yazılımı (hypervisor) üzerinde çalışır ve kaynakların paylaşımını sağlar (8). Vmware yazılımı bu mimariye örnek olarak söylenebilir.

2.2.2 Konteyner Sanallaştırma Mimarisi

Bu mimaride, sanal makineler yerine konteynerler kullanılır. Konteynerler, işletim sistemi düzeyinde izolasyonun yapıldığı çalışma ortamlarıdır. Her bir konteyner, kendi uygulama ve bağımlılıklarını içerir , ancak aynı işletim sistemini paylaşır. Konteynerler, hafif ve hızlı başlatılarak daha verimli çalışır. Aynı zamanda, daha fazla esneklik ve ölçeklenebilirlik sağlar ve uygulamaların farklı ortamlarda sorunsuz çalışmasını sağlar. Genellikle bir konteyner yönetim platformu (örneğin, Docker) üzerinde çalışır ve uygulamaları paketlemek, dağıtmak ve yönetmek için kullanılır (9).

Sağladığı avantajlarından dolayı çalışmamızda sunucu üzerinde Konteyner sanallaştırma mimarisi kullanılmıştır.

2.3 Teknolojik Gelişimin Sağlık Alanındaki Etkileri

Teknolojinin gelişmesiyle birlikte araştırmalar ve sonuçlarına daha kolay erişim sağlanmakta ve bununla birlikte her alanda artarak devam eden önemli iyileşmeler yaşanmaktadır. Gerek vücut değerlerini okuyabilen, ölçebilen, müdahale edebilen cihazların, sensörler, veri iletişimi, bulut bilişim ve yapay zekanın, gerek mobil uygulamaların üretilmesi sağlıkta devrimi meydana getirmiştir. Teknolojinin sağlık alanındaki ilerlemeleri, hastaların yaşam kalitesini artırabilir, tedavi süreçlerini iyileştirebilir ve sağlık hizmetlerinin erişilebilirliğini artırabilir. Teknolojinin sağlık alanındaki etki ve faydalarını birkaç konu altında toplanmak mümkündür.

Tanı ve teşhis: Gelişmiş tıbbi görüntüleme teknolojileri sayesinde, hastalıkların erken teşhisi ve doğru tanısı mümkün hale gelmiştir. Manyetik rezonans görüntüleme, Bilgisayarlı tomografi taramaları, ultrasonografi gibi görüntüleme yöntemleri, doktorlara hastalıkları daha iyi anlama ve tedavi planlamasında yardımcı olma imkanı sağlar (10).

Tedavi ve cerrahi müdahaleler: Robotik cerrahi, minimal invaziv cerrahi ve diğer ileri teknolojiler, cerrahi müdahalelerin hassasiyetini ve etkinliğini artırmıştır. Laparoskopik cerrahi ve Da Vinci robotik cerrahi sistemi gibi teknolojiler, daha az

invaziv olmalarıyla iyileşme sürecini hızlandırabilir ve hastaların ağrı ve komplikasyonlarla karşılaşma riskini azaltabilir.

İlaç geliştirme: Yapay zeka ve veri analitiği, ilaç geliştirmede büyük bir rol oynamaktadır. Büyük veri analizi ve makine öğrenimi, genetik bilgileri ve hastalık verilerini analiz ederek, daha etkili ilaçların geliştirilmesine yardımcı olabilir (11).

Veri ve sağlık analitiği: Sağlık hizmetlerindeki veri analitiği, büyük veri ve yapay zeka teknolojileri sayesinde daha etkili hale gelmiştir. Bu, hastalıkların takibi, epidemiyolojik analizler, hastalık yayılımının kontrolü gibi alanlarda önemli bir rol oynar. Ayrıca, kişiselleştirilmiş tıp, hastaların sağlık verileri ve genetik bilgileri üzerinden daha özelleştirilmiş tedavi planları oluşturulmasını sağlar (12).

Uzaktan sağlık hizmetleri: Teletıp ve uzaktan sağlık hizmetleri, hastaların uzmanlara erişimini kolaylaştırır. Video görüşmeler, uzaktan izleme cihazları ve mobil uygulamalar sayesinde, uzmanlar hastaları uzaktan takip edebilir ve danışmanlık sağlayabilir.

Mobil teknolojiler ve uygulamaları sayesinde bu katkıların gerçekleştirilmesi daha hızlı, maliyetsiz ve her yerden yapılabilir hale gelmiştir.

2.4 Mobil Programlama Dilleri

Telefon, tablet, dizüstü bilgisayarlar ve saat gibi cihazların hedefe yönelik uygulamalar ile hayata temasları daha verimli olmaktadır. Bu uygulamaların gerçekleştirilmesinde işletim sistemlerine özel uygulama dilleri ya da her platformda kullanılabilen ve birden fazla platform için çıktı üretebilen programlama dilleri kullanılmaktadır.

Swift, Yalnızca Apple tarafından özel olarak iOS, macOS, watchOS ve tvOS için geliştirilmiş bir programlama dilidir. Hem hızlı hem de güvenli bir dil olup, iOS uygulama geliştirme için yaygın olarak tercih edilir (13).

Objective-C, Apple'ın geleneksel olarak kullandığı bir programlama dilidir. Özellikle eski iOS projelerinde hala kullanılmaktadır. Swift'e göre daha eski bir dil olsa da, hala mevcut iOS uygulamalarının bakımı ve geliştirilmesi için tercih edilebilir. Fakat diğer programlara çıktı üretememektedir (14).

Kotlin ya da java dilleri ile native Android mobil uygulamalar geliştirilebilmektedir. Kotlin, JetBrains tarafından geliştirilen modern bir programlama dilidir. Genel olarak Java Sanal Makinası üzerinde çalışan uygulamalar için tasarlanmıştır, ancak Kotlin aynı zamanda Android uygulama geliştirmek için de yaygın olarak kullanılan bir dildir. Kotlin, Java ile uyumlu bir şekilde çalışabilir ve mevcut Java koduyla sorunsuz bir şekilde entegre olabilir. Kotlin, okunabilirliği artırmak için basit ve anlaşılır bir söz dizisine sahiptir. Bu da geliştirme sürecini hızlandırır ve kodun daha kolay anlaşılmasını sağlar. Null referans hatalarını en aza indirmek için dil düzeyinde null güvenliği sağlar. Bu, NullPointerException(Boş Değer Döndürme) hatalarının ortaya çıkma olasılığını azaltır. Fonksiyonel programlama tekniklerini destekler (15).

FireMonkey çerçevesi, RAD Studio, Delphi ve C++Builder'ın arkasındaki uygulama geliştirme ve çalışma zamanı platformudur. FireMonkey, Windows, OS X, Android ve iOS için çok cihazlı, gerçek yerel uygulamalar oluşturan ve bunları uygulama mağazalarına ve işletmelere hızla ulaştıran ekipler için tasarlanmıştır. Genel özellikleri itibariyle iyi olmasına rağmen ücretli olması nedeniyle pek tercih edilememekte ve büyük kullanıcı topluluğuna sahip olamamaktadır (16).

React Native, JavaScript'i kullanarak native bileşenlerle birlikte çalışan mobil uygulamalar oluşturulmasına olanak tanır. Bu sayede, iOS ve Android platformları için aynı uygulama kodu kullanılabilmektedir. React Native, performans açısından da oldukça etkilidir ve birçok büyük şirket tarafından kullanılmaktadır (17).

Xamarin, Microsoft tarafından geliştirilen bir mobil uygulama geliştirme platformudur. Bu platform, iOS, Android ve Windows için tek bir kod tabanıyla uygulamalar oluşturmayı sağlar. Xamarin, C# programlama dilini kullanarak mobil uygulamalar geliştirmek için kullanılır ve neredeyse tamamen native performans sunar. Xamarin, Xamarin.Forms ve Xamarin.Native olmak üzere iki farklı yaklaşım sunar. Xamarin.Forms, tek bir kod tabanında kullanıcı arayüzünün paylaşılmasına izin verirken, Xamarin.Native, her platform için ayrı ayrı kullanıcı arayüzü oluşturulmasını sağlar (18).

NativeScript, JavaScript veya TypeScript kullanarak iOS ve Android için

gerçek, yerel mobil uygulamalar geliştirilmesini sağlayan açık kaynaklı bir çerçevedir. NativeScript, uygulama geliştirme sürecini hızlandırmak için paylaşılan bir kod tabanı kullanırken, uygulamanın her platformda doğal performans sergilemesini sağlar. NativeScript, yerel bileşenler ve API'larla doğrudan etkileşim kurulmasını sağlayarak, böylece tamamen özelleştirilebilir ve yüksek kaliteli kullanıcı deneyimleri sunmayı hedeflemektedir (19).

Flutter kullanarak hem Android hem de iOS için uygulama geliştirmek mümkündür. Flutter, Google tarafından geliştirilen bir açık kaynaklı KA (kullanıcı arayüzü) kütüphanesidir ve tek bir kod tabanıyla çoklu platformlarda çalışabilen uygulamalar oluşturmanıza olanak tanır. Flutter, Dart programlama dilini kullanır ve kullanıcı arayüzünü hızlı ve modern bir şekilde geliştirmenize yardımcı olan zengin bir araç koleksiyonuna sahiptir. Flutter, performans açısından da oldukça etkilidir ve platforma özgü bileşenlerle doğrudan çalışabilir (20).

Tablo 1: Mobil programlama dilleri karşılaştırması (21)

	Xamarin	**Native Script**	**React Native**	**Flutter**
Görünüm	Orta	Tatmin Edici	Tatmin Edici	İyi
Performans	Orta	Tatmin Edici	Tatmin Edici	İyi
Özelleştirme	Orta	Tatmin Edici	Tatmin Edici	İyi
Kararlılık	Zayıf	Tatmin Edici	Tatmin Edici	İyi
Web Uyumu	Tatmin Edici	Tatmin Edici	Tatmin Edici	İyi

Amacımıza yönelik kullanabileceğimiz programlama dillerinin bir karşılaştırması Tablo 1'de verilmiştir.

Flutter kendi görünüm oluşturma motorunu kullanır. Bu sayede bazı durumlarda yerel dillerden bile daha hızlı bir kullanıcı arayüzü oluşturabilmektedir. Performans olarak genelde aynı olmakla birlikte bazı durumlarda saf Android IOS uygulamalarından daha hızlı olabilmektedir. Kullanıcı arabirimi son derece esnektir ve yerel platformların yaptığı gibi kullanıcı arabirimi oluşturmaya, özelleştirmeye

olanak tanır. Uygulamalarda sorun çıkarmayan kararlı sürümlere sahiptir ve IOS, Android güncellemelerine hızlı bir şekilde uyum sağlar. Web uygulamaları, mobil uygulamalar ve masaüstü uygulamaları oluşturmak için tam sürüme sahiptir (21).

2.5 Çok Boyutlu Prognostik İndeks Hakkında

Mortaliteyi tahminlemek için uzun süren araştırmalar neticesinde kanser, organ yetmezliği, kalp hastalığı gibi durumlarda kullanılabilecek yöntemler üretilmiş ve geliştirilmiştir. Bu yöntemlerden biri olan ÇBPİ, yaşlı hastalarda klinik, fonksiyonel ve bilişsel durumlarına dayanan çok boyutlu bir ölçek olup, hasta kırılganlığını değerlendirmek ve bazı tahminler yapmak ve buna göre gerekli önlemlerin alınmasını sağlamak amaçlı kullanılan bir indeks olarak ortaya çıkmıştır. ÇBPİ, toplumda yaşayan yaşlı yetişkinlerin sağlık durumlarını değerlendirmek ve prognozlarını tahmin etmek için kullanılan bir ölçektir. ÇBPİ, yalnızca tıbbi faktörleri değil, aynı zamanda psikososyal ve işlevsel faktörleri de hesaba katması ile multidisipliner bir yaklaşım sunar ve yaşlı bireylerin sağlık durumlarını anlamak için kapsamlı bir değerlendirme sağlar. Bu ölçek, yaşlı yetişkinlerin fonksiyonel, bilişsel, tıbbi ve sosyal durumlarını dikkate alarak risk değerlendirmesi yapar ve bunun sonucunda sağlık ve yaşam beklentilerini tahmin eder.

ÇBPİ, sağlık profesyonellerine hastalarının yaşam kalitesini ve sağlık sonuçlarını iyileştirmek için önemli bir araç sağlar. Bu ölçek, yaşlıların sağlık durumlarına uygun tedavi ve bakım planlarının oluşturulmasına yardımcı olur.

ÇBPİ'nin kullanılan parametrelerinden bazıları şunlardır:

Yaş: Kişinin yaşlılık döneminde olması, bazı sağlık sorunlarının ortaya çıkma olasılığını artırır. Yaş, ÇBPİ'nin değerlendirme sürecinde önemli bir faktördür (22).

Fonksiyonel Durum: Kişinin günlük yaşam aktivitelerini bağımsız bir şekilde gerçekleştirebilme yeteneği değerlendirilir. Bu kapsamda, mobilite, banyo yapma, giyinme gibi temel aktiviteler göz önünde bulundurulur (23,24).

Bilişsel Durum: Bireyin bilişsel yetenekleri ve zihinsel sağlığı değerlendirilir. Bellek, dikkat, problem çözme becerileri gibi alanlar incelenir. Bilişsel durum, yaşlı yetişkinlerin sağlık durumunu etkileyen önemli bir parametredir (25).

Tıbbi Durum: Kronik hastalıklar, mevcut sağlık sorunları, ilaç kullanımı ve tıbbi geçmiş ÇBPİ'de dikkate alınır. Bu parametreler, yaşlı yetişkinlerin sağlık risklerini ve prognozlarını belirlemede önemli bir rol oynar (26).

Sosyal Durum: Bireyin sosyal destek sistemleri, yaşadığı ortamın güvenliği, sosyal etkileşim düzeyi gibi faktörler ÇBPİ'de değerlendirilir. Sosyal durum, yaşlı yetişkinlerin sağlık ve yaşam kalitesini etkileyen bir parametredir (26).

ÇBPİ, bu parametreleri bir araya getirerek yaşlı yetişkinlerin sağlık durumlarını kapsamlı bir şekilde değerlendirir. Bu sayede, sağlık profesyonelleri hastalarının risk profillerini anlayabilir, yaşlanma sürecinde ortaya çıkabilecek sorunları önceden tespit edebilir sağlık hizmetlerini kişiselleştirebilir ve uygun tedavi ve bakım planları oluşturabilir. Bu da hastaların yaşam kalitesini artırabilir ve sağlıklarını daha iyi yönetmelerine yardımcı olabilir. ÇBPİ ile ilgili birçok çalışma yapılmış ve sonuçları öngörmede etkili olduğu kanısına varılmıştır.

Örneğin yapılan bir çalışmada 75 yaş üstü 125 hastanın bir yıl boyunca izlenmesi neticesinde, ÇBPİ'nin riskleri ortaya çıkarmada, özel önleyici stratejiler geliştirmede, sağlık hizmetleri kaynaklarının daha yerinde ve verimli kullanılmasını sağlayarak maliyeti düşürmede etkili olduğu ortaya konmuştur (27).

Yapılan büyük bir prospektif çok merkezli çalışma, ÇBPİ'nin hastanede yatan yaşlı hastalarda mortaliteyi tahmin etmede diğer kırılganlık indekslerinden anlamlı derecede daha iyi prognostik doğruluğa sahip olduğunu göstermiştir (28).

Diğer bir çalışmada ÇBPİ'nin COVID-19'dan etkilenen ve yaş ortalaması 80 olan 227 yaşlı hasta üzerinde yapılan incelemede, yatan hastalarda klinik karar vermede yararlı olabileceğini düşündürmüştür (29).

Yine yapılan bir çalışmada ÇBPİ'nin demans veya hafif bilişsel bozukluğu olan yaşlı hastalarda mortaliteyi tahmin etmede kullanılabileceğini göstermektedir. (30).

Tüm bu çalışmalar, ÇBPİ'nin yaşlı hastalarda prognoz tahmini için kullanılan etkili bir araç olduğunu göstermektedir. Fakat uygulanabilirliği parametre fazlalığı açısından zor, karmaşık, vakit alıcı ve maliyetli olabilecek olan bu indeksin bir mobil uygulama ile kısa sürede uygulanabilir olmasının büyük avantajlar getireceği ve

sağlık alanında dijitalleşmeye katkıda bulunacağı, gereksiz prosedürlerin ve bakım maliyetlerinin azaltılabilmesine katkıda bulunacağı düşünülmektedir.

2.6 Sağlıkta Mobil Uygulamalar

Zamanla sağlık alanında ihtiyaç görülen konularda mobil uygulamalar üretilmiş ve kullanımlarıyla birlikte tanı, tedavi ve diğer süreçlerde önemli iyileşmeler kaydedilmiştir. Sağlıkta mobil uygulamalar birkaç başlık altında toplanabilir.

Fitness ve sağlık uygulamaları: Sağlık ve formda kalma konularında yardımcı olan birçok uygulama bulunmaktadır. Örneğin, egzersiz takibi, beslenme planlama ve uyku düzeni gibi konularda destek sağlayan uygulamalar mevcuttur.

Kendini teşhis çözümleri: Belirli semptomlar ve hastalıklar hakkında bilgi sağlayan uygulamalar bulunmaktadır. Bu uygulamalar, kullanıcılara sağlık sorunları hakkında bilgi verirken, doktor ziyareti yerine geçmez ve bir doktora danışmanın yerini tutmaz.

Hatırlatıcı asistan uygulamaları: İlaç alımı, randevular, egzersiz gibi aktiviteler için hatırlatmalar sunan uygulamalar mevcuttur. Bu uygulamalar, kullanıcıların sağlık rutinlerini düzenlemelerine yardımcı olur.

Akıl sağlığı uygulamaları: Zihinsel sağlık ve refahı destekleyen uygulamalar bulunmaktadır. Bu uygulamalar, stres yönetimi, meditasyon, uyku düzeni gibi konularda destek sağlar.

Kadın sağlığı uygulamaları: Kadınlar için özel sağlık konularına odaklanan uygulamalar mevcuttur. Örneğin, adet takibi, gebelik takibi, doğum kontrolü gibi konularda yardımcı olan uygulamalar bulunmaktadır.

Doktorlar için tıbbi uygulamalar: Doktorların tıbbi bilgilere erişmesini ve hastalarıyla etkileşimde bulunmasını sağlayan uygulamalar mevcuttur. Bu uygulamalar, tıbbi kaynaklara erişim, hasta takibi ve tıbbi bilgi paylaşımı gibi işlevlere sahiptir.

ÇBPİ hakkında henüz geliştirilmiş olan bir mobil uygulamaya yönelik çalışmaya rastlanılmamış fakat faklı konularda sağlık sistemini iyileştirmeye yönelik tasarlanmış mobil uygulamalar ve sonuçları çeşitli araştırmalar ile gösterilmiştir.

Yapılan bir araştırmada, mobil sağlık uygulamalarının kullanımının sağlıkla ilgili davranışlar ve klinik sağlık sonuçları üzerinde olumlu bir etkiye sahip olduğunu aynı zamanda uygulama kullanıcılarının, geleneksel yöntemlere kıyasla sağlıklarını yönetmek için mobil sağlık uygulamalarını kullanmaktan daha memnun kaldıklarını göstermektedir (31).

Diğer bir araştırmada, mobil cihazlar ve uygulamaların sağlık profesyonelleri için birçok fayda sağladığı tartışılmış ve özellikle doktorlar için akıllı telefon kullanımının yaygın olduğu belirtilerek, akıllı telefon uygulamalarının verimliliği ve üretkenliği artıracağı düşünülmüştür (32).

Sağlık profesyonelleri tarafından kullanılan araçların veya mobil uygulamaların üretilmesinde ve kullanılmasında dikkat edilmesi gereken en önemli noktalardan birisi hastalara ait şahsi bilgilerin kendilerinin rızası olmadan uygulama ya da sağlık profesyonelleri tarafından kullanılmaması ya da paylaşılmaması olmalıdır. Uygulamalara ait gizlilik politikalarının bulunması ve uygulamaların güvenli hale getirilmesi gerekmektedir.

Klinik karar vermede akıllı telefonların ve dolaysı ile mobil uygulamaların yararlarının, hekimler için umut verici olduğu aynı zamanda hasta mahremiyetine, hak ihlaline dikkat edilmesi gerektiği, akıllı telefon kullanan sağlık çalışanlarının sorumluluklarını korumak için acilen mevzuata ihtiyaç olduğu konusu başka bir araştırmada vurgulanmıştır (33).

2.7 Mobil Uygulamalarda Güvenlik

Kişisel bilgiler bağlamında, mahremiyet kavramları gizlilik ve güvenlik kavramlarıyla iç içedir. Gizlilik, kişisel bilgilere kimlerin hangi koşullar altında erişebileceği konusunu ele alır. Kişisel bilgilerin toplanması, saklanması ve kullanılmasıyla ilgilidir ve verilerin ilk etapta toplanıp toplanamayacağını ve varsa bir amaç için toplanan verilerin başka bir amaç için kullanılabileceği gerekçeleri

inceler (34).

Türkiye'de verilerin mahremiyeti ile ilgili Cumhurbaşkanlığı Dijital Dönüşüm Ofisi tarafından 2019/12 Sayılı Cumhurbaşkanlığı Bilgi ve İletişim Güvenliği Tedbirleri Genelgesi (35) ve Sağlık Bakanlığı tarafından resmi gazetede yayınlanan 30808 sayılı Kişisel Sağlık Verileri Hakkında Yönetmelik (36) ile uygulamaya konulmuştur.

Mobil uygulamalarda mahremiyet için alınması gereken önlemler bulunmaktadır. Geliştiricilerin, verilerinin güvenliği ve gizliliğini sağlayabilmek için bazı önlemleri alması ve gerekli testleri yapması istenmektedir.

Kişiye özel bilgilerin bulunduğu mobil uygulamalarda, bu bilgilere erişim için her kullanıcının bazı doğrulama aşamalarından geçmesi gerekmektedir. Yalnızca kullanıcı adı ve şifre girilerek yapılan giriş sistemi kullanıcı kaynaklı bazı zaaflar barındırabilmektedir. Buna karşın daha güvenli olarak iki adımlı doğrulama ile giriş sistemi bulunmaktadır. Bu doğrulama sistemine örnek olarak kullanıcı adı, şifre ve sms ile veya kullanıcı adı,şifre ve e-posta ile gönderilen kod veya kullanıcı adı, şifre ve parmak izi ile sağlanan girişler verilebilir.

Kişisel verilerin ağ üzerinden iletiminin mevcut olduğu uygulamalarda, bu iletim sırasında veriler şifrelenmemiş ise güvenlik açığı problemi ortaya çıkabilmektedir. Ağdaki kötü amaçlı kişiler tarafından bu veriler ele geçirilebilmektedir. GYK (Güvenli Yuva Katmanı) teknolojisi sayesinde veriler şifrelenmekte ve ele geçirilse bile kullanılamamaktadır. Bu yüzden kişisel verilerin internet üzerinden aktarıldığı uygulamalarda GYK tercih edilmesi gerekmektedir.

Uygulama tarafından kaydedilen kişisel verilerin veri tabanı ortamında şifrelenerek tutulmasının yanı sıra kullanıcı tarafından istendiğinde tamamen silinebilir olması veri gizliliği açısından istenen bir durumdur. Veri silme işlemi uygulama içerisinden kullanıcı tarafından gerçekleştirilemiyorsa bile, uzun süre işlem yapmama, belirli sayıda başarısız oturum açma girişiminden sonra otomatik olarak, ya da destek hattı sayesinde hizmet sağlayıcı tarafından silinebilir.

3. GEREÇ VE YÖNTEM

3.1 Araştırmanın Tipi

Çalışmamız, yaşlı bireylerin mortalite öngörüsü için ÇBPİ kullanımında teknolojiden yararlanarak mobil uygulamanın geliştirilmesini kapsadığı için bir uygulama projesi kapsamındadır.

3.2 Araştırmanın Yeri

Medikal İnformatik Anabilim dalı Tıbbi Bilişim Yüksek lisans tezi olarak yapılmıştır. Herhangi bir mekânsal bağlamı yoktur. Öz kaynaklar vasıtasıyla çalışma yürütülmüştür.

3.3 Araştırmanın Evreni ve Örneklemi

Model ve yapılacak olan uygulama yaşlı bireylerde mortalite riskini öngörmeyi planlanmıştır.

3.4 Araştırmanın Değişkenleri

Herhangi bir nedensel çalışma geliştirilmeyecektir. Sadece bir hesaplayıcı geliştirilerek hedef uygulayıcıların kullanım durumları tanımlanmıştır.

3.5 Araştırmanın Planı

Tablo 2: Araştırma planı

Literatür tarama	15.10.2022 – 15.01.2023
Geliştirme ortamının hazırlanması	16.01.2022 – 15.02.2023
Uygulama kodlarının yazılması	16.02.2023 – 15.05.2023
Hata kontrol ve testleri	16.05.2022 – 15.06.2023
Raporun hazırlanması	16.06.2023 – 15.08.2023

3.6 Veri Toplama Araçları

Veri toplamak için bilimsel çalışmalar ve makaleler taranmış ve değerlendirilmiştir. Aynı zamanda uygulama içi işlemler ve anket değerlendirmesi kullanılan veri toplama araçlarıdır.

Uygulama içinde kullanılan kayıt formları, ÇBPİ parametrelerinin değerlendirme sorularını içeren formlar ve kullanıcı görüşlerini, memnuniyetini değerlendiren anket formu veri toplamak amaçlı kullanılmaktadır. Memnuniyet anketi soruları ve yanıtları Tablo 3'te gösterilmektedir.

Tablo 3: Memnuniyet anketi soruları ve yanıtları

Sorular	Uygulamanın kullanımını kolay buldum	
	Uygulamayı sıklıkla kullanacağımı düşünüyorum	
	Uygulama ÇBPİ değerlendirmesini kolaylaştıracaktır	
	Uygulamayı kullanırken kolayca ilerliyorum	
	Uygulamayı karmaşık buldum	
	Uygulamada hatalar olduğunu düşünüyorum	
	Uygulamayı gereksiz buldum	
	Uygulamayı kullanırken desteğe ihtiyaç duydum	
	Uygulamayı güvenli buldum	
	Uygulamayı başkalarına önerebilirim	
Yanıtlar ve Puanları	Kesinlikle katılmıyorum	1
	Katılmıyorum	2
	Kararsızım	3
	Katılıyorum	4
	Kesinlikle katılıyorum	5

3.7 Çalışma Materyali

ÇBPİ uygulamasının gerçekleştirilebilmesi için gerekli olan materyaller donanımsal ve yazılımsal materyaller olmak üzere iki başlık altında toplanmıştır.

3.7.1 Donanımsal Materyaller

Mobil cihazlar için uygulama yapılabilmesi için uygun donanımın kullanılması, uygulama geliştirimine ve işlemlerin daha hızlı ve uygun şekilde ilerlemesine imkan tanıyacaktır.

ÇBPİ çalışmasında, özellikle Android uygulama geliştirmek için uyumluluğu yüksek olan Linux Mint işletim sistemine sahip 32 gb bellek kapasitesine sahip İntel i7-6700 3.40 ghz işlemcili bir masaüstü bilgisayar, Windows 10 işletim sistemine sahip İntel i7 7700HQ 2.8 ghz işlemcili 32 gb bellek kapasitesine sahip bir dizüstü bilgisayar ve Android 9 işletim sistemine sahip, 6gb ram kapasitesine sahip Xiaomi mi6 cep telefonu kullanılmıştır.

3.7.2 Yazılımsal Materyaller

Çalışmamızda, günümüzde en yaygın olarak kullanılan mobil işletim sistemi Android platformu olduğundan dolayı ve IOS işletim sistemine yazılım üretmenin donanımsal maliyeti daha yüksek olduğundan dolayı öncelikle Android işletim sistemine uygun yazılım geliştirmenin üzerinde durulmuştur.

Uygulama kullanıcı arayüzü, Google firmasının geliştirdiği, Dart dilinde Flutter kütüphanesi kullanılarak üretilecektir. Farklı platformlar için farklı diller öğrenme zorunluluğunu ortadan kaldırması, değişiklikleri hızlı görme, küçük boyutlu, şık uygulamalara imkanı vermesi gibi artıları sebebiyle Flutter tercih edilmiştir.

Editör olarak, Microsoft tarafından ücretsiz olarak sunulan, küçük boyutlu ve ihtiyaca göre ilave araçların eklenebildiği, kararlı olarak çalışan, derleme, hata giderme ve başarım testleri sağladığı için Visual Studio Code ve Yalnızca Android uygulama geliştirmek için Jetbrains tarafından yapılmış Android Studio tercih

edilmiştir.

Uygulama testleri Web, Linux Masaüstü, Windows Masaüstü uygulaması olarak ve Pixel 6 Pro UPA(Uygulama Programlama Arayüzü) 24, Pixel 5 UPA 30 Android YGK (Yazılım Geliştirme Kiti) ve Xiaomi mi 6 da Android uygulaması olarak test edilmiştir.

Sunucu UPA hizmetlerinde, kullanım kolaylığı, esnekliği, güvenlik ilkelerini sağlaması, araç kütüphanesinin tatmin edici olması gibi sebeplerle Python programlama diline ait Django Rest Framework, editör olarak Pyhon ile çok uyumlu, derleme, hata giderme konularında başarılı olan Jetbrains firmasına ait ve ücretsiz sürümü de bulunan PyCharm tercih edilmiştir.

Sunucu kurulumunda Ubuntu işletim sistemine sahip canlı sunucu üzerinde Docker sanallaştırma mimarisi ve onunla beraber sunucu hizmeti vermesi için Nginx web sunucu yazılımı kullanılmıştır.

3.8 İş Akışı Tasarımı

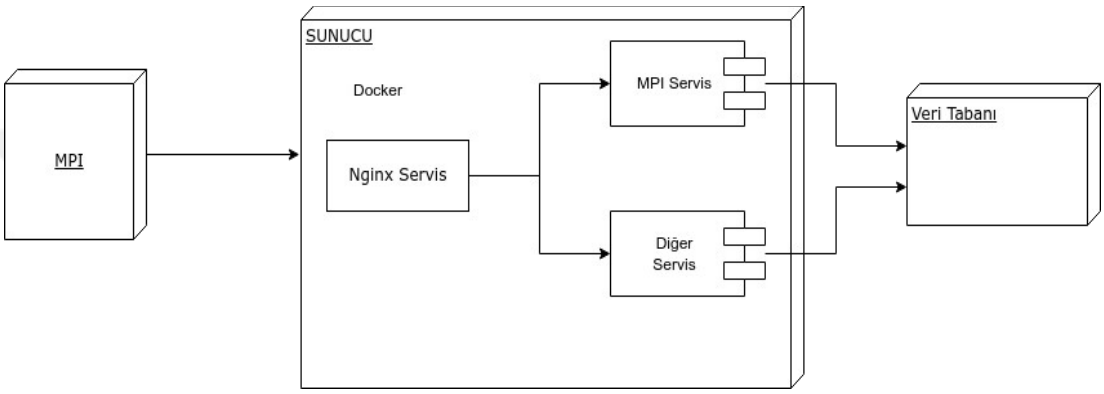
ÇBPİ'nin gerçekleştirilmesinde, mobil uygulamada kullanıcının etkileşimi ve uygulamanın sunucu ile etkileşimi olarak iki farklı yaklaşım ele alınmıştır. Bu yaklaşımların birlikte planlanması ve uygulanması ile ÇBPİ uygulaması gerçekleştirilerek bu tez çalışmasında sunulmuştur.

Ardından uygulama performansı Flutter araçları ve Google değerlendirme sonuçları incelenmiştir. Aynı zamanda içine eklenmiş olan beş seçenekli on sorulu Likert ölçeği kullanılan anket yoluyla kullanıcıların geri dönüşleri alınarak, uygulamanın sağlık alanına katkıları değerlendirilmiştir.

3.8.1 Uygulamanın Sunucu ile Etkileşimi

Uygulamada herhangi bir işlem başlatıldığı anda, Şekil1'de 'MPI' olarak isimlendirilen uygulama, yönlendirme tablosuna bakarak ilgili isteği gönderebileceği fonksiyonu çalıştırmaktadır. Bu fonksiyon, ilgili isteği sunucu tarafına doğru ve şifreli olarak göndermekle yükümlüdür. 'MPI' olarak isimlendirilen mobil uygulama ile yapılan ilgili istek sunucu tarafına ulaştığında, Docker ile oluşturulmuş iki

servisten birisi olan Nginx web sunucusu, isteği karşılayarak bazı kontrollerden geçirdikten sonra gerekli yönlendirmeyi yapmaktadır. İlgili yönlendirmeyi karşılayan Docker ile oluşturulmuş ikinci servis olan 'MPI Servis' olarak isimlendirilen uygulama servisi ise kendi yönlendirme tablosuna bakmakta ve güvenlik kontrollerini uygulayarak isteği ilgili fonksiyona ulaştırmaktadır. Sorumlu fonksiyon kendisine ulaşan bu isteği değerlendirerek, ilgili veri tabanı işlemini gerçekleştirecek ve gerekli cevabı döndürecek şekilde planlanmıştır (Şekil 1).



Şekil 1: Uygulama mantığı

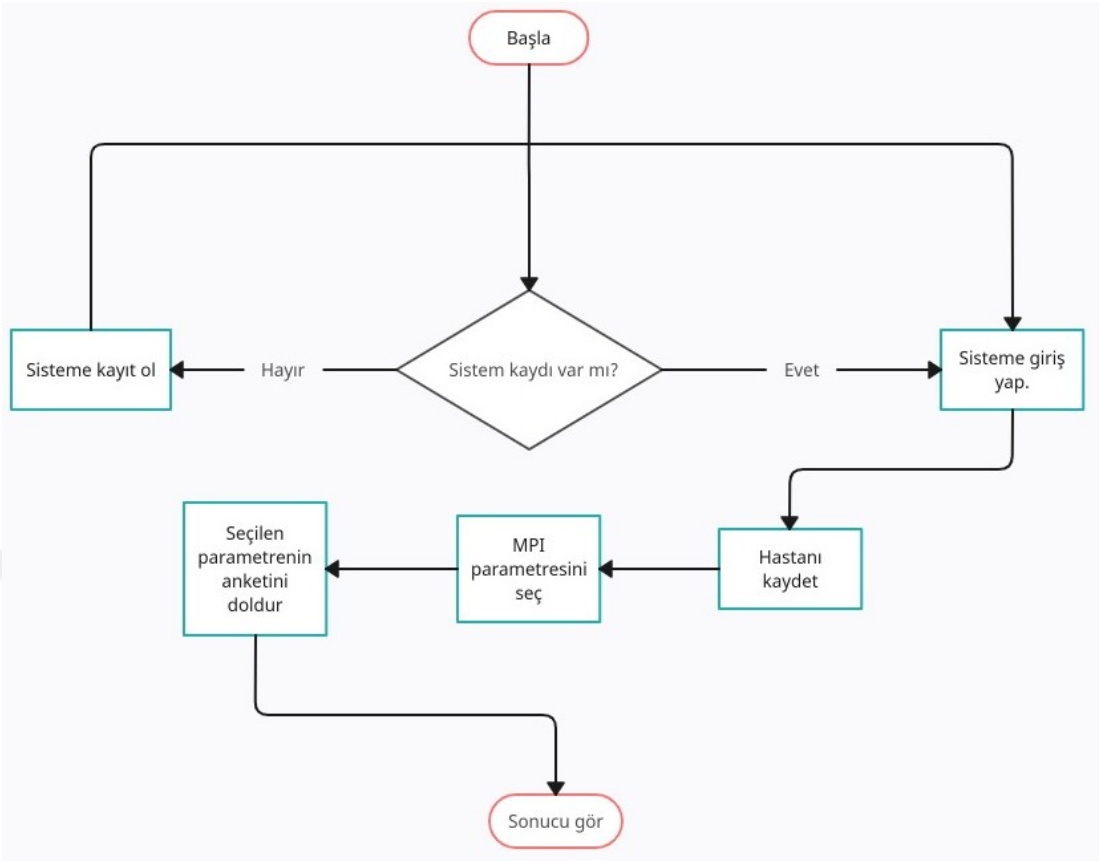
Mobil uygulamalarda veri tabanı, uygulama tarafından kullanılan verilerin depolandığı, yönetildiği ve erişildiği bir yapıdır. Mobil uygulamalar genellikle verileri yerel olarak cihazda saklamak veya sunucu tabanlı bir veri tabanına erişmek için kullanılır. Çalışmamızda veri tabanı olarak SQLite tercih edilmiştir. Mobil uygulamalarda en yaygın kullanılan yerel veri tabanıdır. Hem hafif ve yerel olarak depolanan hem de ilişkisel bir veri tabanı yönetim sistemi sağlar. SQLite, verileri bir dosyada depolar ve SQL sorgularıyla erişim sağlar.

ÇBPİ'nin hesaplama şeması, Tablo 4' de özetlenmiştir. Değerlendirmede parametre olarak, KTMDA (Kısa Taşınabilir Mental Durum Anketi), ESÖ (Exthon Smith Ölçeği), GYA (Günlük Yaşam Aktiviteleri), AGYA (Aletli Günlük Yaşam Akvitileri), KHDÖ (Kümülatif Hastalık Değerlendirme Ölçeği), MBD (Mini Beslenme Değerlendirmesi), Kullanılan ilaç saysı ve hastanın kimlerle yaşadığı

bilgileri ele alınmaktadır. Her bir parametrenin testinin sonucunda bir değer oluşmaktadır. Bu değer düşük, hafif ya da yüksek skora karşılık gelmektedir. Bütün parametreler için karşılık gelen bu değer bulunur ve toplanır. Toplamın sonucunda elde edilen değer, toplam parametre sayısı olan yediye bölünür. Elde edilen sonuç eğer sıfır ile 0,33 arasında ise sonuç düşük, 0,34 ile 0,66 arasında ise sonuç hafif, 0,67 ile bir arasında ise sonuç yüksek olarak kabul edilmektedir.

3.8.2 Kullanıcının Uygulama ile Olan Etkileşimi

Kullanıcının uygulama ile olan etkileşimi , kullanıcının uygulamayı açmasından itibaren bir hastanın ÇBPİ değerini elde edinceye kadar geçen sürecin akışı olarak tasarlanmıştır. Son kullanıcının öncelikle sistemde hesabı yok ise kayıt olması gereklidir. Kayıt olduktan sonra giriş yaparak uygulamanın ana sayfası olan kayıtlı hastaların göründüğü 'Home' sayfasına ulaşması gerekir. Bu sayfada yeni hasta ekleyebilir, eklenmiş hasta üzerinde güncelleme yapılabilir, eklenen hasta silinebilir ya da hastanın ÇBPİ taramasına başlanabilmektedir. ÇBPİ aşaması tercih edildiğinde testini yapmak istediği ÇBPİ parametresini seçerek ilgili testlerin gerçekleştirilmesi planlanmıştır (Şekil 2). 'Home' sayfasında ve diğer ekranlarda hasta veya ÇBPİ parametresi ile ilgili işlem yapılmak istendiğinde, üzerine gelerek sola ve sağa kaydırması istenmektedir. Kaydırma işleminden sonra hastanın ya da parametrenin silinmesi veya güncellenmesi aynı zamanda parametreleri görebilmesi ya da parametre testine geçilebilmesi sağlanmaktadır.



Şekil 2: İş akış diyagramı

3.9 Verilerin Değerlendirilmesi

Uygulamanın, yapılan çalışmaların değerlendirmesinde Flutter Değerlendirme Aracı ve Google Play değerlendirme sonuçları kullanılmaktadır. Bu değerlendirmeler sayesinde uygulamanın mobil ve servis kısımları değerlendirmeye alınmış olur. ÇBPİ uygulamasını kullandığı bellek miktarı, performansı, güvenlik, ağ kullanım durumu, işlemlerde takılma durumları ve farklı cihazlardaki etkileri gibi şekillerde testlere tabi tutulmuştur.

Flutter Değerlendirme Aracı uygulama üzerinde amaçlanan tüm testlerin gerçekleştirilmesini sağlaması (37) ve bu konudaki başarısı yönüyle tercih edilmiş ve ek olarak başka bir aracın bilgisayara kurulmasına gerek kalmamıştır.

Aynı zamanda kullanıcıların memnuniyeti ve uygulamanın başarısı, uygulama içinde tanımlanan beş seçenekli on soru ile oluşturulan anketin sonuçları Likert ölçeği ile değerlendirilmektedir. Kullanımının kolay olması, cevaplarının sayısal olarak ifade edilebilir olması, yanıtlamasının basit olması gibi nedenlerle kullanıcıların görüşlerini, duygularını, memnuniyetlerini değerlendirmek, uygulamanın eksik ya da başarılı yönlerini kullanıcılar bazında tespit edebilmek için beşli Likert Ölçeği kullanılmıştır (38).

3.10 Araştırmanın Sınırlılıkları

Uygulamanın başlangıçta IOS sistemli mobil cihazlarda da çalışması düşünülmüştür. Yazılım ortamları bunu sağlayacak şekilde tercih edilmeye çalışılmış fakat donanımsal ihtiyaçların maliyetli olması sebebiyle bu özellik yerine getirilememiş ve Apple Store'a uygulama eklenememiştir.

Araştırma uygulamamız Google Play Store da yaklaşık 54 ülkede yayınlanmaktadır. Uygulamamızda faklı dil seçeneklerinin olması, uygulamamızın kullanılırlığını artırmak için eklenmesi gereken bir özellik olarak düşünülmüştür. Bu kadar fazla dil seçeneğinin uygulamaya eklenmesi zaman açısından çok maliyetli olacağı düşünüldüğü için dil seçeneği özelliğinin eklenmesinden vazgeçilmiştir. Dünyada en çok konuşulan dillerinden olan İngilizce (39), uygulama dili olarak kabul edilmiştir.

Araştırma uygulamamızın başarısının ya da memnuniyetin tespiti için uygulanan ankete, kullanıcıların uygulamada olmasını istedikleri başka özellikler varsa neler olduğu sorusunun eklenmesi, uygulamada olabilecek, memnuniyeti artırıcı yeni özelliklerin belirlenmesinde önemli bir ipucu olabilirdi.

3.11 Etik Kurul Onayı

Çalışmamız DEÜ Girişimsel olmayan klinik çalışmalar etik kurulu onayı ile başlatılmıştır. Karar Onay no. 2022/11-22 (Ek-1)

4. BULGULAR

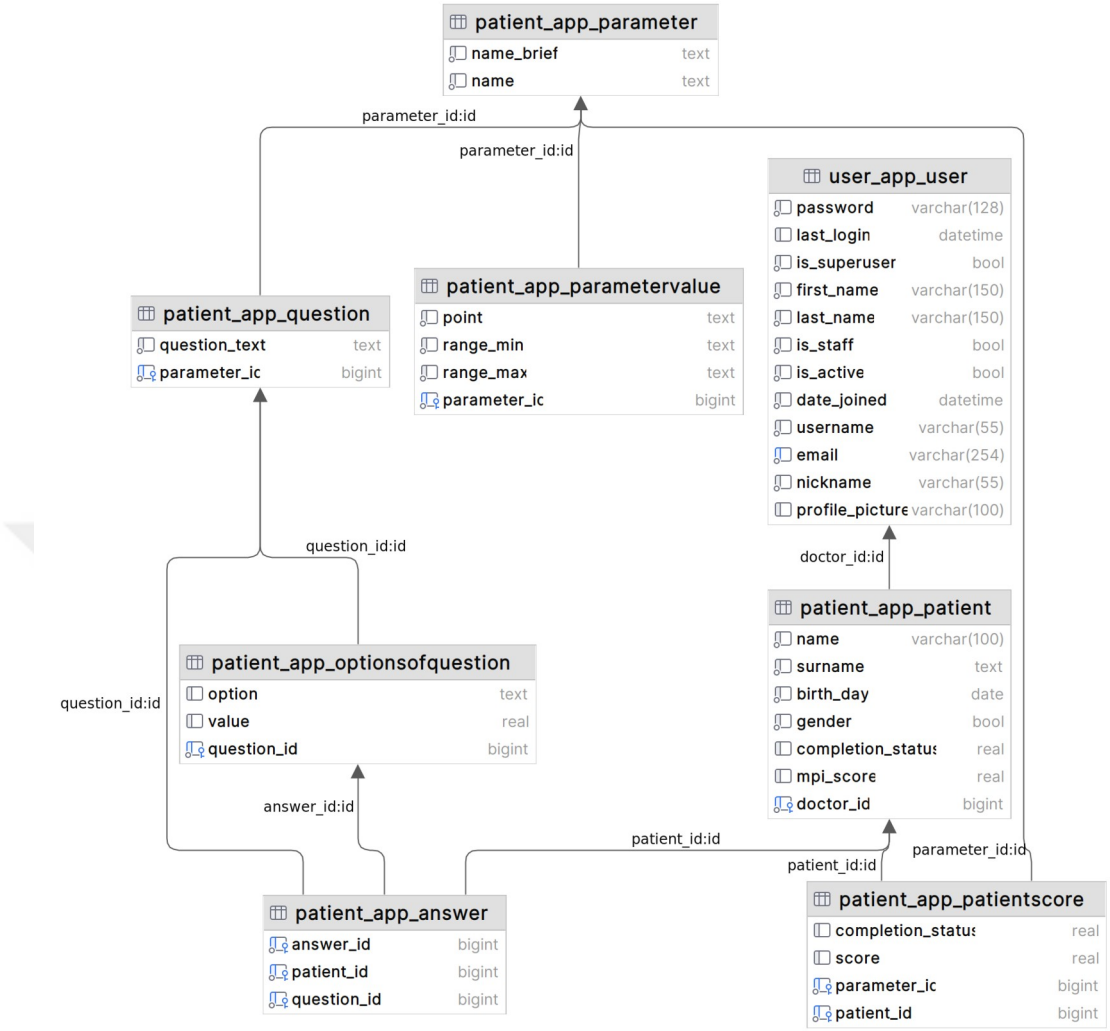
4.1 Veri tabanı Tasarımı

ÇBPİ için veri tabanı oluşturulması için bu tablo baz alınmıştır. Öncelikle elde edilen her bir veri bir hasta için kullanılacağından dolayı hastaların kayıtlarının tutulduğu bir 'patient' tablosu oluşturuldu. Bu tabloda hasta mahremiyeti açısından çok sorun teşkil etmeyecek olan ad, soyad, yaş, cinsiyet bilgileri istendi.

Tablo 4: ÇBPİ hesaplama tablosu

	DÜŞÜK (SKOR = 0)	**HAFİF (SKOR = 0,5)**	**YÜKSEK (SKOR = 1)**
KTMDA	0 – 3	4-7	8-10
ESÖ	16 - 20	10-15	5-9
GYA	6 - 5	4-3	2-0
AGYA	8 - 6	5-4	3-0
KHDÖ	0	1-2	>= 3
MBD	>= 24	17-23,5	<17
İLAÇ SAYISI	0 – 3	4-6	>= 7
KİMİNLE YAŞIYOR	AİLE İLE	BİR KURUMDA	YALNIZ
RİSK	0 – 0,33	0,34 – 0,66	0,67 – 1

Şekil 3'te ÇBPİ uygulamasında kullanılan veri tabanının bir kısım tabloları ilişkisel olarak gösterilmiştir. Veri tabanında, parametreler için parametre kısaltması ve parametre açıklamasının yer aldığı 'parameter' tablosu oluşturulmuştur. Parametrelerin skor aralıkları ve karşılık geldiği puanları içeren veriler 'parametervalue' tablosunda tutulmuştur. Her bir parametrenin ilgili puanını ortaya çıkaran değerlendirme soruları 'question' tablosunda, her soruya karşılık gelen seçenekleri içeren verilerde 'option' tablosunda tutulmaktadır. 'user' tablosunda hasta takibini ve ÇBPİ değerlendirmesini yapan kullanıcılar tutulmaktadır.



Şekil 3: Veri tabanı diyagramı

4.2 Sunucu Uygulaması Tasarımı

Sunucu bilgisayarında istenen verilerin veri tabanından sorgulanarak ÇBPİ uygulamasına aktarılmasını sağlayan, UPA katmanı olarak tanımlayabileceğimiz sunucu uygulamasında; Python programlama dili ile üretilmiş Django Rest Framework (DRF) kullanılmıştır.

ÇBPİ uygulamasında yapılan, veri tabanından bilgi çeken her işlemin burada bir karşılığı bulunmaktadır. DRF' de uygulamanın genel ayarları uygulamanın ana

‘configurations’ dizininde ‘settings.py’ adındaki ayalar dosyasında (Şekil 4) tutulmaktadır. Bu dosyada e-posta ayarları, hesap ayarları, klasör, dizin, zaman ayarları ve uygulamaya ait diğer ayarlar bulunmaktadır. Diğer ülkelerde de kullanılabilmesi için uygulama, İngilizce olarak kodlanmıştır.

```
DATABASES = {
'default': {
    'ENGINE': 'django.db.backends.sqlite3',
    'NAME': BASE_DIR / 'db.sqlite3',
    }
}

DEBUG=FALSE

TIME_ZONE = 'Europe/Istanbul'

LANGUAGE_CODE = 'tr-TR'

STATIC_URL = '/static/'

STATIC_ROOT = os.path.join(BASE_DIR, 'static')

MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

MEDIA_URL = '/media/'

ACCOUNT_EMAIL_REQUIRED = True

ACCOUNT_UNIQUE_EMAIL = True

REST_AUTH_REGISTER_SERIALIZERS = {

    "REGISTER_SERIALIZER": "user_app.serializers.NewRegisterSerializer"

}
```

Şekil 4: Ayarlar dosyası örneği

ÇBPİ servisinin hizmet verebilmesi için birkaç servisin çalışıyor olması gerekmektedir. Nginx’ten yönlendirilen isteğin yönlendirme tablosunda hangi fonksiyonel hizmete gideceği belirlenmiş olur (Şekil 5). Buradaki her tanımlamanın uygulama tarafında karşılığı bulunmaktadır. Bu sayede gelen isteğe, doğru yönlendirme ile doğru bilgi aktarılabilmektedir.

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('user/', include("user_app.urls")),
    path('patient/', include("patient_app.urls")),
    path("privacy/", get_privacy),
    path("deletion-instruction/", get_deletion_instruction),
]
```

Şekil 5: Örnek yönlendirme tablosu

Yönlendirme tablosu yolu ile gönderilen isteği Kontrolcü adında fonksiyonlar karşılar. Bunlar gelen isteğin metoduna göre cevap belirleme yeteneğine sahip olan fonksiyonlardır.

```
@api_view(['GET', 'PUT', 'DELETE'])
def patient_get_update_delete(request, pk):
        try:
                patient = Patient.objects.get(pk=pk)
        except Patient.DoesNotExist:
                return HttpResponse(status=404)
        if request.method == 'GET':
                serializer = PatientSaveSerializer(patient)
                return JsonResponse(serializer.data)
        elif request.method == 'PUT':
                serializer = PatientSaveSerializer(patient, data=request.data)
                if serializer.is_valid():
                        serializer.save()
                        return JsonResponse(serializer.data)
                return JsonResponse(serializer.errors, status=400)
        elif request.method == 'DELETE':
                patient.delete()
                return HttpResponse(status=204)
```

Şekil 6: Hasta kayıtlarını sağlayan kontrolcü

Şekil 6'da GET, PUT, DELETE yöntemlerine cevap vermesi için planlanan bir fonksiyon oluşturulmuştur. Bu fonksiyon sayesinde hasta bilgilerinin görülmesi, güncellenmesi ya da silinmesi mümkün olmaktadır.

DRF'nin sağladığı araçlar ile bunları yapmak daha da kolay olmaktadır. Örneğin 'RetrieveUpdateDestroyAPIView' sınıfını kullanarak nesne üzerinde GET,

PUT, DELETE işlemlerini kolayca yapabilmektedir. Şekil 7'de 'ListCreateAPIView' sınıfı ile listeleme ve yeni kayıt işlemleri kolayca yapılabilmektedir.

```
class PatientAnswersAPI(ListCreateAPIView):
    queryset = Answer.objects.all().prefetch_related("patients", "questions", "answers")
    serializer_class = AnswerSerializer
```

Şekil 7: DRF kütüphanesi ile oluşturulan kontrolcü

Serileştiriciler, karmaşık Python veri türlerinin JavaScript Nesne Gösterimi (json) , Genişletilebilir İşaretleme Dili (xml) gibi basit veri türlerine dönüşümüne ya da tam tersi dönüşümlere olanak tanır. Özellikle DRF de veri iletiminde yoğun olarak kullanılan bir araçtır.

Şekil 8'de ÇBPİ'nin sunucu uygulamasında kullanılan bir kısım Serileştirici örnekleri gösterilmiştir. Buradaki 'PatientSerializerWithDoctors' serileştiricisi doktor bilgileri ile birlikte o doktora ait hasta bilgilerini de uygulama tarafına sunmaktadır. Bu sayede ÇBPİ mobil uygulamasında doktor (kullanıcı) uygulamaya giriş yaptığında hasta bilgileri de otomatik olarak getirilebilmektedir. Şekil 9'da serileştirilmiş bir anonim veri örneği gösterilmiştir.

'DoctorSerializerWithPatients' serileştiricisi ise yeni hasta oluşturulduktan sonra oluşturan doktora ait bütün hastaları json veri formatına sertleştirilerek sunulmasını sağlamaktadır. Serileştiriciler sayesinde okunan veri tabanı tablosunun bilgilerine yeni bilgiler eklenerek istemciye sunulabilir. Şekil 8'deki 'age' değişkeni bu duruma bir örnektir.

```
class DoctorSerializerWithPatients(serializers.ModelSerializer):
    patients = serializers.SerializerMethodField("get_patients")
    class Meta:
        model = get_user_model()
        fields = ["id", "nickname", "profile_picture", "email", "patients"]
    @staticmethod
    def get_patients(model: User):
        patients = Patient.objects.filter(doctor=model)
        return PatientSaveSerializer(patients, many=True).data

class PatientSerializerWithDoctors(serializers.ModelSerializer):
    doctor = serializers.SerializerMethodField("get_doctor")
    age = serializers.ReadOnlyField(source='get_age')
    class Meta:
        model = Patient
        fields = ('id', 'name', 'surname', 'birth_day', 'gender', 'doctor',
                  'completion_status', 'mpi_score', 'age', 'doctor_note')
    @staticmethod
    def get_doctor(model: Patient):
        return DoctorSerializer(model.doctor, many=False).data

class PatientSaveSerializer(serializers.ModelSerializer):
    class Meta:
        model = Patient
        fields = "__all__"
```

Şekil 8: Serileştirici örneği

```
{
    "id": 1,
    "name": "Ege",
    "surname": "Ege",
    "birth_day": "1973-06-19",
    "gender": false,
    "doctor": {
        "id": 49,
        "nickname": "ihsan@deu.edu.tr",
        "profile_picture": "/media/profile/49/43b3e
        "email": "ihsan@deu.edu.tr"
    },
    "completion_status": 0.75,
    "mpi_score": 0.5,
    "age": 50,
    "doctor_note": "Dün gelen hasta"
},
```


Şekil 9: Serileştirilmiş anonim veri örneği

Şekil 10'de veri tabanı ile bağlantılarında ilgili tablolardan veri okuyarak, bunlar üzerinde değişiklikler yapılmasını sağlayacak olan model sınıfı gösterilmiştir. Burada Parametre tablosundan veri okunmasını ve okunan veri üzerinde başka işlemler yapılmasını sağlayacak olan Parametre modeli oluşturulmuştur.

```
class Parameter(models.Model):
    name_brief = models.TextField(max_length=10, null=False, blank=False)
    name = models.TextField(max_length=1000, null=False, blank=False)
```

Şekil 10: Parametre tablosu için model örneği

4.3 Kullanıcı Arayüzü Tasarımı

ÇBPİ uygulamasının tasarımında, önceliklerden biri kullanıcı dostu bir uygulama olmasıdır. Aynı zamanda kararlı yapısı ile kullanıcıların memnuniyeti artırılmak istenmiştir. ÇBPİ'nin işlevselliğinin uygulama üzerinde hayata geçirilmesi hayal gücü ve dikkat gerektirmektedir. Kullanıcı uygulamada istenilen işlemleri akıcı bir şekilde yerine getirebilmelidir. Bunun için uygulamadaki ekranların tasarımı

büyük önem arz etmektedir.

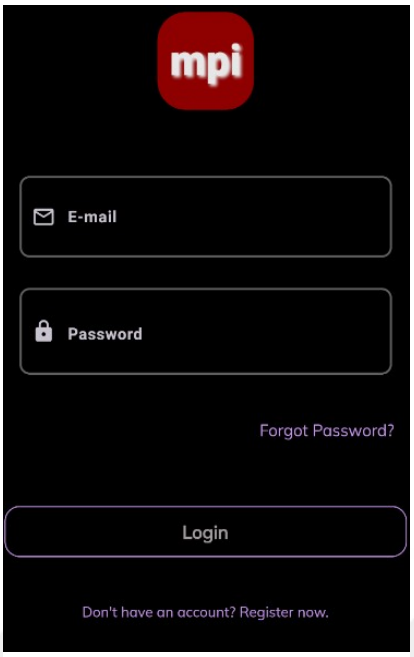
Durum yöneticisi olarak, kullanım kolaylığı ve BLOC (Business Logic Component) yapısı ile uyumlu olması gibi artıları nedeniyle Cubit tercih edilmiştir. Şekil 11'de Patient ve User sınıflarının durumunu takip ederek değişimleri kullanıcı arayüzüne bildiren bir Cubit durum yöneticisi kullanımı gösterilmiştir.

```
class UserCubit extends Cubit<User> {
  UserCubit(super.initialState);
}

class PatientCubit extends Cubit<Patient> {
  PatientCubit(super.initialState);
}
```

Şekil 11: Cubit durum yöneticisi

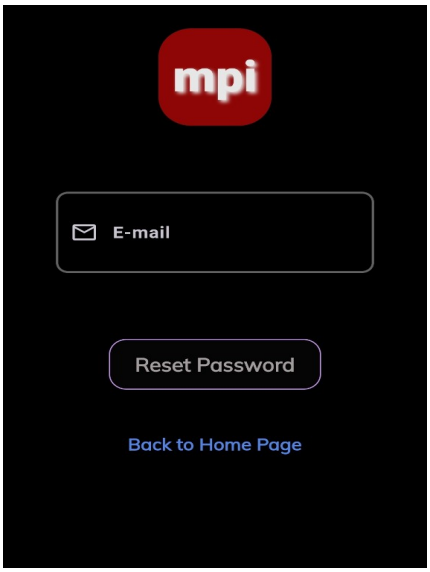
Hasta kayıtlarının doktor bazlı tutulabilmesi, güvenlik amaçlı olarak kullanıcıların yalnızca kendi hastalarını görebilmeleri, her yerden uygulamadaki verilerini oturum açarak görebilecek olmaları, kendi mobil cihazı kaybolsa bile başka bir mobil cihazdan giriş yaparak yine verilerini görebilecek olmaları gibi nedenlerle uygulama hesap sistemi ile giriş yapılacak şekilde oluşturulmuştur.



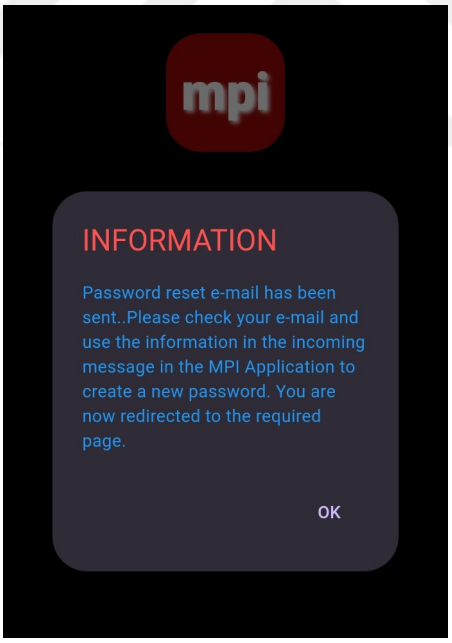
Şekil 12: Kullanıcı giriş ekranı

Şekil 12'de ÇBPİ uygulamasında kullanıcı giriş ekranı gösterilmektedir. Ekranda, uygulama giriş parolasını unutan kullanıcılar için Parola Yenileme ekranı ve hesabı henüz olmayan kullanıcılar için Kayıt ekranı ulaşım linkleri bulunmaktadır. Sayfada hata ve yazım kontrolleri gerçekleştirilmiştir. Formlarda 'flutter_fast_forms' kütüphanesi kullanılmıştır. Bu sayede gerekli formların üretilmesi daha hızlı gerçekleşmektedir.

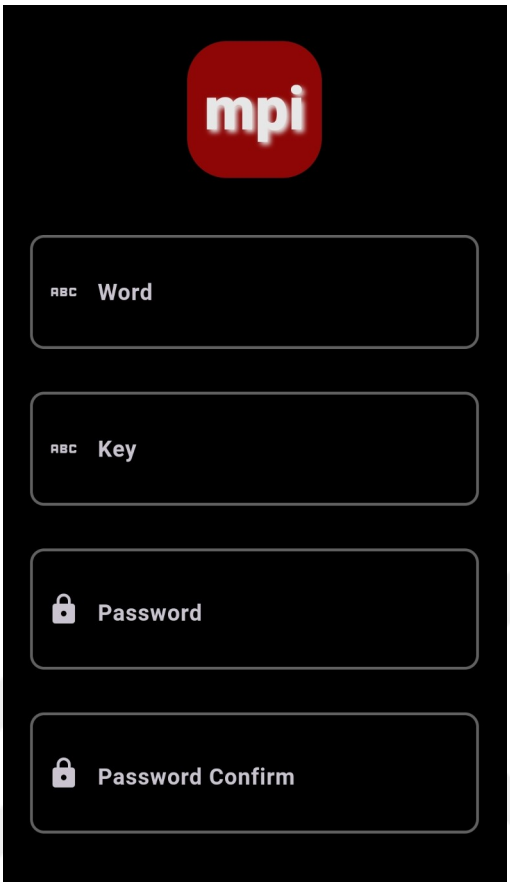
Şekil 13'te şifremi unuttum yazısına tıklandığında gelen sayfanın görünümü verilmiştir. Kullanıcı ilgili alana sisteme kayıt olurken kullandığı e-posta adresini girerek 'Reset Password' butonuna tıkladığında, sistem otomatik olarak şifresini yenileyebileceği bilgi ve yönergeyi, girdiği e-posta adresine gönderecektir. Bu işlemle birlikte kullanıcı, gerçekleşmesi planlanan eylemlerin bilgisini içeren bir uyarı mesajı (Şekil 14) görüp, onayladıktan sonra, e-posta adresine gönderilen bilgileri kullanabileceği bir sayfaya yönlendirilmektedir.



Şekil 13: Şifremi unuttum ekranı



Şekil 14: Şifremi Unuttum tıklandığında görülen uyarı

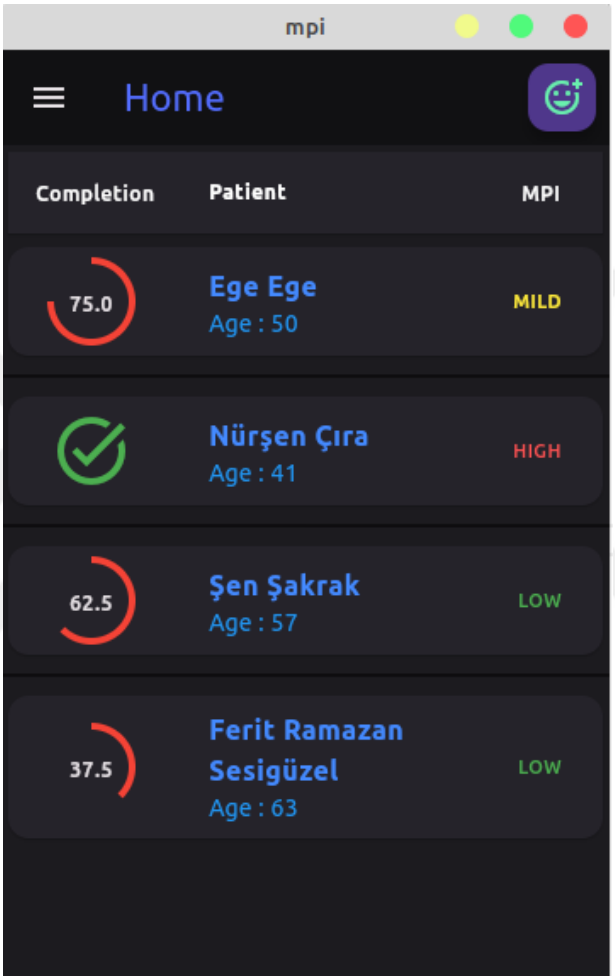


Şekil 15: Şifre yenileme formu

Kullanıcı ‘Reset Password’ butonuna tıkladığında gönderilen bilgiler, kullanıcının doğruluğunu e-posta yoluyla sınamak için kullanılan bir kelime ve bir anahtardan mevcuttur. Kullanıcı bu bilgileri Şekil 15’te gösterilen alanlara girerek unuttuğu var olan şifresini değiştirebilir.

Şekil 16’da kullanıcı başarılı bir şekilde sisteme giriş yaptıktan sonra karşılaştığı Ana ekran gösterilmiştir. Kullanıcı daha önce sisteme girerek hasta oluşturmuş ise onlar bu sayfada görünmektedir. Örnek şekilde anonim olarak oluşturulmuş dört adet hasta bilgisi örneği bulunmaktadır. Bu sayfada hasta ile ilgili tüm işlemler yapılabilmektedir. Hastanın sağa ya da sola kaydırıldığında o hasta ile ilgili yapılabilecek işlemler görülmektedir. Sayfanın sağ üst köşesindeki simgeye tıklandığında yeni hasta kaydının yapılabileceği ekrana ulaşılır.

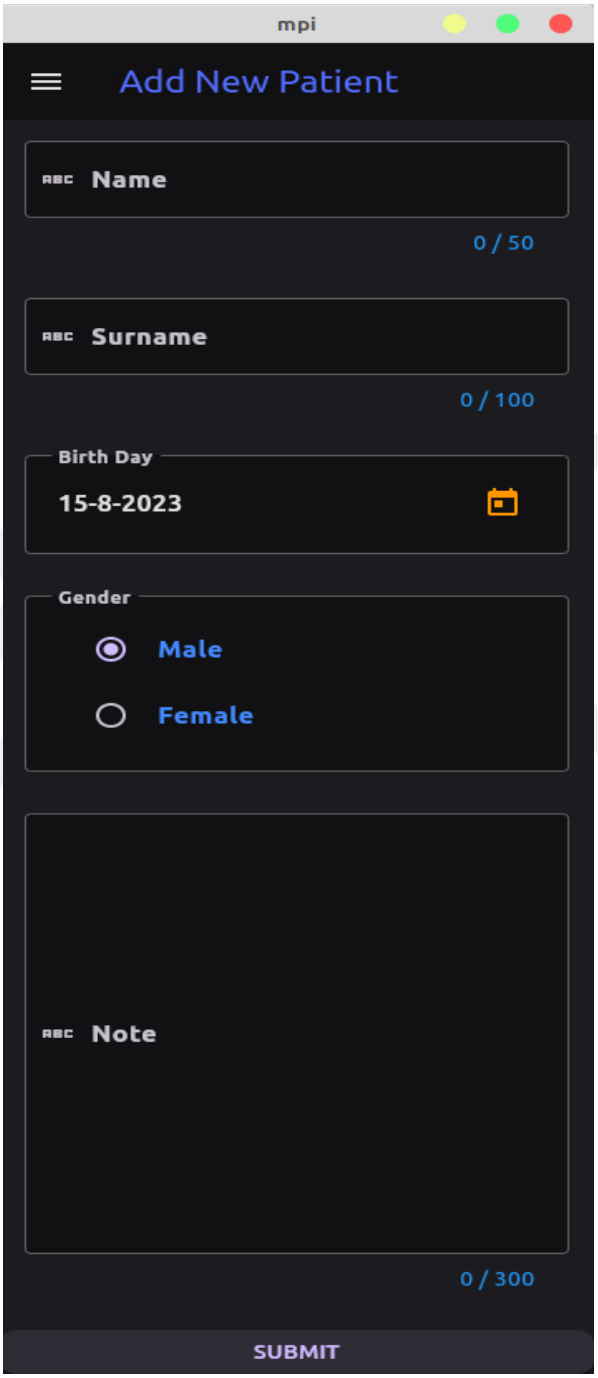
Bu sayfada soldaki alan hastanın ÇBPİ değerlendirmesindeki yüzdelik tamamlanma oranını göstermektedir. Örnek olarak Şen Şakrak hastasının ÇBPİ değerlendirmesindeki tamamlanma oranı %62,5 olarak bulunmuştur. Bu tamamlanma oranı ile ÇBPİ değeri düşük olarak bulunmuştur. Gerçek ÇBPİ değeri ÇBPİ değerlendirmesinin %100 olarak tamamlanmasından sonra ortaya çıkacaktır.



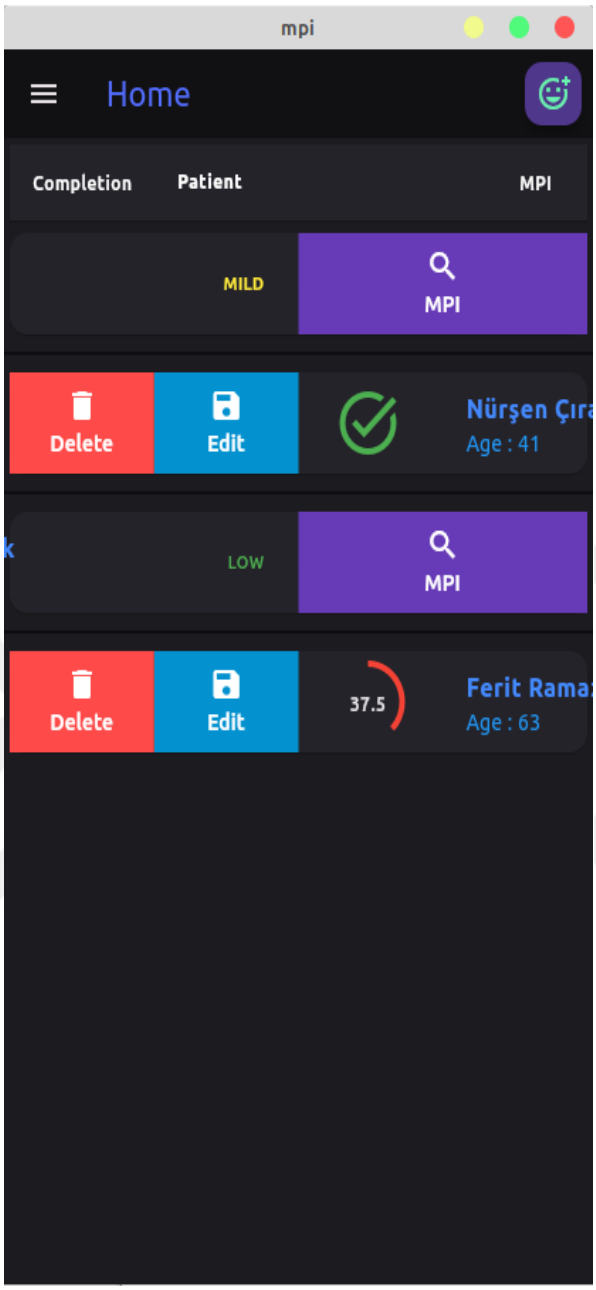
Şekil 16: Kullanıcı karşılama ekranı

Şekil 17'de yeni hasta kaydının yapılabildiği ekran görünmektedir. Hastanın ad, soyad, yaş, cinsiyet bilgileri ile kullanıcının hasta hakkındaki görüşü, notu istenmektedir. Tüm alanlar düzgün bir şekilde doldurulduktan sonra kullanıcı kaydı

yapılabilmektedir. Doldurulmamış boş bir alan ya da alanlar varsa bu alan ya da alanlar doldurulunca kadar hasta kaydı için veri gönderilmemektedir.

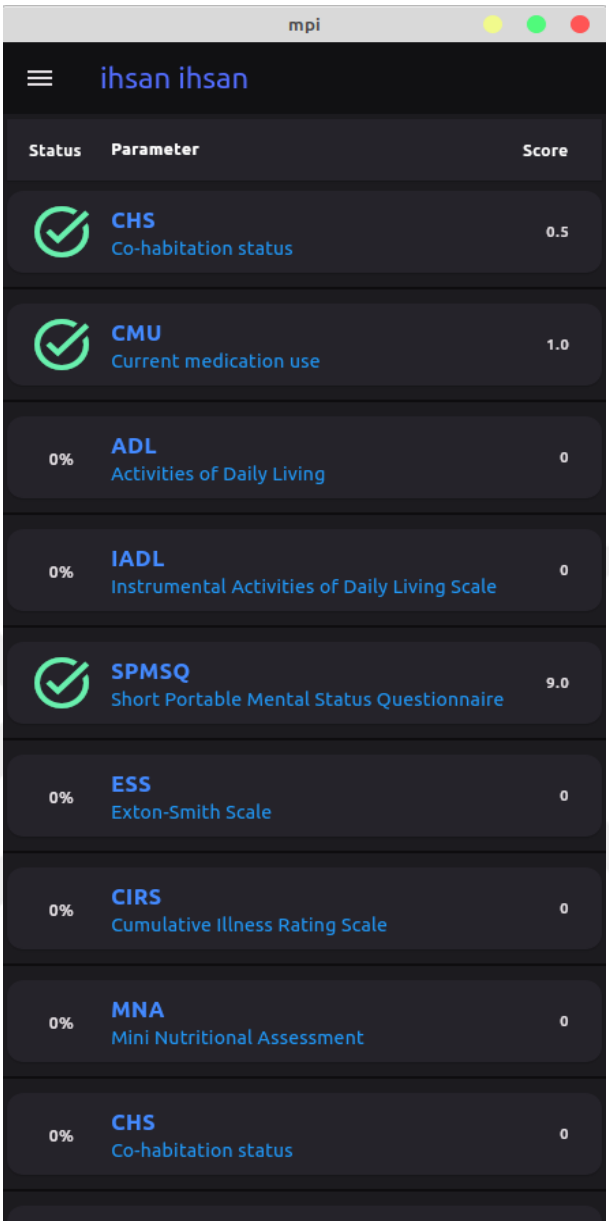


Şekil 17: Yeni hasta kayıt ekranı



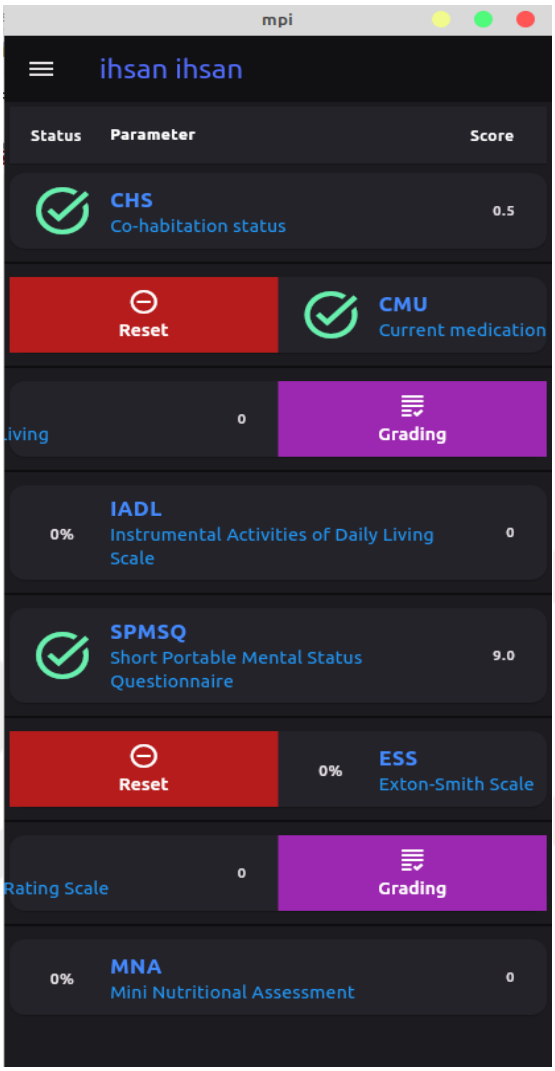
Şekil 18: Kayıtlı hasta işlemleri

Hasta üzerinde yapılabilecek işlemler Silme, Güncelleme ve ÇBPİ işlemleri yapılabilmektedir. Hasta üzerinde parmakla sağa kaydırma işleminden sonra ÇBPİ menüleri görünür hale gelecektir (Şekil 18).



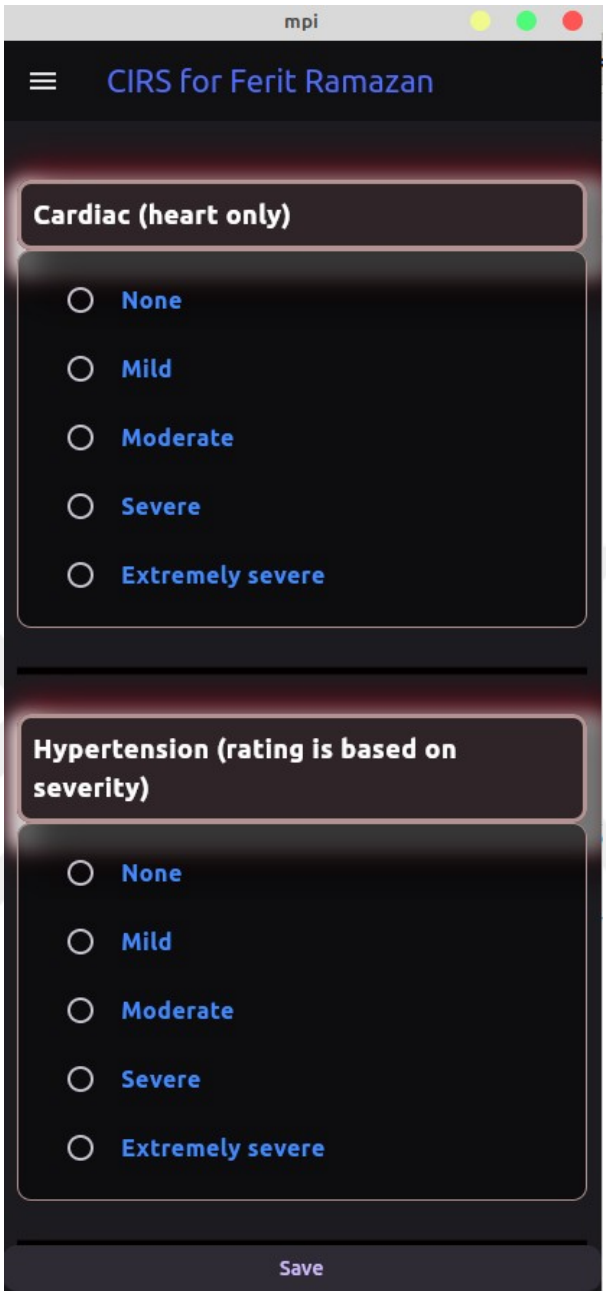
Şekil 19: Hasta parametreleri

Hastanın ÇBPİ değerlendirmesi butonuna tıklanıldığında Şekil 19'da olduğu gibi bir ekran gelmektedir. Solda parametrenin değerlendirmesinin tamamlanıp tamamlanmadığı, ortada parametrenin İngilizce kısaltması ve altında tam adı, sağda da o parametrenin değerlendirme skoru yer almaktadır.



Şekil 20: Parametre üzerindeki işlemler

Parametre üzerinde yapılan işlemler Şekil 20' de olduğu gibidir. Parametrenin üzerinde sağa kaydırıldığında parametre değerlendirmesinin sıfırlanma menüsü, sola kaydırıldığında ise parametrenin değerlendirme menüsü görünmektedir.



Şekil 21: Parametre değerlendirmesi

Her parametrenin uzmanlar tarafından belirlenmiş bazı değerlendirme soruları, kriterleri ve bunların puanları vardır. Bu kriterler, sorular ve puanlar veri tabanına eklenmiştir.

4.4 Sunucu Sistemin Hazırlanması

Sunucu işletim sistemi olarak ücretsiz bir dağıtım olan Ubuntu Sunucu sistemi tercih edildi. Sunucu sistemde kaynak tüketimi, modülerlik ve bakım kolaylığı gibi nedenlerle Docker sanallaştırma teknolojisi kullanılmıştır. Bir uygulamanın servisi ve bir de Nginx web servisi olmak üzere iki adet servis oluşturuldu. Nginx web servisi gelen istekleri uygulama servisine yönlendireceği için öncelikle uygulama servisinin çalıştırılması sonrasında da Nginx web servisinin çalışması gerekmektedir.

```
FROM python:3.11
ENV MPI=/home/app/mpi
RUN mkdir -p $MPI
WORKDIR $MPI
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1
COPY ./requirements.txt $MPI
RUN pip install --upgrade pip
COPY . $MPI
RUN pip install -r requirements.txt
```

Şekil 22: Dockerfile dosyası

Docker teknolojisinde docker-compose ile birden fazla servis planlayabilir ve bunları kolaylıkla yönetilebilmektedir. Şekil 22'de ÇBPİ uygulama servisinin çalışacağı sanal ortamın ayarlarının olduğu Dockerfile dosyasının içeriği gösterilmiştir. Bu dosya ile servisin Python3.11 versiyonu üzerinde çalışması, çalışma yolunun ne olduğu, gerekli paketlerin kurulması gibi işlemler sağlanmaktadır. Bu dosya docker-compose.yml dosyası tarafından kullanılmıştır. Bu sayede docker-compose servisi başlatıp durdurma yeteneğine sahip olmuştur.

```
version: '3'
services:
  mpiapp:
    build: .
    command: bash -c "python manage.py collectstatic --noinput &&
    gunicorn --certfile=/etc/certs/mpi.pem --keyfile=/etc/certs/mpi.key
    configurations.wsgi:application --bind 0.0.0.0:8002  --timeout 1800"
    container_name: mpi
    env_file:
      - docker_config/mpi/mpi-variables.env
    volumes:
      - .:/home/app/mpi:rw
      - ./media:/home/app/mpi/media
      - ./static:/home/app/mpi/static
      - ./docker_config/nginx/certs/:/etc/certs
    expose:
      - "8002"
    restart: always
  mpiNginx:
    image: nginx:latest
    container_name: ng
    ports:
      - "8002:8002"
    volumes:
      - ./docker_config/nginx/:/etc/nginx/conf.d
      - ./static:/home/app/mpi/static
      - ./media:/home/app/mpi/media
    depends_on:
      - mpiapp
```

Şekil 23: docker-compose.yml dosyası

Docker-compose dosyası ‘yml’ ya da ‘yaml’ uzantılı olabilmektedir. Şekil 23’te, sunucu sisteminde kullanılan docker-compose.yml dosyasının içeriği gösterilmiştir. Uygulama servisi dosya içerisinde ‘mpiapp’ olarak, web servisi ise ‘mpiNginx’ olarak adlandırılmıştır. ÇBPİ servisinin konteyner adı ‘mpi’, web servisinin konteyner adı ise ‘ng’ olarak atanmıştır.

‘mpiapp’ servisi başlatılırken, DRF ayarlarına ait bazı işlemlerin öncelikle, medya dosyalarının sunucudan sunulabilmesi, veri modellerindeki değişikliklerin veri tabanına yansıtılabilmesi, 8002 numaralı kanaldan yayın yapılması gibi ayarların gerçekleştirilmesi gerekmektedir. ‘env_file’ satırında servis başlatılırken tanımlanan, hata ayıklama ayarının ‘False’ olması gibi bazı değerler bulunmaktadır.

Servisin güvenli olarak sertifikalandırılması için gerekli olan dosyalar ve klasör işlemleri ‘volumes’ altında ayarlanmıştır. Hata durumlarında otomatik olarak servisin yeniden başlatılması istenmiştir.

```
CONTAINER ID   IMAGE                STATUS       PORTS          NAMES
d23fdc338aed   nginx:latest         running      8002:8002       ng
653ad6a27fe4   tez_backend_mpiapp   running      8002/tcp        mpi
```

Şekil 24: Çalışan konteynerler

Sunucu sistemde çalışan konteynerlerin durumu Şekil 24’te gösterilmiştir. ‘mpi’ servisi 8002 kanalını dinlemektedir.

```
server {
    listen          443 ssl;
    server_name     appservice.deu.edu.tr;
    ssl_certificate /etc/nginx/conf.d/certs/mpi.pem;
    ssl_certificate_key /etc/nginx/conf.d/certs/mpi.key;
}
upstream mpiapp {
    ip_hash;
    server mpiapp:8002;
}
server {
    listen 8002 ssl;
    location / {
        proxy_pass https://mpiapp;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        client_max_body_size 100M;
    }
    server_name appservice.deu.edu.tr;
    ssl_certificate /etc/nginx/conf.d/certs/mpi.pem;
    ssl_certificate_key /etc/nginx/conf.d/certs/mpi.key;
    location /media/ {
        alias /home/app/mpi/media/;
    }
    location /static/ {
        alias /home/app/mpi/static/;
    }
    location /favicon.ico {
        access_log off;
        log_not_found off;
    }
}
```

Şekil 25: Nginx ayar dosyası

Ngnix hizmetinin nasıl çalışması gerektiğini belirleyen ‘ngnix.conf’ dosyasının içeriği Şekil 25’te gösterilmiştir. Bu ayarlar sayesinde Nginx 8002 ve 443 numaralı kanalı dinler ve ‘upstream’ de belirtilen uygulamaya yönlendirir.

4.5 Play Store İşlemleri

Play Store; Android işletim sistemine ait uygulamaların sunulduğu bir uygulama havuzudur. Başkaları tarafından ulaşılması istenen uygulamaların bu havuza konulması gerekmektedir. Fakat bu havuza uygulama atılırken bazı standartların sağlanması ve ayarların yapılması gerekmektedir.

Öncelikle bir Google geliştirici hesabının alınması gerekmektedir. Bunun için Google Play Console (40) web adresine giderek kayıt olunması ve giriş yapılması

gerekmektedir. Giriş yapıldıktan sonra sağlanan kılavuzlar yardımıyla ilgili uygulamanın havuzda görünür olması sağlanabilmektedir.

Google firması bu havuza atılacak uygulamalarda, veri gizliliği açısından, kullanıcıların verilerini silebilme özelliğinin destek hattı yolu ile veya belirli süre aralıkla otomatik olarak, ya da uygulama içerisinden uygulanabilmesini gerekli görmektedir. Bu amaçla, ÇBPİ uygulamasında 'Delete' menusu altında, kullanıcının kendi profil resmini ya da hesap bilgilerini tamamen silmesini sağlayacağı menüler yer almaktadır. Aynı zamanda bununla ilgili bilgi 'https://appservice.deu.edu.tr:8002/deletion-instruction/' adresine yüklenmiştir.

Google tarafından istenen özelliklerden bir diğeri, uygulamada gizlilik politikasının bulunmasıdır. ÇBPİ uygulamasının gizlilik politikası oluşturulmuş ve "https://appservice.deu.edu.tr:8002/privacy/" adresine yüklenmiştir.

Mobil uygulama tarafında yapılması gereken birtakım işlemler bulunmaktadır. Bunlardan birisi, uygulama için uygulamayı temsil edebilecek bir ikon belirlenmesidir. Bu konuda çevrim içi olarak ikon oluşturma sitelerinden (41) faydalanmak mümkündür. İkon oluşturulduktan sonra Flutter için bir ikon uygulama paketi olan 'flutter_launcher_icons' (42) paketi kurulmalıdır. Paket kurulduktan sonra paketin web adresindeki yönergelerin izlenerek ikon kurulumunun tamamlanması gerekir.

Son olarak Flutter belgelerinde gösterilen kılavuzun (43) adımları takip edilerek uygulama tarafındaki hazırlık süreci tamamlanır.



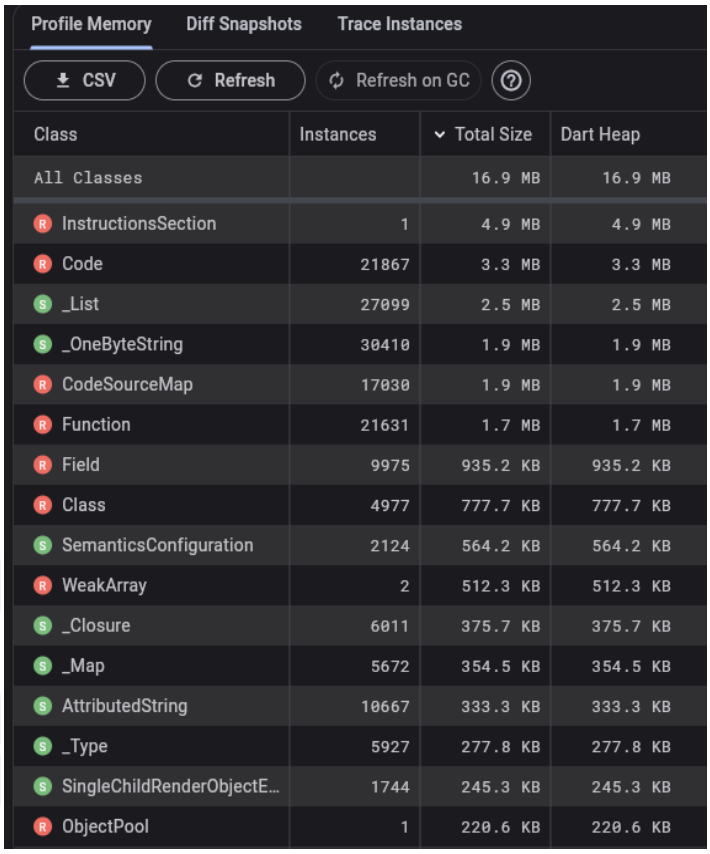
Şekil 26: Performans analizi

4.5.1 Uygulama Performans Değerlendirmesi

Uygulamanın Flutter Geliştirme Aracı ile yapılan testlerinde çıkan sonuçlar değerlendirilmiştir. Bu araç uygulamalardaki düşük performansın nedeninin belirlenmesine yardımcı olacak çeşitli araçlardan oluşur.

Flutterda, normal koşullarda yaklaşık olarak her 16 ms'de bir, kullanıcı arabiriminin animasyonları veya kullanıcı ara birimindeki diğer değişiklikleri yansıtacak şekilde güncelleştirilmektedir. Oluşturulması 16 ms'den uzun süren bir kare, görüntüleme cihazında hatalara neden olabileceği anlamına gelmektedir. Buna göre Şekil 26'daki ÇBPİ uygulamasının performans sonuçlarının başarılı olduğu görünmektedir.

Uygulamaların kullanıldığı cihazlarda, uygulama bellek kullanımı fazla veya tutarsız ise, kullanıldığı cihazın yavaşlamasına, kilitlenmesine ya da kapanmasına neden olabilmektedir. Bu nedenle uygulamaların kullandığı bellek miktarının cihaz tarafından belirlenen sınırlar dışına taşmaması çok önem arz etmektedir.



Şekil 27: Hafıza kullanımı

Uygulamamızın bellek kullanım bilgilerini içeren test sonucu Şekil 27'de gösterilmiştir. Buna göre uygulamamız bellekte toplamda 16,9 mb yer kaplamaktadır. Bu da normal bir uygulamada bellek kullanım miktarı kabul edilebilir bir büyüklüktür ve uygulamanın hafif bir uygulama olduğu anlamını taşımaktadır.

Uygulamamızın ağ analizi sonuçları Şekil 28'de görünmektedir. Bulunulan ağın hızı, sorguların karmaşıklığı, sunucu ve veri tabanı sistemlerinin ve ağlarının o andaki durumları bu tür ağ test sonuçlarını tam olarak etkilemektedir. Bu testin yapıldığı bölgede, internet bağlantının hızının yavaş olmasına rağmen uygulamanın başarısı nedeniyle işlemler seri olarak gerçekleştirilebilmiştir.

Method	Uri	Status	Type	Duration	Timestamp
GET	https://appservice.deu.edu.tr:8002/patient/getPatier	200	json	194 ms	18:56:58.197
GET	https://appservice.deu.edu.tr:8002/patient/getParan	200	json	158 ms	18:57:02.780
GET	InternetAddress('161.9.173.20', IPv4)	101	ws	110 ms	18:57:02.829
GET	https://appservice.deu.edu.tr:8002/patient/getQuest	200	json	195 ms	18:57:05.430
GET	https://appservice.deu.edu.tr:8002/patient/getPatier	200	json	1 s	18:57:19.930

Şekil 28: Ağ analizi

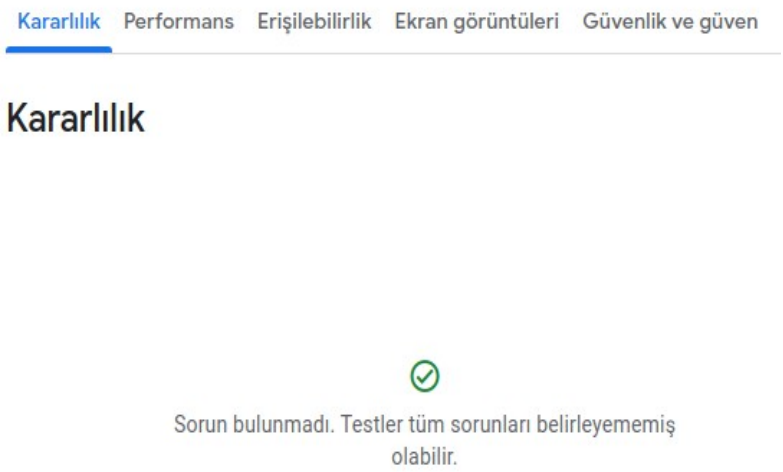
4.5.2 Google Play Analizleri

Uygulamamız Google Play Store'da yayınlanmadan önce, farklı cihazlarda birtakım testlere tabi tutulmuştur.

Cihaz	Ekran boyutu	Android sürümü	RAM (toplam bellek)	ABI
motorola Motorola G20	720x1600	Android 11 (SDK 30)	4.096 MB	arm64-v8a
google Pixel 5	1080x2340	Android 11 (SDK 30)	8.192 MB	arm64-v8a
samsung SM-G981U1	1440x3200	Android 10 (SDK 29)	12.288 MB	arm64-v8a
google Pixel 6	1080x2400	Android 12 (SDK 31)	8.192 MB	arm64-v8a
google Pixel 7	1080x2400	Android 13 (SDK 33)	8.192 MB	arm64-v8a
Google Pixel 2 (sanal)	1080x1920	Android 12 (SDK 31)	-	x86_64
Google Pixel 2 (sanal)	1080x1920	Android 13 (SDK 33)	-	x86_64

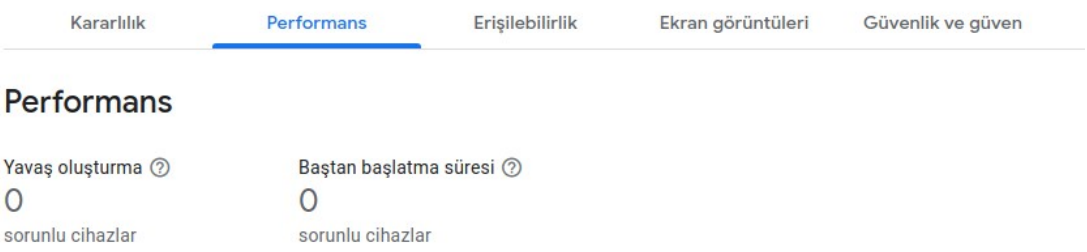
Şekil 29: Uygulamanın test edildiği cihazlar

Google tarafından uygulamanın test edildiği cihazların listesi Şekil 29'da gösterilmektedir. Bu cihazlarda kararlılık, performans, erişilebilirlik, güvenlik gibi testler uygulanmış ve başarılı sonuçlar elde edilmiştir. Her cihazda ayrı ayrı ekran görüntüleri alınarak uygulanan bir test olduğu için Ekran görüntüleri testi ele alınmamıştır fakat test edilen yedi cihazda da ekran görüntüleri alınabilmiş ve test sonucu başarılı bulunmuştur.



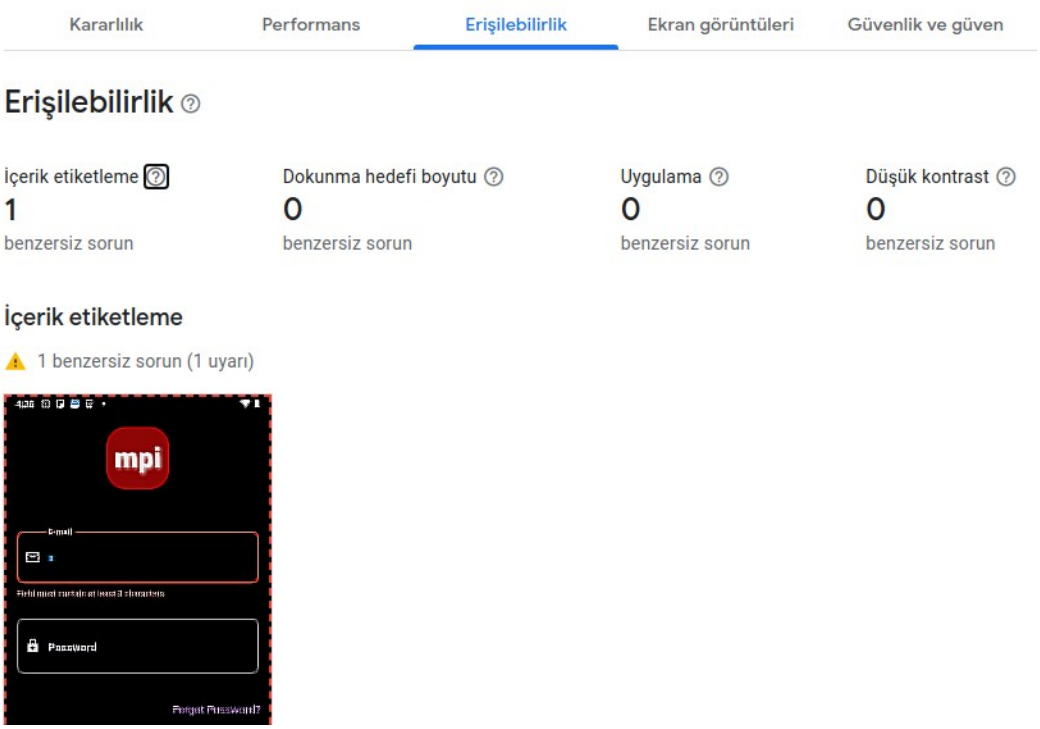
Şekil 30: Google kararlılık testi

Google Play tarafından yapılan kararlılık testi sonucu Şekil 30'da gösterilmektedir. Sonuç uygulamanın başarısını sunmaktadır.



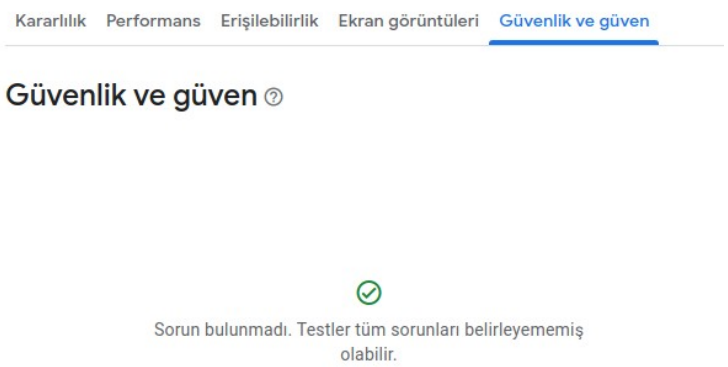
Şekil 31: Google performans testi

Google Play tarafından yapılan performans testi sonuçları da Şekil 31'de gösterilmektedir. Test edilen yedi cihazda da performans sorunu oluşmadığı saptanmıştır.



Şekil 32: Google erişilebilirlik testi

Google tarafından yapılan erişilebilirlik testinin sonucu Şekil 32'de gösterilmiştir. Bu test dört aşamada yapılmış ve yalnızca 'içerik etiketleme' aşamasında, giriş sayfasında yer alan e-posta giriş alanının uyarısının doğru isimlendirilmediğinin bilgisi verilmiştir. Buna istinaden uygulamanın bir sonraki sürümünde bu uyarı düzeltilmiştir. 'Dokunma hedefi boyutu', 'Uygulama' ve 'Düşük kontrast' aşamalarından sorunsuz bir şekilde geçmiştir.



Şekil 33: Google güvenlik testi

Google tarafından yapılan güvenlik testi sonucu Şekil 33'te gösterilmiştir. Google tarafından uygulamamıza yönelik güvenlik açıkları taranmış ve hiçbir güvenlik açığı ile karşılaşılmadığı görülmüştür.

4.5.3 Anket Sonuçları

Geliştirilen ÇBPİ uygulamasında, sistem içerisinde gömülü olarak, çevrim içi yanıtlanabilen ve yanıtlar üzerinden sistemi geliştirmek için bilgi toplayan, 6 olumlu 4 olumsuz likert tipi değerlendirme soruları olan bir değerlendirme anketi oluşturuldu. Bu sorulara 42 kayıtlı kullanıcıdan 28 katılımcının verdiği yanıtların dağılımı aşağıda sunulmuştur.

Uygulama ile ilgili olumlu sorulara (uygulamanın kolaylığı, güvenli olması, başkalarına önerme, kullanımda kolay ilerleme, sıklıkla kullanma, ÇBPİ değerlendirmesini kolaylaştırma) verilen yanıtlarda olumlu yanıt oranı %100.0 ile 64.3 arasındadır, uygulama ile ilgili olumlu yöndeki sorulara olumlu yanıt oranı yüksektir.

Tablo 5: Uygulamanın kolaylığı ile ilgili yanıtların dağılımı

Uygulamanın kullanımını kolay buldum		
	n	%
Katılıyorum	6	21,4
Kesinlikle katılıyorum	22	78,6
Toplam	28	100

Uygulamanın kolaylığı ile ilgili yöneltilen soruya verilen yanıtların dağılımı Tablo 5'te gösterilmiştir. Katılımcıların tümü uygulamayı kolay bulmuştur. Uygulama kolaylığı kullanımı destekleyeceği için bu bulgu önemlidir.

Tablo 6: Uygulamayı önerme ile ilgili yanıtların dağılımı

Uygulamayı başkalarına önerebilirim		
	n	%
Katılıyorum	8	28,6
Kesinlikle katılıyorum	20	71,4
Toplam	28	100

Uygulamanın başkalarına önerilmesini değerlendiren soruya verilen yanıtların dağılımı Tablo 6'te özetlenmiştir. Katılımcıların hepsinin geliştirilen uygulamayı konu ile ilgili diğer kişilere önereceklerini bildirmesi, uygulamanın yaygınlaştırılması ile ilgili önemli bir bulgudur.

Tablo 7: Uygulamanın karmaşıklığı ile ilgili yanıtların dağılımı

Uygulamayı karmaşık buldum		
	n	%
Keskinlikle katılmıyorum	12	42,9
Katılmıyorum	14	50
Kararsızım	2	7,1
Toplam	28	100

Uygulamanın karmaşıklığını sorgulayan soruya verilen yanıtların dağılımı Tablo 7'da özetlenmiştir. Uygulamanın kullanıcılar açısından karmaşıklığını tespit etmek için yöneltilmiş olan bu soruda kullanıcıların neredeyse tamamının bu duruma katılmıyor oluşu, bütünsel olarak uygulamanın kullanımını artıracağını gösteren bir bulgudur.

Tablo 8: Uygulama güvenliği ile ilgili yanıtların dağılımı

Uygulamayı güvenli buldum		
	n	%
Kararsızım	5	17,9
Katılıyorum	10	35,7
Kesinlikle katılıyorum	13	46,4
Toplam	28	100

Uygulamanın güvenlik açısından kullanıcılardaki izlenimini sorgulan soruya verilen yanıtların dağılımı Tablo 8' de özetlenmiştir. Bu soruya verilen cevaplar, uygulamanın, kullanıcıların izni olmadan hiçbir bilgisini toplamaması, sadece kullanıcının isteği dahilinde fotoğraf yüklemek için veri kullanım izin talebi içermesi kullanıcılar açısından uygulama kullanımında güven hissi oluşturarak uygulama kullanımını etkileyecek önemli bir bulgu olmuştur.

Tablo 9: Uygulamanın gereksizliği ile ilgili yanıtların dağılımı

Uygulamayı gereksiz buldum		
	n	%
Kesinlikle katılmıyorum	15	50
Katılmıyorum	12	42,9
Kararsızım	1	7,1
Toplam	28	100

Uygulamanın kullanıcılar açısından gerekli olup olmadığını gösteren yanıtların dağılımını içeren Tablo 9'de kullanıcıların çok büyük çoğunluğu uygulamayı gerekli olarak görmüştür. Bu durum uygulamanın önemli bir boşluğu dolduracağını gösteren değerli bir bulgudur.

Tablo 10: Uygulamanın kullanımında destek ihtiyacı hakkındaki yanıtların dağılımı

Uygulamayı kullanırken desteğe ihtiyaç duydum		
	n	%
Kesinlikle katılmıyorum	9	32,1
Katılmıyorum	16	57,1
Kararsızım	1	3,6
Katılıyorum	2	7,1
Toplam	28	99,9

Tablo 10'da gösterilen yanıtların dağılımına göre kullanıcıların önemli bir bölümü uygulamayı kullanırken desteğe ihtiyaç duymamıştır. Buradan edinilecek bulgular uygulamanın yardımlar veya ipuçları içermesi gerektiğini gösteren önemli bulgulardır. Edinilen bulguya göre, uygulamada kullanım ipuçlarının bulunması, kullanıcıların tamamını kapsayacak bir memnuniyet oluşturması beklenebilmektedir.

Tablo 11: Uygulamayı kullanırken kolay ilerleme hakkındaki yanıtların dağılımı

Uygulamayı kullanırken kolayca ilerliyorum		
	n	%
Katılmıyorum	2	7,1
Katılıyorum	12	42,9
Kesinlikle katılıyorum	14	50
Toplam	28	100

Tablo 11'da, uygulamayı kullanırken kullanıcıların sayfalarda kolayca ilerleyip ilerleyemediğini sorgulayabilmek, işlemlerin yapılabilirliğinin basit olduğunu ve uygulamanın sürekliliği sağlayabildiğini anlamayı kolaylaştıran bulguya ulaşmamızı sağlayan soruya verilen yanıtların dağılımı gösterilmiştir.

Tablo 12: Uygulamanın sık kullanımı hakkındaki yanıtların dağılımı

Uygulamayı sıklıkla kullanacağımı düşünüyorum		
	n	%
Kararsızım	1	3,6
Katılıyorum	7	25
Kesinlikle katılıyorum	20	71,4
Toplam	28	100

Uygulama kullanımının sıklığıyla ilgili yöneltilen soruya verilen yanıtların dağılımı Tablo 12'de gösterilmiştir. Uygulama kullanımının ne kadar sıklıkta olabileceğini, bu ÇBPİ değerlendirmesinde gerekli bir ihtiyacın giderilebileceğini gösteren önemli bir bulgudur.

Tablo 13: Uygulamada tutarsızlık hakkındaki yanıtların dağılımı

Uygulamada tutarsızlıklar olduğunu düşünüyorum		
	n	%
Kesinlikle katılmıyorum	17	60,7
Katılmıyorum	9	32,1
Kararsızım	2	7,1
Toplam	28	99,9

Uygulamadaki tutarsızlıkların değerlendirilmesini sağlayan soruya verilen yanıtların dağılımı Tablo 13'de özetlenmiştir. Uygulamada mantık hatalarının bulunup bulunmadığını gösteren önemli bir bulgudur.

Tablo 14: Uygulamanın süreci kolaylaştırması hakkındaki yanıtların dağılımı

Uygulama ÇBPİ değerlendirme sürecini kolaylaştıracaktır		
	n	%
Katılıyorum	4	14,3
Kesinlikle katılıyorum	24	85,7
Toplam	28	100,0

Uygulamanın ÇBPİ değerlendirmesini kolaylaştırıp kolaylaştırmadığını gösteren soruya verilen yanıtların dağılımı Tablo 14'te özetlenmiştir. Uygulamanın amacına ulaşıp ulaşmayacağını gösteren çok önemli bir bulgudur.

5. TARTIŞMA

Hasta değerlendirme araçları, hastaların gelecek dönemlerine ait sağlık durumları hakkında öngörüler sağladığı için büyük önem arz etmektedir. Elde edilen veriler sayesinde, ileriye dönük sağlık önlemleri alınabilmekte ve oluşma riski bulunan sorunların önüne geçilebilmektedir. Sağlık problemlerinin giderilmesi maliyetleri en aza indirilebilmektedir. Aynı zamanda kişilerin yaşam konforlarını artırılabileceği tahmin edilmektedir.

Çalışmamız, geliştirilen uygulama içerisinde yapılan anket bulgularıyla, sağlık değerlendirmesinde kullanılan ÇBPİ değerlendirmesinin mobil olarak uygulanmasının önemli bir ihtiyacı gidereceği düşüncesini ortaya koymuştur. Geliştirilen uygulama sayesinde hastalar üzerinde daha hızlı ve doğru değerlendirmeler üretilebileceği, zamandan tasarruf sağlanarak daha verimli sağlık hizmeti sunulabileceği düşünülmektedir.

Veri kayıtları uzak sunucularda tutulabileceği gibi yerel olarak uygulamaların kullanıldığı cihazlarda da tutulabilmektedir. Yerel cihazlarda kullanılmasının bazı avantaj ve dezavantajları bulunacaktır. Uzak sunucuda tutulmayan ve yerel cihazda tutulan veriler, uygulamanın internet erişiminden ve yan etkilerinden bağımsız şekilde çalışmasını sağlarken, kullanılan cihazın kaybolması, çalınması, bozulması gibi durumlarda tamamen kaybolacak ya da başka kişilerin eline geçmesi gibi güvenlik sorunlarına neden olacaktır.

İstemci ve sunucu şeklinde tasarlanarak geliştirilen uygulamalar, sunucu bağımlı çalışmasına karşın, güncellemelerde istemci tarafında daha az değişikliğe gidilmesini sağlayabilmektedir. Veriler sunucu sistemlerinde ek güvenlik önlemleri ile tutulur. İstemciler daha az yükle daha performanslı çalışabilir (44). Geliştirilen uygulamamızda veriler veri tabanında tutulmakta, sunucu uygulaması sayesinde istenen verilere ulaşılabilmektedir.

Uygulamanın kullanıcı arayüzünün tasarımında işlevsellik ve kullanıcı dostu tasarım ön planda tutulmuştur. Bu amaçla Flutter seçilmiştir. Flutter geniş bir araç kütüphanesine sahip olmakla birlikte tasarımda geniş bir kullanım alanı sunmaktadır.

Uygulamamız bu sayede hızlı, kullanıcı dostu bir tasarıma sahip olmuştur.

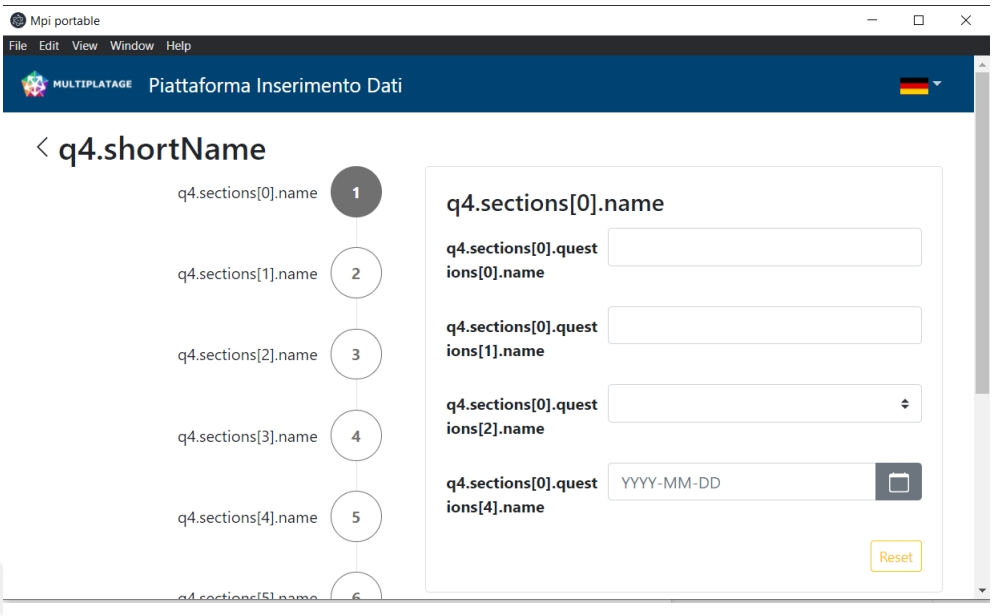
Uygulamanın sunucu tasarımında Docker teknolojisinin kullanılmasıyla sunucu hizmetinin sürekliliği artırılmıştır. Bu teknoloji ile çalıştırılan servisler hata durumlarında sistemlerini otomatik olarak onarmaktadır. Servislerin başka bir ortama taşınmasını oldukça kolay hale getirmektedir. Docker teknoloji kullanılan sistemlerde uygulamalar bu avantajlardan faydalanabilmektedir (45).

Geliştirilen Android uygulamalarının daha çok kullanıcıya ulaşabilmesi için Google Play Store hizmetinden faydalanmak, uygulamalar açısından oldukça gereklidir. Google firması bu hizmeti kullanmak isteyen geliştiricilerden, güvenlik, mahremiyet, doğruluk gibi kriterlerin yerine getirilmesini bekleyerek, uygulamaların belirli bir düzeye çıkarılmasını sağlamaktadır. Aynı zamanda uygulama üzerinde bir den fazla cihazda testler gerçekleştirmekte ve önemli sorunlar içeren uygulamaların düzeltilmesini istemektedir. Geliştirilen mobil uygulamamız beklenen tüm kriterleri sağlamış, tüm testleri geçmiş ve Google Play Store'da yerini almıştır (46).

Uygulama içerisinde uygulanan anket kullanıcıların düşüncelerini görebilmeyi sağlamaktadır. Uygulamanın geliştirilmesi gereken yönlerinin açık olarak belirlenebilmesi için kullanıcılara anket içerisinde uygulamada istedikleri özellikleri rahatça yazabilecekleri alan tanımının bulunması ve buna uygun geliştirmelerin gerçekleştirilmesi, uygulamanın başarısını ve talebini artıracağı düşünülmektedir.

ÇBPİ konusunda daha önce yapılmış 'Mpi portable' isimli Mac ve Windows masaüstü ortamları için oluşturulmuş yalnızca bir uygulama bulunmuştur (47). Uygulamanın incelenmesi Windows sistem üzerinde kurularak yapılmıştır.

Uygulamada dil destekleri eklenmiş fakat uygulamada farklı dillerde hatalar oluşmaktadır (Şekil 34). Varsayılan dil olarak Fransızca tanımlı gelmektedir.



Şekil 34: ‘Mpi portable’ uygulamasında faklı dillerde hata

ÇBPİ uygulamamızda, uygulamanın Google Play Store da 54 farklı ülkede yayınlanması sebebiyle fazla sayıda farklı dillerin desteğinin uygulamaya eklenmesinin getireceği zaman maliyeti ve evrensel bir dil olması nedeniyle varsayılan olarak İngilizce tercih edilmiştir. Başka bir dil desteği bulunmamaktadır. Uygulamamıza Türkçe dil desteğinin eklenmesinin, Türkiye'deki kullanıcılar tarafından daha rahat kullanılmasında faydalı olacağı düşünülmektedir.

‘Mpi portable’ uygulaması yalnızca Mac ve Windows bilgisayarlar için kurulabilir durumdadır. Wine gibi çeşitli araçlar ile Linux, Unix gibi sistemlerde de kurulabilir olsa da, uygulama düzgün olarak çalışmamakta ve çok geç yanıt gelmesi veya linklerin çalışmaması gibi hatalar vermektedir. Oysa, Flutter ile geliştirilen uygulamalar neredeyse bütün işletim sistemlerinde çalıştırılabilir olmaktadır. Geliştirilen uygulamamızda dünyada en çok kullanılan işletim sistemi olması sebebiyle Android sistemi üzerinde yoğunlaşılmıştır (48). Diğer sistemler için uygulama çıktısı üretilmesi ve dağıtımının yapılması, zamansal ve ekonomik olarak daha maliyetli olacağından dolayı tercih edilmemiştir.

‘Mpi portable’ uygulamasında girilen verilerin kontrolleri doğru olarak yapılmamaktadır. Örneğin tarih alanına rastgele uzunlukta veri girilebilmektedir. ÇBPİ için geliştirilen uygulamada ise veri girişlerinde kontroller sağlanmış ve uygulama bütünlüğü korunmuştur.



Şekil 35: 'Mpi portable' uygulamasında veri kontrolü

‘Mpi portable’ uygulamasında anlık olarak sonuçlar üretilmekte ve dosya olarak çıktısının alınması sağlamaktadır. Hasta kaydı yapılmamaktadır. Bu nedenle hasta takibi ve daha sonra yapılması istenebilecek gözlemler de gerçekleştirilemeyecektir. Çıktı olarak verilen dosyaların saklanması ve her yerden ulaşılabilir olamaması gibi maliyetleri de bulunmaktadır. Geliştirilen uygulamamızda hasta kaydı yapılabilmektedir. Böylece uygulamamızı kullanan sağlık hizmeti sağlayıcısı, geriye dönük olarak hastalarını görebilecek ve verilerini inceleme imkanına sahip olacak. Veriler güvenli sunucularda saklandığı için veri kaybı gibi sorunlar ve verileri saklama maliyetleri oluşmayacaktır.

‘Mpi portable’ uygulaması bir masaüstü cihaza bağımlı olarak çalışmaktadır. ÇBPİ değerlendirmelerini hasta başında yapmak zor olarak görünmektedir. Her zaman ve her istenen yerden değerlendirme çalışmaları yapılamayacak ya da gözden geçirilemeyecektir. Geliştirilen uygulamamız mobil cihazlar üzerinde de çalışabildiği

için kullanıcılar her zaman her yerde uygulamayı kullanma, değerlendirme yapma imkanlarına sahip olacaktır.

ÇBPİ konusunda geliştirilmiş bir mobil uygulama bulunmaması nedeniyle, ÇBPİ gibi sağlık alanında kullanılan değerlendirme indekslerinden biri olan, FRS (Framingham Risk Skoru) ve bu konuda üretilmiş ve Google Play'e yüklenmiş 'Framingham Score Heart Age' uygulaması incelenmiştir.

FRS kalp hastalığı riskini değerlendirmek için kullanılır. Yaş, cinsiyet, kan basıncı, kolesterol seviyeleri, sigara içme durumu ve diyabet gibi faktörler göz önüne alınarak bireyin kalp hastalığı geliştirme riski hesaplanır (49).

'Framingham Calc' uygulaması basit bir arayüze sahip ve kullanıcılar için pratik olabileceği düşünülmekte fakat uygulama reklam barındırmaktadır. Bu durum kullanıcıların uygulamayı kullanırken zorlanacağını ve uygulamanın kullanılma isteğini olumsuz yönde etkileyeceğini düşündürmektedir.

'Framingham Calc' uygulaması kullanıcı kayıt ve giriş sistemini, hasta kayıt sistemini içermemektedir. Bu nedenle kullanıcı, değerlendirmesini yaptığı bilgileri görmek istediğinde tekrar aynı işlemleri yapmak zorunda kalacaktır. Kullanıcı kayıt ve giriş sistemi sayesinde uygulamalar kişiye özel veriler taşıyabilme özelliğine sahip olmaktadır. Geliştirilen uygulamamız, kullanıcı kayıt ve giriş doğrulama özellikleriyle diğer uygulamaya nazaran daha fazla ayrıntı içermektedir. Aynı zamanda hasta kayıtları da yapılabilmektedir. Hastalarını görebileceği, yapacağı ve yaptığı değerlendirmeleri görüntüleyebileceği ekranlara sahiptir.

Verilerin takibinin, tekrar gözden geçirmelerin kullanıcılara avantaj getireceği uygulamalarda kayıt sistemlerinin bulunması, bu konudaki ihtiyacı gidererek uygulamanın kullanılma sıklığını artıracaktır.

Gelişen teknoloji ile birlikte cihazların donanımsal özellikleri artarak, bağlantı ve hız problemleri daha az yaşanır olacağı düşünüldüğünde, internet bağlantısı gerektiren uygulamaların da yerel uygulamalar gibi sağlıklı çalışacağı düşünülmektedir. Geliştirilen uygulamamız kullanıldığı aygıt üzerinde veri tutmayarak, uzak bilgisayarda veri tutmaktadır. Bu sayede veri kaybı ve güvenlik gibi risklerin daha az olacağı düşünülmektedir.

Her ne kadar günümüzde internetin olmadığı alanlar çok az olsa da uygulamanın internet olmadan da çalıştırılabilir olması uygulamanın kullanım sıklığını artıracaktır. Bunu sağlayabilmek için kullanıcıların cihazlarında da verilerin tutulması ve cihaz internete bağlandığında güncel verilerin aktarımının sağlanması gerekmektedir. Bu sayede kullanıcı uygulamayı internetin olmadığı konumlarda da kullanabilir hale gelecektir (50). Gerçekleştirilmesi daha çok zaman ve çalışma gerektirmektedir.

6. SONUÇ VE ÖNERİLER

Araştırmamızda uygulamamızın verilerinin tutulmasını, barınmasını sağlayacak veri tabanı ilişkisel olarak başarıyla oluşturulmuştur. Sunucu uygulaması planlandığı bütün işlemleri başarıyla ve hızla yerine getirmektedir. Mobil uygulamadan gelen istekleri doğru şekilde işlemekte, kullanıcı ve veri tabanı etkileşiminde gerekli güvenlik kontrollerini sağlamakta, tutarsız ve yetkisiz işlemleri kısıtlamaktadır. Kullanıcı arayüzü, işlemlerin hızlı, akıcı, tutarlı ve başarılı bir şekilde ilerlemesini sağlayacak şekilde kullanıcı dostu olarak tasarlanmıştır. Sunucu sistemi sürecin devamlılığını sağlayacak yapıda, kararlı olarak oluşturulmuştur. Google Play Store gereksinimleri doğru bir şekilde karşılanmış ve uygulama Google Play Store'a yüklenebilmiştir.

Uygulama performans değerlendirmelerinin hem Flutter araçlarıyla hem de Google tarafından gerçekleştirilmiş olan testlerde başarılı sonuçlar elde edilmiştir. Flutter araçlarıyla yapılan testlerde 16 ms'nin altında hızlı işlem zamanları elde edilmiştir. Uygulamanın test edildiği cihazlarda işlemlerde takılma, kilitlenme, kapanma ya da aşırı bellek kullanımı gibi sorunlar yaşanmamıştır. Uygulama boyutu 20,8 mb ile cihaz deposunda az yer kaplamaktadır. Google tarafından yapılan testlerde kararlılık, performans, erişilebilirlik, güvenlik gibi testler uygulanmış ve herhangi bir sorun görülmemiştir.

Uygulama üzerinden yapılan anket sonuçlarında, uygulamanın gereksizliği hakkında verilen yanıtlar %92 oranda olumsuz olarak cevaplanarak yapılan çalışmanın gerekliliğini doğrulamıştır. Uygulamanın başkalarına önerilmesi hakkında verilen cevapların tamamı olumlu olarak ile uygulamanın kullanım sıklığına verilen cevaplar %96 olumlu olarak uygulamanın başarısının ve kullanıcıların memnuniyetinin bulunduğu çıkarımını vermektedir.

Böyle kapsamlı bir mobil uygulama çerçevesinin tamamlanması, çok fazla literatür taraması ve zaman gerektirmektedir. Bu yüzden çok yönlü ve doğru planlama yapılması önerilmektedir. Araştırmada mobil uygulama, sunucu uygulaması ve sunucu sistemi olmak üzere üç farklı geliştirme yapılmıştır.

Sunucu sistem desteğinin bu hizmeti veren Hosting firmaları ile sağlanması çalışmanın aşamalarını ve süresini kısaltacaktır. Aynı zamanda sunucu sistemin bakımı, güvenliği, güncellenmesi gibi konulardan ilgili firma sorumlu olacaktır.

7. KAYNAKLAR

1. Pallavi L, Jagan A, Rao DrBT. Mobile Cloud Computing : Overview & Current Research Challenges. 2017.

2. Dinh HT, Lee C, Niyato D, Wang P. A survey of mobile cloud computing: architecture, applications, and approaches. Wirel Commun Mob Comput. 2013;13(18):1587-611.

3. Fernando N, Loke SW, Rahayu W. Mobile cloud computing: A survey. Future Gener Comput Syst. Ocak 2013;29(1):84-106.

4. Shravanthi C, Guruprasad HS. Mobile Cloud Computing as Future for Mobile Applications. IJRET Int J Res Eng Technol. 2014;3(5):253-6.

5. Suliman M. A Brief Analysis of Cloud Computing Infrastructure as a Service(IaaS). 14 Şubat 2021;6:1409-1412.

6. Yasrab R. Platform-as-a-Service (PaaS): The Next Hype of Cloud Computing. 2018.

7. Al-Madhagy TH, Alanzi A, Mohd Yusof S, Alruwaili M. Software as a Service (SaaS) Cloud Computing: An Empirical Investigation on University Students' Perception. Interdiscip J Inf Knowl Manag. 01 Ocak 2021;16:213-53.

8. Kamla R, Ali Yahiya T, Mustafa N. An Implementation of Software Routing for Building a Private Cloud. Int J Comput Netw Inf Secur. 08 Mart 2018;10.

9. Morabito R, Beijar N. Enabling Data Processing at the Network Edge through Lightweight Virtualization Technologies. 2016.

10. Laal M. Health information technology benefits. AWERProcedia Inf Technol Comput Sci. 01 Kasım 2012;2:224-8.

11. Mamyrbekova S, Nurgaliyeva Z, Saktapov A, Zholdasbekova A, Kudaibergenova A. Medicine of the Future: Digital Technologies in Healthcare. Ziyadin S, Shaikh A, Dinis De Sousa R, Borodin A, Mottaeva A, editörler. E3S Web Conf. 2020;159:04036.

12. Erdal E. The Impact of Technology Trends on Healthcare Systems: A Study on Opportunities and Threats. Int J Trend Sci Res Dev. 31 Ekim 2018;Volume-2:1574-8.

13. Singh S. How is Swift Language Changing iOS Application Development [Internet]. Appinventiv. 2017 [a.yer 08 Ağustos 2023]. Erişim adresi: https://appinventiv.com/blog/swift-shaping-future-ios-app-development/

14. What Is Objective-C: Definition, Usages & App Examples [Internet]. Designveloper. 2022 [a.yer 08 Ağustos 2023]. Erişim adresi: https://www.designveloper.com/blog/what-is-objective-c

15. Kotlin Help [Internet]. [a.yer 09 Ağustos 2023]. Kotlin Docs | Kotlin. Erişim adresi: https://kotlinlang.org/docs/home.html

16. FireMonkey. Içinde: Wikipedia [Internet]. 2023 [a.yer 08 Ağustos 2023]. Erişim adresi: https://en.wikipedia.org/w/index.php?title=FireMonkey&oldid=1167435637

17. React Native. Içinde: Wikipedia [Internet]. 2023 [a.yer 08 Ağustos 2023]. Erişim adresi: https://en.wikipedia.org/w/index.php?title=React_Native&oldid=1165572712

18. profexorgeek. What is Xamarin? - Xamarin [Internet]. 2022 [a.yer 08 Ağustos 2023]. Erişim adresi: https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin

19. Introduction | NativeScript [Internet]. [a.yer 08 Ağustos 2023]. Erişim adresi: https://docs.nativescript.org/#bring-your-own

20. Flutter - Build apps for any screen [Internet]. [a.yer 09 Ağustos 2023]. Erişim adresi: //flutter.dev/

21. Luchaninov Y. MobiDev. [a.yer 09 Ağustos 2023]. 2023 Cross-Platform Mobile App Development Frameworks Compared. Erişim adresi: https://mobidev.biz/blog/cross-platform-mobile-development-frameworks-comparison

22. Özşaker E. Yaşlı Cerrahi Hastaları ve Kırılganlık Sendromu. Samsun Sağlık Bilim Derg. 29 Nisan 2023;8(1):49-62.

23. Bonnekoh SI, Meyer AM, Pickert L, Schulz RJ, Becker I, Polidori MC. The multidimensional prognostic index in hospitalized older adults: practicability with regard to time needs. Aging Clin Exp Res. Mart 2023;35(3):711-6.

24. Pilotto A, Ferrucci L, Franceschi M, D'Ambrosio LP, Scarcelli C, Cascavilla L, ve ark. Development and validation of a multidimensional prognostic index for one-year mortality from comprehensive geriatric assessment in hospitalized older patients. Rejuvenation Res. Şubat 2008;11(1):151-61.

25. Mendez M. General Mental Status Scales, Rating Instruments, and Behavior Inventories. Içinde: The Mental Status Examination Handbook [Internet]. Elsevier; 2022 [a.yer 07 Ağustos 2023]. s. 181-99. Erişim adresi: https://linkinghub.elsevier.com/retrieve/pii/B9780323694896000164

26. Ungar A, Mannarino G, van der Velde N, Baan J, Thibodeau MP, Masson JB, ve ark. Comprehensive geriatric assessment in patients undergoing transcatheter aortic valve implantation - Results from the CGATAVI multicentre registry. BMC Cardiovasc Disord. 2018;18(1).

27. Meyer AM, Siri G, Becker I, Betz T, Bödecker AW, Robertz JW, ve ark. The Multidimensional Prognostic Index in general practice: One-year follow-up study. Int J Clin Pract. 13 Ağustos 2019;e13403.

28. Pilotto A, Rengo F, Marchionni N, Sancarlo D, Fontana A, Panza F, ve ark. Comparing the prognostic accuracy for all-cause mortality of frailty instruments: a multicentre 1-year follow-up in hospitalized older patients. PloS One. 2012;7(1):e29090.

29. Pilotto A, Azzini M, Cella A, Cenderello G, Castagna A, Pilotto A, ve ark. The multidimensional prognostic index (MPI) for the prognostic stratification of older inpatients with COVID-19: A multicenter prospective observational cohort study. Arch Gerontol Geriatr [İnternet]. 01 Temmuz 2021 [a.yer 05 Ağustos 2023];95. Erişim adresi: https://pubmed.ncbi.nlm.nih.gov/33882420/

30. Overbeek FCMS, Goudzwaard JA, van Hemmen J, van Bruchem-Visser RL, Papma JM, Polinder-Bos HA, ve ark. The Multidimensional Prognostic Index Predicts Mortality in Older Outpatients with Cognitive Decline. J Clin Med. 23 Nisan 2022;11(9):2369.

31. Han M, Lee E. Effectiveness of Mobile Health Application Use to Improve Health Behavior Changes: A Systematic Review of Randomized Controlled Trials. Healthc Inform Res. Temmuz 2018;24(3):207-26.

32. Ventola CL. Mobile Devices and Apps for Health Care Professionals: Uses and Benefits. Pharm Ther. Mayıs 2014;39(5):356-64.

33. Lee M, Bin Mahmood ABS, Lee ES, Smith HE, Tudor Car L. Smartphone and Mobile App Use Among Physicians in Clinical Practice: Scoping Review. JMIR MHealth UHealth. 31 Mart 2023;11:e44765.

34. Nass SJ, Levit LA, Gostin LO, Rule I of M (US) C on HR and the P of HITHP. The Value and Importance of Health Information Privacy. Içinde: Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research [Internet]. National Academies Press (US); 2009 [a.yer 09 Ağustos 2023]. Erişim adresi: https://www.ncbi.nlm.nih.gov/books/NBK9579/

35. Türkiye Cumhuriyeti Cumhurbaşkanlığı Dijital Dönüşüm Ofisi - 2019/12 Sayılı Cumhurbaşkanlığı Bilgi ve İletişim Güvenliği Tedbirleri Genelgesi [İnternet]. [a.yer 09 Ağustos 2023]. Erişim adresi: https://cbddo.gov.tr/bgrehber/2019-12-sayili-bilgi-guvenligi-tedbirleri-cumhurbaskanligi-genelgesi/

36. resmigazete.gov.tr/eskiler/2019/06/20190621-3.htm [İnternet]. [a.yer 09 Ağustos 2023]. Erişim adresi: https://www.resmigazete.gov.tr/eskiler/2019/06/20190621-3.htm

37. DevTools [Internet]. [a.yer 23 Ağustos 2023]. Erişim adresi: https://docs.flutter.dev/tools/devtools/overview

38. Joshi A, Kale S, Chandel S, Pal D. Likert Scale: Explored and Explained. Br J Appl Sci Technol. 10 Ocak 2015;7:396-403.

39. Play Console [İnternet]. [a.yer 16 Ağustos 2023]. Erişim adresi: https://play.google.com/console/developers/?pli=1

40. Android Asset Studio [İnternet]. [a.yer 16 Ağustos 2023]. Erişim adresi: https://romannurik.github.io/AndroidAssetStudio/index.html

41. Dart packages [Internet]. [a.yer 16 Ağustos 2023]. flutter_launcher_icons | Dart Package. Erişim adresi: https://pub.dev/packages/flutter_launcher_icons

42. Build and release an Android app [Internet]. [a.yer 16 Ağustos 2023]. Erişim adresi: https://docs.flutter.dev/deployment/android

43. Duchessi P, Chengalur-Smith I. Client/server benefits, problems, best practices. Commun ACM. 01 Mayıs 1998;41:87-94.

44. Amanda. What is Docker and what does it do? Advantages and benefits [Internet]. Chakray. 2023 [a.yer 29 Ağustos 2023]. Erişim adresi: https://www.chakray.com/what-is-docker-and-what-does-it-do-exploring-the-advantages-and-benefits-of-this-powerful-platform/

45. Google Play'de Android Uygulamaları [İnternet]. [a.yer 29 Ağustos 2023]. Erişim adresi: https://play.google.com/store/games?hl=tr&gl=TR

46. Intro MPI [Internet]. [a.yer 06 Ağustos 2023]. Erişim adresi: http://multiplat-age.it/index.php/en/tools

47. StatCounter Global Stats [Internet]. [a.yer 24 Ağustos 2023]. Operating System Market Share Worldwide. Erişim adresi: https://gs.statcounter.com/os-market-share

48. Jahangiry L, Farhangi MA, Rezaei F. Framingham risk score for estimation of 10-years of cardiovascular diseases risk in patients with metabolic syndrome. J Health Popul Nutr. 13 Kasım 2017;36:36.

49. Gazzawe F. Comparison of Websites and Mobile Applications for E-Learning. N Y. 2017;

8. EKLER