

DOKUZ EYLÜL ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ



STAJ RAPORU

ÖĞRENCİNİN

BÖLÜMÜ : **BİLGİSAYAR MÜHENDİSLİĞİ**

ADI SOYADI : **EMRE ÖZKAYA**

NUMARASI : **2020510137**

İZMİR

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü		
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024	Bitiş Tarihi : 09/02/2024	Sayfa : 2

İÇİNDEKİLER

Sayfa

İçindekiler	ii
Şekil Listesi	v

Bölüm 1

Staj Kurumu

1.1 Hakkında Genel Bilgi	6
1.2 Misyon ve Vizyon	6
1.3 Yapısal Organizasyon.....	7
1.4 Bilgisayarla İlgili Birimlerin Rolü.....	7
1.5 İletişim	8

Bölüm 2

Staj Projesi

2.1 Proje Tanımı	9
2.2 Proje ve Kullanıcı Özellikleri	10
2.2.1 Kullanıcı Profili	10
2.2.2 Proje Özellikleri.....	10

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 3

2.3 Kullanılan Teknolojiler..... 10

Bölüm 3 Staj Haftalık Özeti

3.1 Birinci Hafta	12
3.2 İkinci Hafta	13
3.3 Üçüncü Hafta	13
3.4 Dördüncü Hafta	13

Bölüm 4 Staj Haftalık Detay

4.1 Birinci Hafta Detay	14
4.1.1 Jira Nedir ?.....	14
4.1.2 Excalidraw Nedir ?	15
4.1.3 Kod Dökümantasyonu Nedir ?	16
4.1.4 UML Diyagram Nedir ?	17
4.1.5 Aktivite (Acitivity) Diyagramı Nedir ?	18
4.1.6 Kullanım Şekli (Use Case) Diyagramı Nedir ?	19
4.1. Sequence Diyagram Nedir ?	20
4.2 İkinci Hafta Detay	21
4.2.1 React.js ve React-Native Nedir?.....	21
4.2.2 Cross-Platform Nedir ?	22
4.2.3 Expo Nedir ?	22
4.2.4 Expo-SQLite Nedir ?	23
4.2.5 CSS Nedir ?	24
4.2.6 Text Recognition (OCR) Nedir ?	25
4.2.7 Google Cloud Vision OCR Nedir?	26
4.3 Üçüncü Hafta Detay	26

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 4

4.3.1 UI/UX Nedir ?	26
4.4 Dördüncü Hafta Detay	32
4.4.1 OCR Ekranı	32
4.4.2 Geçiş Ekranı	34
4.4.3 Çözüm Yönergeleri Ekranı	35

Bölüm 5

Kaynakça

Kaynakça	37
----------------	----

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 5

Şekil Listesi

Şekil 2.1 Hedef Ürün	9
Şekil 3.1 Haftalık Toplantı Planı	12
Şekil 4.1 Jira Sprint Planlama Ekrani	14
Şekil 4.2 Excalidraw Proje Ekrani	16
Şekil 4.3 Kod Dökümantasyonu	17
Şekil 4.4 Aktivite Diyagramı	19
Şekil 4.5 Use Case Diyagramı	20
Şekil 4.6 Ürün Ekleme Akış Diyagramı	21
Şekil 4.7 Expo-Sqlite Kullanımı	24
Şekil 4.8 Wireframe UI Tasarımı	28
Şekil 4.10 Proje Mimarisi	30
Şekil 4.11 Giriş Ekranı Tasarımı	31
Şekil 4.12 Giriş Ekranı Kamera Açılmaya Kodu	32
Şekil 4.13 Ocr Ekranı Kodu	33
Şekil 4.14 Ocr Ekranı Tasarımı	33
Şekil 4.15 Geçiş Ekranı Tasarımı	34
Şekil 4.16 Geçiş Ekranı Kodu	35
Şekil 4.17 Çözüm Yönergeleri Ekranı	36
Şekil 4.18 Veritabanı Sorgusu Ve Yazı Filtreleme Kodu	36

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü		
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024	Bitiş Tarihi : 09/02/2024	Sayfa : 6

BÖLÜM 1

Staj Kurumu

1.1 Hakkında Genel Bilgi

Dokuz Eylül Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, bilgisayar bilimleri ve mühendisliği alanındaki akademik çalışmaları ve araştırmaları ile tanınan bir eğitim kurumudur. Bu bölüm, öğrencilere bilgisayar mühendisliği alanında teorik bilgilerin yanı sıra pratik uygulamalar ve projeler üzerinde çalışma fırsatı sunar.

Dokuz Eylül Üniversitesi, 1982 yılında İzmir'de kurulmuş olup, Türkiye'nin önde gelen üniversiteleri arasında yer alır. Bilgisayar Mühendisliği Bölümü, öğrencilere yazılım geliştirme, sistem yönetimi, veritabanı yönetimi ve siber güvenlik gibi çeşitli alanlarda derinlemesine bilgi ve beceri kazandırmayı hedefler.

Bu bölüm, özellikle yazılım mühendisliği, yapay zeka, bilgisayar ağları ve güvenliği gibi alanlarda yenilikçi eğitim ve araştırma faaliyetleri yürüterek öğrencilerin sektörde rekabetçi olmalarını sağlar. Geniş bir araştırma yelpazesi sunarak, öğrencilere akademik ve profesyonel gelişim için zengin fırsatlar sunar.

Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü, akademik mükemmeliyeti ve endüstri ile işbirliğini ön planda tutarak bilgisayar mühendisliği eğitiminde liderlik yapmayı amaçlar.

1.2 Misyon ve Vizyon

Misyonu: Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü, öğrencilerine yüksek kalitede eğitim ve araştırma imkanları sunarak, bilgisayar mühendisliği alanında bilgi ve becerilerini geliştirmelerini sağlamak ve onları sektörün ihtiyaç duyduğu nitelikli profesyoneller olarak yetiştirmeyi amaçlar.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü		
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024	Bitiş Tarihi : 09/02/2024	Sayfa : 7

Vizyonu: Bölüm, ulusal ve uluslararası düzeyde tanınan, yenilikçi araştırmaları ve teknolojik gelişmeleri takip eden, öğrenci ve öğretim üyelerinin sürekli gelişimine önem veren bir eğitim kurumu olmayı hedefler. Ayrıca, endüstri ile işbirliğini güçlendirerek öğrencilerine pratik deneyimler sunar ve mezunlarının global çapta rekabet edebilir olmalarını sağlar.

Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü, öğrencilerine akademik ve profesyonel kariyerlerinde başarılı olmaları için gereken desteği ve kaynakları sunarak, bilgi toplumunun ihtiyaç duyduğu nitelikli bireylerin yetiştirilmesine katkıda bulunmayı amaçlar.

1.3 Yapısal Organizasyon

Bölümün organizasyon yapısı, akademik ve araştırma birimleri olarak düzenlenmiştir. Özellikle bilgisayar mühendisliği ile ilgili birimler, bölümün akademik ve araştırma faaliyetlerine odaklanır.

Örnek birimler:

- Yazılım Mühendisliği: Yazılım geliştirme metodolojileri ve projeler üzerinde çalışır.
- Bilgisayar Ağları ve Güvenliği: Ağ teknolojileri ve siber güvenlik konularında eğitim ve araştırma yapar.
- Yapay Zeka ve Veri Bilimi: Yapay zeka ve makine öğrenimi tekniklerini uygular ve araştırır.
- Sistem Analizi ve Tasarımı: Bilgisayar sistemlerinin tasarımları ve analizi üzerine odaklanır.

1.4 Bilgisayarla İlgili Birimlerin Rolü

Bilgisayarla ilgili birimler, Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü'nün akademik ve araştırma hedeflerine ulaşmasında kritik öneme sahiptir. Bu birimler, öğrencilere güncel bilgisayar mühendisliği uygulamaları üzerine derinlemesine bilgi sağlar ve sektördeki teknolojik gelişmeleri takip etmelerine olanak tanır.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü		
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024	Bitiş Tarihi : 09/02/2024	Sayfa : 8

İşlevleri şunlardır:

- Yazılım Mühendisliği: Güncel yazılım geliştirme araçları ve teknikleri üzerinde eğitim sağlar.
- Bilgisayar Ağları ve Güvenliği: Ağ teknolojileri ve siber güvenlik önlemleri konusunda bilgi ve beceri kazandırır.
- Yapay Zeka ve Veri Bilimi: Yapay zeka ve veri bilimi projeleri üzerinde çalışma fırsatı sunar.
- Sistem Analizi ve Tasarımı: Sistemlerin etkin tasarımı ve analizi için gerekli yöntemleri öğretir.

1.5 İletişim

Bölüm Adı: Dokuz Eylül Üniversitesi Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

E-Posta : web@cs.deu.edu.tr

Adres : Tınaztepe Kampüsü, Buca 35390, Izmir, Turkiye

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 9

BÖLÜM 2

Staj Projesi

2.1 Proje Tanımı

Teknik Servis Destek (mobil) Uygulaması, buzdolabı teknik servis elemanlarının cihazların ekrana bastığı hata kodlarını hızlı ve doğru bir şekilde tanımlayıp ilgili sorunu çözümlemelerine yardımcı olacak bir mobil uygulama üzerine kuruludur. Uygulama, teknik servis elemanlarının telefonlarının kamerası aracılığıyla hata kodlarını okutmasını, OCR (Optik Karakter Tanıma) tekniği ile bu kodları ayırtmasını ve ilgili hata kodunun çözümüne dair yönlendirmeler sunmasını sağlar. Bu sayede, servis elemanları arızanın onarımı için gerekli adımları öğrenebilir ve sorunu etkin bir şekilde çözebilirler.



Şekil 2.1 Hedef Ürün

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü		
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024	Bitiş Tarihi : 09/02/2024	Sayfa : 10

2.2 Proje ve Kullanıcı Özellikleri

2.2.1 Kullanıcı Profili

- Yeni başlayan teknik servis elemanları:** Uygulama, buzdolabı teknik servisine yeni başlayan elemanların hızlı bir şekilde eğitim almasını ve pratik yapmasını sağlar.
- Deneyimli teknik servis elemanları:** Karmaşık hata kodlarıyla karşılaşıklarında hızlı çözüm üretmek için bu uygulamadan yararlanabilirler.
- Normal kullanıcılar:** Uygulama, teknik servis çağrımadan önce veya küçük sorunları kendileri çözmek isteyen buzdolabı sahiplerine de açılabilir. Bu, hem kullanıcıların vakit tasarrufu sağlamasına hem de teknik servis elemanlarının iş yükünü hafifletmeye yardımcı olur.

2.2.2 Proje Özellikleri

- OCR Tekniği:** Uygulama, Google tarafından sağlanan OCR tekniği ile hata kodlarını görüntü üzerinden algılayıp ayırtır. Bu, uygulamanın hatayı hızlı ve doğru bir şekilde tanımlamasını sağlar.
- Hata Kodlarına Dair Yönlendirmeler:** Uygulama, her hata kodu için detaylı çözüm adımlarını sunar, böylece teknik servis elemanı veya kullanıcı sorunu etkin bir şekilde çözebilir.
- Veri Güvenliği ve Gizlilik:** Uygulama, KVKK gibi veri koruma yasalarına uyum sağlamak için kullanıcının cihazında herhangi bir kişisel veri depolamaz. Tüm veriler bellek tabanlı olup, sadece kullanıcının cihazında saklanır. OCR aşamasında da herhangi bir fotoğraf depolaması yapılmaz, bu da veri güvenliğini artırır.
- Uygulama Boyutu:** Uygulamanın boyutu düşük tutularak, cihazların depolama alanından tasarruf edilmesi ve daha hızlı çalışması hedeflenir.

2.3 Kullanılan Teknolojiler

- Proje, cross-platform olarak kodlanmıştır. Bu sayede uygulama hem Android hem de IOS cihazlarda kullanılabilmektedir.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 11

- Kullanılan Teknolojiler
 - Css
 - React
 - Expo
 - Javascript
 - Expo-SQLite
 - React-Native
 - Google Cloud Vision

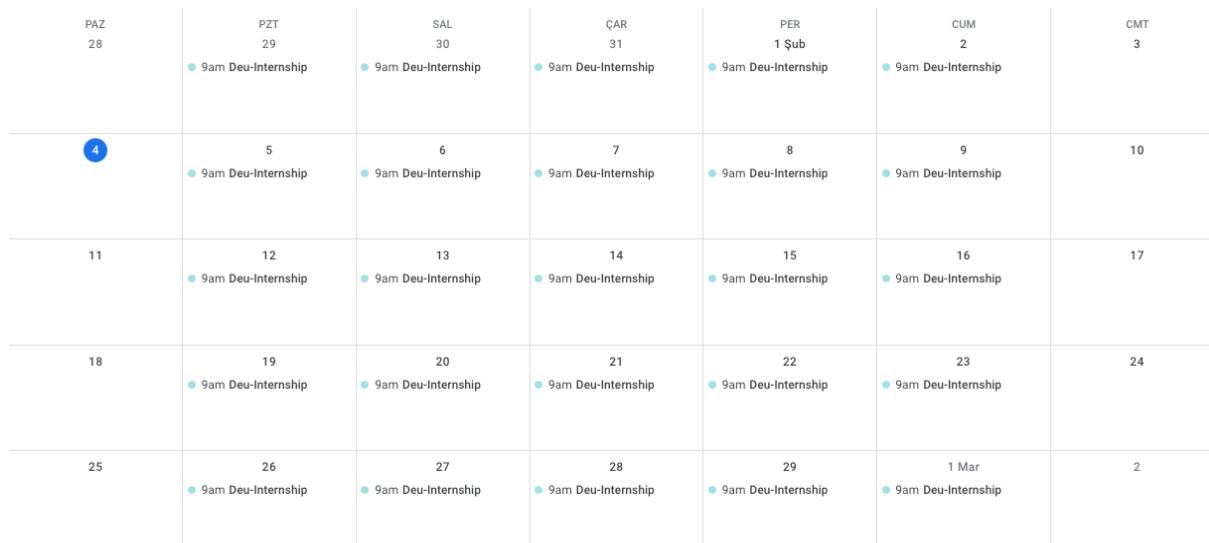
Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 12

BÖLÜM 3

Staj Haftalık Özeti

3.1 Birinci Hafta

Stajın ilk gününde, projede bulunan çalışanlarla tanışıldı, Yapılacak projenin sektördeki yeri hakkında bilgi edinildi ve üyelikler oluşturularak hesaplar aktif hale getirildi. İkinci gün, proje başlangıcına odaklandı; mentor ile toplantı yapılarak proje detayları ve yol haritası belirlendi, dökümantasyon gereksinimleri ve kullanılacak diyagram çeşitleri üzerine çalışıldı. Üçüncü günden itibaren, proje kavramsal dizayn çalışmalarına başlandı; activity diagramları, sequence diagramları oluşturuldu ve kavramsal dizayn devam ederek use case diagrama geçildi. Günlük toplantılar ve haftalık raporlama toplantıları ile stajın başlangıcı organize bir şekilde ilerledi.



Şekil 3.1 Haftalık Toplantı Planı

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü		
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024	Bitiş Tarihi : 09/02/2024	Sayfa : 13

3.2 İkinci Hafta

Stajın dördüncü gününde, kavramsal dizayn çalışmaları hız kesmeden devam etti ve use case diagram tamamlandı, ayrıca haftalık planlamayı içeren bir toplantı gerçekleştirildi. Beşinci ve altıncı günler, Söz konusu işlerin yanı sıra projede kullanılacak teknoloji yiğini belirlendi. Bu süre zarfında günlük toplantılar yapıldı. Daha sonraki üç gün, projede kullanılacak teknolojiler için eğitimlere başlandı, bu eğitimler esnasında demo projeler oluşturularak teknoloji için pratikler kazanıldı. OCR teknikleri araştırılmaya başlandı ve daha önce benzer işlerin yapıldığı projeler incelendi ve son olarak mentor ile birlikte haftalık raporlama toplantısı gerçekleştirildi.

3.3 Üçüncü Hafta

Stajın onuncu gününde, projenin UI/UX tasarımları yapıldı. On birinci günde, daha önce “wireframe” olarak tasarlanan UI/UX’ler ihtiyaca göre artık canlı bir tasarıma çevirildi. On ikinci ve on üçüncü günde, projeye ait genel gereksinimler belirlendiği ve eğitimler de alındığı için proje kodlanması başladı. On dördüncü gün, projenin detaylı dokümantasyonu hazırlandı; bu doküman, kullanılacak mimariyi, yazılım dillerini ve araçlarını içeriyordu ve kodlamaya başlamak için gerekli zemini oluşturdu.

3.4 Dördüncü Hafta

Stajın on beşinci gününde, geçen hafta oluşturulan altyapı üzerinden projenin önemli bir kısmı olan OCR teknikleri uygulanmaya başladı. Burada çıkan sürüm uyumsuzlukları ile süreç beklenenden uzun sürdü ancak başarıyla tamamlandı. On yedinci gün, on sekizinci, on dokuzuncu günler günlük toplantı yapıldı ve proje uçtan uca test edildi ve bazı iyileştirmeler yapıldı. Yirminci gün, proje tamamıyla bitmiş hale geldi ve mentore gösterilmek üzere demo hazırlıkları yapıldı.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 14

BÖLÜM 4

Staj Haftalık Detay

4.1 Birinci Hafta Detay

4.1.1 Jira Nedir ?

Jira, Atlassian tarafından geliştirilen özel bir sorun takip ürünüdür ve aynı zamanda çevik proje yönetimi için kullanılır. Temel olarak, Jira, yazılım geliştirme süreçlerinde görevlerin takip edilmesi, hataların yönetilmesi, özelliklerin planlanması ve projelerin izlenmesi için kullanılır.

- Hata takibi: Yazılım projelerindeki hataların kaydedilmesi, izlenmesi ve çözülmesi için kullanılır.
- Görev yönetimi: Görevlerin atanması, takibi ve süreçlerin optimize edilmesi için kullanışlıdır.
- Çevik proje yönetimi: Scrum ve Kanban gibi çevik yöntemlerle çalışmayı destekler.
- Özelleştirilebilirlik: Farklı projeler ve ekipler için özelleştirilebilir ve genişletilebilir.

Jira, mühendisler, yazılım geliştiricileri, proje yöneticileri ve birçok sektörde çalışan ekipler tarafından kullanılan bir araçtır. Ayrıca, çeşitli eklentiler ve entegrasyonlarla genişletilebilir, böylece farklı ihtiyaçlara uyum sağlayabilir.

The screenshot shows a Jira board with several columns and rows of issues. The columns include 'Infrastructure', 'Order', 'Mobile', 'Dashboard', and 'Product'. The rows represent individual issues with titles like 'TOKSTORE-1465', 'TOKSTORE-1428', 'TOKSTORE-1490', etc., each with its own description and status indicators.

Issue	Description	Status
TOKSTORE-1465	DB analysis	Open
TOKSTORE-1428	Order memory usage analysis	In Progress
TOKSTORE-1490	Db Performans Araştırması İçin Servislere Datadog Modülü Ekleme	Open
TOKSTORE-1512	WebPOS uygulamasında gramaj desteği bulunmuyor	Open
TOKSTORE-1132	[CrossPlatform] Axios interceptor, get access token with refresh token	Open
TOKSTORE-1433	Acr'a düşen TransactionTooLarge hatası	Open
TOKSTORE-1498	POS Authentication	Open
TOKSTORE-1158	CrossPlatformdan silinen çek'in 400TR cihazında Liste düzenden kaldığı görülmüştür.	Open
TOKSTORE-1160	[CrossPlatform] Masasız çekin silinemediği görülmüştür.	Open
TOKSTORE-1154	[CrossPlatform] Ürünne not eklenmek istendiğinde kaytt edilemediği görülmüştür	Open
TOKSTORE-1139	[CrossPlatform] Çek içinde değişiklik yapıp güncelleme yapılmadan geri gelmemek istendiğinde os ve android	Open
TOKSTORE-1124	Şube bazlı ürün isimleri gelmediği görülmüştür.	Open
TOKSTORE-1502	KNV ürününü ekledikten sonra başka bir ürün eklemek istendiğinde istisna kodu seçimi yapılmamasına rağmen istisna	Open
TOKSTORE-1501	Ürün eklemeye tıklanıldıkten sonra ürün eklemeden ürün ekleme sayfasından çıkış yapılmamaktadır	Open
TOKSTORE-1491	Divan Stadda karşılaşılan Acr'a da yer almayan TokenStore durdu hatası	Open

Şekil 4.1 Jira Sprint Planlama Ekranı

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 15

4.1.2 Excalidraw Nedir ?

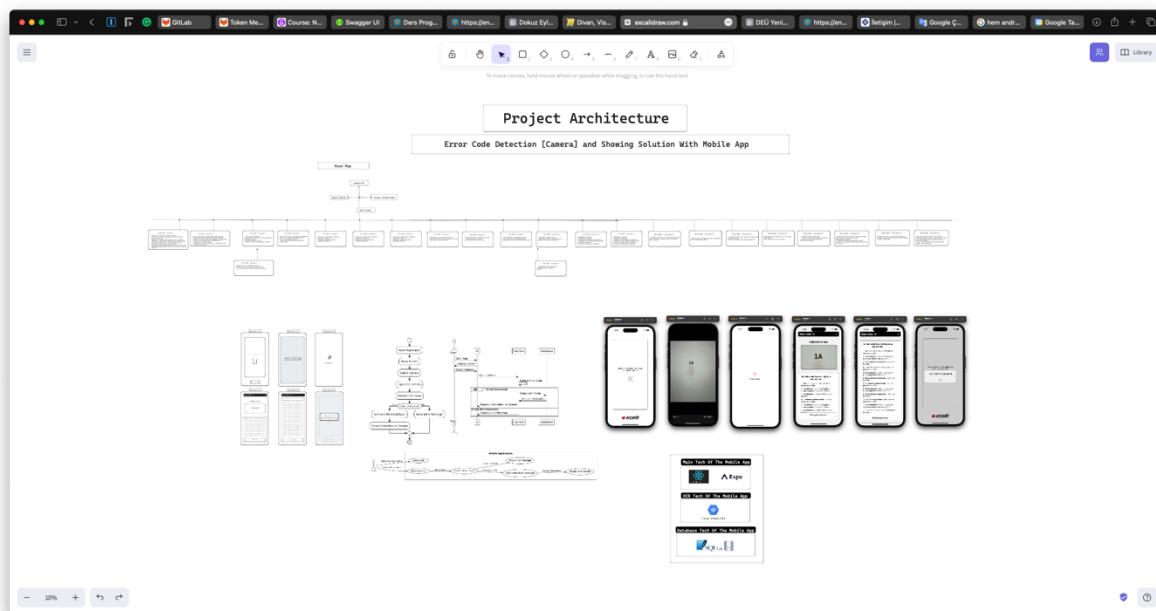
Excalidraw, web tabanlı bir çizim ve diyagram oluşturma uygulamasıdır ve kullanıcıların fikirlerini, süreçleri ve sistemleri görsel bir biçimde ifade etmelerine olanak tanır. Bu platform, basit arayüzü ve elle çizilmiş gibi görünen çizim stiliyle öne çıkar. Yazılım ve sistem tasarıımı, eğitim materyalleri oluşturma, beyin fırtınası yapma ve daha fazlası için kullanılabilir.

- Çizim Araçları:** Kullanıcılar, şekiller, oklar, metin kutuları ve özgür çizim araçları kullanarak fikirlerini çizebilirler. Elle çizilmiş görünüm, diyagramların daha insancıl ve erişilebilir görünmesini sağlar.
- Şablonlar ve Semboller:** Sık kullanılan şekiller ve diyagram elementleri için şablonlar ve semboller sunar, böylece kullanıcılar sık sık tekrar eden öğeleri hızla çizebilirler.
- İşbirliği:** Excalidraw, eşzamanlı işbirliğini destekler, bu da birden fazla kullanıcının aynı çizim üzerinde gerçek zamanlı olarak çalışmasına olanak tanır. Bu özellik, takımların uzaktan çalışırken bile fikirleri paylaşmaları ve üzerinde işbirliği yapmaları için mükemmelidir.
- Dışa ve İçe Aktarma:** Kullanıcılar, çizimlerini PNG veya SVG gibi popüler formatlarda dışa aktarabilir veya daha sonra düzenlemek üzere Excalidraw formatında kaydedebilirler.
- Gizlilik ve Güvenlik:** Çizimler, kullanıcının cihazında işlenir ve saklanır, bu da kullanıcı verilerinin gizliliğini artırır. İşbirliği sırasında bile, veriler doğrudan kullanıcılar arasında aktarılır ve bir sunucuda saklanmaz.

Excalidraw, kullanım kolaylığı, esnekliği ve işbirliğine olanak tanıyan özellikleri ile yazılım geliştirme ekipleri arasında popüler bir araçtır. Bu özellikler, projelerin daha verimli ve etkili bir şekilde yönetilmesine katkıda bulunur.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 16



Şekil 4.2 Excalidraw Proje Ekranı

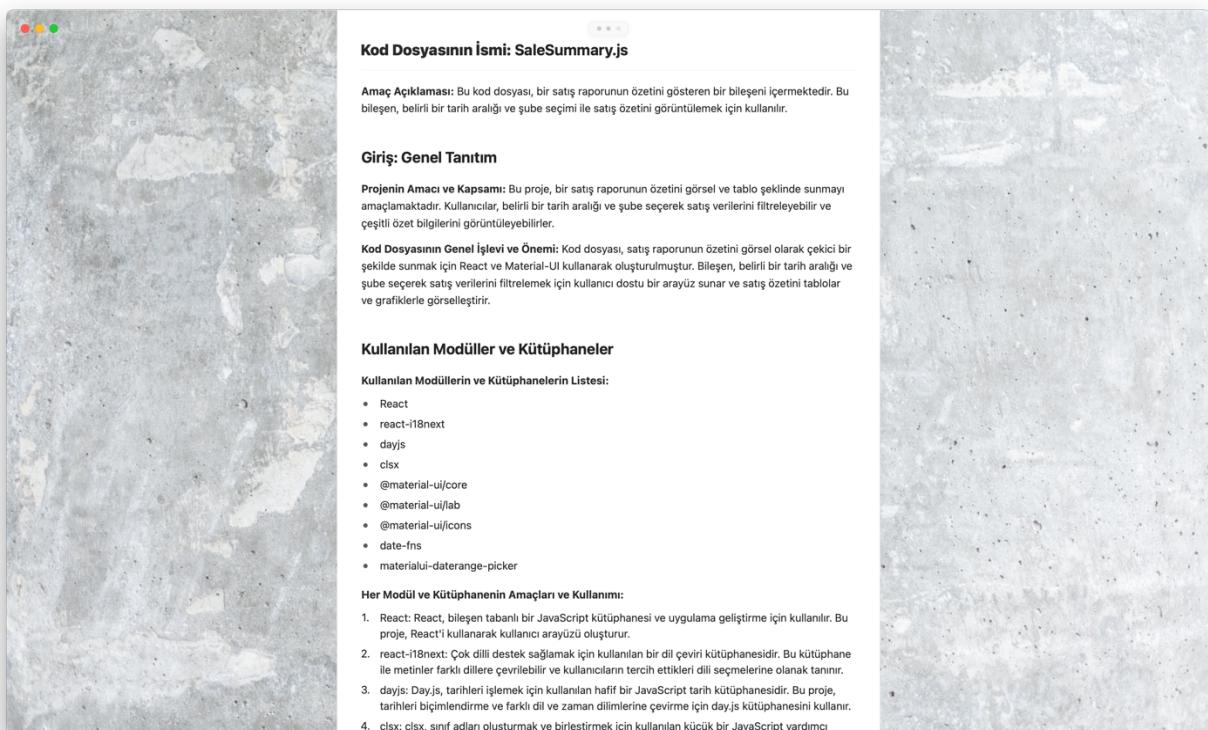
4.1.3 Kod Dökümantasyonu Nedir ?

Kod dökümantasyonu, yazılım geliştirme sürecinde yazılan kodun anlaşılır ve belgelendirilmiş hali olarak tanımlanır. Bu belgeler, yazılımın nasıl çalıştığını, kullanılan fonksiyonların ve değişkenlerin ne işe yaradığını anlatan açıklamalar içerir. Kod dökümantasyonu, yazılım projelerinin yönetimi ve bakımı sırasında büyük bir öneme sahiptir.

- API Dokümantasyonu:** API (Application Programming Interface) kodun dökümantasyonunun bir parçasıdır ve diğer yazılım bileşenlerinin bu kodla nasıl etkileşimde bulunacaklarını açıklar.
- Belgeleme Aracı Kullanımı:** Çeşitli belgeleme araçları, yazılım geliştiricilerine kodlarını daha kolay ve etkili bir şekilde belgeleme olanağı sunar.
- Proje Yönetimi:** Kod dökümantasyonu, yazılım projelerinin takip edilmesi, bakımı ve geliştirilmesi sırasında ekip üyelerine yol gösterir.

Kod dökümantasyonu, yazılım projelerinin daha sürdürülebilir ve işbirliği odaklı olmasına yardımcı olur. Aynı zamanda yeni geliştiricilerin projeye katılmasını kolaylaştırır ve kodun daha iyi anlaşılmasını sağlar.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 17



Şekil 4.3 Kod Dökümantasyonu

4.1.4 UML Diyagram Nedir ?

UML (Unified Modeling Language), iş sistemlerini modellemek ve açıklamak için kullanılan bir dil ve bir standarttır. Genellikle yazılım geliştirme alanında kullanılan bu dil, karmaşık sistemlerin tasarımını, geliştirilmesini ve belgelenmesini kolaylaştırmak amacıyla kullanılır.

UML, görsel bir dildir ve karmaşık sistemleri anlamak ve tasarlamak için grafik notasyonlar içerir. UML diyagramları, sistemlerin farklı yönlerini ve bileşenlerini temsil eden çeşitli tiplerde gelir.

Örneğin:

- Sınıf Diyagramları
- Durum Diyagramları
- Sıralama Diyagramları
- Kullanım Durumu Diyagramları

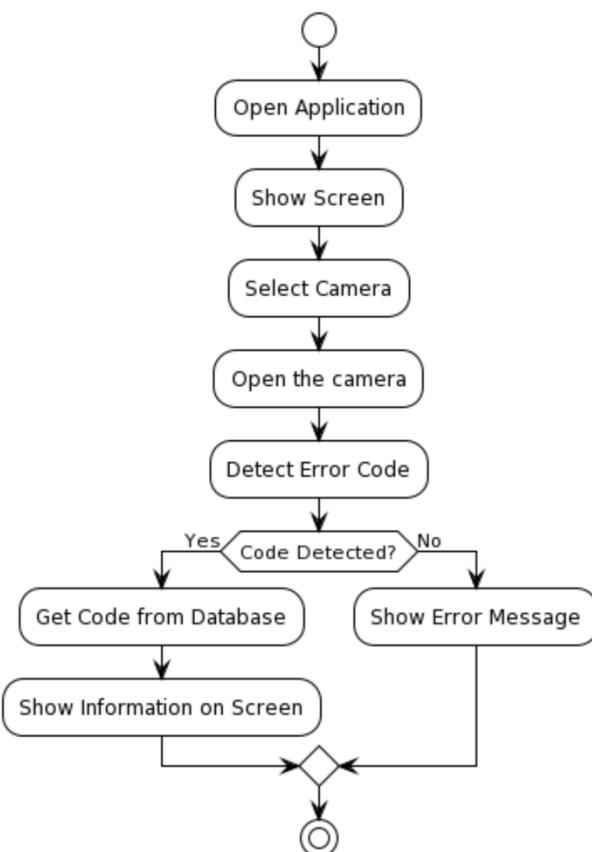
Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 18

UML, yazılım geliştirme sürecini daha anlaşılır ve yönetilebilir hale getirir. Geliştiriciler, sistemlerini daha iyi planlayabilir, tasarlayabilir ve belgeleyebilirler. Ayrıca, UML diyagramları yazılım projelerinde iletişimini artırır ve hataları önlemeye yardımcı olur. Bu nedenle, yazılım mühendisliği alanında önemli bir araçtır.

4.1.5 Aktivite (Activity) Diyagramı Nedir ?

Aktivite Diyagramı, yazılım geliştirme sürecinde iş akışlarını ve faaliyetleri görsel olarak temsil eden bir Birleşik Modelleme Dili (UML) diyagram türüdür ve bir sistem içindeki veya bir işlem sırasındaki akışı, fonksiyonları ve karar noktalarını adım adım göstermek amacıyla kullanılır. Yazılım geliştiricilere ve proje yöneticilerine, iş akışlarının ve süreçlerin nasıl işlediğini derinlemesine anlamalarını sağlayarak, daha etkin planlama ve iş birliği olanakları sunar. Karmaşık iş akışlarını ve operasyonları basitleştirecek, proje ekip üyelerinin projelerin işleyiş mekanizması hakkında açık ve net bir perspektife sahip olmalarını sağlayan bu diyagramlar, başlangıç ve bitiş noktaları, eylem düğümleri, karar düğümleri (koşullu dallanmalar), eş zamanlılık (paralel işlemler) ve senkronizasyon çubukları gibi temel bileşenleri içerir. Aktivite diyagramları, yazılım geliştirme sürecindeki temel faaliyetlerin ve süreçlerin modellenmesi amacıyla kullanılmakta olup, ekiplerin iş akışlarını optimize etmesine, potansiyel sorunları önceden tespit etmesine ve daha verimli bir yazılım geliştirme süreci tasarlamasına imkan tanırken, sistem analizi ve tasarım aşamasında kullanıcı hikayelerinin ve gereksinimlerin nasıl gerçekleştirileceğinin anlaşılmasına katkı sağlar.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 19



Şekil 4.4 Aktivite Diyagramı

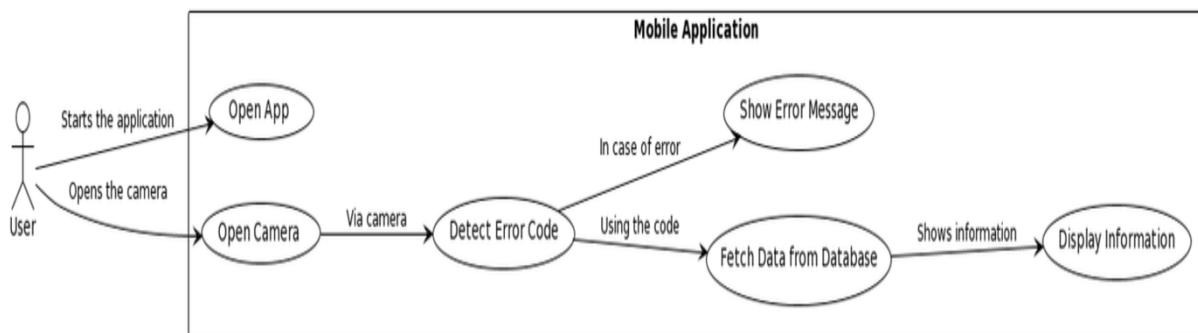
4.1.6 Kullanım Şekli (Use Case) Diyagramı Nedir ?

Kullanım Şekli (Use Case) Diyagramı, bir yazılım veya sistemin işlevsellliğini ve kullanım senaryolarını gösteren bir UML (Unified Modeling Language) diyagram türüdür. Bu diyagram, bir sistemin nasıl çalıştığını, hangi kullanıcıların sistemi nasıl kullanabileceğini ve bu kullanım senaryolarının nasıl etkileşime girdiğini görselleştirir.

Use Case Diyagramları, genellikle bir sistem veya yazılımın işlevsellliğini anlamak, gereksinimleri belirlemek ve tasarım sürecini başlatmak için kullanılır. Bu diyagramlar, ihtiyaç duyulan tüm işlevleri (use case'leri) ve bu işlevleri gerçekleştiren aktörleri (kullanıcılar veya sistemler) tanımlar. Ayrıca, her bir kullanım durumu (use case) ile aktörler arasındaki ilişkiyi gösterir.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 20

Use Case Diyagramları, yazılım geliştirme sürecinin erken aşamalarında kullanılarak gereksinimlerin netleştirilmesine yardımcı olur. Bu sayede, projenin kapsamı ve kullanıcı ihtiyaçları daha iyi anlaşılır ve daha sonra detaylı tasarım ve geliştirme süreçlerine geçiş sağlanır.



Şekil 4.5 Use Case Diyagramı

4.1.7 Sequence Diyagram Nedir ?

Bir "sequence" diyagramı veya sıra diyagramı, yazılım mühendisliği ve sistem tasarımları alanlarında kullanılan bir UML (Unified Modeling Language) diyagram türüdür. Bu tür bir diyagram, işlem etkileşimlerini ve süreçlerin zaman içindeki sıralamasını göstermek için kullanılır. Özellikle yazılım sistemlerinin tasarımları ve analizi sırasında kullanılır.

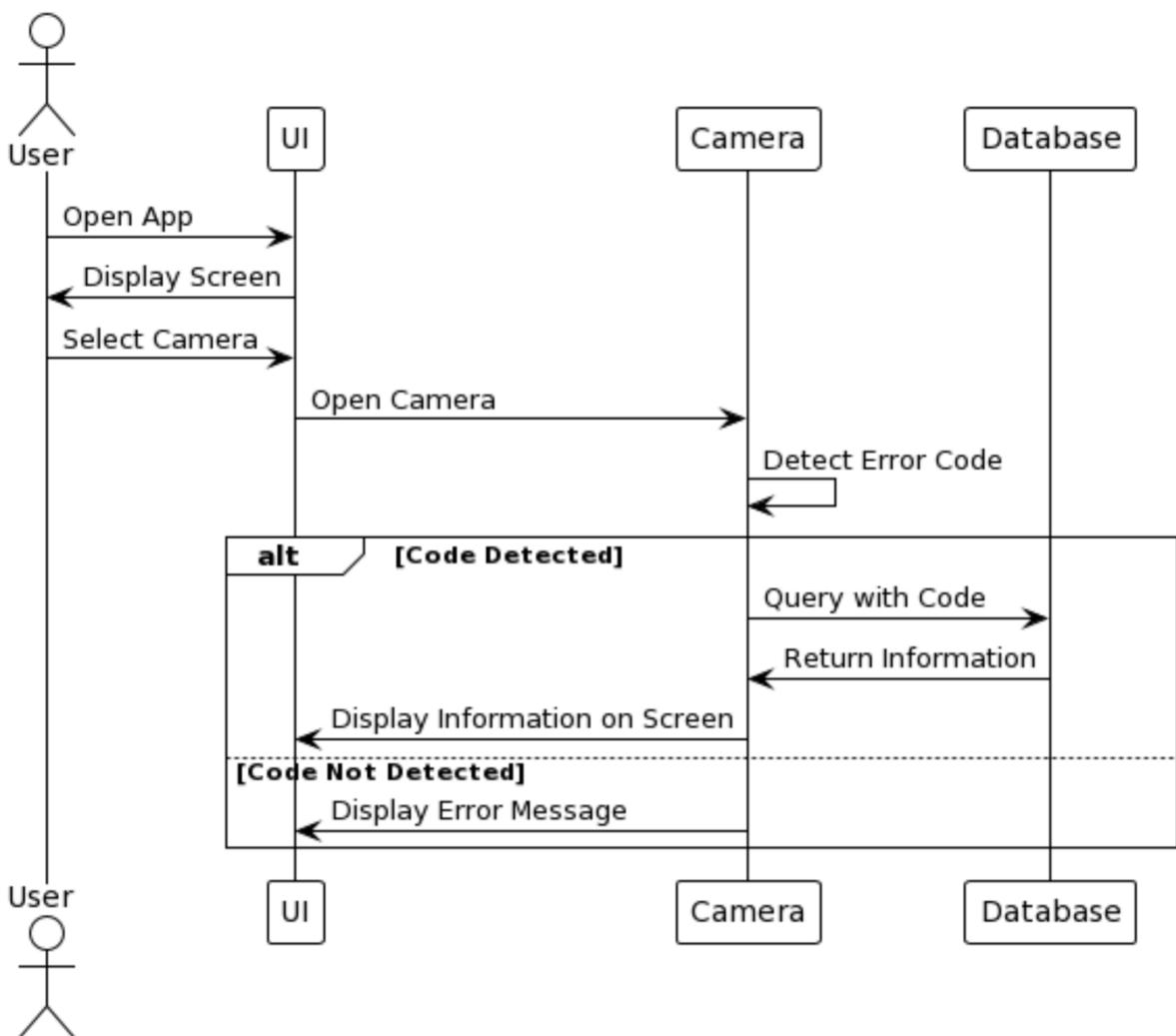
Bir sequence diyagramı aşağıdaki temel öğeleri içerir:

- Nesneler:** Sistemdeki veya yazılımin parçalarını temsil eden nesneler.
- Mesajlar:** Nesneler arasında iletilen mesajlar veya işlem çağrıları.
- Zaman Eksenisi:** Zamanın ilerlemesini temsil eden bir eksende nesnelerin etkileşimleri sıralanır.

Bu diyagramlar, bir sürecin veya işlemin nasıl gerçekleştiğini, nesnelerin nasıl etkileşimde bulunduğu ve bir işlemin nasıl ilerlediğini görsel olarak açıklar. Yazılım geliştirme sürecinde, sistem tasarnımını belgelemek, analiz etmek ve iletişim sağlamak için önemli bir araçtır.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 21



Şekil 4.6 Ürün Ekleme Akış Diyagramı

4.2 İkinci Hafta Detay

4.2.1 React.js ve React-Native Nedir?

React.js ve React Native, modern web ve mobil uygulama geliştirme için kullanılan iki popüler JavaScript kütüphanesi ve çerçevesidir. Her ikisi de Facebook tarafından geliştirilmiş olup, geliştiricilere kullanıcı arayüzleri oluşturmada büyük kolaylıklar sağlar. Ancak, her birinin kullanım alanları, avantajları ve temel özellikleri farklılık gösterir.

React.js (genellikle sadece "React" olarak adlandırılır), kullanıcı arayüzleri tasarlamak için kullanılan bir JavaScript kütüphanesidir. Özellikle tek sayfa uygulamaları (SPA - Single Page Applications) oluşturmak için tasarlanmıştır. React, veri değişiklerinin olduğu durumlarda

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 22

kullanıcı arayüzüünü etkili bir şekilde güncellemek için sanal DOM kullanır. Bu, uygulamanın performansını artırır ve geliştiricilerin daha hızlı ve verimli bir şekilde çalışmalarını sağlar.

React Native, React'ın temelleri üzerine kurulmuş bir mobil uygulama çerçevesidir. Android ve iOS platformları için gerçek, yerel mobil uygulamalar geliştirmek için kullanılır. React Native, geliştiricilere tek bir JavaScript kod tabanı üzerinden iki platformda hitap edebilme imkanı sunar. Bu, geliştirme sürecini hızlandırır ve kod tekrarını azaltır.

4.2.2 Cross-Platform Nedir ?

Cross-platform uygulama geliştirme, geliştiricilerin tek bir kod tabanı kullanarak birden fazla işletim sistemi için uygulamalar oluşturmasını sağlayan bir yaklaşımdır. Bu yaklaşım, özellikle Android ve iOS gibi farklı mobil platformlarda çalışacak uygulamalar geliştirmek isteyenler için büyük avantajlar sunar. Cross-platform uygulama geliştirmenin temel amacı, geliştirme sürecini hızlandırmak, maliyetleri azaltmak ve kod tekrarını minimize etmektir.

- Tek Kod Tabanı:** Geliştiriciler, farklı platformlarda çalışacak uygulamalar için aynı kodu kullanır. Bu, uygulama geliştirme sürecini hızlandırır ve kod yönetimini kolaylaştırır.
- Maliyet Etkin:** Tek bir uygulama geliştirmek, birden fazla platform için ayrı ayrı uygulamalar geliştirmekten genellikle daha az maliyetlidir.
- Daha Hızlı Piyasaya Sürme:** Tek bir geliştirme çalışmasıyla birden fazla platforma uygulama sunulabilir, bu da uygulamanın daha hızlı piyasaya sürülmесini sağlar.
- Bakım ve Güncellemeler:** Uygulamanın bakımı ve güncellemeleri daha kolaydır, çünkü değişiklikler tek bir yerden yönetilir ve tüm platformlara dağıtilır.

4.2.3 Expo Nedir ?

Expo, React Native ile mobil uygulamalar geliştirmeyi kolaylaştıran açık kaynaklı bir platformdur. Geliştiricilere, yerel kod yazmadan iOS ve Android için uygulamalar oluşturma imkanı sunar. Expo, uygulama geliştirme sürecini hızlandıran bir dizi araç ve hizmet içerir. Bunlar arasında uygulama yayılama, hızlı yineleme ve yerel API'lere kolay erişim gibi

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 23

özellikler bulunur. Expo, özellikle React Native ile yeni başlayanlar için mükemmel bir başlangıç noktasıdır, çünkü karmaşık yerel geliştirme ortamlarının kurulumunu gerektirmez.

- **Hızlı Prototipleme:** Expo, hızlı prototipleme ve uygulama geliştirmeyi mümkün kılar. Kod değişiklikleri arasında önizlenebilir.
- **Yerel API'lere Kolay Erişim:** Kamera, konum, bildirimler gibi yerel cihaz özelliklerini ve API'lere kolay erişim sağlar.
- **Hassas Geliştirme Araçları:** Expo, hata ayıklama, loglama ve uygulama performansını izleme gibi geliştirme süreçlerini destekleyen araçlar sunar.
- **Expo Go Uygulaması:** Geliştiriciler, Expo Go uygulamasını kullanarak, kodlarını gerçek cihazlarda hemen test edebilirler. Bu, bir uygulamanın nasıl görüneceği ve çalışacağı hakkında hızlı geri bildirim almayı sağlar.
- **Otomatik Güncellemeler:** Expo, uygulamaları otomatik olarak güncelleyebilme özelliğine sahiptir, bu sayede kullanıcılar en son sürümeye hızla erişebilir.

4.2.4 Expo-SQLite Nedir ?

Expo-SQLite, Expo platformu üzerinde çalışan uygulamaların SQLite veritabanlarını yönetmelerine olanak tanıyan bir modüldür. SQLite, yerleşik bir SQL veritabanı motorudur ve diskte depolanan hafif, dosya tabanlı bir veritabanı sistemidir. Expo-SQLite modülü, geliştiricilerin React Native uygulamalarında veritabanı işlemleri gerçekleştirmesine, verileri saklamasına, sorgulamasına ve yönetmesine imkan tanır. Bu, hem iOS hem de Android platformlarında veri depolama ve yönetimi için etkili bir çözüm sunar.

- **Hafif Veritabanı Yönetimi:** SQLite, hafif bir veritabanı yönetim sistemidir, bu da mobil cihazlarda verimli çalışmasını sağlar.
- **Yerel Veri Depolama:** Uygulama verilerini kullanıcının cihazında yerel olarak depolamak için kullanılır. Bu, çevrimdışı veri erişimi ve kalıcı veri depolama ihtiyaçları için idealdir.
- **Çapraz Platform Desteği:** Expo-SQLite, hem iOS hem de Android platformlarında çalışır, bu da geliştiricilere tek bir kod tabanı ile iki platformda da veritabanı işlemleri yapma imkanı sunar.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 24

- **SQL Sorguları:** Geliştiriciler, veritabanı işlemlerini gerçekleştirmek için SQL sorguları kullanabilirler. Bu, veri sorgulama, güncelleme, silme ve daha fazlası için geniş bir esneklik sağlar.

Expo-SQLite'ı kullanmak için, öncelikle Expo uygulamanıza **expo-sqlite** modülünü eklemeniz gerekmektedir. Modül, Expo'un yönettiği proje yapılandırmasının bir parçasıdır, bu yüzden yerel yapılandırmalar veya bağlantılar gerektirmez. Kurulumdan sonra, SQLite veritabanları oluşturabilir, veritabanına bağlanabilir ve SQL sorguları yoluyla veritabanı üzerinde işlemler gerçekleştirebilirsiniz.

```
import * as SQLite from 'expo-sqlite';

// Veritabanı dosyasını açmak veya oluşturmak
const db = SQLite.openDatabase("veritabani_adi.db");

// Bir tablo oluşturmak ve veri eklemek
db.transaction(tx => {
  tx.executeSql(
    "CREATE TABLE IF NOT EXISTS kullanıcılar (id INTEGER PRIMARY KEY AUTOINCREMENT, ad TEXT, yas INTEGER);",
    [],
    () => console.log("Tablo başarıyla oluşturuldu"),
    (_, hata) => console.log("Tablo oluşturulurken hata oluştu", hata)
  );

  tx.executeSql(
    "INSERT INTO kullanıcılar (ad, yas) VALUES (?, ?);",
    ["Ahmet", 25],
    () => console.log("Veri başarıyla eklendi"),
    (_, hata) => console.log("Veri eklenirken hata oluştu", hata)
  );
});
```

Şekil 4.7 Expo-SQLite Kullanımı

4.2.5 CSS Nedir ?

CSS (Cascading Style Sheets), web sayfalarının görünüm ve biçimini tanımlamak için kullanılan bir stil sayfası diliidir. HTML ile oluşturulan web sayfası yapılarının (yazı tipi boyutları, renkler, aralıklar, sütunların düzeni, ve diğer görsel efektler gibi) nasıl görüneceğini belirlemek için CSS kullanılır. CSS, web geliştirmede önemli bir teknolojidir ve modern web tasarımının temel yapı taşlarından biridir.

- **Ayrıştırma:** CSS, içerik (HTML) ile tasarım ve biçimlendirmeyi (CSS) birbirinden ayırrır. Bu, içeriğin nasıl görüneceğini değiştirmek istediğinizde, sadece CSS dosyasını

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 25

güncellemeniz gerektiği anlamına gelir. Bu yaklaşım, web sayfasının bakımını ve yönetimini kolaylaştırır.

- **Cascading (Kademelendirme):** CSS, birden fazla stil kaynağının bir arada kullanılmasına izin verir ve stil kurallarının hangi sırayla uygulanacağını belirler. Bu, global stil sayfaları, sayfa içi stiller ve tarayıcı varsayılanlarını birleştirmeyi mümkün kılar.
- **Tasarım Kontrolü:** CSS, web sayfalarının genel görünümü üzerinde detaylı kontrol sağlar. Metinlerin, arka planların, kutuların ve diğer elementlerin nasıl görüneceğini detaylı bir şekilde ayarlayabilirsiniz.
- **Yanıt Veren Tasarım:** CSS, medya sorguları aracılığıyla farklı ekran boyutları ve çözünürlükleri için farklı stil kuralları uygulayarak yanıt veren tasarımların oluşturulmasını sağlar. Bu, bir web sayfasının mobil cihazlar, tabletler ve masaüstü bilgisayarlarda uygun şekilde görüntülenmesini sağlar.

4.2.6 Text Recognition (OCR) Nedir ?

Text recognition, yani metin tanıma, görüntüler üzerindeki yazılı metni algılayıp dijital metne dönüştüren bir teknolojidir. Bu süreç, genellikle Optik Karakter Tanıma (OCR) olarak bilinir ve fotoğraflar, taramış belgeler, veya video kareleri gibi çeşitli kaynaklardan metin çıkarmak için kullanılır. Metin tanıma teknolojisi, yapay zeka ve makine öğrenimi algoritmalarından yararlanarak, görüntülerdeki karakterleri, kelimeleri ve cümle yapılarını tanır ve anlamlandırır.

- **Çeşitli Formatlardan Metin Çıkarma:** Yazılı metni, el yazısı, basılı dokümanlar, işaretler, ve hatta web sayfaları gibi çeşitli kaynaklardan tanıyıp dijitalleştirebilir.
- **Dil ve Karakter Destekleri:** Çok sayıda dilde ve yazı tipinde metni tanıabilme yeteneği. Modern metin tanıma sistemleri, Latin alfabetesinden Çince karakterlere kadar geniş bir yelpazede destek sunar.
- **Yüksek Doğruluk Oranları:** Gelişmiş algoritmalar sayesinde, metin tanıma teknolojisi yüksek doğruluk oranlarına ulaşabilir, hatta bazı durumlarda el yazısını bile doğru bir şekilde tanıyalabilir.
- **Hızlı İşleme:** Büyük miktarlardaki veriyi kısa sürede işleyebilme yeteneği, metin tanıma teknolojisini veri girişi ve doküman yönetimi için ideal kılar.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 26

4.2.7 Google Cloud Vision OCR Nedir?

Google Cloud Vision OCR, Google Cloud Vision API'nin bir parçası olarak sunulan, görüntülerdeki metni tanıma ve dijital metne dönüştürme yeteneğine sahip bir özelliktir. OCR (Optik Karakter Tanıma) teknolojisi kullanarak, kullanıcıların fotoğraflar, PDF'ler ve diğer görüntü formatlarındaki yazılı metni algılamalarını ve bu metni düzenlenebilir ve aranabilir bir formata çevirmelerini sağlar. Bu hizmet, Google'in gelişmiş makine öğrenimi algoritmalarından faydalananarak geniş bir dizi yazı tipi ve dili destekler, bu sayede yüksek doğruluk oranları ile metin tanıma işlemleri gerçekleştirebilir.

- Yüksek Doğruluk:** Google'in gelişmiş makine öğrenimi modelleri, farklı yazı tipleri, boyutları ve dillerdeki metni yüksek doğrulukla tanır.
- Çoklu Dil Desteği:** Google Cloud Vision OCR, çok sayıda dilde metin tanıma yeteneğine sahiptir, bu da onu küresel uygulamalar için uygun kılar.
- Esnek Kullanım:** Geliştiriciler, REST API aracılığıyla Google Cloud Vision OCR özelliklerine erişebilir ve uygulamalarına entegre edebilirler.
- Güçlü Entegrasyon:** Google Cloud Platform'un diğer hizmetleriyle (örneğin, Google Cloud Storage) sorunsuz entegrasyon sağlar, bu sayede geliştiriciler görüntüleri kolaylıkla saklayabilir ve işleyebilir.
- Geniş Uygulama Alanı:** Faturalar, makbuzlar, sokak işaretleri, kimlik belgeleri ve hatta el yazısı metinler gibi çeşitli kaynaklardan metin çıkarma işlemleri için kullanılabilir.

4.3 Üçüncü Hafta Detay

4.3.1 UI/UX Nedir ?

UI (User Interface) ve UX (User Experience), kullanıcıların bir ürün, sistem veya hizmetle etkileşimlerinin tasarımlıyla ilgili iki temel kavramdır. Bu terimler genellikle birlikte anılır, ancak farklı yönleri temsil ederler. UI, kullanıcı arayüzünün görsel, işitsel ve dokunsal özelliklerini kapsarken; UX, bir kullanıcının bir ürün veya hizmetle etkileşim sırasında yaşadığı deneyimin bütününe odaklanır.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 27

User Interface (Kullanıcı Arayüzü), kullanıcıların bir ürün, uygulama veya web sitesiyle etkileşime girebildiği noktaların tümünü ifade eder. UI, düğmeler, ikonlar, renkler, yazı tipleri, düzenler ve navigasyon gibi görsel öğeleri içerir. Bir UI tasarımcısının görevi, bu öğeleri estetik açıdan hoş ve kullanım kolay bir şekilde düzenlemektir.

- Görsel Tasarım:** Renkler, yazı tipleri ve görüntüler gibi görsel unsurlar.
- Etkileşim Tasarımı:** Kullanıcıların nasıl etkileşime gireceği (örneğin, düğmeler, menüler).
- Kullanılabilirlik:** Arayüzün kullanım kolaylığı.

User Experience (Kullanıcı Deneyimi), kullanıcıların bir ürünü veya hizmeti kullanırken yaşadıkları deneyimin bütünüdür. UX tasarımı, bir ürünün kullanımı, etkili, kolay erişilebilir ve kullanıcı dostu olmasını sağlamayı hedefler. UX, kullanıcının ürünle veya hizmetle etkileşiminin her yönünü, ilk karşılaşmadan hedeflerine ulaşana kadar geçen süreci kapsar.

- Kullanıcı Araştırması:** Kullanıcı ihtiyaçlarını ve davranışlarını anlama.
- Senaryo ve Akış Şemaları:** Kullanıcıların ürünle nasıl etkileşime gireceklerini tasarlama.
- Prototipleme ve Test Etme:** Tasarımları test ederek kullanılabilirliği değerlendirme.

UI ve UX birbirinden ayrı olsa da, birlikte çalışarak kullanıcıların bir ürün veya hizmetle etkileşimini optimize ederler. Etkili bir UI, kullanıcıların gözüne hoş görünür ve onlara kolay bir navigasyon sağlarken; iyi bir UX, bu etkileşimin sorunsuz, verimli ve tatmin edici olmasını sağlar. Bir ürünün başarısı, genellikle bu iki alanın ne kadar iyi entegre edildiğine ve birlikte çalıştığına bağlıdır.

4.3.2 Proje Tasarımları

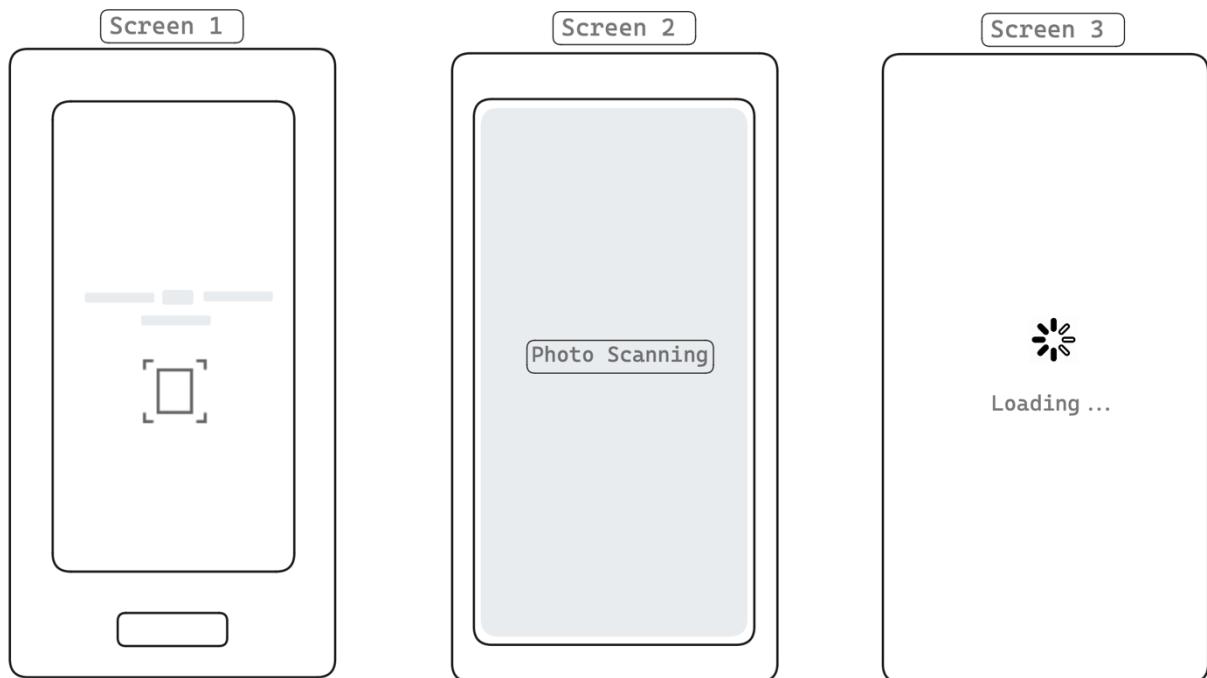
Proje tasarımları iki fazda ele alınmıştır ilk faz wireframe ikinci faz ise marka giydirilmiş, kullanıcının gördüğü tasarımlardır. Marka giydirilmiş tasarımlar ilgili başlıklar altında kodları ile birlikte verilecektir.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 28

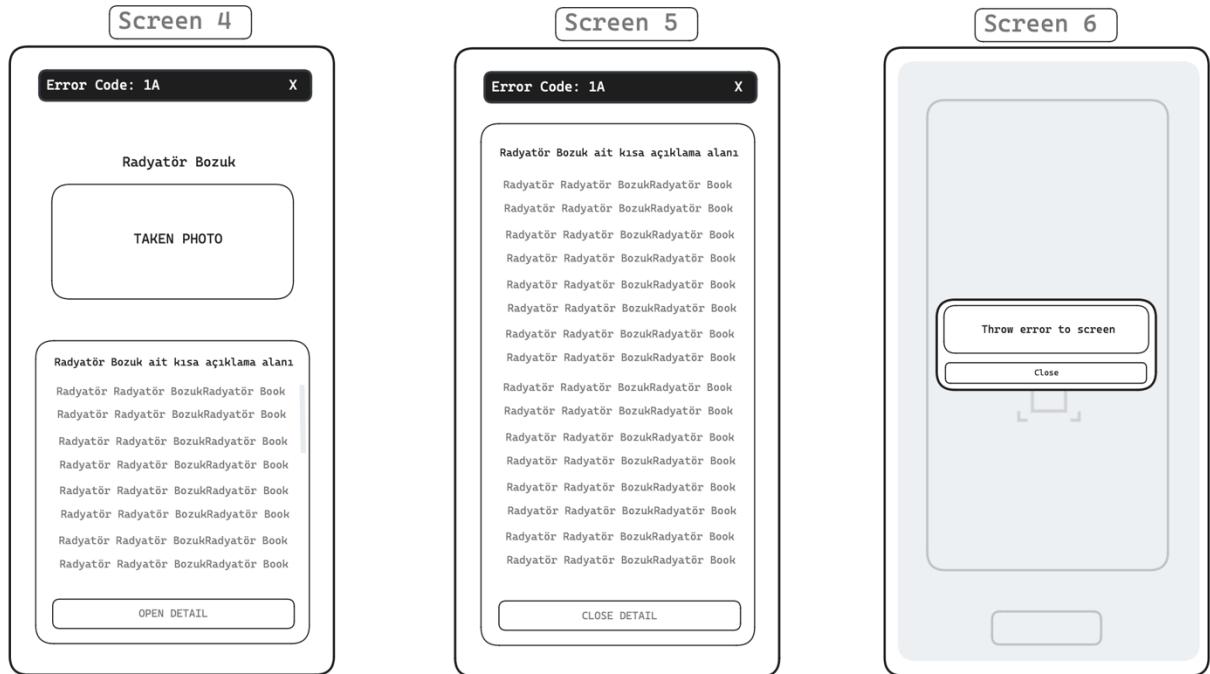
Wireframe, bir web sitesi, uygulama veya diğer dijital ürünlerin tasarım sürecinde kullanılan, ürünün yapısal bir çerçevesini veya iskeletini temsil eden düşük detaylı bir çizimdir. Genellikle siyah beyaz renklerde hazırlanan wireframe'ler, sayfa düzeni, içerik yerleşimi, kullanıcı arayüzü elemanları (menüler, butonlar, resimler) ve navigasyon akışı gibi temel öğeleri gösterir. Wireframe'lerin ana amacı, bir ürünün fonksiyonel yönlerine odaklanarak tasarım ve geliştirme ekipleri arasında iletişimini kolaylaştırmak ve projenin temel yapısını belirlemektir.

- Düşük Detaylı:** Renk, tipografi ve görsel tasarım öğeleri gibi detaylardan ziyade, sayfa düzeni ve işlevsellik üzerine odaklanır.
- Hızlı ve Ekonomik:** Wireframe'ler, tasarım sürecinin erken aşamalarında hızlıca oluşturulabilir ve değişiklikler için esneklik sağlar.
- İletişimi Kolaylaştırır:** Proje paydaşları ve geliştirme ekipleri arasında fikir birliği oluşturmak ve geri bildirim almak için kullanılır.
- Kullanıcı Deneyimine Odaklanır:** Kullanıcıların nasıl etkileşime gireceği, navigasyon ve bilgi mimarisi gibi unsurları vurgular.



Şekil 4.8 Wireframe UI Tasarımı

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 29



Şekil 4.9 Wireframe UI Tasarımı

Marka giydirmesi yapılmış UI (User Interface), bir şirketin veya markanın görsel kimliğini ve stil rehberini, kullanıcı arayüzü tasarımına entegre ettiği bir yaklaşımdır. Bu, markanın renkleri, logoları, yazı tipleri, ikonlar ve diğer görsel öğeleri kullanılarak yapılan bir uygulama, web sitesi veya dijital ürünün tasarımını içerir. Marka giydirmesi yapılmış bir UI, kullanıcılara sunulan dijital arayüzler aracılığıyla markanın kimliğini güçlendirir ve marka tutarlığını sağlar.

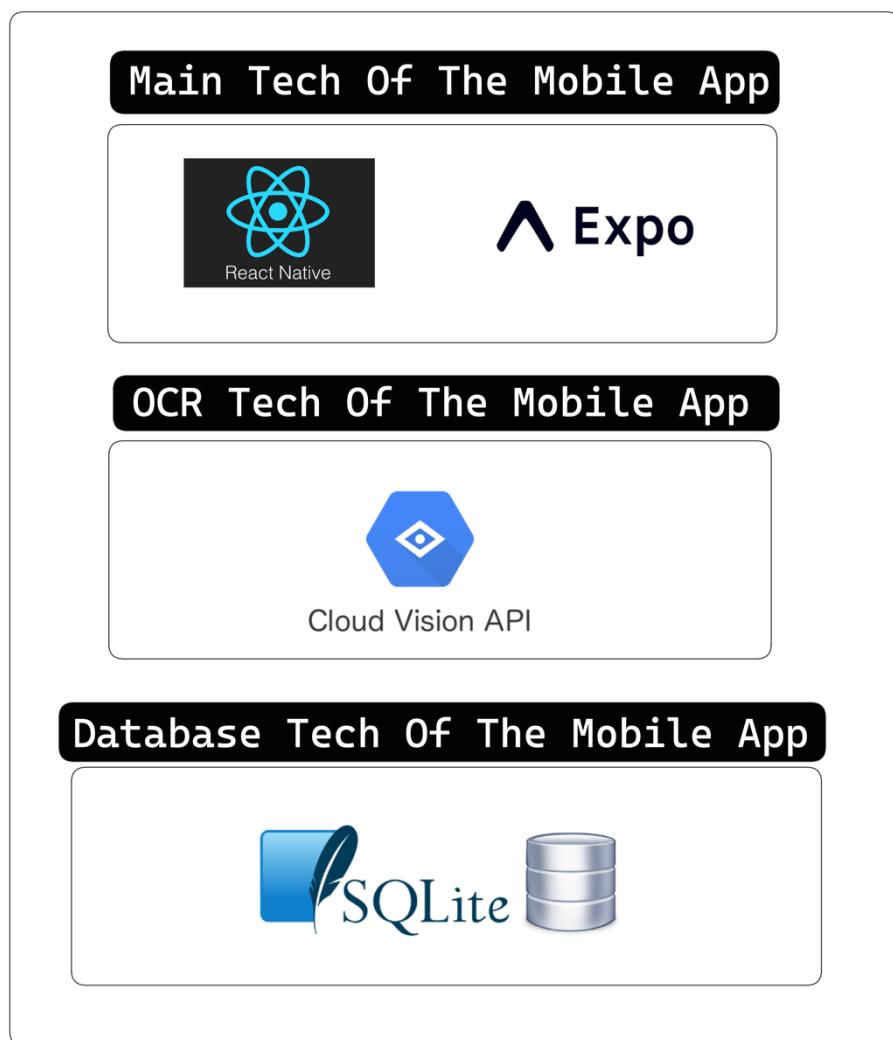
- **Marka Kimliğinin Entegrasyonu:** Markanın görsel kimliği, kullanıcı arayüzüne derinlemesine entegre edilir. Bu, renkler, yazı tipleri, logo kullanımı ve diğer görsel öğelerle markanın tanınabilirliğini artırır.
- **Tutarlılık:** Kullanıcı arayüzü boyunca görsel ve işitsel öğelerde tutarlılık sağlanır. Bu, kullanıcıların markayı tanımasına ve onunla bir bağ kurmasına yardımcı olur.
- **Kullanıcı Deneyimi ile Uyum:** Marka giydirmesi, kullanıcı deneyimini olumsuz etkilemeyecek şekilde yapılır. Kullanılabilirlik ve estetik, marka kimliği ile uyumlu bir şekilde dengelenir.
- **Marka Değerlerinin Yansıtılması:** UI üzerinden markanın değerleri ve kişiliği kullanıcılara aktarılır. Bu, duygusal bir bağ oluşturarak marka sadakatini artırabilir.

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 30

4.3.3 Proje Mimarisi

Proje, React Native ve Expo çerçevesi kullanılarak geliştirilmiş bir mobil uygulamadır. Bu uygulama, görüntülerdeki metni tanıma ve dijital metne dönüştürme yeteneği kazandırmak için Google Cloud Vision OCR tekniğinden yararlanmaktadır. Bu sayede kullanıcılar, fotoğraflar üzerindeki yazılı metni kolayca tanıyalabilir ve işleyebilirler. Uygulama içerisinde veri depolama ihtiyaçları için Expo-SQLite kullanılmıştır; bu, hem iOS hem de Android platformlarında verimli bir şekilde yerel veri depolama imkanı sunar. Bu mimari, gelişmiş metin tanıma özelliklerini mobil cihazlarda sorunsuz bir şekilde kullanabilme ve elde edilen verileri yerel olarak saklayıp yönetebilme kabiliyeti ile kullanıcı dostu bir deneyim sağlamaktadır.



Şekil 4.10 Proje Mimarisi

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 31

4.3.4 Giriş Ekranı

Giriş ekranı, uygulama ilk kez yüklendiğinde kullanıcıların karşılaşacağı ve doğrudan kameralı aktive edebilecekleri arayüzdür. Bu ekranın temel amacı, kullanıcıların ihtiyaçlarını sade ve etkin bir biçimde karşılamaktır. Tarama alanına dokunulduğu anda, uygulama kullanıcının kamerasını erişimi için izin ister. Onay alındıktan sonra, kameralı uygulama içinde kullanıma sunulur.



Şekil 4.11 Giriş Ekranı Tasarımı

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 32

```

const openCamera = async () => {
    const granted = await ImagePicker.requestCameraPermissionsAsync();
    if (granted) {
        setIsLoading(true)
        let data = await ImagePicker.launchCameraAsync({
            mediaTypes: ImagePicker.MediaTypeOptions.All,
            base64: true,
        });
        if (data.canceled) {
            console.log("cancelled")
            setIsLoading(false)
        }
        if (!data.canceled) {
            const responseData = await callGoogleVisionAsync(data.assets[0].base64);
            const text = responseData.text;
            const image = data.assets[0].uri;
            console.log("text", text)
            const response = checkErrorCodeInText(text);
            if (response) {
                setText(response);
                setImage(image);
            }else{
                setIsLoading(false)
                Alert.alert(`Hata Kodu ${text} Bulunamadı`)
            }
        }
    } else {
        Alert.alert('Permissions required to access camera roll.')
    }
};

```

Şekil 4.12 Giriş Ekranı Kamera Açılmama Kodu

4.4 Dördüncü Hafta Detay

4.4.1 OCR Ekranı

Kullanıcı, bu erişim ile hata kodunun yer aldığı bölgenin fotoğrafını çekebilir; çekilen fotoğraf, OCR (Optik Karakter Tanıma) teknigi kullanılarak analiz edilmek üzere Google Cloud Vision'a gönderilir. Fotoğraf üzerindeki metinler ayırtıldıktan sonra, elde edilen veriler, veritabanında kayıtlı olan hata kodları ile karşılaştırılır. Eğer bir eşleşme tespit edilirse, veritabanından ilgili bilgiler çekilir ve kullanıcının sorunu çözmeye yardımcı olacak yönergeler içeren bir ekrana yönlendirilir. Bu süreç, kullanıcılarla hızlı ve doğrudan bir çözüm sunarak etkili bir kullanıcı deneyimi sağlar.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 33

```
//file name: OCR.js

const API_KEY = 'key'; //put your key here.
//this endpoint will tell Google to use the Vision API. Also, we are passing in our key.
const API_URL = `https://vision.googleapis.com/v1/images:annotate?key=${API_KEY}`;

function generateBody(image) {
  const body = {
    requests: [
      {
        image: {
          content: image,
        },
        features: [
          {
            type: 'TEXT_DETECTION', //we will use this API for text detection purposes.
            maxResults: 1,
          },
        ],
      },
    ],
  };
  return body;
}

async function callGoogleVisionAsync(image) {
  const body = generateBody(image); //pass in our image for the payload
  const response = await fetch(API_URL, {
    method: 'POST',
    headers: {
      Accept: 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify(body),
  });
  const result = await response.json();
  const detectedText = result.responses[0].fullTextAnnotation;
  return detectedText
    ? detectedText
    : {text: "This image doesn't contain any text!"};
}

export default callGoogleVisionAsync;
```

Şekil 4.13 OCR Ekranı Kodu



Şekil 4.14 OCR Ekranı Tasarımı

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 34

4.4.2 Geçiş Ekranı

Söz konusu ekran OCR teknigi sonucunda elde edilen verilere dayalı olarak kullanıcıyı sorun çözüm önerilerinin bulunduğu ekrana yönlendirirken veya herhangi bir hata ile karşılaşıldığında devreye giren bir ara yüzdür. Bu ekran, kullanıcı dostu bir deneyim sunmak amacıyla özel olarak tasarlanmıştır. Kullanıcıların, uygulama içerisindeki geçişler sırasında veya potansiyel hata mesajlarıyla karşılaşıklarında, sürecin sorunsuz ve anlaşılır olduğundan emin olmak için önemlidir. Geçiş ekranı sayesinde, kullanıcılar işlem devam ederken bilgilendirilir ve beklenen süreç hakkında sabırı olmaları teşvik edilir. Bu, kullanıcılarla uygulama içerisindeki navigasyonun kesintisiz ve etkili bir şekilde devam ettiğine dair güven verir, böylece kullanıcılar gerekli çözümlere ulaşmak için gereken süreyi daha rahat kabul edebilirler.



Şekil 4.15 Geçiş Ekranı Tasarımı

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 35

```

import React from 'react';
import { View, ActivityIndicator, Text, StyleSheet } from 'react-native';

const LoadingScreen = () => {
    return (
        <View style={styles.container}>
            <ActivityIndicator size="large" color="#fa4b51" />
            <Text style={styles.text}>Yükleniyor...</Text>
        </View>
    );
};

const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center',
        backgroundColor: '#fff',
    },
    text: {
        marginTop: 10,
        fontSize: 20,
        color: 'black',
    },
});

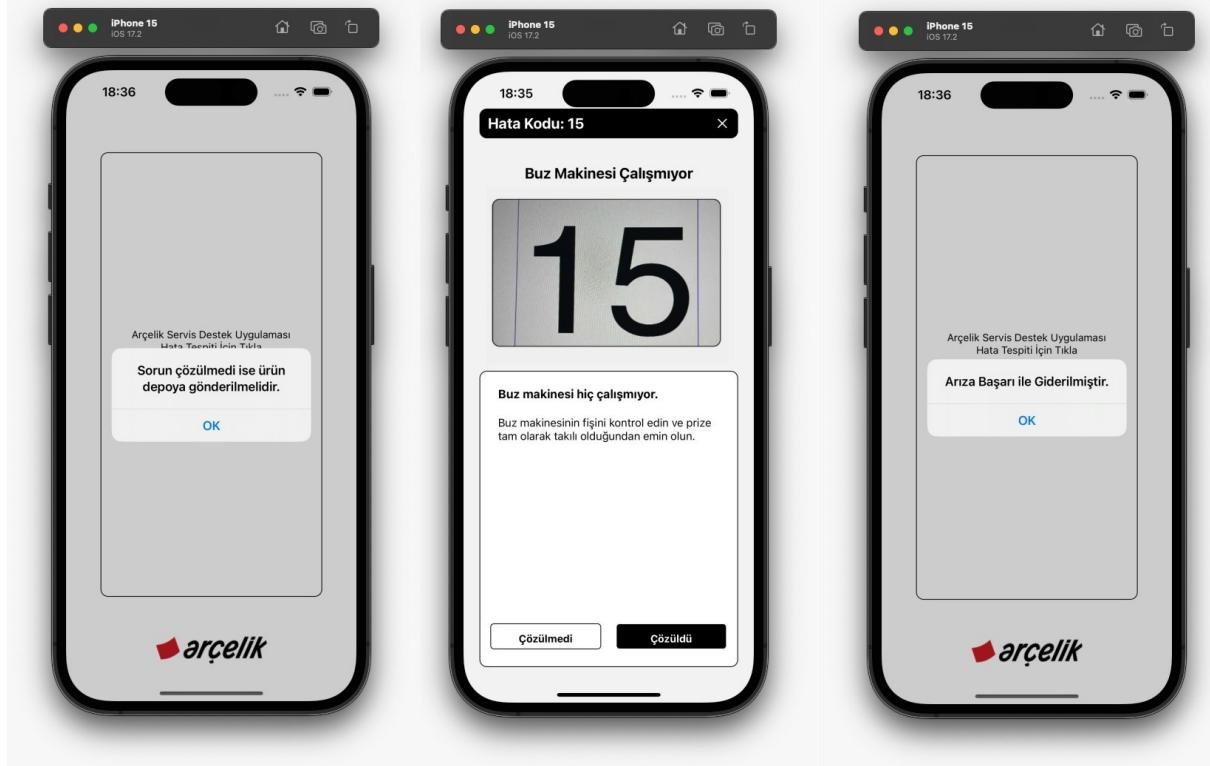
export default LoadingScreen;
|
```

Şekil 4.16 Geçiş Ekranı Kodu

4.4.3 Çözüm Yönergeleri Ekranı

Söz konusu ekranada, kullanıcılar kamera ile çektileri kodun fotoğrafını ve bu fotoğrafta tespit edilen ilgili hata koduna yönelik gerekli arıza çözüm adımlarını inceleyebileceklerdir. Bu ekranada sunulan bilgiler arasında hata kodunun genel adı, kısa bir açıklaması ve detaylı çözüm adımlarından ilki yer alacaktır. Ekranın alt kısmında bulunan “Çözülmedi” seçeneğini seçerse kullanıcı bir sonraki çözüm adımını görmüş olacaktır. Her bir adımda çözülmeli seçeneğini seçerse kullanıcı en sonunda: “Sorun Çözülemedi ise ürün depoya götürülmelidir” uyarı ekranada bulunuyor. Eğer herhangi bir adımda “Çözüldü” seçeneğini seçerse: “Arıza başarı ile giderilmiştir” mesajını ekran'a getiriyor. Kullanıcının bir diğer seçeneği ise ekranın sağ üst köşesinde yer alan “X” simgesine tıklayarak giriş ekranına dönebilirler. Bu işlem sayesinde, çekilen fotoğrafın depolanmayacak olması, uygulamanın ilk yüklenliğinde sahip olduğu hız ve güvenilirliği korumasını sağlar. Bu tasarım, kullanıcı deneyimini ön planda tutarak, arıza çözüm sürecinin hem verimli hem de kullanıcı dostu bir şekilde ilerlemesine olanak tanır.

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 36



Şekil 4.17 Çözüm Yönergeleri Ekranı

```

const getErrorByName = (code) => {
    return new Promise((resolve, reject) => {
        db.transaction(tx => {
            tx.executeSql(
                'SELECT * FROM errors WHERE error_code = ?',
                [code],
                (_, {rows}) => {
                    const data = rows._array;
                    resolve(data); // Veriyi resolve ediyoruz
                },
                (_, error) => {
                    reject(error); // Hata durumunda reject ediyoruz
                }
            );
        });
    });
}

-----
const error_code_set = [
    "0", "1", "2", "3", "4", "5", "6", "7", "8", "9",
    "10", "11", "12", "13", "14", "15", "17", "18",
    "19", "1A", "1B", "1C", "1D", "1E", "1F", "A",
    "B", "C", "D", "E", "F"
]

function checkErrorCodeInText(text) {
    let match = null;
    error_code_set.some(code => {
        const regex = new RegExp(`\\b${code}\\b`, 'i');
        const result = regex.test(text);
        if (result){
            match = code;
            return true;
        }
    });
    return match || null;
}
export default checkErrorCodeInText;

```

Şekil 4.18 Veritabanı Sorgusu ve Yazı Filtreleme Kodu

Staj Yetkilisi :Mehmet Hilal Özcanhan	İmza :
---------------------------------------	--------

Adı Soyadı : Emre Özkaya	Firma Ünvanı : Dokuz Eylül Üniversitesi Bilgisayar Mühendisliği Bölümü
Staj Türü : Donanım	Başlangıç Tarihi : 18/01/2024 Bitiş Tarihi : 09/02/2024 Sayfa : 37

BÖLÜM 5

Kaynakça

Kaynakça

- [1] [https://en.wikipedia.org/wiki/Jira_\(software\)](https://en.wikipedia.org/wiki/Jira_(software))
- [2] <https://appmaster.io/tr/blog/jira-nedir>
- [3] <https://www.dopinger.com/tr/blog/jira>
- [4] <https://excalidraw.com>
- [5] <https://reactnative.dev>
- [6] <https://expo.dev>
- [7] <https://docs.expo.dev/versions/latest/sdk/sqlite/>
- [8] <https://cloud.google.com/vision/docs>
- [9] <https://www.lucidchart.com>