

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

ISE 402 BİTİRME ÇALIŞMASI

FLUTTER İLE MOBİL UYGULAMA GELŞİTİRME
NATURE SAVE

B17120008 - Emine DEMİRCAN

B171200025 - Gülşah SEVİNEL

Fakülte Anabilim Dalı : BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

Tez Danışmanı : Dr.Öğr.Üyesi Burcu ÇARKLI YAVUZ

Mayıs 2021

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

FLUTTER İLE MOBİL UYGULAMA GELŞİTİRME
NATURE SAVE

ISE 402 BİTİRME ÇALIŞMASI

Emine DEMİRCAN
Gülşah SEVİNEL

Fakülte Anabilim Dalı : BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ

**Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oy çokluğu ile
kabul edilmiştir.**

.....

Jüri Başkanı

.....

Üye

.....

Üye

ÖNSÖZ

Bu çalışmada görüş ve önerileri ile katkıda bulunan, çalışma boyunca desteğini esirgemeyen değerli hocamız Dr.Öğr.Üyesi Burcu ÇARKLI YAVUZ'a teşekkürlerimizi sunarız. Ayrıca bu süreçte bizden maddi manevi desteklerini esirgemeyen ailemiz ve arkadaşlarımıza da bir teşekkürü borç biliriz.

ÖNSÖZ	iii
SİMGELER VE KISALTMALAR LİSTESİ	vi
ŞEKİLLER LİSTESİ	vii
ÖZET	viii
BÖLÜM 1. GİRİŞ.....	1
1.1. Günümüz Plastik Sorunu.....	2
1.2. Projenin İşleyişi.....	4
BÖLÜM 2. YÖNTEMLER	5
2.1. Flutter Nedir?	5
2.2. Dart Nedir?.....	6
2.2.1. Dart Paketleri (Dart Packages) Nedir?.....	6
2.2.2. Plugin Paketleri Nedir?	7
2.2.3. Kullanılan Paketler.....	7
2.2.3.1. Cached Network Image.....	7
2.2.3.2. Cloud Firestore.....	8
2.2.3.2.1. Firestore’a Veri Ekleme	9
2.2.3.3. device_info.....	11
2.2.3.4. easy_localization.....	12
2.2.3.5. Flutter Phoenix.....	12
2.2.3.6. Flutter Platform Widgets.....	13
2.2.3.7. get_it	13
2.2.3.8. Flutter Location Plugin	14
2.2.3.9. Pedantic.....	15
2.2.3.10. Provider.....	15
2.2.3.11. Timeago	15
2.2.3.12. QRCode Plugin for Flutter.....	16
2.3. Firebase Nedir?	16
2.3.1. Firebase’in Sağladığı Özellikler	17
2.4. Adobe XD Nedir?	17
BÖLÜM 3. ARAYÜZ	18
3.1. Giriş Sayfasının Kodları	19
BÖLÜM 4. SONUÇLAR VE ÖNERİLER	22
PROJENİN İŞ ZAMAN ÇİZELGESİ	23

KAYNAKÇA.....	24
ÖZGEÇMİŞ	25
ISE 402 BİTİRME ÇALIŞMASI.....	26

SİMGELER VE KISALTMALAR LİSTESİ

SKA	: Sürdürülebilir Kalkınma Amaçları
SDK	: Software Development Kit (Yazılım Geliştirme Kiti)
iOS	: iPhone Operating System
DB	: Database
UI	: User Interface
UX	: User Experience
API	: Application Programming Interface (Uygulama Programlama Ara yüzü)
REST	: Representational State Transfer (Temsili Durum Aktarımı)
RPC	: Remote Procedure Call (Uzaktan Yordam Çağırıcısı)

ŞEKİLLER LİSTESİ

Şekil 1. 1 Sürdürülebilir Kalkınma Amaçları	2
Şekil 1. 2 Projenin İşleme Şeması.....	4
Şekil 1. 3 Yer tutucu ile Cached Network Image Kullanımı	7
Şekil 1. 4 İlerleme göstergesi ile Cached Network Image Kullanımı.....	8
Şekil 1. 5 Image builder ile Cached Network Image Kullanımı.....	8
Şekil 1. 6 State sınıfından Firestore örneği edinme	9
Şekil 1. 7 Firestore’a veri eklenmesi.....	9
Şekil 1. 8 Future yöntemi.....	10
Şekil 1. 9 Flutter’da kimlik doğrulaması	10
Şekil 1. 10 veri güncelleme.....	11
Şekil 1. 11 Merge değeri true olan verinin Firebase üzerinde gösterimi	11
Şekil 1. 12 device_info kullanımı	12
Şekil 1. 13 Phoenix widget kullanımı	12
Şekil 1. 14 Phoenix’in çağırılması	13
Şekil 1. 15 daha sonra erişilmek istenen verilerin kaydedilmesi	14
Şekil 1. 16 AppModel sınıfından get_it’e erişme	14
Şekil 1. 17Konum hizmeti ve İzin durumunun kontrol edilmesi.....	14
Şekil 1. 18 Değişken konumlarda geri arama yapma	15
Şekil 1. 19 Timeago kullanımı.....	15
Şekil 1. 20 QRCode kullanımı	16
Şekil 1. 21 Android için QRCode kullanımı.....	16
Şekil 1. 22 Ana Giriş Sayfası.....	18
Şekil 1. 23 Giriş ve Kayıt sayfaları	18
Şekil 1. 24 Profil sayfası	18
Şekil 1. 25 Anasayfa	18
Şekil 1. 26 Harita	18
Şekil 1. 27 Firebase App Id ve Package name’in bulunduğu sayfa.....	19
Şekil 1. 28 Firebase Authentication izinleri sayfası.....	19
Şekil 1. 29 Kullanıcılara kayıt izni verilmesi.....	20
Şekil 1. 30 Firebase’te kullanıcıların tutulduğu yer	20
Şekil 1. 31 Uygulamayı veri tabanına bağlama	20
Şekil 1. 32 Giriş yapma fonksiyonu.....	21
Şekil 1. 33 Giriş yapılan bilgilerin kontrolünü yapan fonksiyon.....	21
Şekil 1. 34 Girişin başarılı olursa ya da olmazsa gösterilecek mesajları tanımlayan class	22

ÖZET

Anahtar kelimler: Mobil Uygulama, Plastik, Geri Dönüşüm, Sürdürülebilir Kalkınma Amacı

Günümüzde internet ve elektronik ticaret uygulamaları her alanda yaygınlaşmakta ve sıklıkla kullanılmaktadır. Mobil uygulamalar da en fazla kullanılan uygulama türlerinden birisidir. Özellikle de akıllı telefon kullanımı artımı ile herkese ulaşabilme kolaylığı daha da artmıştır. Bu çalışma da günümüz problemlerinden biri olan geri dönüşüm sistemini dolaylı bir ödül haline getirerek tüketici sınıfı sürece tamamen aktif bir şekilde dahil etmeyi sağlayan proje anlatılacaktır. Bu uygulamanın tamamı Flutter ile geliştirilecektir. Veri tabanı olarak Firebase kullanılacaktır. Ara yüz tasarımları Adobe XD üzerinden yapılmıştır.

BÖLÜM 1. GİRİŞ

Yüzyılda başlayan Sanayi Devrimi tüm dünya ülkelerinin üretim ve ekonomik yapısını etkilemiş, sanayileşen ülkelerin hem iç ekonomileri hem de dış ilişkileri köklü bir şekilde değiştirmiştir. Üretimin artmasına bağlı olarak üretim fazlası meydana gelen ülkeler yeni pazar arayışlarına başlamıştır. Ayrıca yoğun üretim nedeniyle çevresel bozulmalar meydana gelmiştir. Sürdürülebilir kalkınma ile büyümenin yoksulluğunu azaltmak, dünya üzerinde yaşayanlar için ortak refahı tesis etmek ve gelecek nesillerin ihtiyaçlarını karşılamaya devam etmek için hem kapsayıcı hem de çevreci olması gerektiği kabul edilmektedir. Bu durum sürdürülebilir kalkınmanın üç temel dayanağını ön plana çıkarmaktadır. Bunlar; ekonomik büyüme, çevresel yönetim ve sosyal içermelidir.

2015 yılında kabul edilen Gündem 2030 belgesinde 17 Sürdürülebilir Kalkınma Amacı ve 169 Sürdürülebilir Kalkınma Hedefi belirlenmiştir. Özellikle yoksulluğun arttığı, açlığın ölümcül seviyelere geldiği, tarımsal alanların verimini kaybettiği, okyanus asitleşmesinin geri dönülemez seviyelere geldiği, küresel iklim değişikliğinin etkilerinin şiddetli bir şekilde hissedildiği, biyolojik çeşitliliğin ve genetik kaynakların azaldığı, çölleşme ve ormansızlaşma faaliyetlerinin sonlandırılmadığı bir dünya tüm dünya devletleri için küresel ortak sorundur.

Bu sorunlara çözüm için SKA altında: SKA-1 (yoksulluğun bitirilmesi), SKA-2 (açlığa son), SKA-3 (sağlıklı bireyler), SKA-4 (eğitim), SKA-5 (kadının güçlendirilmesi ve cinsiyet eşitliği), SKA-6 (içme ve kullanma suyu), SKA-7 (sürdürülebilir enerji), SKA-8 (istihdam ve kalkınma), SKA-9 (endüstri ve yenilik), SKA-10 (eşitsizliklerin azaltılması), SKA-11 (şehirler), SKA-12 (sürdürülebilir üretim ve tüketim), SKA-13 (iklim değişikliğiyle mücadele), SKA-14 (sudaki yaşam), SKA-15 (karadaki yaşam), SKA-16 (barış ve adalet) ve SKA-17 (uygulama araçları) amaçlarının küresel ölçekte belirlenmesi önemli bir ilerlemedir. Kısaca SKA dünyadaki sorunların çözümü için 2030 yılına kadar ulaşılması için gereken amaç ve hedefleri içermektedir. ^[1]



Şekil 1. 1 Sürdürülebilir Kalkınma Amaçları

Proje kapsamında SKA-12 “sorumlu üretim ve tüketim” amacı incelenmiştir. Bu projenin amacı, üretim sektörünü düzenli olarak üretilen plastik israfından kurtarmak, geri dönüşüm sistemini dolaylı bir ödül haline getirerek tüketici sınıfı sürece tamamen aktif bir şekilde dahil etmek ve kapsamlı ve başarılı geri dönüşümden gelen kazancı topluma yansıtabilmek şeklinde sıralanabilir. Bu amaçlar kapsamında inovatif çözüm, üretim, tüketim ve tedarik zincirlerinin sorunlarıyla mücadelede süreci kolaylaştırma ve sorundan çıkan zararı kazançla telafi etme amacıyla geliştirilecek.

1.1. Günümüz Plastik Sorunu

Dünyada plastik üretiminin büyük bir bölümü Asya ülkelerinde üretiliyor. İkinci sırada ise Avrupa gelmektedir. Avrupa ülkelerinde sadece Norveç ve İsviçre 2016 yılında 60 milyon ton plastik üretmiştir.^[2]

Dünyada her yıl 500 milyar ile 1 trilyon arasında plastik poşet kullanılmakta, bu sayının 400 milyarı sadece Amerika Birleşik Devletleri’nde tüketilmektedir.^[3] Türkiye’de ise bir

kiři, yılda ortalama 440 plastik pořet kullanmaktadır.^[4] Ayrıca dünya genelinde yılda 5 milyar plastik pořetin řöplere atıldığı belirtilmektedir.

Plastik biyolojik olarak parçalanamadığından ötürü çevreye ciddi derecede zarar vermektedir ve plastik pořet kullanımının bu kadar sık olması ile de bazı önlemler alma gerekliliğı doğurmuřtur. Bu önlemlerden birisi plastik pořet yasaklamak olmuřtur. Bu önlemi alan ölkelerden bazıları Amerika, İngiltere, Meksika ve Hindistan'dır. Diğer bir önlem ise plastik pořetleri ücretli hale getirmek olmuřtur.

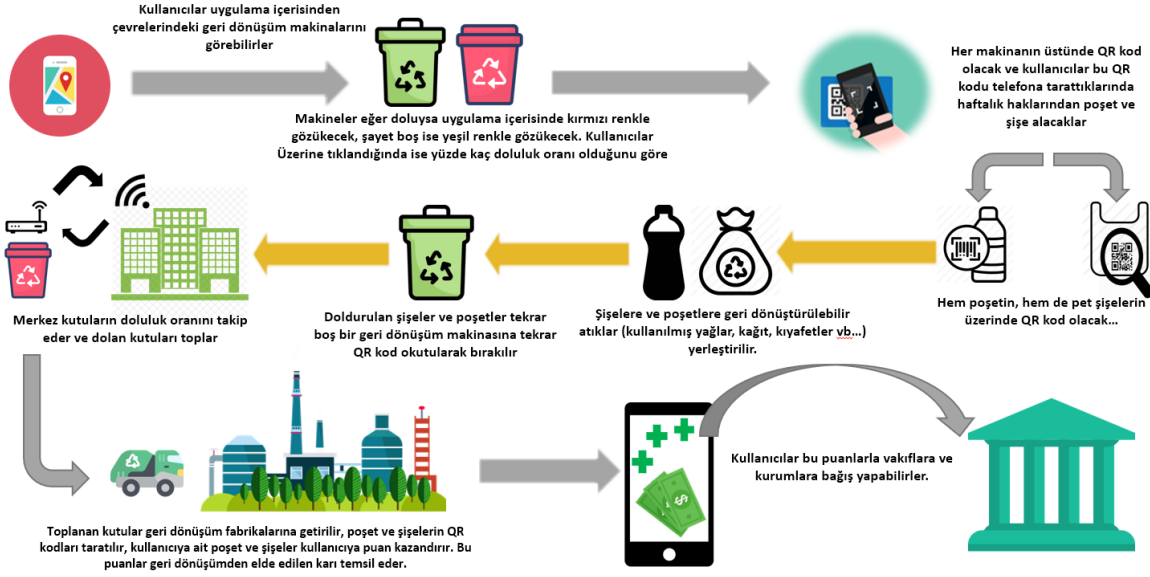
Türkiye'de ise, 1 Ocak 2019 tarihinden itibaren ücretli pořet uygulamasına geçilmiş, marketlerde ve diğer mağazalarda plastik pořetler, tüketicilere ücret karşılığında verilmektedir. Plastik pořetler, 25 kuruřtan az olmamak üzere Bakanlık tarafından oluşturulacak komisyonca belirlenmektedir.^[5] Böylece plastik pořet kullanımının azaltılması amaçlanmaktadır.

Belediyelerin geri dönüşüm sahaları da plastik atık miktarını ve diğer geri dönüřtürülebilir ürünlerin atık miktarını azaltmak için kullanılan girişimlerden biridir. Değerlendirilebilir Atık Malzemeler Sanayicileri Derneğı (TÜDAM) tarafından yapılan çalışmada, Türkiye'de yerleşim birimlerinde yılda yaklaşık 6 milyon geri dönüřtürülebilir nitelik evsel oluřturmakta ancak yılda bunun yaklaşık 5 milyon tonu geri dönüřtürölmek yerine řöpe gitmektedir.^[6] Bunun Türkiye ekonomisine zararı 2,25 milyar Türk lirasıdır.

Böyle girişimler ne kadar etkili olsa da hala yeterli gelmemektedir. Science dergisinde yayımlanan arařtırmaya göre, Türkiye plastik atıkları dönüřtürme konusunda en başarısız ilk 20 ölkedir. 2016 yılında řöpleri toplamakla yükümlü olan belediyeler 31 milyon ton řöp topladılar. Türkiye İstatistik Kurumu'na göre bu řöplerin sadece yüzde 9,8'i geri dönüşüme gönderilmiştir.^[7]

1.2.Projenin İşleyişi

Projenin İşleme Şeması



Şekil 1. 2 Projenin İşleme Şeması

Proje kapsamında geri dönüşüm için anlaşılan belediyelerle, o ilçelerde en yaygın kesimlerine özel çöp kutuları yerleştirilir. Kullanıcılar uygulama içerisinde çevrelerindeki geri dönüşüm makinalarını görebilirler. Makineler eğer dolu ise uygulama içerisinde kırmızı renkle gözükecek, şayet boş ise yeşil renkle gözükecek. Kullanıcılar üzerine tıkladığında ise doluluk oranını görebilecektir.

Her çöp kutuları üzerinde QR kod olacak ve kullanıcılar bu QR kodu telefona tarattıklarında haftalık haklarından poşet ve şişe alacaklardır. Hem poşetin hem de pet şişelerin üzerinde QR kod olacaktır. Kullanıcılar bu poşet ve şişelere geri dönüştürülebilir atıkları (kullanılmış yağlar, kâğıt, kıyafetler vb.) yerleştirir. Doldurulan şişeler ve poşetler yakındaki çöp kutusuna bırakılır. Merkez kutuların doluluk oranını takip eder ve dolan çöp kutularını toplar.

Toplanan çöp kutuları geri dönüşüm fabrikalarına getirilir, poşet ve şişelerin QR kodları taratılır. Kullanıcıya ait poşet ve şişeler kullanıcıya puan kazandırır. Bu puanlar geri

dönüşümden elde edilen karı temsil eder. Kullanıcılar bu puanlarla belediye ile anlaşmalı vakıf ve kurumlara bağış yapabilirler.

BÖLÜM 2. YÖNTEMLER

2.1. Flutter Nedir?

Flutter, Google tarafından geliştirilen açık kaynak kodlu bir UI yazılım geliştirme kitidir. Android, iOS, Windows, Mac, Linux ve web için uygulama geliştirmek için kullanılır. Flutter'ın ilk sürümü “Sky” olarak biliniyordu ve Android işletim sisteminde çalışıyordu.^[8]

Mobil uygulama oluşturmanın kolaylığı nedeniyle Flutter, mobil uygulamanın daha hızlı tasarlanmasına ve geliştirilmesine katkıda bulunur. Zaman tasarrufu sağlar. Flutter, widget kütüphanesindeki kodun %95'inin tekrar kullanılmasını sağlar. Ana ekranda uygulama değişikliklerinin güncellenmesi bir dakikadan daha az bir zaman sürer. Uygulama ne kadar hızlı pazara ulaşırsa geliştirici de daha çabuk geri dönüş alır ve eksiklerini daha hızlı bir şekilde giderir.

Ücretsiz erişilebilen yüzlerce üçüncü taraf paket, reklam, video, bulut vb. birçok yardımcı eklentiler içerir. Açık kaynak olması sebebi ile de bir sorun ile karşılaşıldığında internet üzerindeki topluluklardan kolay bir şekilde soruna çözüm bulunabilmektedir.

Flutter çapraz bir platformdur, aynı altyapıyı kullanarak hem iOS hem de Android tabanlı uygulamalar geliştirilebilir. Bu her açıdan geliştiriciye tasarruf sağlayacaktır. Artık mobil uygulamalarda animasyon daha çok kullanılmaktadır. En sıradan görünen uygulamada bile listeler, switchler, checkbox'lar, sayfa geçişleri, diyaloglar, gesture efektleri vs. aslında hepsinde animasyonla vardır.

Flutter'da ara yüz katmanı ne native SDK'lere bağımlı ne de onların üzerine kurulan birtakım soyutlamalara, ara yüzlere, köprülere vs. bağımlıdır. Tek bir dille, donanım seviyesinde native, tek bir codebase'den aklınıza gelebilecek her türlü kullanıcı ara yüzü mümkündür. Üstelik Flutter çok zengin bir hazır ara yüz bileşenleri kataloğu sunmaktadır. Android ve iOS SDK'leri birbirinden çok farklıdır. cross-platform çözümlerde geliştirici kodu olabildiğince paylaşmak ister ve bu gibi platform farklılıklarında zorlanabilir. Kod ayırımı yapmak, ne zorluk varsa ondan en az 2 tane, hatta 3 tane demektir. Flutter'da platformlar arasında ne kadar ayırım yapılacağı geliştiricinin tercihinine bağlıdır. Çünkü hazır ara yüz bileşenleri native SDK'leri bire bir taklit edecek şekilde ayrılmıştır ama cross-platform da çalışabilir ve daha da önemlisi geliştirici kendi çözümlerini de üretebilir.

2.2. Dart Nedir?

Dart, native derlemeyi Linux, macOS ve Windows masaüstü platformlarına kapsamaktadır. Daha önceki geliştiriciler yalnızca Android veya IOS cihazlarını kullanarak yeni araçlar oluşturabiliyorlardı. Dahası, bu uzantı ile bağımsız bir yürütülebilir programa bir Dart programı oluşturmak mümkün hale gelmiştir. Bu nedenle, proje temsilcilerine göre, Dart SDK'nın kurulması artık zorunlu değil, bağımsız çalıştırılabilir dosyalar birkaç saniye içinde çalışmaya başlayabilmektedir.

Ayrıca Flutter araç seti ile de entegre edilmiştir, böylece derleyiciyi küçük servislerde (örneğin arka uç desteği) kullanmayı mümkün kılmaktadır.

2.2.1. Dart Paketleri (Dart Packages) Nedir?

Native platformlardan bağımsız sadece Flutter ve Dart SDK'leri kullanarak yazılmış paketlerdir. Dart paketine örnek vermek gerekirse: `meet_network_image` paketi internet üzerinden resim alacaksanız, resim yüklenirken veya hata verdiğinde başka widget gösterebilmenize yarıyor.

2.2.2. Plugin Paketleri Nedir?

Dart kodunda yazılmış bir API içeren ve Android için (Java veya Kotlin kullanarak) ve / veya iOS (ObjC veya Swift kullanarak) platforma özgü bir uygulama içeren özel bir Dart paketi. Pluginlere örnek vermek gerekirse : image_picker plugini , iOS ve Android için ayrı ayrı API'ları kullanarak sizin resim çekebilmenize veya galerinizden var olan bir resmi alabilmenize yarıyor.

2.2.3. Kullanılan Paketler

2.2.3.1. Cached Network Image

Bazı durumlarda, çevrimdışı kullanılabilecekleri için görüntüleri WEB 'den indirilirken önbellege almak kullanışlıdır. Bunun için cached_network_image paketi kullanılır.

CachedNetworkImage, doğrudan veya ImageProvider aracılığıyla kullanılabilir. CachedNetworkImage olarak CachedNetworkImageProvider'ın her ikisi de web için minimum desteğe sahiptir.

Yer tutucu ile (placeholder):

```

CachedNetworkImage(
  imageUrl: "http://via.placeholder.com/350x150",
  placeholder: (context, url) => CircularProgressIndicator(),
  errorWidget: (context, url, error) => Icon(Icons.error),
),

```

Şekil 1. 3 Yer tutucu ile Cached Network Image Kullanımı

Veya bir ilerleme göstergesiyle (progress indicator):

```

CachedNetworkImage(
  imageUrl: "http://via.placeholder.com/350x150",
  progressIndicatorBuilder: (context, url, downloadProgress) =>
    CircularProgressIndicator(value: downloadProgress.progress),
  errorWidget: (context, url, error) => Icon(Icons.error),
),

```

Şekil 1. 4 İlerleme göstergesi ile Cached Network Image Kullanımı

Hem yer tutucu işlevine sahip olmak hem de görüntü sağlayıcının başka bir pencere ögesinde kullanmasını sağlamak istediğinizde bir imageBuilder sağlayabilirsiniz:

```

CachedNetworkImage(
  imageUrl: "http://via.placeholder.com/200x150",
  imageBuilder: (context, imageProvider) => Container(
    decoration: BoxDecoration(
      image: DecorationImage(
        image: imageProvider,
        fit: BoxFit.cover,
        colorFilter:
          ColorFilter.mode(Colors.red, BlendMode.colorBurn)),
    ),
  ),
  placeholder: (context, url) => CircularProgressIndicator(),
  errorWidget: (context, url, error) => Icon(Icons.error),
),

```

Şekil 1. 5 Image builder ile Cached Network Image Kullanımı

2.2.3.2. Cloud Firestore

Cloud Firestore, Firebase ve Google Cloud'dan mobil, web ve sunucu geliştirme için esnek, ölçeklenebilir bir veritabanıdır. Firebase Gerçek Zamanlı Veritabanı gibi, verilerinizi gerçek zamanlı dinleyiciler aracılığıyla istemci uygulamaları arasında senkronize tutar ve ağ gecikmesinden veya İnternet bağlantısından bağımsız olarak çalışan duyarlı uygulamalar oluşturabilmeniz için mobil ve web için çevrimdışı destek sunar. Cloud Firestore, Cloud Functions dahil olmak üzere diğer Firebase ve Google Cloud ürünleriyle de sorunsuz entegrasyon sunar.

Cloud Firestore, iOS, Android ve web uygulamalarınızın yerel SDK'lar aracılığıyla doğrudan erişebileceği, bulutta barındırılan bir NoSQL veritabanıdır. Cloud Firestore, REST ve RPC API'lerine ek olarak yerel Node.js, Java, Python, Unity, C++ ve Go SDK'larında da mevcuttur.

Cloud Firestore'un NoSQL veri modelini izleyerek verileri, değerlerle eşlenen alanlar içeren belgelerde depolarsınız. Bu belgeler, verilerinizi düzenlemek ve sorgular oluşturmak için kullanabileceğiniz belgeleriniz için kapsayıcılar olan koleksiyonlarda depolanır. Belgeler, basit dizeler ve sayılardan karmaşık, iç içe nesnelere kadar birçok farklı veri türünü destekler. Ayrıca belgeler içinde alt koleksiyonlar oluşturabilir ve veritabanınız büyüdükçe ölçeklenen hiyerarşik veri yapıları oluşturabilirsiniz. Cloud Firestore veri modeli , uygulamanız için en uygun veri yapısını destekler.

2.2.3.2.1. Firestore'a Veri Ekleme

Cloud Firestore'a veri eklemenin iki yolu vardır; birinci yol, belge adını belirlemektir ve ikinci yol, Cloud Firestore'un rastgele bir kimlik oluşturmaktır, her iki durumu da görelim. Bu nedenle, State sınıfınızda ilk önce bir Cloud Firestore örneği edinmeniz gerekir:

```
1 final firestoreInstance = FirebaseFirestore.instance;
```

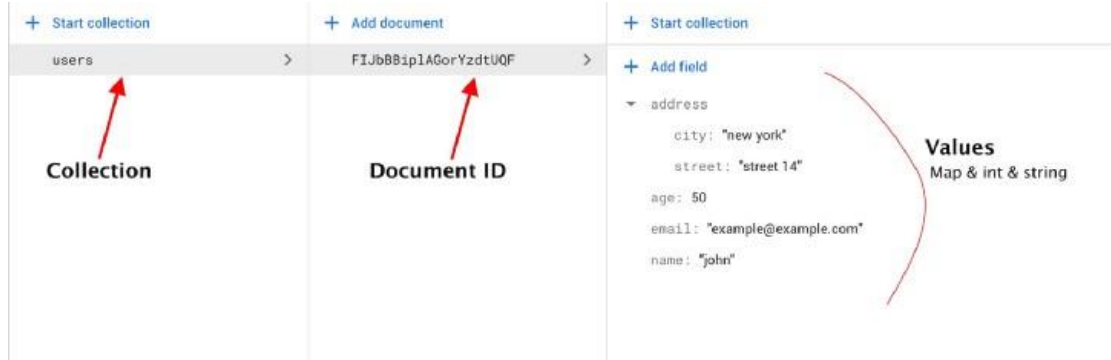
Şekil 1. 6 State sınıfından Firestore örneği edinme

Verilerin eklenmesi:

```
1 void _onPressed() {
2   firestoreInstance.collection("users").add(
3     {
4       "name" : "john",
5       "age" : 50,
6       "email" : "example@example.com",
7       "address" : {
8         "street" : "street 24",
9         "city" : "new york"
10      }
11    }).then((value){
12      print(value.id);
13    });
14 }
```

Şekil 1. 7 Firestore'a veri eklenmesi

Yukarıda görebileceğiniz gibi, bir butona basılarak bir kullanıcı koleksiyonu add() oluşturuyoruz ve rastgele bir id üretecek yöntemi kullanıyoruz . Yana add() yöntemi döndürür. Future<DocumentReference> bu nedenle kullanabilir yöntem then(), bir geri arama içerecektir çağrılacak Future bitirir. Geri aramaya value aktarılan bir parametre olan değişken tiptedir, DocumentReference bu nedenle özelliği id otomatik olarak oluşturulan kimliği almak için kullanabiliriz.



Şekil 1. 8 Future yöntemi

Şimdi Flutter'da Firebase Kimlik Doğrulaması kullandığınızı varsayalım, otomatik olarak oluşturulan bir kimlik kullanmak yerine userIddoküman kimliği olarak kullanabilirsiniz, bu şekilde verileri almak daha kolay olacaktır:

```
1 void _onPressed(){
2   var firebaseUser = FirebaseAuth.instance.currentUser;
3   firestoreInstance.collection("users").doc(firebaseUser.uid).set(
4     {
5       "name" : "john",
6       "age" : 50,
7       "email" : "example@example.com",
8       "address" : {
9         "street" : "street 24",
10        "city" : "new york"
11      }
12    }).then((_) {
13      print("success!");
14    });
15 }
```

Şekil 1. 9 Flutter'da kimlik doğrulaması

Burada, userIddoküman kimliği olarak kullandığımız için, bu nedenle set() belgeye veri eklemek için yöntemi kullanıyoruz. Bir belge zaten varsa ve onu güncellemek istiyorsanız, isteğe bağlı adlandırılmış parametreyi kullanabilir merge'ye şu şekilde true

ayarlayabilirsiniz:

```
1 void _onPressed() {
2   var firebaseUser = FirebaseAuth.instance.currentUser;
3   firestoreInstance.collection("users").doc(firebaseUser.uid).set(
4     {
5       "username" : "userX",
6     }, SetOptions(merge: true)).then((_) {
7       print("success!");
8     });
9 }
```

Şekil 1. 10 veri güncelleme

Bu şekilde belge içindeki verilerin üzerine yazılmaz.

+ Add field

▼ address

city: "new york"

street: "street 14"

age: 50

email: "example@example.com"

name: "john"

userName: "userX"

Şekil 1. 11 Merge değeri true olan verinin Firebase üzerinde gösterimi

2.2.3.3. device_info

Flutter uygulamasında güncel cihaz bilgilerini alır.

```
import 'package:device_info/device_info.dart';

DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
AndroidDeviceInfo androidInfo = await deviceInfo.androidInfo;
print('Running on ${androidInfo.model}'); // e.g. "Moto G (4)"

IosDeviceInfo iosInfo = await deviceInfo.iosInfo;
print('Running on ${iosInfo.utsname.machine}'); // e.g. "iPod7,1"
```

Şekil 1. 12 device_info kullanımı

2.2.3.4. easy_localization

easy_localization eklentisi ile uygulama uluslararasılaştırması hızlı ve kolaydır. Özellikle son güncelleme ile. Dil değiştirme işlevine sahip basit bir uygulama sayfası oluşturmaya olanak sağlar.

2.2.3.5. Flutter Phoenix

Phoenix, uygulamanızı uygulama düzeyinde yeniden başlatır, uygulama pencere ögesi ağacınızı sıfırdan yeniden oluşturarak önceki durumunu kaybeder. Phoenix, uygulama sürecinizi işletim sistemi düzeyinde tam olarak yeniden başlatmaz.

Kullanımı için öncelikler uygulama widget'ınızı Phoenix widget'ına sarıyoruz.

```
void main() {
  runApp(
    Phoenix(
      child: App(),
    ),
  );
}
```

Şekil 1. 13 Phoenix widget kullanımı

Uygulamanızı yeniden başlatmak istediğinizde yeniden doğuş statik yöntemini çağırmanız gerekiyor.

```
Phoenix.rebirth(context);
```

Şekil 1. 14 Phoenix'in çağırılması

2.2.3.6. Flutter Platform Widgets

Bu widget seti, tek bir çapraz platform widget seti kullanarak hedef platforma dayalı olarak işlemeye izin verir.

Tema parçacığıyla birlikte kullanılan çarpıntı ThemeData nesnesinin bir platform özelliği vardır. Bu, varsayılan olarak Android'de TargetPlatform.android ve iOS'ta TargetPlatform.ios (ayrıca macos vb. için) şeklindedir, ancak bir ThemeData nesnesi oluştururken programlı olarak ayarlanabilir. Theme.of(context).platform ögesinin çağırılması mevcut platformu döndürür. Birkaç Flutter kitaplığı widget'ı, nasıl oluşturulduklarını değiştirmek için bu alanı kullanır ve tüm Flutter Platform Widget'ları kitaplık widget'ları, şeylerin platforma özel sürümlerini oluşturmak için bu alanı kullanır.

2.2.3.7. get_it

Bu, Dart ve Flutter projeleri için Splat'tan oldukça ilham alan bazı ek özellikler içeren basit bir servis bulucudur. Nesnelere erişmek için InheritedWidget veya Provider yerine kullanılabilir.

Uygulamanız büyürse, bir noktada uygulamanızın mantığını Widget'larınızdan ayrılmış sınıflara koymanız gerekecektir. Widget'larınızı doğrudan bağımlılıklardan uzak tutmak, kodunuzu daha iyi organize eder ve test etmeyi ve bakımını kolaylaştırır. Ancak şimdi bu nesnelere UI kodunuzdan erişmenin bir yoluna ihtiyacınız var. .Net dünyasından Flutter'a geldiğimde bunu yapmanın tek yolu InheritedWidgets kullanmaktı. Bunları StatefulWidget'a sararak kullanmanın yolunu buldum; oldukça hantal ve sürekli çalışma sorunları var.

```
final getIt = GetIt.instance;

void setup() {
  getIt.registerSingleton<AppModel>(AppModel());

  // Alternatively you could write it if you don't like global variables
  GetIt.I.registerSingleton<AppModel>(AppModel());
}
```

Şekil 1. 15 daha sonra erişilmek istenen verilerin kaydedilmesi

```
MaterialButton(
  child: Text("Update"),
  onPressed: getIt<AppModel>().update // given that your AppModel
),
```

Şekil 1. 16 AppModel sınıfından get_it'e erişme

2.2.3.8. Flutter Location Plugin

Flutter için bu eklenti, Android ve iOS'ta konum almayı yönetir. Ayrıca konum değiştirildiğinde geri aramalar sağlar. Flutter 1.12 ile tüm bağımlılıklar projenize otomatik olarak eklenir. Projeniz Flutter 1.12'den önce oluşturulduysa, bunu takip etmeniz gerekebilir. Konum talep etmek için, Konum Hizmeti durumunu ve İzin durumunu her zaman manuel olarak kontrol etmelisiniz.

```
Location location = new Location();

bool _serviceEnabled;
PermissionStatus _permissionGranted;
LocationData _locationData;

_serviceEnabled = await location.serviceEnabled();
if (!_serviceEnabled) {
  _serviceEnabled = await location.requestService();
  if (!_serviceEnabled) {
    return;
  }
}

_permissionGranted = await location.hasPermission();
if (_permissionGranted == PermissionStatus.denied) {
  _permissionGranted = await location.requestPermission();
  if (_permissionGranted != PermissionStatus.granted) {
    return;
  }
}

_locationData = await location.getLocation();
```

Şekil 1. 17 Konum hizmeti ve İzin durumunun kontrol edilmesi

Ayrıca konumunuz değişirken sürekli geri aramalar alabilirsiniz:

```
location.onLocationChanged.listen((LocationData currentLocation) {
  // Use current location
});
```

Şekil 1. 18 Değişken konumlarda geri arama yapma

2.2.3.9. Pedantic

Bu paket üç amaca hizmet eder: Yazdığımız kod için en iyi uygulamalar da dahil olmak üzere, Dart statik analizinin Google'da dahili olarak nasıl kullanıldığını belgeler. Belgeler bu README.md'dir. Karşılık gelen bir analiz_options.yaml örneği içerir. Bu en iyi uygulamaların uygulanmasında kullanılan ara sıra küçük Dart kodu parçacıkları içerir.

2.2.3.10. Provider

InheritedWidget'ı manuel olarak yazmak yerine sağlayıcıyı kullanarak şunları elde edersiniz: kaynakların basitleştirilmiş tahsisi/imhası yavaş yüklenme her seferinde yeni bir sınıf oluşturmaya kıyasla büyük ölçüde azaltılmış bir kazan plakası devtools dostu bu InheritedWidget'ları tüketmenin yaygın bir yolu (Bkz. Provider.of/Consumer/Selector) karmaşıklığı katlanarak artan bir dinleme mekanizmasına sahip sınıflar için artırılmış ölçeklenebilirlik (bildirimleri göndermek için $O(N^2)$ olan ChangeNotifier gibi).

2.2.3.11. Timeago

Bu kütüphaneyi üst düzey fonksiyon formatı (tarih) ile kullanmanın en kolay yolu.

```
import 'package:timeago/timeago.dart' as timeago;

main() {
  final fifteenAgo = new DateTime.now().subtract(new Duration(minutes: 15));

  print(timeago.format(fifteenAgo)); // 15 minutes ago
  print(timeago.format(fifteenAgo, locale: 'en_short')); // 15m
  print(timeago.format(fifteenAgo, locale: 'es')); // hace 15 minutos
}
```

Şekil 1. 19 Timeago kullanımı

2.2.3.12. QRCode Plugin for Flutter

Kamera ile QR Kodlarını okumak için bir Flutter eklentisi.

```
Future<String> futureString = new QRCodeReader()  
    .setAutoFocusIntervalInMs(200) // default 5000  
    .setForceAutoFocus(true) // default false  
    .setTorchEnabled(true) // default false  
    .setHandlePermissions(true) // default true  
    .setExecuteAfterPermissionGranted(true) // default true  
    .scan();
```

Şekil 1. 20 QRCode kullanımı

```
Future<String> futureString = new QRCodeReader().scan();
```

Şekil 1. 21 Android için QRCode kullanımı

2.3. Firebase Nedir?

Firebase, Google tarafından mobil ve web uygulamaları oluşturmak için geliştirilmiş bir platformdur. Bağımsız bir şirket olarak 2011 senesinde kuruldu. Google 2014'te platformu satın aldı. Şirket, uygulama geliştirme konusunda Google'ın öncü hizmetidir

Herhangi bir platformdan Firebase kullanarak kullanıcı ekleyebilir, silebilir kullanıcının bilgileri anlık olarak değiştirebiliriz (Yani IOS uygulamasında eklediğimiz kullanıcıyı hem webden hem Android 'den hızlıca takip edebiliriz). Bunun dışında yazdığınız bir uygulamanın analizlerini ve raporlarını rahatlıkla takip edebilirsiniz.

Örnek olarak uygulama ne zaman nasıl hata almış gibi. Hangi şehirde daha çok uygulamanızı kullanıyor gibi birçok analize ulaşabilirsiniz. Bence en büyük özelliklerinden birisi bir platformda değiştirdiğiniz herhangi bir bilgiyi (DB içindeki) diğer platformdan kolayca kullanıcıya yansıtabilirsiniz.

2.3.1. Firebase'in Sağladığı Özellikler

- Eşzamanlı veritabanı
- Kullanıcı girişlerinde yetkilendirme
- Depolama
- Makine öğrenme kiti
- Performans test ortamı
- Platform arası ortak biçimde uygulama analizleri
- Ortak fonksiyon
- Reklam materyalleri
- Bildirim gönderimi
- Genel uygulama yönetimi

2.4. Adobe XD Nedir?

Web siteleri ve mobil uygulamalar için (UI ve UX) yani kullanıcı ara yüzü ve kullanıcı deneyimini tasarlamak ve prototip hale getirerek test etmek için Adobe tarafından geliştirilen Adobe XD, vektör tabanlı bir araçtır. Üstelik bu aracı MacOS, iOS, Windows ve Android işletim sistemlerinde kullanılabilir.

Bilindiği üzere Google, arama sonuçlarında ön plana çıkaracağı sitelerde kullanıcı odağı ve kullanıcı deneyimi ön planda tutuyor. Kullanıcı memnuniyeti istiyor. Bunu sağlamak içinde web sitesi sahiplerinin bazı faktörleri yerine getirmesi gerekiyor. Bu faktörler içerisinde de kullanıcı deneyimi önemli bir yer ediniyor. İşte Adobe XD (Experience Design) ile kullanıcı deneyimini farklı açılardan test edilebiliyor.

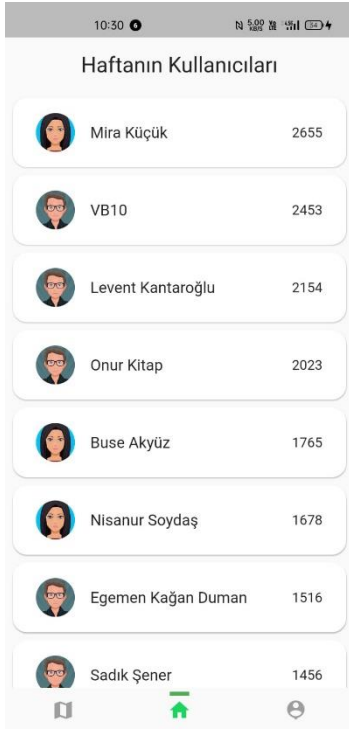
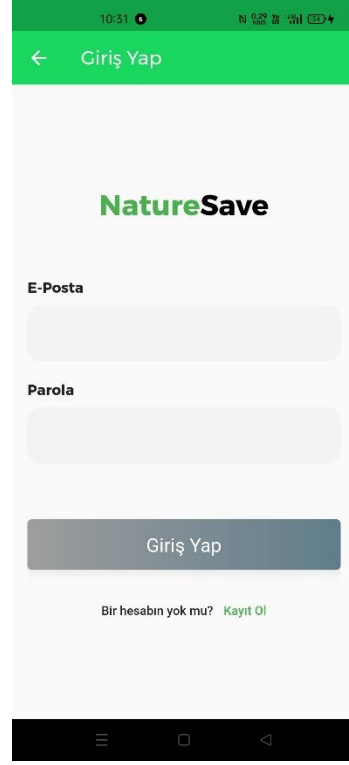
BÖLÜM 3. ARAYÜZ



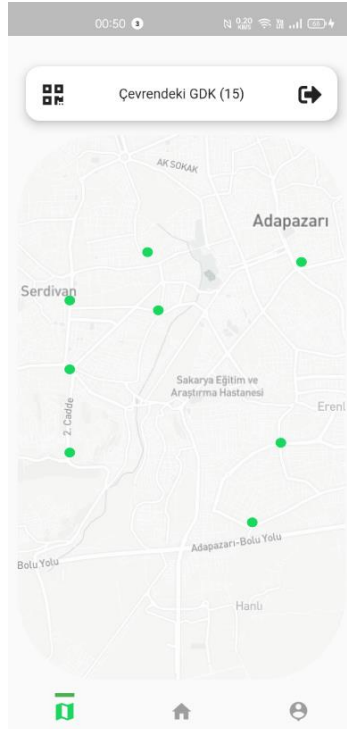
Şekil 1. 22 Ana Giriş Sayfası



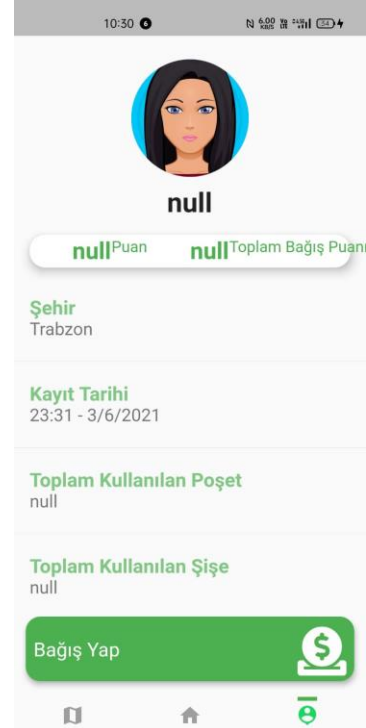
Şekil 1. 23 Giriş ve Kayıt sayfaları



Şekil 1. 25 Anasayfa



Şekil 1. 26 Harita



Şekil 1. 24 Profil sayfası

3.1. Giriş Sayfasının Kodları

App ID ⓘ
1:387939042496:android:2560d3bc01e1b6ca651925

App nickname
Add a nickname ✎

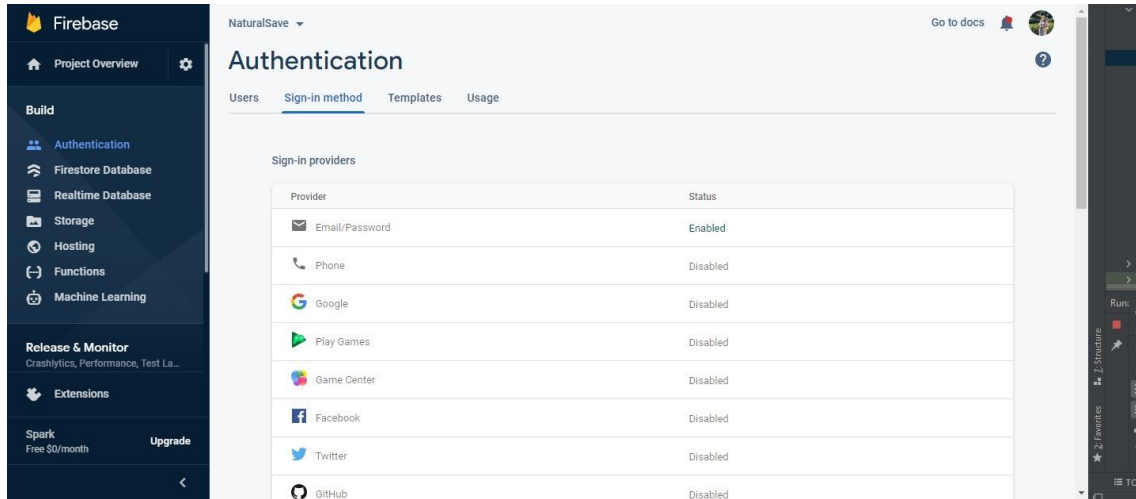
Package name
com.example.naturesave

SHA certificate fingerprints ⓘ
fb:b1:ce:29:b2:d4:38:e0:4d:6e:16:be:9f:c2:27:09:36:77:04:5d

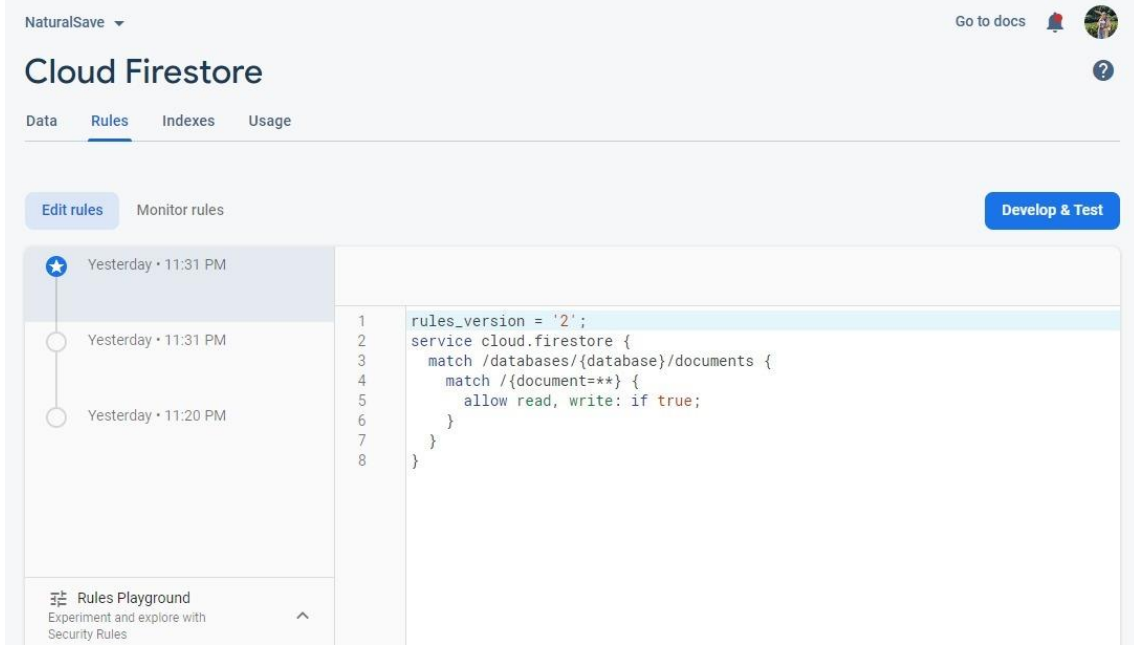
Type ⓘ
SHA-1

Şekil 1. 27 Firebase App Id ve Package name'in bulunduğu sayfa

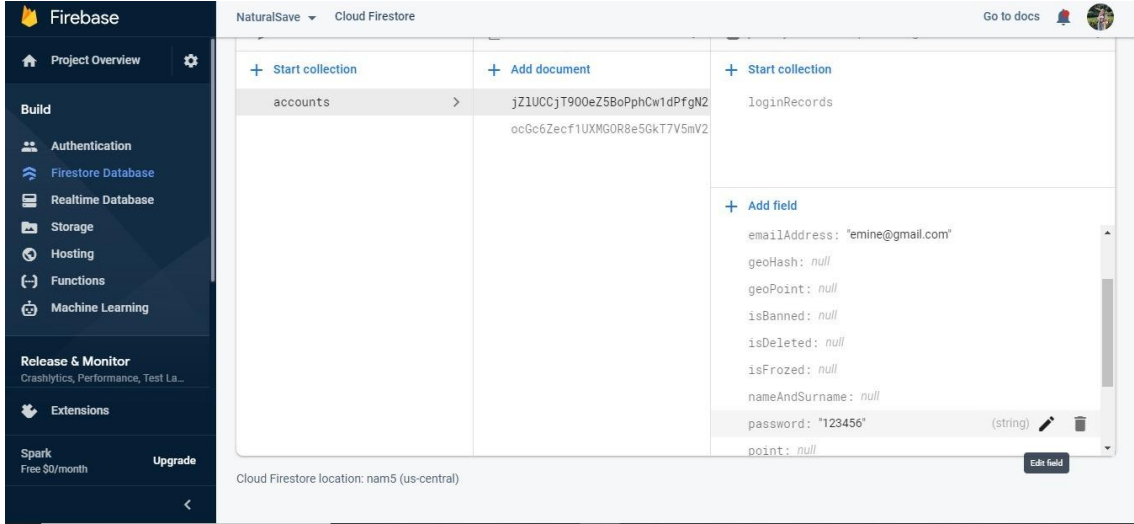
Sha! parmak izidir. Uygulama paket yöneticisi ve elektronik medya dağıtım/satış platformlarında yayınlanınca buna ihtiyaç duyar. com.example.naturesave projenin paket ismidir.



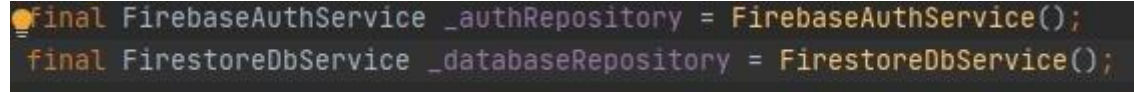
Şekil 1. 28 Firebase Authentication izinleri sayfası



Şekil 1. 29 Kullanıcılara kayıt izni verilmesi



Şekil 1. 30 Firebase'te kullanıcıların tutulduğu yer



Şekil 1. 31 Uygulamayı veri tabanına bağlama

```

Future<AccountQuery> signInWithEmailAndPassword(
    {String email, String password}) async {
    state = AccountState.Logining;
    var aQuery = await _authRepository.signInWithEmailAndPassword(
        email: email, password: password);

    if (aQuery.isSuccess) {
        currentUserId = aQuery.account.userId;

        var dQuery = await _databaseRepository.signInToDatabase(
            userId: aQuery.account.userId, password: password);

        if (dQuery.isSuccess) {
            state = AccountState.Idle;
            listenCurrentUser();
            return dQuery;
        } else {
            await signOut();
            state = AccountState.Idle;
            return AccountQuery.failed(cErrorType: 0);
        }
    }
    state = AccountState.Idle;
    return aQuery;
}

```

Şekil 1. 32 Giriş yapma fonksiyonu

```

void _login() async {
    var cQuery = await _accountProvider.signInWithEmailAndPassword(
        email: _emailController.text, password: _passwordController.text);

    if (cQuery.isSuccess) {
        _execHomeScreen();
    } else {
        /*
        Hata ele alınacak base widgetdan
        */
    }
}

```

Şekil 1. 33 Giriş yapılan bilgilerin kontrolünü yapan fonksiyon

```
import 'account.dart';

class AccountQuery {
  Account account;
  dynamic error;
  int errorType;
  bool isSuccess;

  AccountQuery.success({Account cAccount}) {
    account = cAccount;
    isSuccess = true;
  }

  AccountQuery.failed({dynamic cError, int cErrorType}) {
    error = cError;
    errorType = cErrorType;
    isSuccess = false;
  }
}
```

Şekil 1. 34 Girişin başarılı olursa ya da olmazsa gösterilecek mesajları tanımlayan class

BÖLÜM 4. SONUÇLAR VE ÖNERİLER

Flutter'ın gelişmesi ile birlikte bu alanda geliştirilen mobil uygulama sayısı da her geçen gün artmakta, bu uygulamalara olan talep hızla büyümektedir. Mobil uygulama sistemlerin gelişmesi ve yaygınlaşması nedeniyle bir çalışma yapılmış ve ihtiyaçlar giderilmeye çalışılmıştır. Mobil cihazların yaygın olarak kullanılması ve açık kaynak işletim sistemlerinin tanınmış olduğu imkânı tanınması sebebi ile Flutter seçilmiştir. Günümüzde birçok teknoloji firması, kendi mobil uygulamalarını Flutter ile geliştirmeye ve dönüştürmeye başlamıştır. Flutter'ın kullanımı dünden güne artmaktadır.

Uygulamamız ülkemizde büyük bir çevre sorunu haline gelen plastik atık sorununu yetkili belediyelerin geri dönüşüm sistemleri aracılığı ve ödül sistemi ile azaltma amacıyla geliştirilmiştir.

Artık neredeyse herkesin bir mobil cihazı olduğu için mobil uygulama geliştirilmiştir.

Çalışmamız ileriki aşamalarda geliştirilmeye açık bir çalışmadır

PROJENİN İŞ ZAMAN ÇİZELGESİ

İşin Tanımı	Aylar			
	MART	NİSAN	MAYIS	HAZİRAN
Konu Belirlenmesi	x			
Kullanılacak Yöntem Seçimi ve Yol Haritası	x			
Ara yüz Tasarımı	x			
Teknik Eğitimler	x	x		
Login İşlemleri		x		
Veri Tabanı		x	x	
Harita İşlemleri		x	x	
Raporlama			x	x

KAYNAKÇA

- [1] <https://www.kureselamaclar.org/>, Eriřim Mayıs 2021
- [2] <http://bilimveaydinlanma.org/content/images/pdf/rapor/kapitalizmin-yol-actigi-plastik-sorunu-nedir-nasil-basedecegiz.pdf>, Eriřim Mayıs 2021
- [3] Kayan, A. Küçük, A. (2020) Plastik Kirliliğın Çevresel Zararları ve Çözüm Önerileri. Ankara Hacı Bayram Veli Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi (Nisan 2020)
- [4] <https://www.aa.com.tr/tr/turkiye/her-yil-5-milyon-ton-evsel-atik-cope-gomuluyor/820803>, Eriřim Haziran 2021
- [5] <https://www.aa.com.tr/tr/turkiye/her-yil-5-milyon-ton-evsel-atik-cope-gomuluyor/820803>, Eriřim Haziran 2021
- [6] <https://www.birgun.net/haber/geri-donusturmede-en-basarisiz-20-ulke-icindegiz-260901>, Eriřim Haziran 2021
- [8] <https://tr.wikipedia.org/wiki/Flutter>, Eriřim Mayıs 2021

ÖZGEÇMİŞ

Emine Demircan, 10.04.1999 da Tokat'ta doğdu. İlk, orta ve lise eğitimi İstanbul'da tamamladı. 2017 yılında Vefa Poyraz Anadolu Lisesi Sayısal bölümünden mezun oldu. 2017 yılında Sakarya Üniversitesi Bilişim Sistemleri Mühendisliği Bölümü'nü kazandı. 2019 yılında Fırat Plastik Kauçuk Sanayi ve Ticaret Anonim Şirketine işletme stajını, 2020 yılında da Yıldız Teknoparkta yazılım stajını yapmıştır. Şu an Softinyo şirketinde Junior Mobile Developer olarak çalışmaktadır. SAÜ Bilişim Sistemleri Mühendisliği Bölümünden 2021 yılında mezun olacaktır.

Gülşah Sevinel, 24.06.1998 de İstanbul'da doğdu. İlk, orta ve lise eğitimini Büyükçekmece'de tamamladı. 2016 yılında Özel Uğur Temel Lisesi'nden mezun oldu. 2017 yılında Sakarya Üniversitesi Bilişim Sistemleri Mühendisliği Bölümü'nü kazandı. 2020 yılında Erdoğanlar Alüminyum San. Tic. Aş. işletme stajını yapmıştır. SAÜ Bilişim Sistemleri Mühendisliği Bölümünden 2021 yılında mezun olacaktır.

ISE 402 BİTİRME ÇALIŞMASI

Değerlendirme ve Sözlü Sınav Tutanağı

KONU :

ÖĞRENCİLER (Öğrenci No/AD/SOYAD):

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
Yazılı Çalışma			
Çalışma klavuza uygun olarak hazırlanmış mı?	x	0-5	
Teknik Yönden			
Problemin tanımı yapılmış mı?	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
Yapılan işlerin zorluk derecesi?	x	0-25	
Sözlü Sınav			
Yapılan sunum başarılı mı?	x	0-5	
Soruları yanıtlama yetkinliği?	x	0-20	
Devam Durumu			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
Diğer Maddeler			
Toplam			

DANIŞMAN (JÜRİ ADINA):

DANIŞMAN İMZASI: