**BİTİRME ÖDEVİ**

**STUDENTLINK**

**Gülistan ÖZTÜRK**

**Doç. Dr. M. Emin TENEKECİ**

**BİTİRME ÖDEVİ**

**STUDENTLINK**

**Gülistan ÖZTÜRK**

**Doç. Dr. M. Emin TENEKECİ**

Harran Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü Bitirme Ödevi Yönergesi uyarınca hazırlanmış ve anılan bölüme sunulmuştur.

Şanlıurfa 2023 Gülistan ÖZTÜRK

ONAY

**Doç. Dr. M. Emin TENEKECİ** Doç. Dr. İ. Berkan AYDİLEK

**Danışman Bölüm Başkanı**

Dr. Öğr. Üyesi Kemal GÜNER. Arş. Gör. Songül AKDAĞ

**Komisyon Başkanı Üye**

**İÇİNDEKİLER**

**Sayfa No**

İçindekiler……………………………………………………………………………..i

TEŞEKKÜR………………………………………………………………………….ii

ŞEKİLLER DİZİNİ………………………………………………………………….iii

TABLOLAR DİZİNİ………………………………………………………………...iv

SİMGELER DİZİNİ…………………………………………………………………..v

1. GİRİŞ………………………………………………………………………………1

1.1. Flutter……………........................…………………………………………….…2

1.2. Android Ve Mobil Uygulama Özellikleri.…………………………………….…4

2. BENZER ÇALIŞMALAR LİTERATÜR TARAMASI……..…………………….5

2.1. Harran Üniversitesi Mobil Uygulaması.…………………………………………6

2.2. Haruzem………….…………………..……………………………………..……7

3. ÖNERİLEN YÖNTEM……………………..……………………………………...8

3.1. Hedefler……………………..……………………………………………………8

3.1.1. Mobil Uygulama İle Kolaylık Sağlamak…………..…………………………...8

3.1.2. Zaman Kaybını Önlemek…………………………..…………………………...9

3.1.3. Hızlı İletişim Sağlamak…………………………..………………………….....9

3.1.4. Gereksiz Bildirimlerin Önüne Geçmek.………………………..………………9

3.2. Öğrenci Filtreleme İşlemi…..……………………..…………………………….10

3.2.1 Algoritma Adımları……………………………..………………….………….10

4. KULLANILAN TEKNOLOJİLER….…………………………………...……….10

4.1 Dart……….…………………………..………………………………………….10

4.2. Flutter………………………….………………………………………………. 13

4.2.1 Mobil Uygulama Arayüzü….…..………..…………………………………….16

4.3. Firebase…………………...…………………………………………………….26

5. BULGULAR……………………………………..……………………………….30

5.1. Öğrenci Filtrelme......…………………………………..……………………….30

5.2. Mobil Uygulama …………………………………..………...…………………32

6. SONUÇ VE ÖNERİLER.……………….……………………….……………….33

KAYNAKLAR…………………………………..………………………………….34

ÖZGEÇMİŞ….……………………………………..……………………………….35

Bitirme projesi konusu seçerken isteklerimi göz önünde bulundurup, projenin yürütülmesi sırasında bana yardımcı olan proje danışmanım Doç. **Dr. Öğr. M. Emin TENEKECİ’ye, bölüm başkanımız Doç. Dr. İ. Berkan AYDİLEK’e, danışmanım Kemal GÜNER’e ve eğitim hayatım boyunca bana katkısı olan tüm hocalarıma ve aileme teşekkürlerimi sunuyorum.**

Gülistan ÖZTÜRK

2023

***Şekil-1 – Widget Mimarisi***

***Şekil-2 – Flutter Mimarisi***

***Şekil-3 – Harran Üniversitesi Mobil Uygulaması***

***Şekil-4 – Haruzem Web Sitesi***

***Şekil-5 – Öğrenci Giriş Ekranı***

***Şekil-6 – Öğrenci Kayıt Ekranı***

***Şekil-7 – Mobil Uygulama Ana Sayfası***

***Şekil-8 – Bildirim Menüsü***

***Şekil-9 – Öğrenci Profil***

***Şekil-10 – Çıkış Butonu***

***Şekil-11 – Firebase Arayüzü***

***Şekil-12 – Firebase Kullanılan Platformlar***

***Şekil-13 – Firebase Proje***

***Şekil-14 – Kullanıcı Tablosu***

***Şekil-15 – Kullanıcı Doğrulama Yöntemi***

***Şekil-16– Firebase İmport***

***Şekil-17– Veri Tabanına Bağlanma***

***Şekil-18– Filtreleme Seçenekleri***

***Şekil-19– Verilerin Filtrelenmesi***

***Şekil-20– Filtreleme Kullanımı***

***Tablo-1 – Firebase Analitik***

SDK : Software Development Kit

CLI : Command Line Interface

VM : Virtual Machine

API : Application Programming Interface

FCM : Firebaese Cloud Messaging

1. **GİRİŞ**

Mobil uygulama teknolojisi, günümüz dijital çağında önemli bir evrim geçirmiş ve kullanıcıların günlük yaşamlarını şekillendiren kritik bir rol oynamıştır. Bu evrim, teknolojinin hızla gelişmesi, taşınabilir cihazların yaygınlaşması ve sürekli bağlantı imkanlarının artması gibi faktörlerle tetiklenmiştir. Bu bağlamda, mobil uygulamalar, teknolojik yeniliklerin hemen her kesimde etkili bir şekilde kullanılmasına olanak tanıyan önemli araçlar haline gelmiştir. Özellikle 2000'li yılların başlarına damgasını vuran akıllı telefon devrimi, mobil uygulama geliştirmenin kapılarını son kullanıcıya açmıştır. Apple'ın AppStore'unu 2008 yılında tanıtması ve ardından Android ve diğer platformların benzer uygulama mağazalarını oluşturması, mobil uygulama ekosistemini hızla büyütmüştür. Günümüzde mobil uygulamalar, hemen her sektörde ve kullanıcı profiline hitap eden çeşitli uygulama türleriyle günlük yaşantımızın ayrılmaz bir parçası haline gelmiştir. Eğitimden sağlığa, alışverişten sosyal medyaya kadar birçok alanda mobil uygulamalar, işlevselliği artırarak ve kullanıcı deneyimini optimize ederek önemli katkılarda bulunmaktadır. Ayrıca, mobil uygulama pazarı, geliştiricilere geniş bir kullanıcı kitlesine ulaşma ve yenilikçi fikirlerini hayata geçirme imkanı tanıyan bir platform oluşturarak dijital girişimciliği teşvik etmektedir. Eğitim alanında mobil uygulama gelişimi, öğrencilere daha etkili ve etkileşimli bir deneyimi sunmak amacıyla mobil teknolojilerin kullanılması anlamına gelir. Mobil uygulamalar, öğrencilere esneklik, erişim kolaylığı ve öğrenmeyi eğlenceli hale getirme potansiyeli gibi avantajlar sunar.

Bu projede mobil uygulama üzerinden akademisyenlerin ve öğrencilerin iletişim kurması hedefleniyor. Akademisyenler öğrencilere mail/bildirim gönderme gereği duyduklarında öğrenci filtreleme seçenekleri yetersiz kalmaktadır. Öğrenciler de bu bildirimlerden çoğunlukla geç haberdar olmaktadırlar. Flutter, dart ve firebase kullanarak geliştirilecek olan bu proje ile öğrenciler ve akademisyenler mobil uygulamaya giriş yaparak uygulamanın getirdiği kolaylıklardan faydalanabileceklerdir. Akademisyenlerin öğrencileri filtreleme yöntemiyle seçip iletişim kurabilmelerini sağlayan bu mobil uygulama, eğitim süreçlerini daha etkili ve kişiselleştirilmiş hale getirebilir. Bu uygulama, akademisyenlere öğrenci gruplarını belirli kriterlere göre filtreleme ve sınıflandırma imkanı sunabilir. Akademisyenler, bu filtreleme seçenekleri sayesinde öğrenci popülasyonunu daha iyi anlayabilir ve öğrencilere özel geri bildirim veya duyurular yapabilir. Uygulama, akademisyenlere öğrencilere özel olarak e-posta veya bildirim gönderme özelliği sağlayarak iletişimi daha hızlı ve etkili bir hale getirebilir. Örneğin, belirli bir konuyla ilgili bilgileri paylaşmak veya bir etkinlik hakkında hatırlatmalar yapmak için bu özellik kullanılabilir. Bu, öğrencilerin dikkatini çekerken, akademisyenlerin de öğrencilerle daha yakın bir ilişki kurmasına yardımcı olabilir.

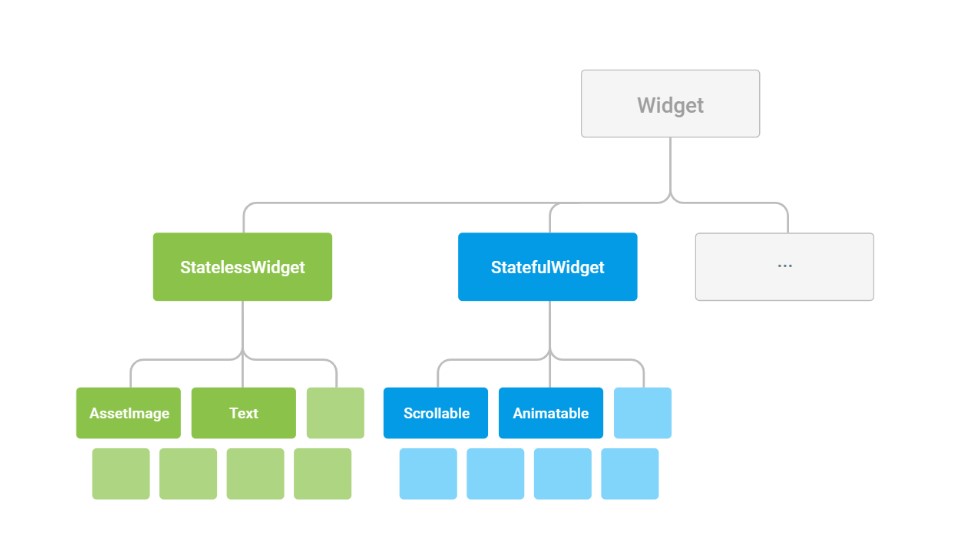
Sonuç olarak, akademisyenlerin öğrencileri filtreleme ve iletişim kurma süreçlerini kolaylaştıran bu tür bir mobil uygulama, eğitimde daha etkili bir öğrenme deneyimi oluşturabilir ve akademik iletişimi güçlendirebilir.

* 1. **Flutter**

Flutter, Google tarafından geliştirilen açık kaynaklı bir kullanıcı arayüzü (UI) yazılım geliştirme framework'üdür. Dart programlama dilini kullanarak, tek bir kod tabanı üzerinden hem iOS hem de Android platformları için native uygulamalar oluşturmanıza olanak tanır. Flutter, özellikle hızlı geliştirme, güçlü performans ve çekici kullanıcı arayüzleri oluşturma yetenekleri ile dikkat çeker. Flutter, desteklediği altı platform genelinde bir uygulama için tutarlı ve cazip kullanıcı arabirimleri oluşturma sürecini basitleştirir. Widget tabanlı yapısı sayesinde, uygulama içindeki her öğeyi bir bileşen olarak düşünerek modüler ve esnek bir tasarım sağlar.

Flutter'ın en büyük avantajlarından biri, "hot reload" özelliği sayesinde kodunuzu değiştirdiğinizde uygulamanızı anında güncelleyebilme imkanı sunmasıdır. Bu, geliştirme sürecini hızlandırır ve anlık geri bildirim almanızı sağlar. Ayrıca, zengin bir widget koleksiyonu ve geniş bir topluluk desteği, geliştiricilere çeşitli görsel ve işlevsel özellikleri uygulamak için geniş bir araç seti sunar. Flutter, hem küçük ölçekli projelerde hızlı bir başlangıç yapma olanağı sunar hem de büyük ölçekli uygulamaların karmaşıklığıyla başa çıkabilen bir yapıya sahiptir. Bu nedenle, mobil uygulama geliştirmek isteyen geliştiriciler için güçlü bir seçenek olarak öne çıkar.

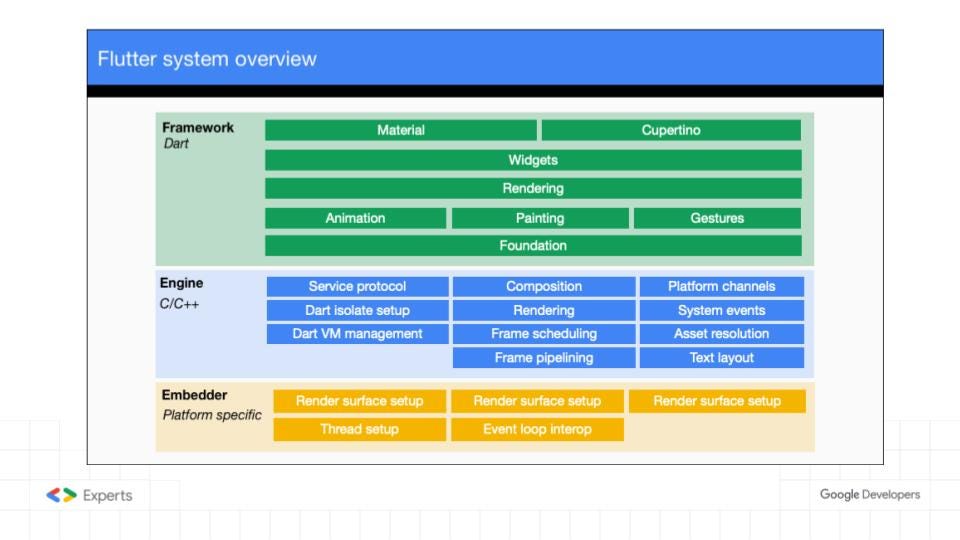
Flutter, kullanıcı arayüzünü oluşturmak için widget'ları temel alır. Her şey bir widget'dır ve widget'lar birleştirilerek kompleks arayüzler oluşturulur. Widget'lar, kullanıcı arayüzünü oluşturan temel yapı taşlarıdır ve zengin bir widget kütüphanesi, geliştiricilere çeşitli arayüz öğelerini özelleştirmek ve birleştirmek için geniş olanaklar sunar. Ayrıca, Flutter'ın açık kaynaklı olması, sürekli olarak güncellenmesini ve geliştirilmesini sağlar, bu da geliştiricilere en son teknolojileri ve özellikleri kullanma imkanı tanır. Aşağıda Şekil-1’de Flutter Widget Mimarisi de verilmiştir.



***Şekil-1 Widget Mimarisi***

Flutter'ın bu widget tabanlı mimarisi, modüler tasarımı ve performans avantajları sayesinde geliştiricilere hızlı, esnek ve etkili bir mobil uygulama geliştirme deneyimi sunar.

Flutter Framework, her katman bir önceki katman üzerine inşa edilecek şekilde bir dizi katman halinde düzenlenir. Şekil-2’deki Flutter mimarisinde verildiği gibi çerçevenin üst katmanları alt katmanlara göre daha sık kullanılır. Bu tasarımın amacı, daha az kodla daha fazlasını yapmaya yardımcı olmaktır.



***Şekil-2 Flutter Mimarisi***

Flutter uygulamaları, derlenmiş native kod gibi çalıştığı için yüksek performans sunar. Ayrıca, GPU hızlandırması ve özel animasyonlar gibi özelliklerle zengin, çekici kullanıcı arayüzleri oluşturabilir.

Flutter, geliştirici araçlarıyla birlikte gelir. Dart SDK, Flutter CLI (Command Line Interface), Visual Studio Code ve Android Studio gibi popüler entegre geliştirme ortamlarıyla uyumludur.

Flutter, açık kaynaklı bir projedir ve sürekli olarak geliştirilmektedir. Bu, geliştiricilerin topluluk tarafından sağlanan güncellemeleri ve özellikleri kullanmalarını sağlar. Flutter topluluğu da oldukça aktif ve destekleyicidir.

Flutter, geniş bir geliştirici kitlesi tarafından benimsenmiş ve birçok başarılı uygulamanın temelini oluşturmuştur. Hızlı prototipleme, platform bağımsızlık ve zengin widget seti gibi özellikleriyle, mobil uygulama geliştirme süreçlerini iyileştirmek isteyen geliştiriciler için güçlü bir tercih haline gelmiştir.

* 1. **Android Ve Mobil Uygulama Özellikleri**

Mobil teknolojinin hızla ilerlemesi ve kullanıcı taleplerinin artmasıyla birlikte, Android işletim sistemine sahip cihazlar için geliştirilen mobil uygulamalar da büyük bir önem kazanmıştır. Android, dünya genelinde milyonlarca kullanıcı tarafından tercih edilen bir işletim sistemidir ve mobil uygulama geliştiricilerine geniş bir platform sunar. Android mobil uygulamaları, çeşitli özellikler ve avantajlar sunarak kullanıcı deneyimini zenginleştirir. Bu özellikler arasında çoklu görev yetenekleri, kişiselleştirilebilir arayüzler, geniş uygulama marketi (Google Play Store), güçlü çoklu medya desteği ve yüksek performanslı donanım entegrasyonu bulunmaktadır. Ayrıca, Android uygulama geliştirme süreci, açık kaynaklı olması sayesinde geniş bir geliştirici topluluğunu destekler ve sürekli olarak güncellenen SDK (Software Development Kit) ile geliştiricilere yenilikçi araçlar sunar. Bu sayede, mobil uygulama geliştiricileri, kullanıcı dostu ve işlevsel uygulamalar tasarlayarak Android ekosisteminde benzersiz deneyimler sunma fırsatına sahiptirler. Bu hızla evrim geçiren alan, mobil uygulama geliştirme dünyasına sürekli yeni özellikler ve gelişmeler ekleyerek kullanıcılarına daha fazla değer sunmaya devam etmektedir.

Android işletim sistemi ve mobil uygulama özellikleri, günümüz mobil cihaz kullanımının temelini oluşturur. Özellikleri:

* Açık Kaynak ve Ücretsiz: Android, açık kaynaklı bir işletim sistemidir ve bu sayede geliştiricilere geniş bir özgürlük sağlar. Ücretsiz olarak kullanılabilir, bu da cihaz üreticileri ve geliştiriciler için maliyet avantajı sağlar.
* Çeşitli Cihazlarda Kullanılabilirlik: Android, birçok farklı cihazda kullanılabilir, bu da kullanıcılara geniş bir cihaz yelpazesi sunar (telefonlar, tabletler, akıllı saatler vb.)
* Kişiselleştirilebilir Arayüzler: Android, kullanıcıların cihazlarını kişiselleştirmelerine olanak tanır. Ana ekran düzeni, tema ve widget'lar gibi birçok özellik kişiselleştirilebilir.
* Geniş Uygulama Marketi (Google Play Store): Google Play Store, milyonlarca uygulamayı içeren geniş bir uygulama marketidir. Kullanıcılar, oyunlardan üretkenlik araçlarına kadar birçok farklı kategoride uygulamalara erişebilirler.
* Gelişmiş Çoklu Görev Yetenekleri: Android, çoklu görev yetenekleri sayesinde kullanıcılara aynı anda birden fazla uygulamayı çalıştırma imkanı tanır. Uygulamalar arasında hızlı geçişler yapmak mümkündür.
* Güçlü Medya Desteği: Yüksek kaliteli ses ve görüntü desteği, Android'in güçlü medya yeteneklerini ortaya koyar. Çeşitli medya formatlarını destekleme kabiliyeti, kullanıcılara geniş bir multimedya deneyimi sunar.
* Gelişmiş Konum Hizmetleri: Android cihazlar, GPS ve diğer konum tabanlı hizmetlere entegre olarak yer tabanlı uygulamaların geliştirilmesini sağlar.
* Yüksek Performanslı Donanım Entegrasyonu: Android, çeşitli donanım bileşenleriyle entegre çalışabilir. Bu, yüksek performanslı oyunlar ve grafik yoğun uygulamalar için ideal bir platform sunar.
* Geliştirici Topluluğu ve Sürekli Güncellemeler: Geniş bir geliştirici topluluğu, Android'in sürekli olarak güncellenmesini ve yeni özelliklerin eklenmesini sağlar. Geliştirici topluluğu, sorunları hızla çözmek ve yenilikçi uygulamalar geliştirmek için bir araya gelir.

Android ve mobil uygulama özellikleri, kullanıcı dostu bir deneyim sağlamak ve geliştiricilere esneklik sunmak amacıyla sürekli olarak geliştirilmektedir.

1. **Benzer Çalışmalar Literatür Taraması**

Akademisyenlerin mobil uygulama üzerinden öğrencilere mesaj gönderdiği, öğrencilerin bu bildirimleri ve duyuruları takip ettiği sistemlere benzer projeler, özellikle eğitim kurumlarında akademisyen ve öğrenciler arasındaki iletişimi daha iyi hale getirmek için yaygın olarak uygulanmaktadır. Bunun gibi projeler, hem akademisyenlerin doğru öğrencilere ulaşmasını sağlamayı hem de öğrencilerin doğru ve hızlı bir şekilde bu duyuru ve bildirimlerden haberdar olmalarını hedeflemektedir.

Asıl amaç hızlı ve doğru haberleşme olduğundan sadece eğitim alanı değil aynı zamanda iş hayatında da buna benzer özel mobil uygulamalar bulunmaktadır. Mail üzerinden haberleşmede kullanıcılar bazen yanıltıcı ve yanlış yönlendirilebiliyor. Kullanıcılar spam veya dolandırıcılık amaçlı maillerle karşılaşılabiliyor. Şirketler ve kurumlar bunun önüne geçmek için çalışanlarına şirket veya kuruma ait özel mobil uygulamalar tasarlayabiliyorlar. Bu tasarladıkları mobil uygulamalara kullanıcılar kendilerine ait id ve şifreler ile giriş yapabiliyorlar. Böylece hem kullanıcıya doğrudan güvenli bir şekilde ulaşılabiliyor hem de kullanıcı gelen bildirimlerin güvenli bir gönderici tarafından geldiğini anlayabiliyor. Ve mesajların bütünlüğü de bozulmuyor. Bu şekildeki uygulamaları daha çok mesajların içeriğinin çok gizli olduğu ve üçüncü taraf kişiler tarafından bilinmemesi gereken bilgileri içeren yazışmalar kullanan şirket ve kurumlar kullanmaktadır. Bu uygulamalar yine sadece şirket veya kurum kullanıcıları ulaşabiliyor.

Bu tezdeki uygulama daha çok eğitim ile alakalı olduğundan daha doğrusu öğrenci ve akademisyen arasında bir iletişimi ele aldığından bu alandaki örnekler incelenmiştir.

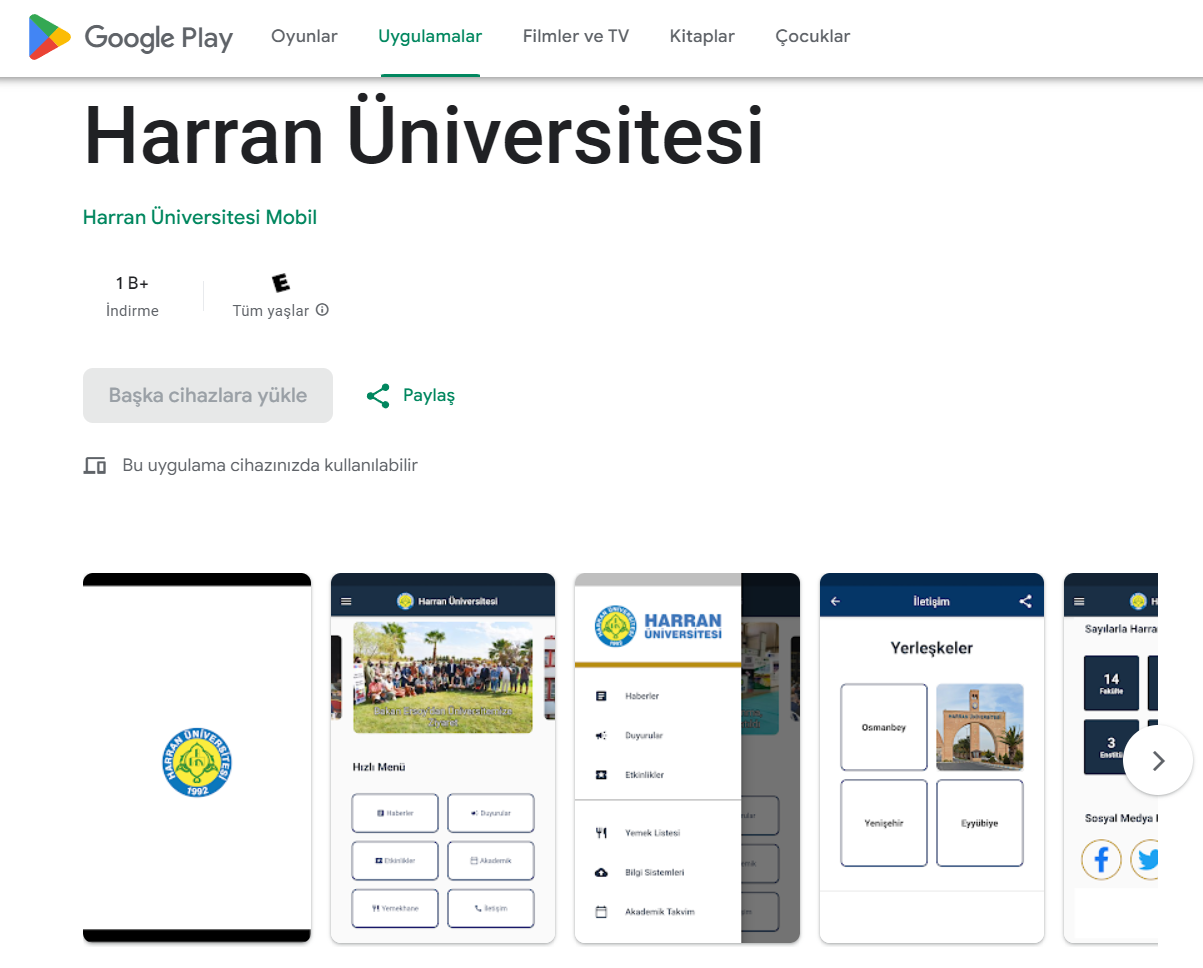
* 1. **Harran Üniversitesi Mobil Uygulaması**

Harran Üniversitesi Mobil Uygulaması öğrencilerin mobil uygulamayı

Google Play Store’dan telefonlarına yüklediği bir mobil uygulamadır. Şekil-3’te verilen Harran Üniversitesi mobil uygulamasında öğrenciler giriş yaptıktan sonra duyuru,haber, etkinlikler, yemekhane ve bilgi sistemlerine ulaşabiliyorlar. Uygulamada ayrıca üniversite ile ilgili iletişim ve sosyal medya hesapları bilgiler bulunmaktadır. Uygulamadaki bu bölümlere tıklandıktan sonra, uygulama web sitelerine yönlendirme yapmaktadır. Yani uygulamadan çıkılıp tarayıcıdan devam edilmektedir. Yemekhane para yükleme işlemlerinde kullanıcılar direkt web tarayıcısına yönlendiriliyor. Yani öğrenciler Harran Üniversitesine ait web sayfasındaki işlemleri uygulamada görebiliyor fakat uygulama kullanıcıları tekrar web sayfasına yönlendirme yapmaktadır.

Uygulamada kullanıcı giriş bilgileri bulunmamaktadır. Yani kullanıcı uygulamayı yükledikten sonra direkt anasayfaya yönlendiriliyor. Herhangi bir kullanıcı doğrulama veya giriş bölümü bulunmamaktadır.

Uygulamada öğrencilerin akademisyen hocalardan direkt bildirim aldığı ve bu bildirim-mailleri anlık olarak görüntüleyebildiği bir bölüm bulunmamaktadır.

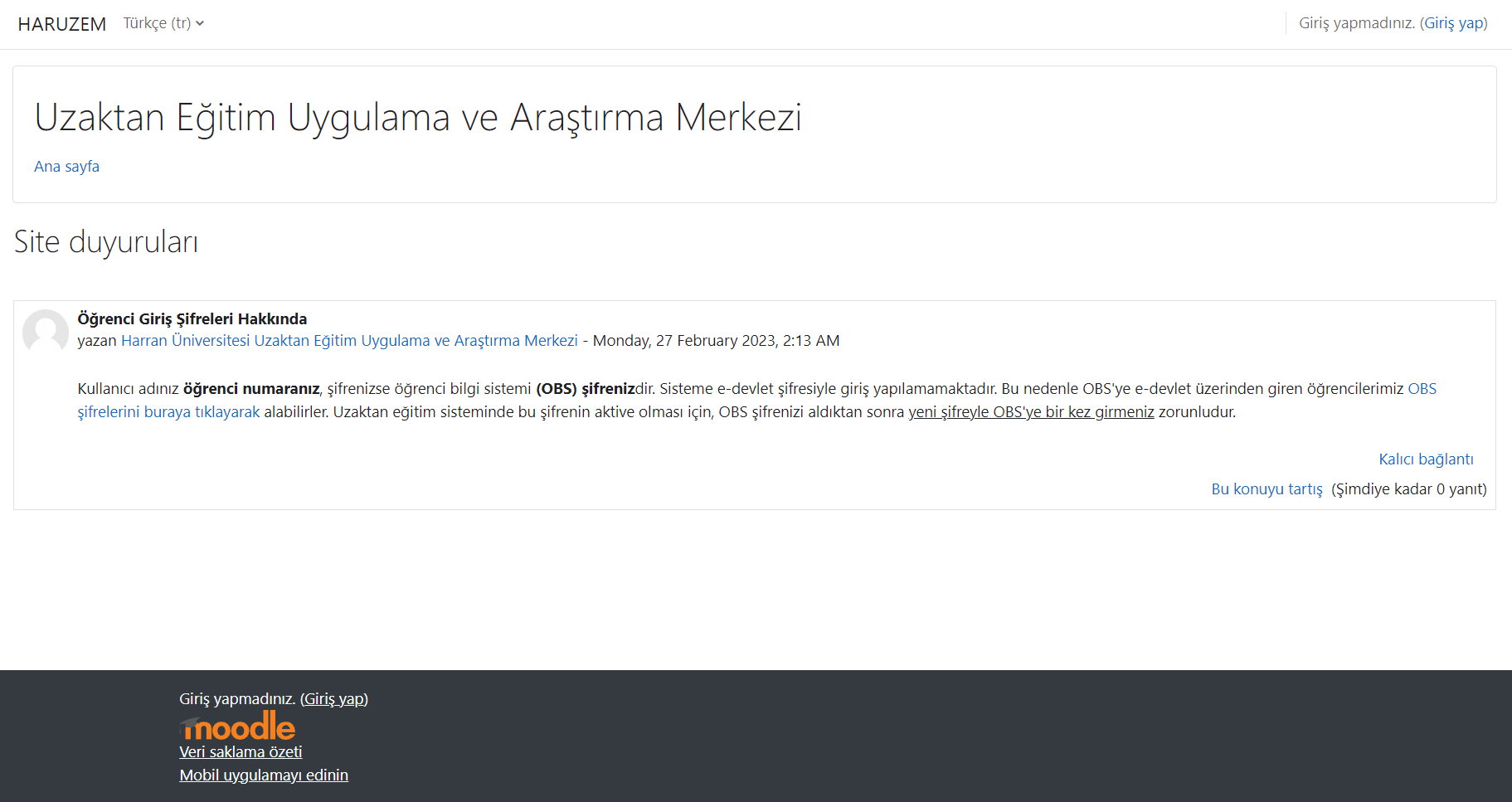


***Şekil-3 Harran Üniversitesi Mobil Uygulaması***

Mobil uygulamada öğrencilerin herhangi bir geri dönüş alabilecekleri bir bölüm bulunmamaktadır. Akademisyenler için de herhangi bir bölüm bulunmamaktadır. Kullanıcılar mobil uygulamaları web sitelerinden daha kolay ve kullanışlı bulmaktadır. Bu uygulama ile de kolaylık sağlanmıştır. Ancak kullanıcılar çoğu bölümde tekrar web sitesine yönlendiriliyor.

* 1. **Haruzem**

Harran Üniversitesi Uzaktan Eğitim Uygulama Ve Araştırma Merkezi (HARUZEM) web sitesi Harran Üniversitesi uzaktan eğitim için kullandığı bir web sitesidir. Şekil-4’te verilen Haruzem Web Sitesine öğrenciler 9 haneli öğrenci numaraları ve öğrenci bilgi sisteminde (obs) kullandıkları şifreleri ile giriş yapabilmektedirler.



***Şekil-4 Haruzem Web Sitesi***

Öğrenciler uygulamaya öğrenci numaraları ve ve obs şifreleri ile giriş yaptıktan sonra ana sayfada aldığı dersler ile ilgili akademisyen hocaların paylaştığı ders içeriklerini veya dokümanları görüntüleyebilecekleri panel bulunmaktadır. Akademisyenler bu sistem üzerinden öğrencilere ödev de gönderebilmektedir. Öğrenciler ise ilgili ödevlerini yine bu sistem üzerinden yükleyip gönderebilmektedir. Özellikle Covid-19 pandemi döneminde akademisyen hocalar bu platform üzerinden ders işlemişlerdir. Ödevlendirme ve ders notu paylaşımı bu sistem üzerinden yapılmıştır. Öğrenciler de ödev gönderme işlemlerini bu sistem aracılığı ile yapmışlardır.

Bu web sitesinde akademisyen hocaların mesaj gönderebileceği ileti menüsü de bulunmaktadır. Fakat bu bölüm anlık olarak bildirim göndermiyor. Ve sistem bir web sitesi olduğundan kullanıcılar bu iletileri ve bildirimleri sisteme giriş yaptıktan sonra görebilmektedirler.

1. **Önerilen Yöntem**

Bu proje sayesinde öğrenciler tek bir uygulama üzerinden hem duyurulara hem de akademisyen hocaların gönderdiği bildirimlere anında ulaşabilecektir. Akademisyen hocalar da filtreleme seçenekleri ile sadece ilgili öğrencilere bildirim gönderebileceklerdir. Böylece öğrenci ve akademisyen hocalar arasında sağlıklı bir iletişim sağlanmış olacak. Öğrenciler de önemli bildirimleri anında aldıkları için herhangi bir gecikme sorunu olmadan mesaj içeriğinden anında haberdar olabilecekler. Ve mail üzerinden değil de mobil uygulama kullanıldığından bu bildirimler mail sistemindeki gibi spama düşme veya gözden kaçma gibi sorunların da önüne geçmiş olacaktır.

* 1. **Hedefler**
     1. **Mobil Uygulama İle Kolaylık Sağlamak**

StudentLink mobil uygulaması, öğrencilerin tek bir mobil uygulama ile duyuru ve bildirimlerden anında haberdar olmasını sağlar. Bu sayede, öğrenciler herhangi bir duyuru ve bildirimi kaçıramayacak böylece akademisyen ve öğrenci arasında bir iletişimsizlik olmayacaktır. Mobil uygulama ile:

* Anında Bildirim Alma: Öğrenciler akademisyen hocaların gönderdiği ilgili mesajları anında bildirim seçeneği ile anlık görüntüleyebilecekler.
* Öğrenci Filtreleme: Akademisyen hocalar gönderecekleri mesajlar sadece ilgili öğrencilerini geniş filtreleme seçenekleri ile seçip gönderebilecekler.
* Kullanıcı Girişi ve Güncellemesi: Öğrenciler uygulamaya öğrenci numaraları ve kendi belirledikleri şifre ile kayıt olduktan sonra giriş yapabilecekler. Böylece öğrenci numaraları ile ilişkili veri tabanından akademisyen hocaların gönderdiği bildirimleri mobil uygulama üzerinden anında alabileceklerdir.
* Öğrenci Numarası İle ilişkili Veri Tabanı: Mobil uygulama ile alınan öğrenci numaraları bir veri tabanına kaydedilecek. Böylece filtreleme sırasında gerekli olacak veri tabanı oluşturulacaktır.

Mobil uygulamaların kullanılırlığı kullanıcı açısından daha kolay ve çok kullanılan bir sistem olduğundan projeyi mobil uygulama şeklinde yapmak hem geniş bir kullanıcı kitlesi oluşturacak hem de bu kullanıcı kitlesine kolaylıklar sağlayacaktır. Web sitelerine nazaran mobil uygulamalar daha çok kullanıldığından proje mobil uygulama ile şeklinde oluşturulmuştur.

* + 1. **Zaman Kaybını Önlemek**

StudentLink uygulaması ile akademisyenlerin ve daha çok da öğrencilerin zaman kaybını önlemek hedeflenmektedir. Akademisyen hocalar önemli mesajlarını ve öğrencilere yapması için hazırladıkları ödevleri mail yolu ile göndermektedirler. Kimi zaman öğrenciler bu maillerden çok geç haberdar olabiliyorlar. Bu da mesaj içerini veya ödevden çok geç haberdar olmalarına yol açabiliyor. Çoğu zaman da ödev süresi bittikten sonra mailleri görebiliyorlar. Bu da öğrenciler için olumsuz bir duruma neden olmaktadır. Mail içeriğinden geç haberdar olan öğrenci kendisine verilen sürenin tamamını kullanamadığından ya ödevini hiç yapamıyor ya da eksik yapmak sorunda kalıyor. Bu durumda akademisyen ve öğrenci arasında iletişim sıkıntısına yol açmaktadır.

StudenLink mobil uygulaması ile bunun gibi zaman kaybının önüne geçilmesi hedeflenmektedir.

* + 1. **Hızlı İletişim Sağlamak**

StudentLink mobil uygulaması ile öğrenci ve akademisyenler arasındaki mesajlar anında bildirim yoluyla uygulamada görüntülenebilecektir. Bu da akademisyenlerin gönderdiği mesaj içeriğinin öğrenciler tarafından görünmemesi durumunu ve bu nedenlerden doğacak olan iletişim sıkıntısını ortadan kaldıracaktır.

* + 1. **Gereksiz Bildirimlerin Önüne Geçmek**

StudentLink mobil uygulaması ile öğrenciler kendileri ile alakalı olmayan mail ve mesajlar almayacaklar. Akademisyenlere sunulan geniş filtreleme seçenekleri ile hocalar dersi alan veya mesajın ilgili kişilerini filtreleme ile seçip öğrencilere gönderebilecektir. Gereksiz maillerin önüne geçilip enerji tüketimi, karbon ayak izini ve elektronik atık oranını da bu şekilde az da olsa azaltmış olacaktır.

Bu nedenlerle, anında bildirim gönderen mobil uygulamalar öğrenci ve akademisyenlerin sorun ve isteklerine göre şekillendirildiğinden hem akademisyenler hem de öğrenciler için kolay, kullanışlı ve hızlı bir çözüm olacaktır.

* 1. **Öğrenci Filtreleme İşlemi**
     1. **Algoritma Adımları**

1. Öğrenciler 9 haneli özel öğrenci numaraları ile mobil uygulamaya kayıt olacaktır.
2. Kaydolan öğrencilerin 9 haneli numaraları veri tabanında kaydedilecektir.
3. Kaydedilen numaralar veri tabanında öğrencilerin giriş yılı, fakülte, bölümü ve bölüm sıralaması şeklinde özelliği çıkartılmış şekilde düzenlenecektir.
4. Akademisyen hocalar mesaj gönderirken bu özelliklere göre öğrencileri filtreleyip mesaj gönderebilecektir.
5. **Kullanılan Teknolojiler**
   1. **Dart**

Dart, ilk kez Google tarafından 2011 yılında tanıtılan çok maksatlı ve nesne yönelimli, smalltalk tarzını benimsemiş bir programlama dilidir. Google’ın JavaScript’in yerini almasını düşünerek geliştirdiği Dart, sonraları Ecma International’ın çabalarıyla bir standart haline getirilmiştir. Şimdilerde Google’ın Flutter Framework’ü ile ön plana çıkmaya başlayan Dart’ı kullanarak; web sunucuları, web uygulamaları, native mobil uygulamalar ve IoT geliştirmeleri yapılabilmektedir.

Dart dili sınıf-temelli, tekil-kalıtımlı C-tarzında bir kod dizilimine sahiptir ve JavaScript diline veya çalıştığı sistemdeki özgün (native) dile çevrilebilir. Interface'leri,single inheritance'ı, mixin'leri, soyut sınıfları, generic'leri ve opsiyonel tipleri (dynamic) destekler. Dart ilk kez Danimarka'nın Aarhus şehrinde, 2011 yılının Ekim ayının 10-12. günlerinde düzenlenen GOTO etkinliği kapsamında tanıtıldı. Dart projesi Lars Bak ve Kasper Lund tarafından kuruldu.

Ecma International, Dart'ın standartlaştırılmasını sağlamak ve JavaScript olarak derlenebildiği için modern tarayıcılarda üzerinde efektif bir şekilde çalışmasının gerçekleştirilmesi adına TC52 adında teknik komisyonu kurdu. Dart dilinin ilk spesifikasyonu, 2014 yılının Temmuz ayında Ecma International'ın 107. Genel Kongre'sinde kabul edildi. O zamandan beri 2. beyannamesi de kabul görmüş durumdadır.

Dart kodu, 3 farklı şekilde kullanılabilir:

JavaScript Olarak Derleme: Dart kodunun yaygın internet tarayıcılarında çalışabilmesi için JavaScript'e koddan-koda derlenmesi gerekmektedir. Resmi sitesinde yer alan bilgiye göre Dart, "Modern uygulama geliştirimi ve yüksek performanslı programlar yazmak adına kolayca geliştirim araçları yazmak için tasarlanmıştır. Dart kodu web tarayıcısında çalışacağı zaman, dart2js derleyicisi ile Javascript'e derleme yapılır. JavaScript olarak derlenmiş Dart kodu, bütün modern tarayıcılarda hiçbir değişiklik gerektirmeksizin uyumlu hale gelir. Derlenmiş Javascript kodunun gereksiz kontrol ifadelerinden ve işlemlerinden arındırılmış bir şekilde optimize olması sebebiyle Dart ile yazılan kod bazı durumlarda eşdeğer olarak elle yazılan Javascript kodundan daha performanslı olabilir.

Dartium Tarayıcısı'nda : Dart SDK'sı, Dart VM (sanal makine) içeren Chromium web tarayıcısı ile birlikte gelir. Bu tarayıcıyı diğer tarayıcılardan ayıran fark ise Dart kodunu JavaScript'e derlemeksizin çalıştırabilir olmasıdır. Bu tarayıcı, genel amaçlı değildir ve Dart uygulamaları için bir geliştirim aracı olarak kullanılır. Önceden de Chrome içerisine tümleşik olarak bu tarayıcının getirilmesi amaçlandı fakat daha sonra iptal edildi.

Platform bağımsızdır : Dart SDK'sı ayrıca dart kodunun komut satırında çalıştıırlabilmesini sağlayan ve platformdan bağımsız olan Dart VM ile gelir. Dart SDK'sıne dahil olan Dart VM ve dart ile yazılan dil araçları, SDK'nin kritik bir bölümünü oluşturur. Bu dil araçları yalnızca dart2js derleyicisini içermekle kalmaz, ayrıca pub adında paket yöneticisini de içerir. Dart, özelleştirilmiş web sunucuları gibi tam fonksiyonel sistem uygulamalarının yazılması için standart kütüphanesi ile birlikte gelir.

Dart uygulamaları iki türden birinde çalışır. Ön tanımlı olmayan ve aktif hale getirilerek kullanılan "kontrollü mod"'da dinamik olarak tür kontrolleri yapılır. Kontrollü modda eğer statik türler varsa ve tür uyuşmazlığına dair bir durum olduğunda bu hataların yakalanması sağlanır. Örneğin, eğer bir metot String dönecek şekilde yazılmışsa ve tam sayı döndürmeye çalışıyorsa dinamik tür belirtimi bu hatayı yakalar ve exception olarak fırlatır. Kontrollü modda çalışma, uygulama geliştirimi ve test aşaması için tavsiye edilen yöntemdir. Dart programları, dinamik tür kontrollerinin varsayılan olarak kapalı olduğu "üretim modu"'nda gelir. Bu modun varsayılan olmasının sebebi Dart programının en hızlı şekilde çalıştırılabilmesi içindir.

Eş zamanlı çalışmak için Dart, birbirinden bağımsız ve bellek paylaşımı olmayan, bunun yerine mesajlarla birbirleriyle haberleşen işçi bileşenler olan isolate'leri kullanır. Bu durum, Erlang dilindeki aktörler ile benzerlik gösterir. Her dart programı en azından bir isolate'e ihtiyaç duyar ki bu main isolate'dir. Dart kodu JavaScript olarak derlendiğinde, isolate'ler Web Worker'lara dönüştürülür.

Snapshot’lar: Snapshot'lar Dart VM'in çekirdeğini oluşturur. Aslında her bir snapshot, nesneleri ve diğer çalışma zamanı verilerini tutan tipik bir dosyadır.

1. Betik snapshot'ları: Dart programları snapshot dosyaları olarak derlenebilirler. Bu dosyalar program kodunun tümünü içerir ve bağımlı olduğu bileşenler de önceden ayrıştırılmış halde gelir ve çalışmak için hazırdırlar. Bu sayede programların başlaması oldukça hızlıdır.
2. Tam snapshot'lar:Dart'ın çekirdek kütüphaneleri, programların çok hızlı açılması için snapshot dosyaları olarak derlenebilir. Dart VM'in birçok standart dağıtımında, çalışma zamanında yüklenen ana kütüphaneler önceden derlenmiş snopshot'lar halinde bulunur.
3. Nesne snapshot'ları :Dart oldukça asenkron bir dildir. Bunun için eş zamanlı çalışmayı sağlayan isolate'leri kullanır. Bu işçilerin mesajları birbirine aktarabilmeleri için, mesajı öncelikle uygun hale getirmeleri (serialization) gerekmektedir. Bu işlem, gönderilecek nesnenin verilmesiyle oluşturulan ve daha sonra deserialization yapılması için diğer isolate'e gönderilen snapshot kullanarak gerçekleştirilir.

Dart, eş zamanlı çalışma için farklı modeller sunan Fletch adındaki sanal makineye sahiptir. Bununla birlikte herhangi bir uygulamada Dart kodunun gömülebilmesi için basit bir API sunar. Google, Dart'ın tüm kısımlarının Android ve iOS üzerinde native mobil uygulama geliştirimi yapılabilmesi için Flutter üzerinde uğraşmaktadır.

dart2js mevcut Dart kodundan JavaScript koduna derlemeyi sağlayan 2015 yılında Google'ın ürettiği ve Dart dili ile yazılan bir derleyicidir. dart2js, Dart dilinin tüm spesifikasyonlarını ve semantiğini içermek üzere geliştirilmiştir. Önceki derleyicilerin evrimi niteliğindedir: dartc, Dart kodundan Javascript koduna derlemeyi amaçlayan ilk derleyicidir fakat artık kullanılmamaktadır. Frog ise bu alandaki ikinci Dart'tan JavaScript'e çevirim yapan derleyicidir ve Dart ile yazılmıştır. Frog hiçbir zaman dilin tüm spesifikasyonlarını uygulayamamıştır ve dart2js'in çıkışına ön ayak olmuştur. 2013 yılının 28 Mart'ında Dart ekibi, blog'unda Dart kodunun JavaScript'e dönüştürülmesini sağlayan dart2js derleyicisini paylaşmışlardır ve Chrome'un V8 JavaScript motorunda Deltablue benchmark testlerinin yürütülmesi sonucu elle yazılan JavaScript kodundan daha hızlı olduğunu dile getirmişlerdir.

Dart Dilinin Avantajları:

* Performans: Dart, hızlı ve etkili çalışma yetenekleriyle bilinir. Bu özellik, özellikle mobil ve web uygulamaları gibi performansın önemli olduğu alanlarda tercih edilmesini sağlar.
* Nesne Yönelimli Programlama (OOP): Dart, nesne yönelimli programlamaya odaklanmıştır, bu da kodun düzenli ve modüler olmasını sağlar. Bu, büyük ölçekli projelerde daha iyi bakım ve geliştirme sağlar.
* Asenkron Programlama: Dart, asenkron programlamayı destekler. Bu, uygulamaların daha etkili bir şekilde çalışmasına ve kullanıcı deneyimini artırmasına yardımcı olur.
* Flutter ile Uyumlu: Dart, özellikle Flutter framework'ü ile birlikte kullanıldığında popülerdir. Flutter, özellikle çok platformlu uygulamalar geliştirmek için kullanılan bir UI toolkit'ini içerir.
* Açık Kaynak ve Topluluk Desteği: Dart, açık kaynak bir projedir ve geniş bir geliştirici topluluğuna sahiptir. Bu, dilin sürekli olarak geliştirilmesini sağlar.
* Güçlü Araç Desteği: Dart, zengin bir set araç ve kütüphane desteğine sahiptir. Geliştiricilere uygulamalarını hızlı bir şekilde oluşturma ve yönetme konusunda yardımcı olur.

Dart Dilinin Dezavantajları:

* Ekosistem: Dart'ın popülerliği artsa da, diğer dillerle karşılaştırıldığında hala daha küçük bir ekosistema sahiptir. Bu, dilin kullanılabilirliğini etkileyebilir.
* Adoption (Benimseme) Süreci: Dart'ın benimseme süreci, özellikle mevcut projelerdeki diğer dillerle uyumlu hale getirilmesi gerektiğinde bazen zorlu olabilir.
* Dilin Yaşı: Dart, diğer bazı dillere göre daha genç bir dil olarak kabul edilebilir. Bu, dilin uzun vadeli dayanıklılığı ve benimsenme oranı konusunda belirsizlik yaratabilir.
* Platform Bağımlılığı: Dart, özellikle Flutter dışında daha geniş bir platform bağımlılığına sahip olabilir. Bu, dilin kullanılabilirliğini sınırlayabilir.
* Dökümantasyon Zayıflıkları: Bazı geliştiriciler, Dart'ın dökümantasyonunun eksik olduğunu veya yetersiz olduğunu iddia edebilir. Bu, yeni başlayanlar veya dilde daha fazla deneyime sahip olmayan geliştiriciler için bir zorluk olabilir.
  1. **Flutter**

Flutter, Google tarafından oluşturulan açık kaynaklı bir UI yazılım geliştirme kitidir. Android, iOS, Windows, Mac, Linux ve web için uygulamalar geliştirmek için kullanılıyor. Flutter'ın ilk sürümü "Sky" olarak biliniyordu ve Android işletim sisteminde çalışıyordu. Flutter ilk olarak 2015 Dart geliştirici zirvesinde tanıtıldı. Tanıtımında sabit bir şekilde saniyede 120 FPS çalışan uygulamalar geliştirilebileceği belirtildi.4 Aralık 2018'de Flutter 1.0, Flutter Live etkinliğinde ilk "kararlı sürüm" olarak yayımlandı. 11 Aralık 2019'da Flutter Interactive etkinliğinde Flutter 1.12 yayımlandı.6 Mayıs 2020'de, 2.8 sürümündeki Dart SDK ve 1.17.0 sürümündeki Flutter, Metal API'a desteğin eklendiği ve iOS cihazlarındaki (yaklaşık %50) ana widget'larındaki performansı büyük ölçüde artırdı. Ağ etkinlikleri izleme aracı ve çok daha fazlası eklendi.

Flutter'ın ana bileşenleri şunlardır:

* Dart platformu
* Flutter motoru
* Temel kütüphane
* Tasarıma özel widget'lar

Flutter'ın en önemli özelliği, özelleştirilebilir ve etkileyici kullanıcı arayüzleri oluşturmak için kendi çizim motorunu kullanmasıdır. Bu çizim motoru, donanım hızlandırmasını kullanarak kullanıcı arayüzlerinin hızlı bir şekilde oluşturulmasını ve animasyonlarla zenginleştirilmesini sağlar.

Ayrıca, Flutter'ın sahip olduğu zengin widget koleksiyonuyla kullanıcı arayüzü bileşenlerini kolayca oluşturabilir, düzenleyebilir ve özelleştirebiliriz. Flutter, her şeyin bir widget olduğu widget tabanlı bir yapıya dayanır. Bir "widget", kullanıcı arayüzü öğelerini temsil eden birinci sınıf bir nesnedir. Örneğin, bir düğme, bir metin kutusu veya bir sayfa birer widget'tir. Flutter'ın temel felsefesi, küçük ve tekrar kullanılabilir widget'ları bir araya getirerek karmaşık kullanıcı arayüzlerini oluşturmayı kolaylaştırmaktır.

Flutter, sıcak yeniden yükleme (hot reload) özelliği sayesinde hızlı iterasyonlarla çalışmaya olanak tanır. Bu özellik, kodu değiştirdiğimizde uygulamanın anında güncellenmesini sağlar, böylece hızlıca değişiklikler yapabilir ve sonuçlarını hemen görebiliriz. Bu, geliştirme sürecini hızlandırır ve zaman kazandırır.

Flutter, tek bir kod tabanıyla birden çok platformda çalışabilme yeteneğine sahiptir. Bu, aynı kodu kullanarak hem iOS hem de Android için uygulama geliştirmeyi sağlar. Ayrıca, Flutter'ın sunduğu zengin ve hızlı widget'lar, kullanıcı deneyimini iyileştirmek için çeşitli animasyonlar, efektler ve geçişler eklemeyi sağlar.

Flutter, büyük bir topluluğa ve geniş bir ekosistema sahiptir. Bu, sorunlarınızı çözmek, öğrenmek ve geliştirmek için geniş bir kaynak havuzuna erişim sağlar.

Flutter, hızlı geliştirme, platformlar arası uyumluluk ve güzel kullanıcı arayüzleri oluşturma yetenekleri ile dikkat çeken bir geliştirme aracıdır. Dart programlama dilini kullanarak, widget tabanlı yapısıyla ve çeşitli özellikleriyle, geliştiricilere esnek ve güçlü bir çerçeve sunar.

Flutter, geniş bir kütüphane ekosistemine sahiptir ve bu kütüphaneler, çeşitli ihtiyaçlara yönelik önceden oluşturulmuş çözümler sunar. İşte Flutter'da kullanılan bazı önemli kütüphanelerden bazıları:

1. Material Design ve Cupertino Widgets:

* material.dart: Google'ın Material Design tasarım diline uygun widget'ları içerir.
* cupertino.dart: Apple'ın iOS tasarım diline uygun widget'ları içerir.

1. HTTP İsteği İçin Kütüphaneler:

* http.dart: HTTP istekleri yapmak ve almak için kullanılan kütüphane.
* dio.dart: Yüksek performanslı, güçlü bir HTTP istek kütüphanesi.

1. State Yönetimi İçin Kütüphaneler:

* provider.dart: Widget ağacında durum yönetimi sağlayan basit ve etkili bir kütüphane.
* riverpod.dart: Provider'ın gelişmiş versiyonu, Dart'ın Sound Null Safety özelliğini destekler.

1. Navigation İçin Kütüphaneler:

* flutter\_bloc.dart: BloC (Business Logic Component) mimarisi ile birleştirilmiş Flutter için durum yönetimi.
* get.dart: Hızlı ve basit sayfa yönetimi için kullanılan kütüphane

1. JSON Serileştirme İçin Kütüphaneler:

* json\_serializable.dart: Dart sınıflarını JSON'a dönüştürmek ve JSON'dan Dart sınıflarına dönüştürmek için kullanılır.
* freezed.dart: Immutable (değiştirilemez) sınıfları ve union türlerini oluşturmak için kullanılır.

1. Veritabanı İşlemleri İçin Kütüphaneler:

* moor.dart: SQLite veritabanı ile etkileşimde bulunmak için kullanılan kütüphane.
* hive.dart: Hızlı, hafif ve açık kaynaklı bir NoSQL veritabanı.

1. Grafik ve Animasyon İçin Kütüphaneler:

* fl\_chart.dart: Çeşitli grafik türlerini oluşturmak için kullanılan kütüphane.
* animations.dart: Animasyonlar oluşturmak ve yönetmek için kullanılır.

1. UI Tasarım İçin Kütüphaneler:

* styled\_widget.dart: Widget'ları hızlı ve okunabilir bir şekilde özelleştirmek için kullanılır.
* font\_awesome\_flutter.dart: FontAwesome ikonlarını kullanmak için.

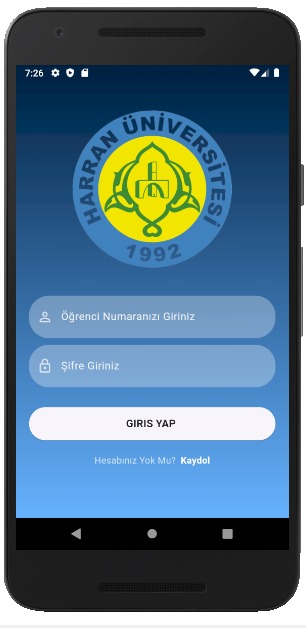
1. Test İçin Kütüphaneler:

* flutter\_test.dart: Flutter uygulamalarını test etmek için gerekli olan kütüphane.
* mockito.dart: Dart dilinde mock nesneleri oluşturmak ve test etmek için kullanılır.

1. Firebase ve Diğer Backend Entegrasyonları İçin Kütüphaneler:

* firebase\_core.dart: Firebase'i Flutter uygulamalarına entegre etmek için.
* cloud\_firestore.dart: Firebase Firestore veritabanı ile etkileşimde bulunmak için.
  + 1. **Mobil Uygulama Arayüzü**
       1. **Giriş Sayfası**

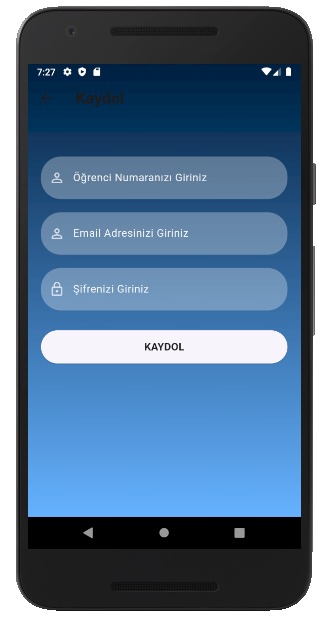
Öğrencilerin uygulamayı açtıklarında karşılaştıkları ilk ekran Şekil-5’teki gibidir. Öğrenci numaraları ve şifresi kullanıcıdan istenir. Burada kayıtlı kullanıcıların doğru şekilde giriş yapması beklenir. Yanlış şifre veya öğrenci numarası girilmesi sonucunda kullanıcılar uyarı alacaktır.

****

***Şekil-5: Öğrenci Giriş Ekranı***

* + - 1. **Kayıt Sayfası**

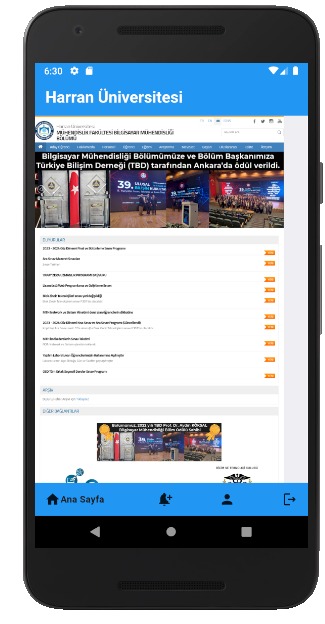
Öğrenciler eğer ilk defa uygulamaya giriş yapıyorlar ise kayıt yapmaları gerekir. Çeşitli bilgiler alınır bu bilgiler; Şekil-6 da öğrenci numarası, email adresi ve şifredir. Alınan bu bilgiler Firebase’e kaydedilir. Başarılı şekilde kayıt yapanlar giriş sayfasına yönlendirilir.

****

***Şekil-6: Öğrenci Kayıt Ekranı***

* + - 1. **Ana Sayfa**

Öğrencilerin uygulamaya başarılı bir şekilde giriş yaptıktan sonra Şekil-7’de karşılaştıkları ana sayfa aşağıdaki gibidir.



***Şekil-7: Mobil Uygulama Ana Menüsü***

* + - 1. **Bildirim**

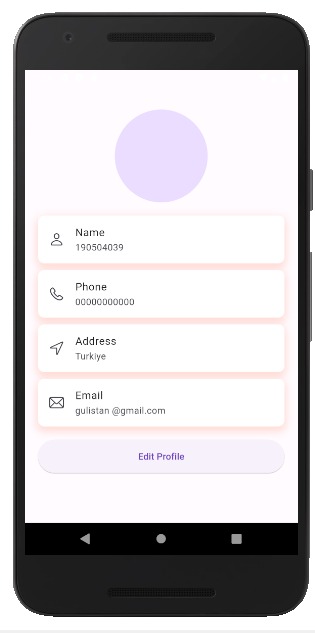
Öğrencilerin akademisyenlerden aldığı bildirimleri görüntüleyebildikleri Şekil-8’deki ekran gibidir. Burada gelen bildirimler anlık olarak kullanıcıya bildirilir.

****

***Şekil-8: Bildirim Menüsü***

* + - 1. **Profil**

Öğrencilerin profil bilgileri Şekil-9’daki gibidir. Burada kullanıcılar profillerini düzenledikleri takdirde bilgiler Firebase’e kaydedilir.

****

***Şekil-9: Öğrenci Profil***

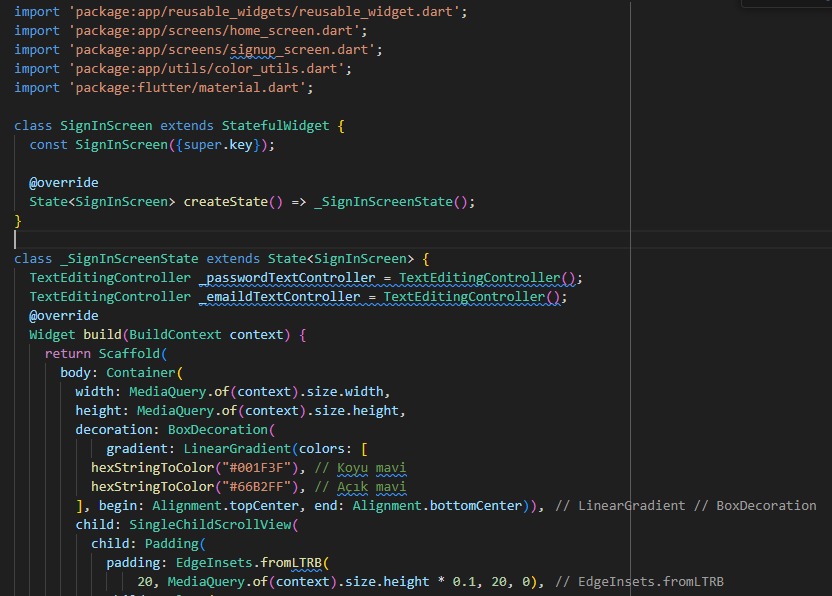
* + - 1. **Çıkış**

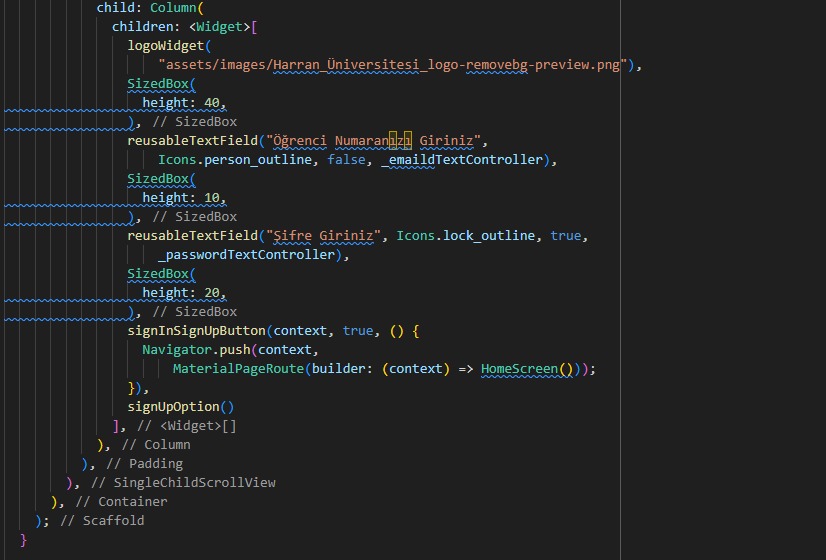
Kullanıcılar mobil uygulama sayfasında Şekil-10’daki gibi altta bulunan çıkış simgesine tıkladıklarında uygulamadan çıkış yapabilmektedirler. Çıkış yaptıkları zaman yine giriş sayfasına yönlendirilirler.

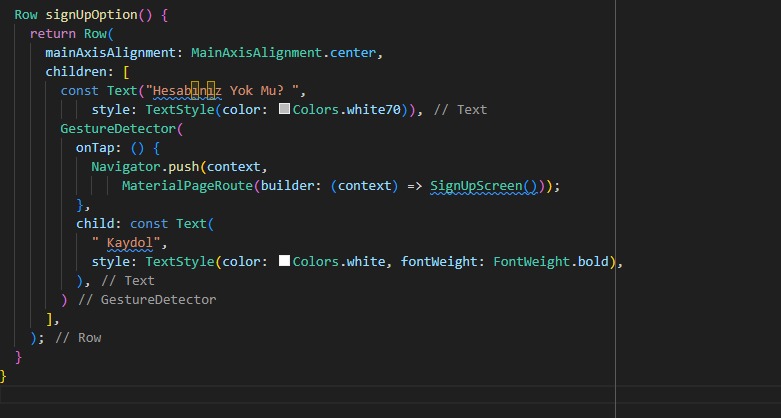


***Şekil-10: Çıkış Butonu***

* + 1. **Dart Kodlarından Örnekler**
       1. **Giriş Ekranı**

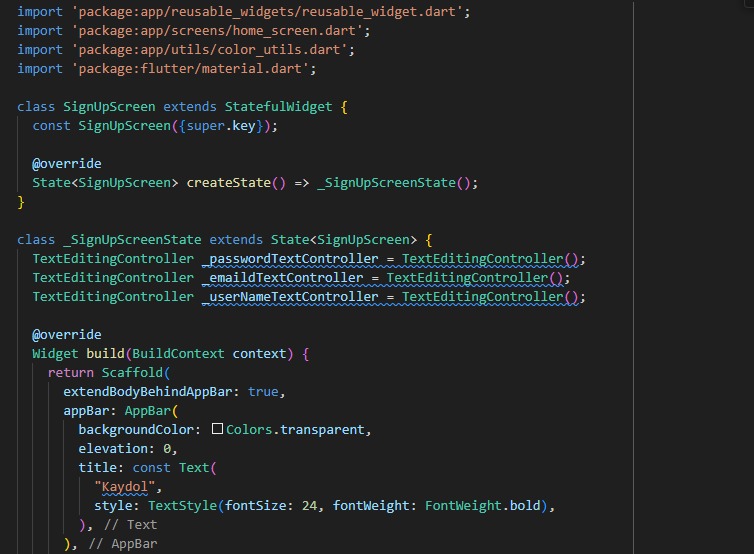


****

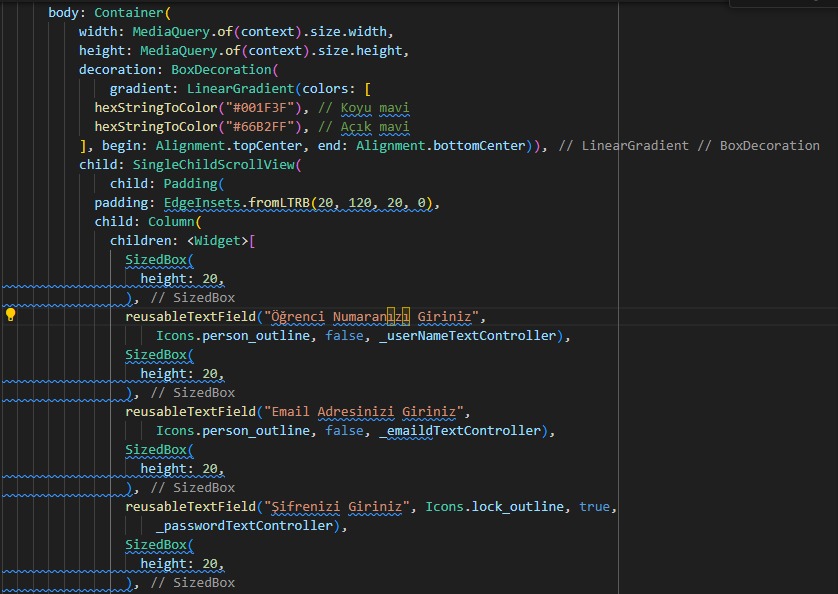
**(Giriş Ekranı Kodları)**

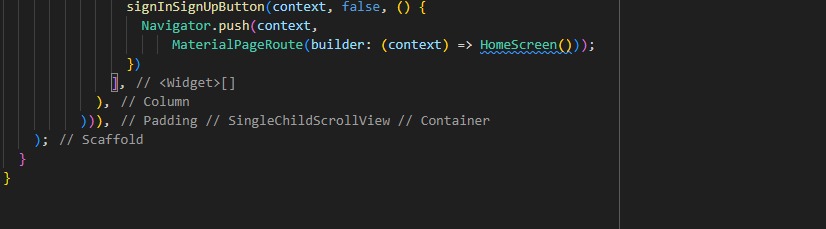
**(Giriş Ekranı Kodları)**

* + - 1. **Kayıt Ekranı**



**(Kayıt Ekranı Kodları)**



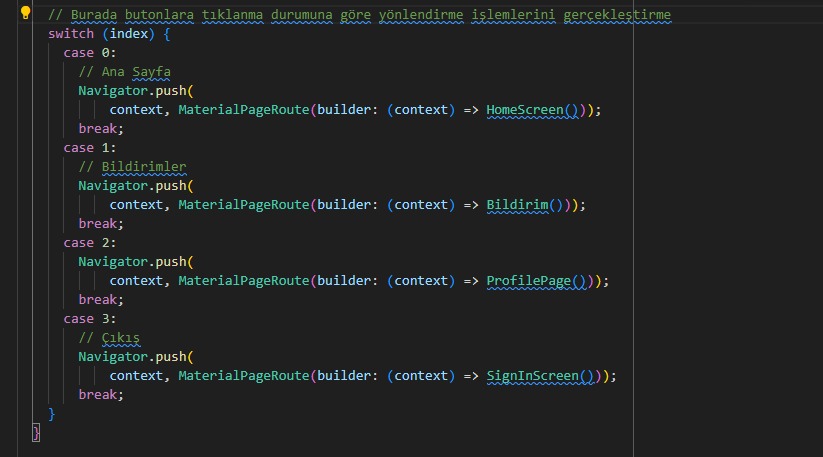


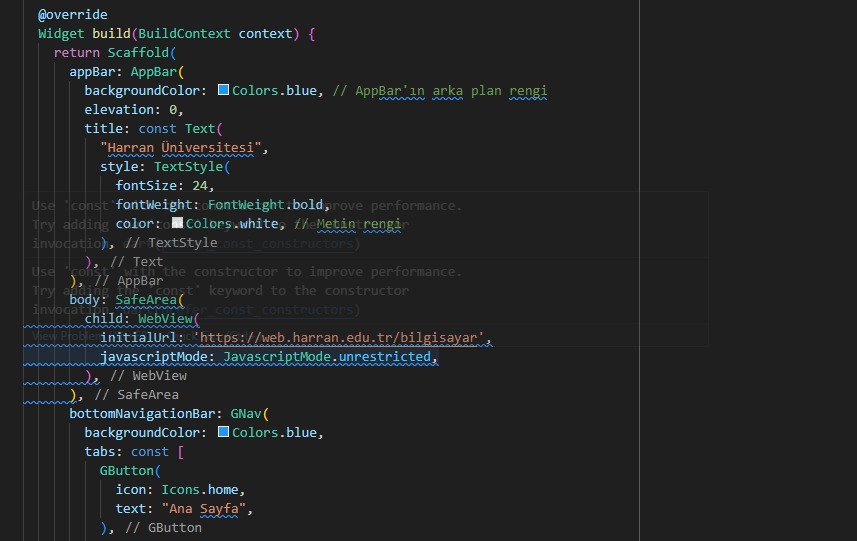
**(Kayıt Ekranı Kodları)**

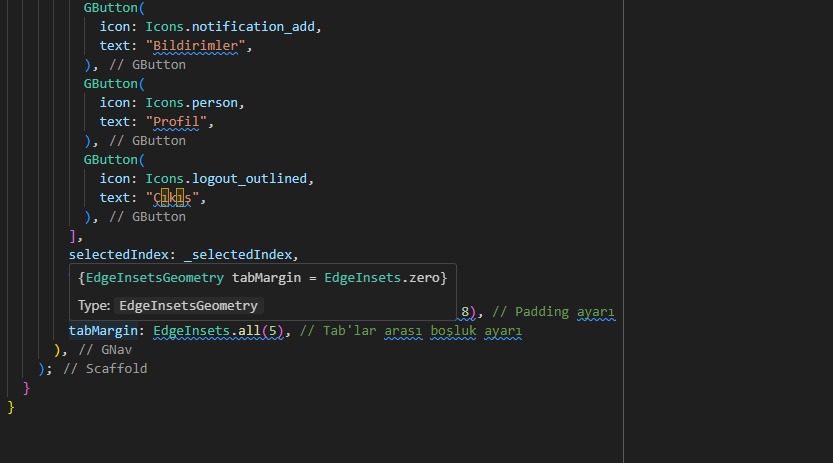
* + - 1. **Ana Menü**



**(Ana Menü Ekranı Kodları)**

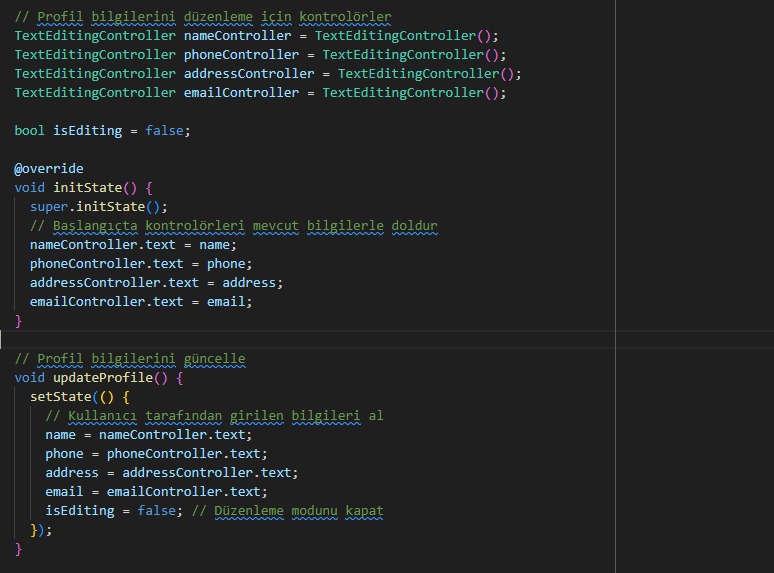
****

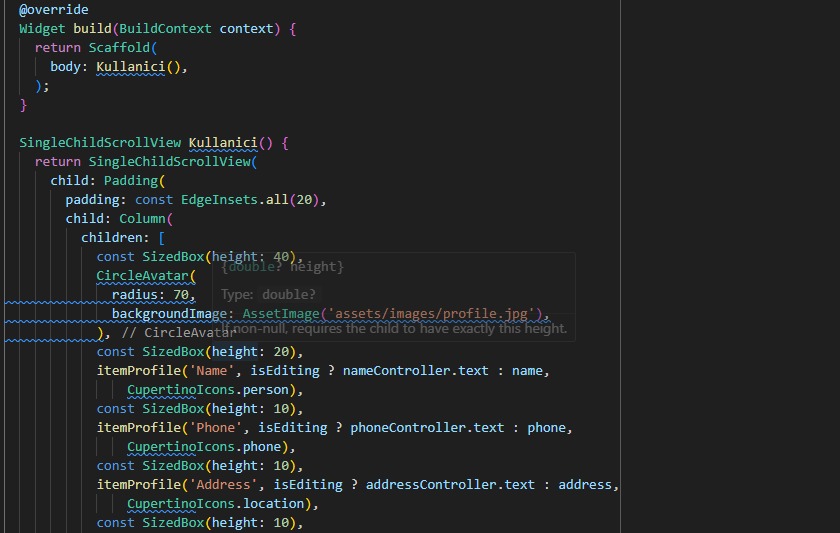


****

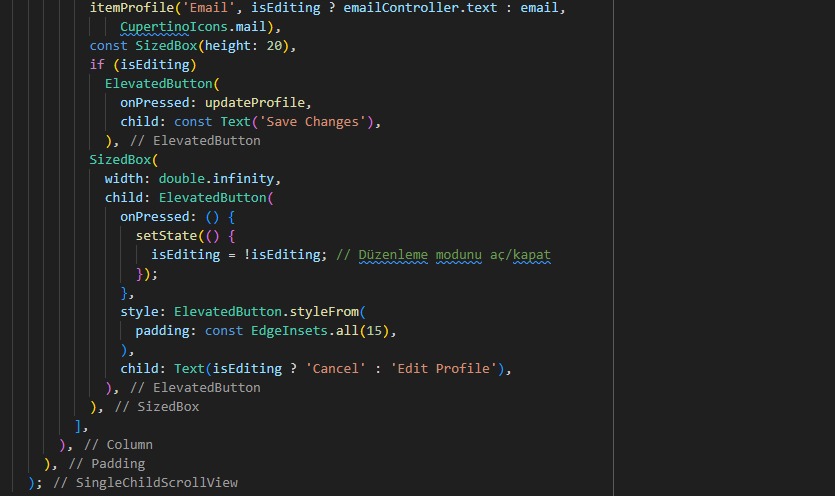
**(Ana Menü Ekranı Kodları)**

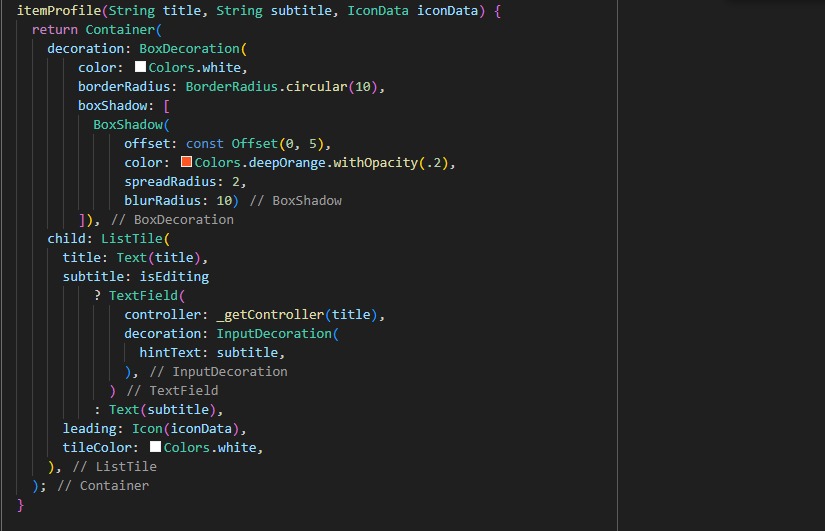
* + - 1. **Profil**





**(Kullanıcı Profili Ekranı Kodları)**

****

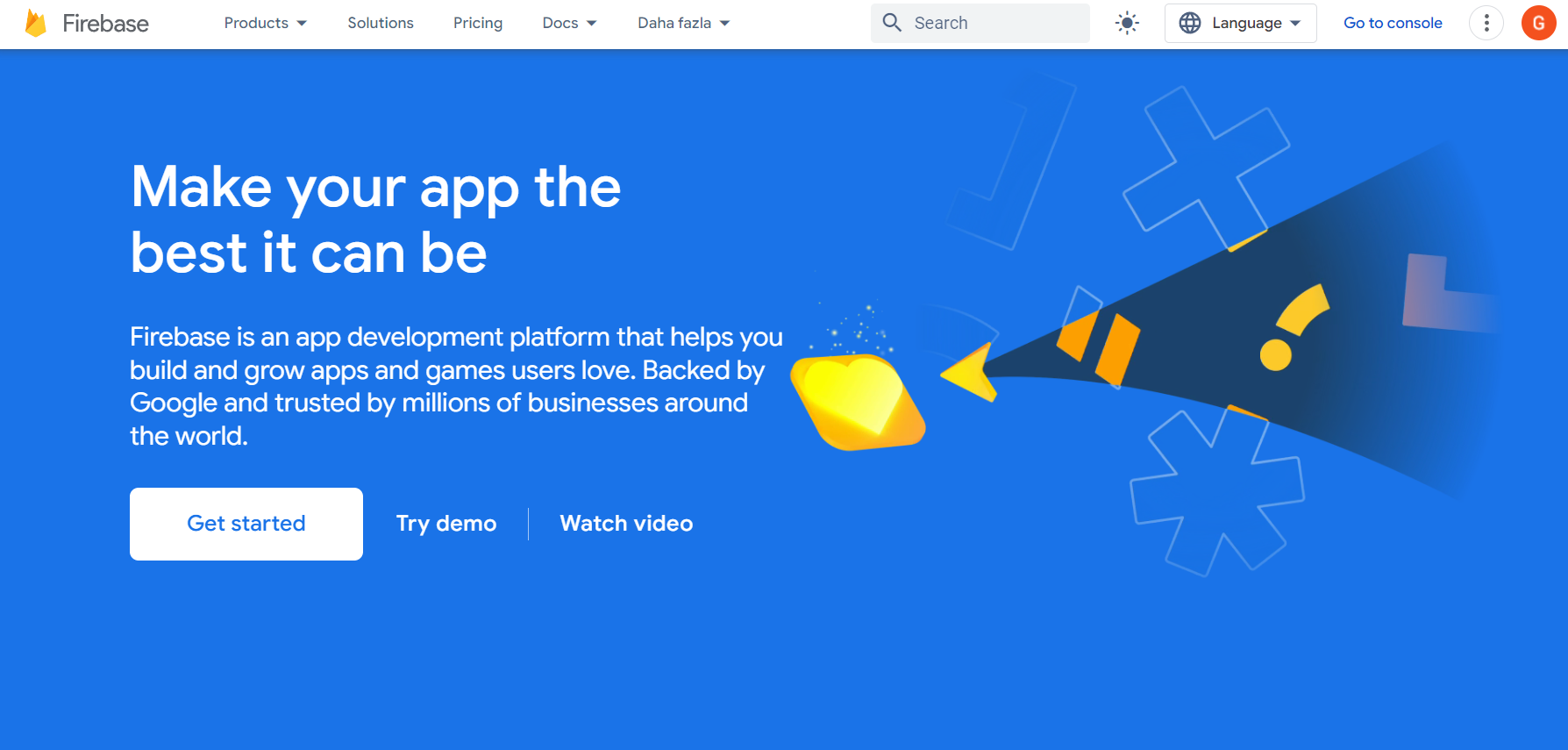
****

****

**(Kullanıcı Profili Ekranı Kodları)**

* 1. **Firebase**

Firebase, Google tarafından geliştirilen ve sunduğu bir bulut tabanlı mobil ve web uygulama geliştirme platformudur. Firebase, geliştiricilere uygulamalarını daha hızlı ve daha verimli bir şekilde oluşturmaları, dağıtmaları ve ölçeklendirmeleri için bir dizi araç ve hizmet sunar. İşlevselliği geniş bir yelpazede kapsar, özellikle kullanıcı kimlik doğrulama, gerçek zamanlı veritaban, depolama, sunucu tarafı işlevselliği, test ve analitik gibi alanlarda hizmet verir. Firebase servisinin geçmişi alsında James Tamplin ve Andrew Lee tarafından 2011 yılında kurulan Envolve şirketine dayanmakta. Geliştiricilerin web sayfalarına API aracılığıyla online chat özelliği kazandırabilmeleri amacıyla kurulan Envolve geliştiriciler tarafından planlanandan farklı bir şekilde (uygulama datalarını gerçek zamanlı olarak kullanıcılarla senkronize bir şekilde paylaşılması gibi) kullanılır. Kurucular bu sebeble 2012 yılında bildiğimiz Firebase’in de temellerinin atıldığı servis ayrımı kararını verirler. 2012 yılından itibaren pek çok gelişmeyi barındıran, mobil ve web uygulama geliştirme platformu olarak konumlandırılan Firebase 2014 yılında Google tarafından satın alınır.



***Şekil-11: Firebase Arayüzü***

**Tarihçe:**

2011: Firebase, başlangıçta 2011 yılında kurulan bir şirket tarafından oluşturuldu. İlk başta, mobil ve web uygulama geliştiricilerine gerçek zamanlı senkronizasyon sağlamak amacıyla kullanılan bir araç seti olarak başladı.

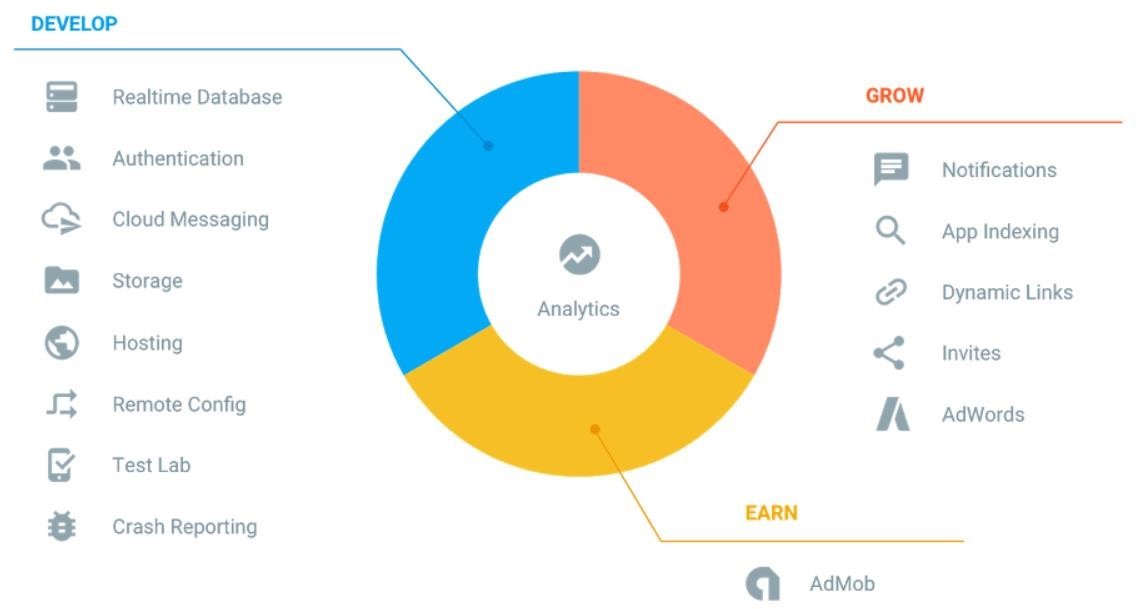
2014: Google, Firebase'i satın aldı ve platformu daha da genişletmeye başladı. Firebase, bir dizi yeni özellik ve hizmetle güçlendirildi.

2016: Firebase, Google I/O etkinliğinde büyük bir güncelleme aldı. Bu güncelleme, geliştiricilere sunulan araçları ve hizmetleri genişleterek, platformun daha kapsamlı bir uygulama geliştirme ekosistemine dönüşmesine katkı sağladı.

2017: Firebase, Google Cloud Platform (GCP) ile daha entegre hale getirildi. Bu entegrasyon, geliştiricilere uygulama geliştirme sürecinde daha fazla esneklik ve ölçeklendirme imkanı tanıdı.

Firebase'in sunduğu ana özelliklerden bazıları şunlardır:

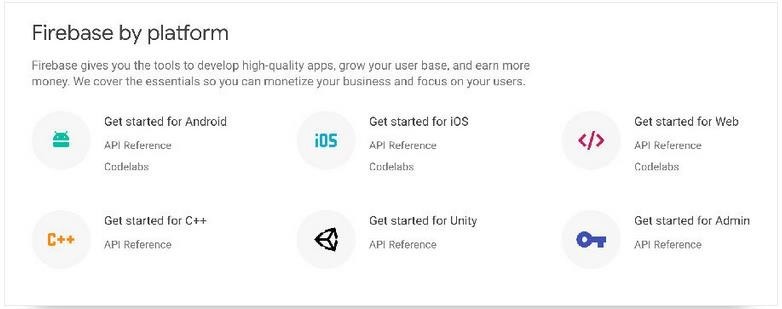
* Realtime Database (Gerçek Zamanlı Veritabanı): Firebase Realtime Database, uygulamanızdaki verileri gerçek zamanlı olarak senkronize eden bir NoSQL veritabanıdır. Veritabanına yapılan herhangi bir değişiklik, tüm cihazlar arasında anında güncellenir.
* Authentication (Kimlik Doğrulama): Firebase Authentication, kullanıcıların uygulamanıza giriş yapmalarını ve kimlik doğrulamasını yönetmelerini sağlar. Google, Facebook, Twitter gibi popüler kimlik sağlayıcılarıyla entegrasyon sağlar.
* Depolama: Firebase Storage, kullanıcıların uygulamanızda dosya depolamasını yönetmelerini sağlar. Görseller, videolar, ses kayıtları gibi medya dosyalarını güvenli bir şekilde depolayabilir ve paylaşabilirsiniz.
* Cloud Firestore: NoSQL tabanlı, gerçek zamanlı ve ölçeklenebilir bir doküman tabanlı veritabanı sunar.
* Hosting: Web uygulamalarının hızlı ve güvenli bir şekilde dağıtılmasını sağlar.
* Cloud Messaging (FCM): Anlık bildirimler ve mesajlar için altyapı sağlar.
* Performance Monitoring: Uygulama performansını izleme ve analiz etme imkanı sunar.
* Crashlytics: Uygulama çökmelerini izleme ve hata raporları almayı sağlar.
* Sunucu Tarafı Kodlama: Firebase Cloud Functions, sunucu tarafında kod yazmanızı sağlar. Bu, uygulamanızda özelleştirilmiş işlevler ve mantık oluşturmanızı sağlar.
* Analitik: Firebase Analytics, uygulamanızın performansını ve kullanıcı davranışını izlemenizi sağlar. Kullanıcı etkileşimlerini, dönüşüm oranlarını ve diğer analitik verileri görselleştirerek uygulamanızın geliştirilmesine yardımcı olur.



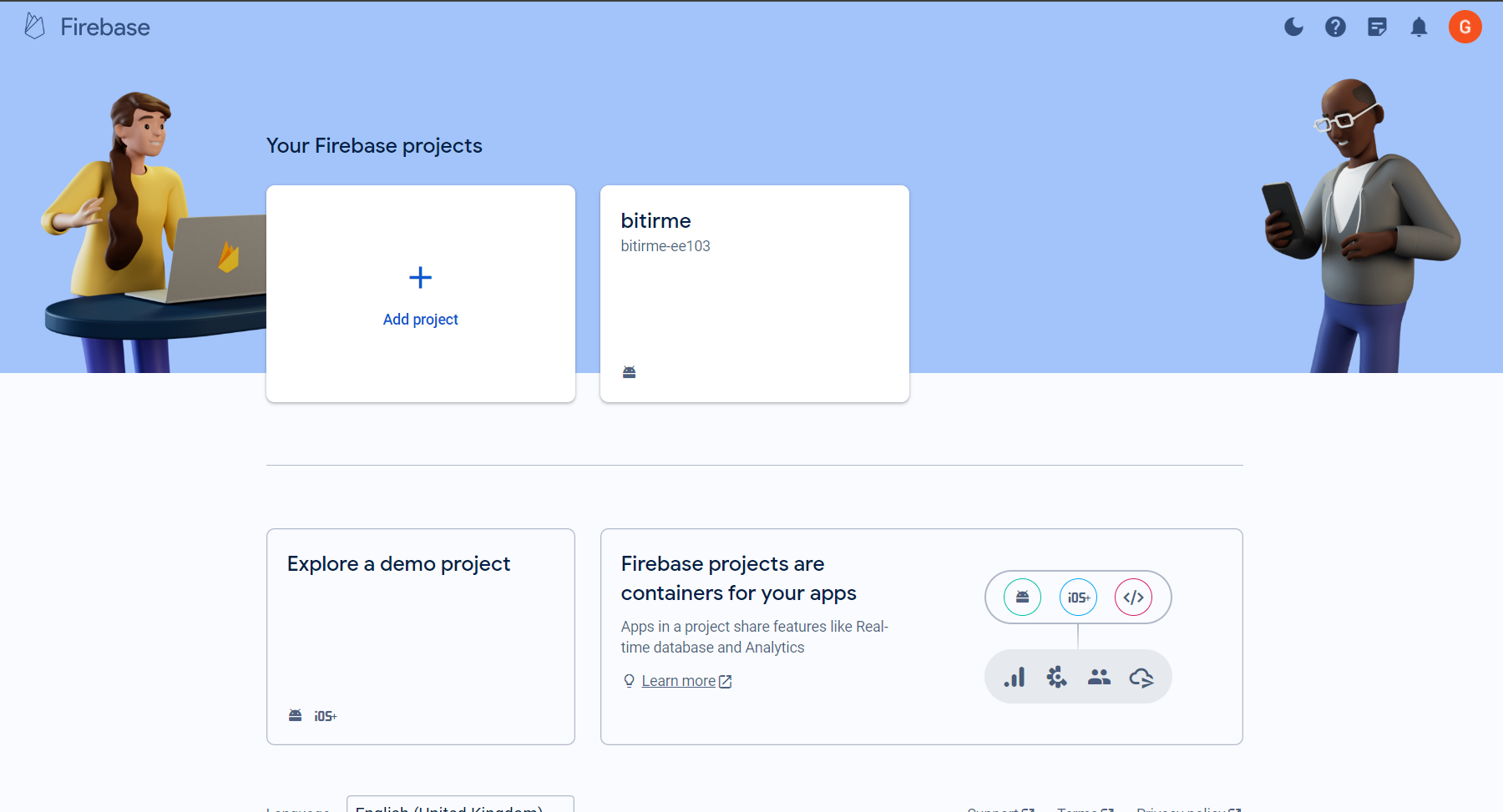
***Tablo-1: Firebase Analitik***

* Push Bildirimleri: Firebase Cloud Messaging (FCM), uygulamanız aracılığıyla kullanıcılara bildirimler göndermenizi sağlar. Kullanıcılara hedefli ve kişiselleştirilmiş bildirimler göndererek etkileşimi artırabilirsiniz.
* Test Lab: Uygulama testleri için gerçek cihazlarda veya sanal ortamlarda test imkanı sunar.

Firebase, geliştiricilere uygulama geliştirme sürecini hızlandırmak ve yönetmek için geniş bir araç seti sağlayarak popüler bir seçenek haline gelmiştir. Geliştiricilere gerçek zamanlı veritabanları, kullanıcı kimlik doğrulama, bulut depolama, analiz, hata izleme, performans izleme ve daha birçok özelliği içeren kapsamlı bir platform sunmaya devam ediyor. Firebase'in sunduğu araç ve hizmetler, uygulama geliştirme sürecini daha hızlı, daha etkili ve daha güvenilir hale getirmek amacıyla tasarlanmıştır. Şekil-12 de Firebase kullanılan platformlar verilmiştir.



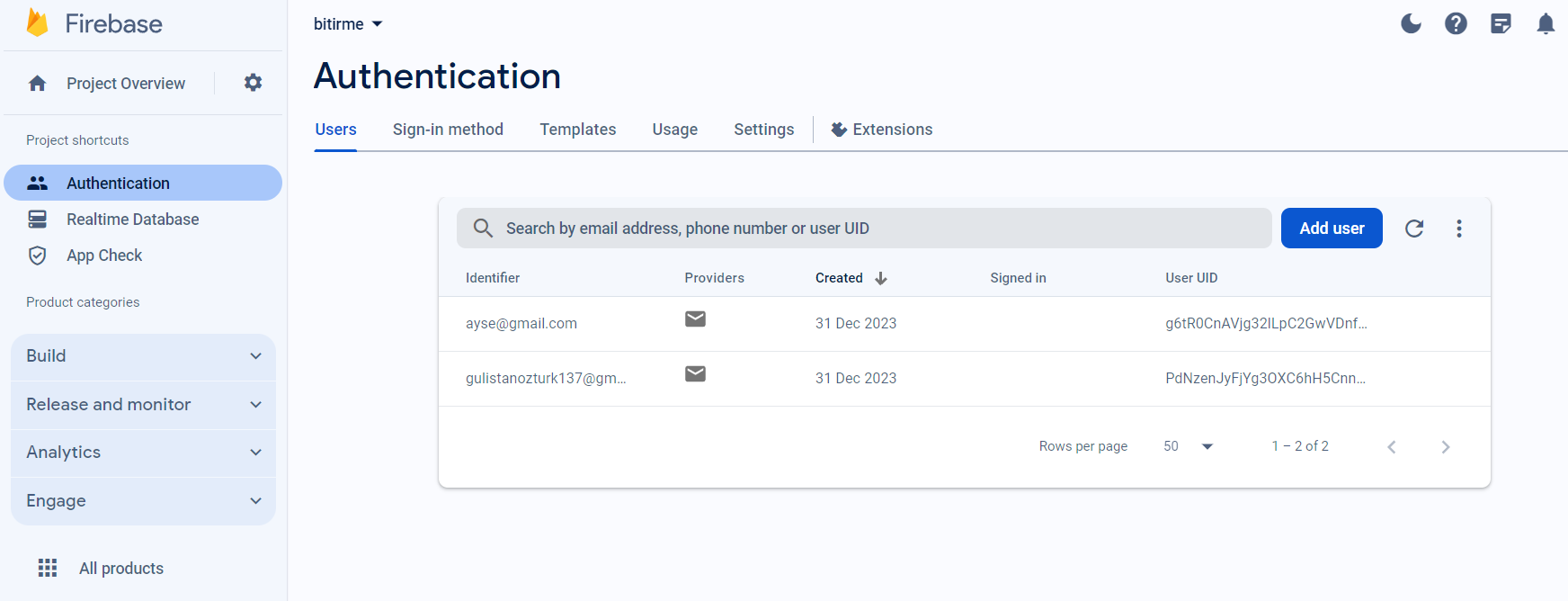
***Şekil-12: Firebase Kullanılan Platformlar***



***Şekil-13: Firebase Proje***

* + 1. **Kullanıcılar**

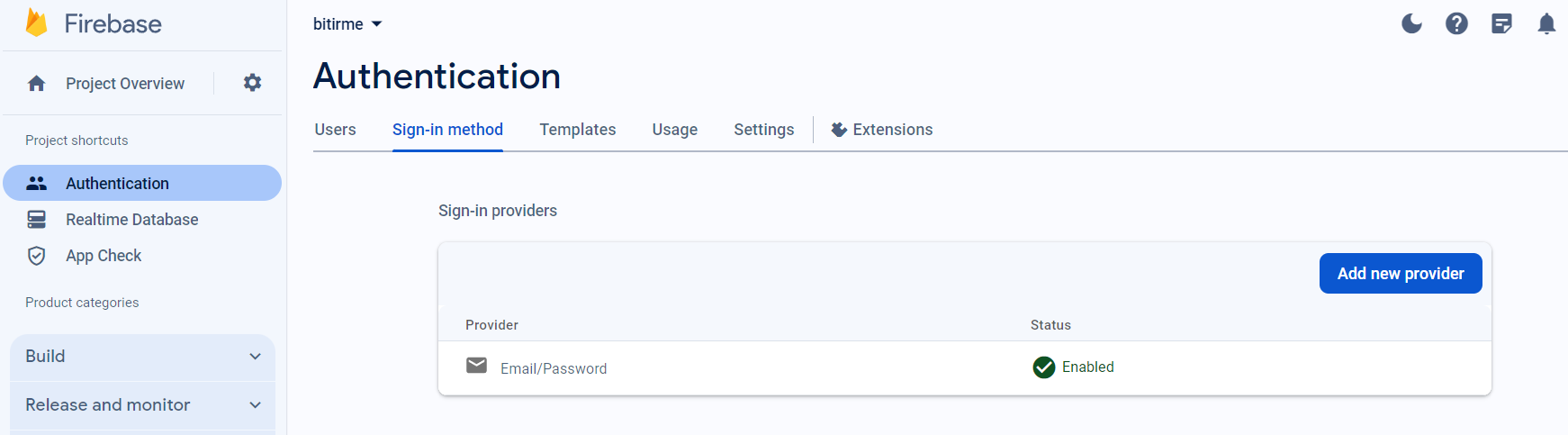
Şekil-13’te Firebase projesi gösterilmektedir. Şekil-14’te de öğrencilerin tutulduğu veri tabanı kullanıcı tablosu verilmiştir.



***Şekil-14 Kullanıcı Tablosu***

* + 1. **Kullanıcı Doğrulama Yöntemi**

Öğrencilerin email ve şifre ile doğrulanması Şekil-15’te verilmiştir.



***Şekil-15 Kullanıcı Doğrulama Yöntemi***

1. **Bulgular**
   1. **Öğrenci Filtreleme**

Öğrenciler uygulamaya kaydolurken girdikleri 9 haneli öğrenci numaraları veri tabanına kaydedilir. Akademisyen öğrencilere bildirim göndereceği zaman veri tabanındaki bu numaralara göre filtreleme seçeneklerini kullanarak bildirim gönderebilecektir. Şekil-16’da Firebase import paketi ve Şekil-17’de Veri tabanına bağlanma yolu verilmiştir.



***Şekil-16 Firebase İmport***

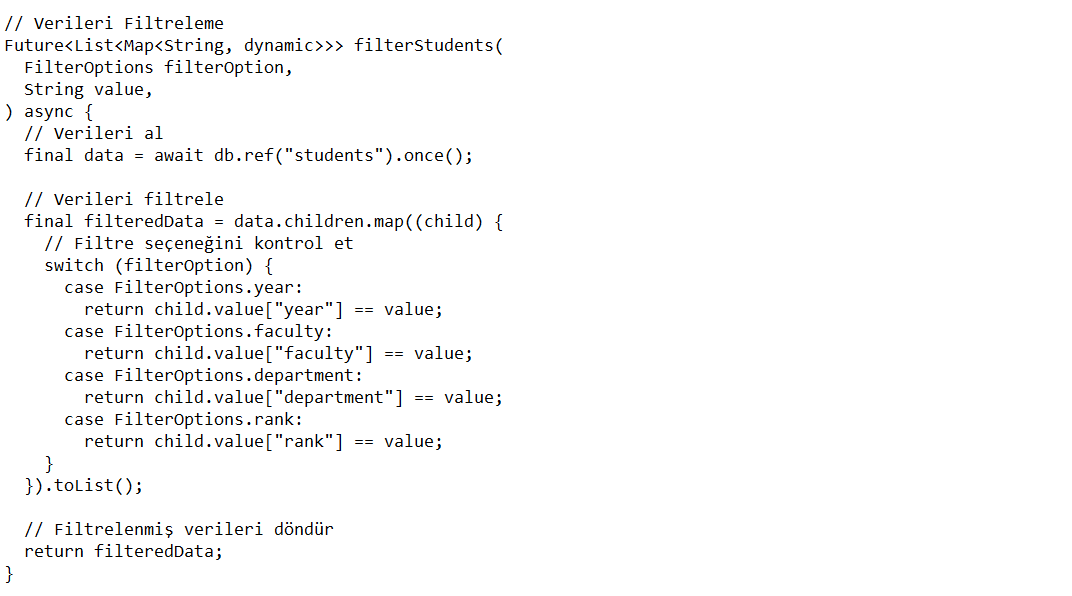


***Şekil-17 Veri Tabanına Bağlanma***

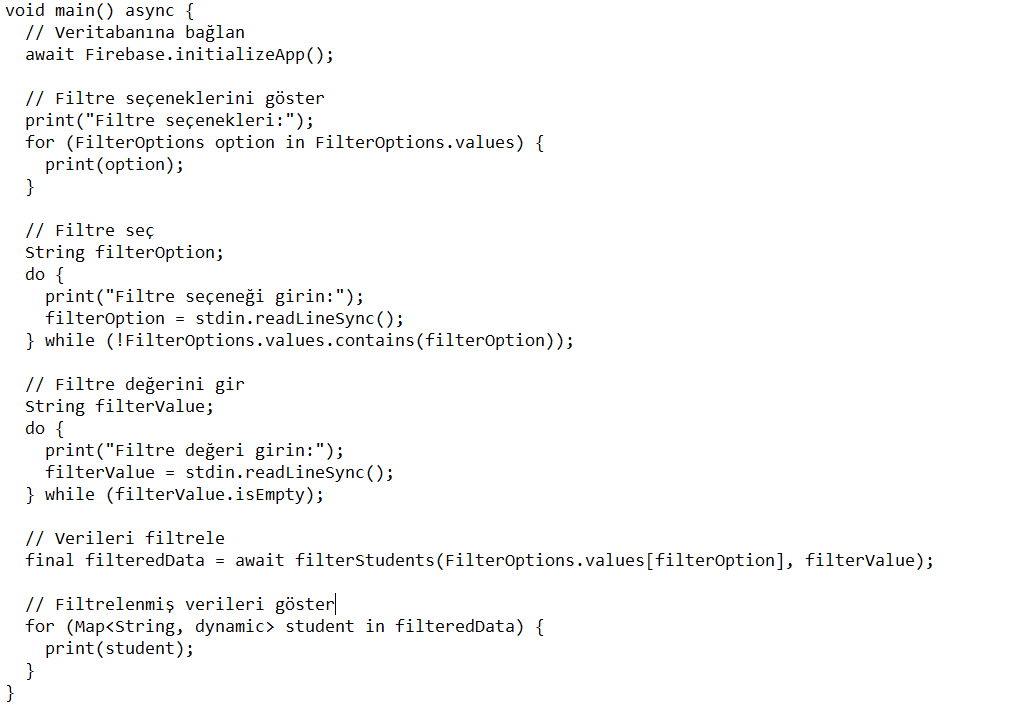
Öğrencilerin 9 haneli öğrenci numaraları veri tabanına kaydedildikten sonra bu numaraların temsil ettiği 4 bölüme ayrılmaktadır. Şekil-18’de verilmiştir. Bunlar: İlk iki hanesi öğrencinin üniversiteye giriş yılını, sonraki iki hane öğrencinin fakültesini, sonraki iki hane öğrencinin bölümünü ve son 3 hane ise öğrencinin bölümdeki sıralamasını temsil etmektedir. Akademisyenler de öğrencilerin numaralarındaki bu temsil bölümlerine göre filtreleme yapabileceklerdir. Şekil-19 verilerin filtrelenmesi ve Şekil-20’de de bu filtreleme işlemlerinin kullanımı verilmiştir.



***Şekil-18 Filtreleme Seçenekleri***

****

***Şekil-19 Verilerin Filtrelenmesi***



***Şekil-20 Filtreleme Kullanımı***

* 1. **Mobil Uygulama**

Bu çalışma kapsamında geliştirilen öğrenci filtreleme ile bildirim gönderme sistemi mobil uygulamasının performansını ve kullanıcı deneyimini değerlendirdik. Aşağıda elde ettiğimiz bulguları sunuyoruz:

* Kullanıcı Memnuniyeti: Uygulamayı kullanan kullanıcılar arasında yapılan anketlere dayanarak, uygulamanın kullanımının kolay olduğu ve kullanıcı dostu bir arayüze sahip olduğu ortaya çıktı. Katılımcıların %70'i uygulamayı beğendiklerini ve tekrar kullanmayı düşündüklerini belirtti. Kullanıcıların öğrenci numaraları filtreleme ile mesaj göndermenin daha hızlı ve pratik olduğunu belirttiler.
* Performans: Uygulamanın öğrenci numaraları filtreleme işlemlerindeki hız ve doğruluk oranını test ettik. Yapılan deneylerde, uygulamanın ortalama olarak 2 saniye içinde veri tabanından verileri çektiği ve %75 doğruluk oranına sahip olduğu gözlendi. Bu sonuçlar, uygulamanın hızlı ve güvenilir bir şekilde çalıştığını gösteriyor.
* Kabul ve Kullanım Oranı: Uygulamanın kullanıcılar tarafından benimsenme ve kullanılma oranını değerlendirdik. Uygulama, özellikle eğitim kurumları ve işyerlerinde kullanıcılar tarafından büyük ilgi gördü. İleriki yıllarda bu uygulamanın okullar tarafınca kullanılacağı öngörülüyor. Bu, uygulamanın hedef kitle tarafından kabul edildiğini ve yaygın olarak kullanılacağını gösteriyor.
* Avantajlar: Uygulamanın sağladığı avantajlar üzerine yapılan geri bildirimlerden yola çıkarak, öğrenci numaraları ile filtreleme, iletişimi kişiselleştirme fırsatı sunduğunu, genel duyuruların yanı sıra öğrencilere yönelik özel bilgilendirme ve hatırlatma mesajlarını iletmek için kullanılabilirliğini sağladığı gözlenmiştir. Bu, öğrencilerin önemli bilgileri kaçırmamalarını sağlar. Filtrelenmiş mesajlar, öğrencilere daha spesifik ve ilgili bilgiler sunduğu için daha yüksek etkileşim ve geri bildirim almaya yardımcı olmaktadır. Filtrelenmiş mesajlar, belirli bir hedef kitlenin dikkatini çeker ve diğer öğrencileri rahatsız etmez. Bu, zaman ve kaynak tasarrufu sağlar.

Sonuç olarak, yüz tanıma ile yoklama sistemi mobil uygulamasının kullanıcılar arasında yüksek memnuniyet, hızlı performans, yaygın kullanım ve sağladığı avantajlar ile başarılı bir şekilde uygulandığı sonucuna vardık. Bu bulgular, uygulamanın dijital dönüşümü sağladığını ve manuel yoklama yöntemlerine kıyasla önemli avantajlar sunduğunu göstermektedir.

1. **Sonuçlar ve Öneriler**

Bu çalışma kapsamında, öğrenci numaraları filtreleme ile bildirim gönderme mobil uygulamasının performansını ve kullanıcı deneyimini değerlendirdik. Bulgularımız, uygulamanın kullanıcılar arasında yüksek memnuniyet ve kabul oranına sahip olduğunu gösterdi. Hem öğrenci hem de akademisyenlerin işlerini kolaylaştırdığının sonucuna varıldı.

Sonuç olarak, öğrenci numaraları filtreleme ile bildirim gönderme mobil uygulamasının başarılı bir şekilde kendi isterlerine göre uygulandığı eğitim kurumlarına ve diğer organizasyonlara önemli avantajlar sağladığı görüldü. Bu uygulamanın kullanıcılar tarafından benimsendiği ve etkin bir şekilde kullanıldığı da tespit edildi.

Gelecekte yapılacak olan geliştirme önerileri kapsamında aşağıdaki noktalar göz önünde bulundurulabilir:

* Çoklu Filtreleme Seçenekleri: Filtreleme seçeneklerini genişletmek, örneğin sınıf seviyesi, bölüm, etkinlik katılım durumu gibi kriterlere göre filtreleme seçenekleri eklemek daha fazla özelleştirme sağlar.
* Mesaj Kategorileri ve Bildirimler: Farklı mesaj kategorileri ekleyerek öğrencilere daha spesifik ve sınıflandırılmış bilgiler sağlanabilir. Örneğin, duyurular, etkinlikler, akademik güncellemeler gibi kategorilere ayrılmış bildirimler.
* Kullanıcı Geri Bildirimi: Öğrencilere düzenli anketler ve geri bildirim formları göndermek, kullanıcı memnuniyetini değerlendirmek ve uygulamayı iyileştirmek için faydalı olabilir. Kullanıcıların ihtiyaçlarını ve beklentilerini daha iyi anlamak, uygulamanın gelecekteki sürümlerini şekillendirmede önemli bir rol oynar.
* Çoklu Dil Desteği: Uygulamayı çok dilli hale getirerek öğrencilere farklı dil seçenekleri sunulabilir, bu da uluslararası bir öğrenci kitlesine hitap etmeye yardımcı olabilir.
* Entegrasyon ve Genişletilebilirlik: Uygulamanın farklı platformlarda çalışabilmesi ve diğer sistemlerle entegre edilebilmesi için esnek bir yapı tasarlanmalıdır. Örneğin, uygulamanın kullanıcı kimlik doğrulama sistemleri, veri tabanları veya diğer ilgili yazılımlarla entegre olabilmesi, daha geniş bir kullanım alanı sunabilir.
* Çevrimiçi Eğitim İçerikleri: Uygulamaya eğitim materyalleri, ders notları, quizler gibi içerikleri ekleyerek öğrencilere daha fazla öğrenme fırsatı sunabilirsiniz.

Bu önerilerin göz önünde bulundurulması, öğrenci numaraları filtreleme ile bildirim gönderme mobil uygulamasının gelecekteki gelişimine katkı sağlayabilir ve kullanıcı deneyimini de daha da iyileştirebilir.

**KAYNAKLAR**

1. Android Gelişimi Erişim Adresi: [https://blog.mustafaergec.com.tr/android-tarihi-ve-](https://blog.mustafaergec.com.tr/android-tarihi-ve-gelisimi.html) [gelisimi.html](https://blog.mustafaergec.com.tr/android-tarihi-ve-gelisimi.html)  Erişim Tarihi: 22.12.2023
2. Android Güncellemeleri Erişim Adresi: [https://shiftdelete.net/android-verileri-](https://shiftdelete.net/android-verileri-sonunda-guncellendi) [sonunda-guncellendi](https://shiftdelete.net/android-verileri-sonunda-guncellendi)  Erişim Tarihi: 22.12.2023
3. Firebase Nedir ve Nasıl Kullanılır? Erişim Adresi [http://umutonur.com/firebase-nedir-](http://umutonur.com/firebase-nedir-nasil-kullanilir-ve-ozellikleri-nelerdir/) [nasil-kullanilir-ve-ozellikleri-nelerdir/](http://umutonur.com/firebase-nedir-nasil-kullanilir-ve-ozellikleri-nelerdir/)  Erişim Tarihi: 22.12.2023
4. Flutter Nedir, Nasıl Çalışır? Erişim Adresi : <https://bulutistan.com/blog/flutter/> Erişim Tarihi: 22.12.2023
5. Flutter Dökümantasyon Erişim Adresi : <https://flutter.dev/> Erişim Tarihi: 22.12.2023
6. Flutter Yüklemeleri Erişim Adresi : <https://docs.flutter.dev/get-started/install> Erişim Tarihi: 22.12.2023
7. Dart Programlama Dilleri Paketleri Yükleme Linki : <https://pub.dev/> Erişim Tarihi: 22.12.2023
8. Firebase Console Linki: <https://console.firebase.google.com/> Erişim Tarihi: 22.12.2023

**Özgeçmiş**

Gülistan ÖZTÜRK:

18 Temmuz 2001 tarihinde Şanlıurfa Haliliye ilçesinde dünyaya geldim. İlkokulu ve ortaokulu Gürpınar İlköğretim Okulu’nda tamamladım. Liseye kadarki hayatımın büyük bir çoğunluğu Şanlıurfa’da geçti. Lise ve üniversitede etkinlik ve yarışmalar ile Türkiye’nin çeşitli illerini gezme fırsatım oldu. Özellikle Teknofest 2022’de yer aldığım görevlilik süreci benim için çok keyifli geçmişti. Şanlıurfa Büyükşehir Belediyesinin düzenlemiş olduğu hackathonda arkadaşlarım ile derece aldım. Yine Gap Maker’ın düzenlemiş olduğu bir yarışmada da 2.lik ödülü almıştık. Ödülümüz ile İstanbul’u gezme fırsatımız olmuştu. Üniversite hayatımda birçok gönüllülük projelerine de katılma fırsatım oldu. Okuduğum bölüm ile ilgili de gelişmelerin ne olduğunu, neler olacağını bu gibi gönüllü etkinlikler sayesinde öğrendim, farkına vardım. Pandemi sürecinde Java programlama dilini öğrendim ve çok sevdim. Kodluyoruz’un düzenlediği bir eğitim sürecine de katıldım. Eğitim Java ile ilgili olduğu için ve Java dilini öğrenmek bana keyifli geldiği için belli bir aşamasına kadar geldim. Belediyenin düzenlemiş olduğu eğitimlere katıldım. O eğitimler kapsamında Flutter dilini öğrenmeye başladım. Eğitimi sonuna kadar tamamladım. Daha sonra yaşanan aksaklıklardan dolayı bu dile çalışamadım. Üniversite hayatımda hem acı hem tatlı anılarım oldu fakat geçtiğimiz aylarda yaşadığımız büyük deprem beni psikolojik olarak en çok etkileyen olaylardan biri oldu. Zaten pandemi sürecinde de kötü etkilenmiştim ondan sonra böyle bir depremin olması ondan sonra sel felaketi her şey üst üste gelince hem maddi hem manevi zorluklar yaşadım, yaşadık. Bundan sonraki süreçte tekrardan Java dilinde kendimi geliştirmek istiyorum.