

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI**

**PATHFINDER**  
**Turistler ve Rehberleri Online Bir**  
**Platformda Buluşturan Platform**  
**Tasarımı**

**B191210309 - Serhat BİLAL**  
**B191210301 - Hilal NARMAN**

**Bölüm : BİLGİSAYAR MÜHENDİSLİĞİ**  
**Danışman : Prof.Dr.Nejat YUMUŞAK**

**2020-2021 Yaz Dönemi**

## ÖNSÖZ

Her Bir köşesi cennet olan bu memleketin gezilecek görülecek çok fazla yeri var. Elbette hepsini saymak pek mümkün değil. Ama konu ile ilgili bilgili ve tecrübeli kişileriz sizlere gezmek istediğiniz bölge ile ilgili detayları paylaşımları ve sizleri yönlendirmesi gezinin çok daha keyif dolu ve sizleri tatmin edecek şekilde olmasını sağlayacaktır. Bu konu hem yerli turist hemde yabancı turistin gündemindedir. Ancak bazen bölgede sizlere rehberlik edecek tecrübeli personelin bulunmaması bazen de rehberlik ücretlerini pahalı olması gibi bir çok sebepten dolayı rehberlik hizmeti tercih edilmiyor.

## İÇİNDEKİLER

ÖNSÖZ.....	II
İÇİNDEKİLER .....	III
ŞEKİLLER LİSTESİ .....	IV
ÖZET.....	V

## BÖLÜM 1

GİRİŞ.....	1
1.1. TUR REHBERLERİ SİTELERİ.....	4
1.2. MOBİL UYGULAMALAR .....	5
1.3. İLETİŞİM BİLGİSİ PAYLAŞIMI.....	6
1.3.1.Email.....	6
1.3.2.Telefon .....	6

## BÖLÜM 2

VERİ TABANI.....	7
2.1. FIREBASE .....	8
2.1.1. FIREBASE HANGİ KİTLEYE HITAP EDİYOR? .....	9
2.1.1.GOOGLE FIREBASE NASIL KULLANILIR ? .....	10

## BÖLÜM 3

API VE GÜVENLİK.....	13
3.1. ARAYÜZ VE KOD BLOKLARI .....	13
3.1.1. SplashScreen Arayüzü .....	13
3.1.2. Kullanıcı Karşılama Ekranı Oluşturulması .....	16
3.1.4. Üye Ol Sayfasının Oluşturulması .....	19
3.1.5. Giriş Yap Sayfasının Oluşturulması .....	26
3.1.5. Ana Sayfasının Oluşturulması .....	32
3.1.6. Ücretsiz İlan Ver Sayfasının Oluşturulması .....	35
3.1.6. Profil Sayfasının Oluşturulması.....	40
3.2. PROGRAM DOSYA YERLEŞİM YAPISI VE BAĞLANTILARI .....	43
3.2.1. Flutter Dosya Yapısı .....	43
3.2.2. Firebase Veritabanı Özellikleri .....	44

## BÖLÜM 4

SONUÇLAR VE ÖNERİLER .....	49
KAYNAKLAR.....	50
EKLER .....	51
ÖZGEÇMİŞ .....	51
BSM401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI.....	52

## ŞEKİLLER LİSTESİ

Şekil 2.1	Firestore Hizmet Yapısı	7
Şekil 2.2	Firestore Logosu	8
Şekil 2.3	Firestore Ağı	9
Şekil 2.4	Yeni Proje Oluşturma Ekranı	10
Şekil 2.5	Giriş Ekranı	11
Şekil 2.6	Özellik Ekranı	11
Şekil 3.1	SplashScreen Arayüzü Görseli	13
Şekil 3.2	SplashScreen Arayüzü Kodları	14
Şekil 3.3	Kullanıcı Karşılama Ekranı Görseli	16
Şekil 3.4	Kullanıcı Karşılama Ekranı Kodları	16
Şekil 3.5	Üye Ol Sayfası	20
Şekil 3.6	Üye Ol Sayfası Kodları	20
Şekil 3.7	Giriş Yap Sayfası	26
Şekil 3.8	Giriş Yap Sayfası Kodları	26
Şekil 3.9	Ana Sayfa Arayüzü	32
Şekil 3.10	Ana Sayfa Kodları	32
Şekil 3.11	Ücretsiz İlan Ver Sayfasının Arayüzü	35
Şekil 3.12	Ücretsiz İlan Ver Sayfasının Kodları	36
Şekil 3.13	Profil Sayfasının Arayüzü	40
Şekil 3.14	Profil Sayfasının Kodları	41
Şekil 3.15	Flutter Dosya Yapısı	43
Şekil 3.16	Firestore Kayıt Yapısı	44
Şekil 3.17	Firestore Kayıt Yapısı	45
Şekil 3.18	Firestore Kayıt Yapısı	45
Şekil 3.19	Firestore Kullanıcı Koleksiyon Yapısı	46
Şekil 3.20	Firestore Rehber Koleksiyon Yapısı	46
Şekil 3.21	Firestore Analiz Yapısı	47
Şekil 3.22	Firestore Analiz Yapısı	47
Şekil 3.23	Firestore Analiz Yapısı	48
Şekil 3.24	Firestore Sürüm Analizi Yapısı	48
Şekil 3.25	Firestore Kullanılan Cihaz Analizi Yapısı	48

## ÖZET

Anahtar kelimeler: Mobil Uygulama, Veri Güvenliği, Bulut Bilişim

Dört bir yanı birbirinden zengin kültüre sahip olan ülkenin gezilmesi ve görülmesi gereken bir çok mekan bulunmaktadır. Bu şekilde bir gezi düşünen kişiler gezmek için internet arama motorlarına hemen gidecekleri bölgelerin adını yazıp nereleri gezeyim yazmaktadırlar. Fakat her ne kadar elimizin altında bulunan internet ile sürekli bir arama yapabiliyor olsak dahi bu bazen çokta başarılı sonuçlar vermemektedir. Bunun için tur rehber acentaları mevcut olmaktadır. Bu noktada da yine ne yazık ki gezginlerin karşısına ya rehberin donanımı veya verdiği hizmet karşılığı alabileceği ücret gezgini memnun etmemektedir.

Bu tasarım çalışması ile Google tabanlı Flutter ile oluşturmuş olduğumuz tur rehberine ait olan bilgiler yine Google tabanlı olan Firebase Veritabanı ile saklanacaktır. Mobil uygulama projesi Flutter SDK'lar kullanılarak Dart programlama dilinde yazılmıştır. Yazılmış olan bu API'lar tamamen güvenli bir şekilde kullanılmıştır.

Uygulamaya kayıtlı olan gezginler ilgilendikleri bölge ile alakalı olarak uygulamaya kayıtlı olan tur rehberleri arasından kendileri için en uygun olan rehber ile iletişime geçebileceklerdir.

Sonuç olarak uygulama sayesinde turist veya diğer adı ile gezginler ile tur rehberlerini online bir platform üzerinde güvenli bir şekilde buluşturmuş oluyoruz.

## BÖLÜM 1 GİRİŞ

Dünyada yazılı kaynaklara geçmiş ilk rehberler Mısırlı rahipler olarak anılmaktadır. Ancak, İ.Ö. 7. yüzyılda başlayan olimpiyat oyunları organize seyahatlerin başlangıcı olmuş ve beraberinde turist rehberliğini de getirmiştir.

Anadolu’da doğup büyüyen tarihçi ve coğrafyacı Herodot, gezdiği yerleri ve geleneklerini anlatan ilk rehberlerdendir ve bir anlamda kültür turizminin öncüsü sayılabilir. Daha sonraları ticaret ve seyahatlerin artışına paralel olarak Anadolu kıyılarını ziyaret edenlerin de sayısı artmıştır. Ancak, bu dönemlerde rehberlik yapanlar kulaktan dolma bilgileri aktarmaktadırlar. Ortaçağ’da Haçlı seferleri ile kitlesel olarak seyahatler başlamış olup, ilkel biçimde de olsa bu seyahatlerde ulaşım, konaklama, güvenlik ve rehberlik organizasyonu görülmektedir. İzleyen dönemlerde ise seyahat büyük masraf gerektirdiğinden zenginlerin yararlanabildiği ayrıcalıklı bir zevk ve statü kaynağı olmuştur.

Turizmin ve turist rehberliğinin günümüzdekine benzer bir niteliğe kavuşması ancak Sanayi Devrimi’ni izleyen yıllardadır. 1851’de Thomas Cook’un İngiltere’de düzenlediği bir günlük tren gezisi modern turizmin başlangıcı kabul edilmektedir.

20. yüzyıl başlarında İngiltere’de Londra Kulesi ve Windsor Şatosu rehberli gezilere açılmış, bazı seyahat acenteleri rehber istihdam etmeye başlamışlardır. Ancak, bölgesel olarak rehberlerin eğitimi ve örgütlenme çalışmaları 1930’lu yıllardan itibaren başlamıştır. Turizmin gelişmesi ve kitle turizminin başlaması ile profesyonel turist rehberliği hizmeti zorunluluğu doğmuştur.

Ülkemizde rehberliğin geçmişi 19. yüzyılın sonlarına kadar uzanmaktadır. Rehber örgütleri de elli yılı aşkın bir süredir turist rehberlerinin sunduğu hizmetin niteliğinin arttırılması, hizmet içi eğitim, uzmanlaşma alanlarında çalışmalarını sürdürmekte, ülke

turizminin geliştirilmesi konusunda yapılan çalışmalara etkin bir biçimde katılmaktadırlar.

1839'da Tanzimat'ın ilanı ile eğitim, güzel sanatlar ve ticaret alanında Avrupa ile ilişkiler gelişmiş ve İstanbul'da yaşayan yabancı sayısında artış olmuştur. Avrupa'da seyahat alanındaki gelişmelere paralel olarak Paris-İstanbul arasında Orient-Express tren seferlerinin 5 Haziran 1883'de başlaması ve Kırım Savaşı'nda Osmanlıların batılı güçleri desteklemesi, batılıların ülkeye sempati duymasına ve ticaret, gezi amaçlarıyla ziyaretlerin yoğunlaşmasına neden olmuştur.

Bu gelişmeyle birlikte Osmanlı'da 19. yüzyıl sonunda görülmeye başlayan turist rehberliği, ülkenin tanıtımı ve ulusal güvenlik açısından büyük önem taşımaktaydı. Mesleğin kurumsallaştırılması çabalarının ilk örneği, 29 Ekim 1890 tarihli 190 sayılı Nizamname'dir. Rehberliği belli kurallara bağlayan bu nizamnamenin yürütülmesi görevi İçişleri tarafından belediyelere verilmişti. Nizamnamede mesleğin icrası sırasında iyi niyet gözetmek, düzgün bir Türkçe ve yabancı dil bilgisi yeterliliğinin sınavla tespiti öngörülmekteydi.

Türkiye Cumhuriyeti'nin kurulmasından iki yıl sonra 8 Kasım 1925 tarih ve 2730 sayılı Ecnebi Seyyahlara Tercümanlık ve Rehberlik Edecekler Hakkında Kararname ile rehberlik konusunda sıkı kurallar getirilmiştir. Kararnamenin üzerinde durduğu en önemli konu tercüman rehberlerin eğitimi idi, ancak turistik faaliyetlerle ilgili bir kuruluş olmadığından rehberlerin yeterliklerinin saptanması konusundaki sınav koşulu kağıt üzerinde kalmıştır. 20'li yılların sonlarında ekonomik krizin etkisiyle Türkiye'de de turizmin durma noktasına geldiği bir sırada 1928 yılı Aralık ayında İstanbul İktisat Müdürlüğü bir rehberlik kursu açılacağını basın aracılığı ile duyurmuş, bu ilanla kursa sınavla alınan adaylardan 50'si tercüman-rehber olmaya hak kazanmıştır. 1935 yılında belediyelerde turizm şubelerinin kurulmasının ardından bir rehberlik kursu daha açılmış ve bu kurstan da 53 kişi mezun olmuştur. Ancak, 40'li yıllarda II. Dünya Savaşı'nın etkisi ile turizm durmuş ve rehberlerin büyük bölümü mesleği bırakmıştır. 50'li yıllarda modern turizm önem kazanmış ve 1950 yılındaki İkinci Turizm Danışma

Kurulu'na Basın-Yayın ve Turizm Genel Müdürlüğü tarafından sektör için önemini vurgulayan Turist Tercüman Rehberliği başlıklı bir rapor sunulmuştur. Bunun ardından 1951 yılında Türkiye Milli Talebe Federasyonu (TMTF) rehberliğin günün koşullarına uygun hale getirilmesi için bir çalışma yapmıştır. Üniversite öğrencileri için açılan rehberlik kursunu tamamlayan 129 genç mesleğe adım atmıştır. 1955-60 yılları arasında çeşitli öğrenci derneklerinin organizasyonu ile başlatılan ve İstanbul Belediyesi ile Basın Yayın İstanbul İl Müdürlüğü tarafından gerçekleştirilen iki ayrı kurstan 128 öğrenci mezun olarak çeşitli dillerde rehberlik yapma hakkı kazanmıştır.

20 Ağustos 1963 tarihinde kurulan ve çalışma alanına giren konularda personel yetiştirmek üzere kurslar, uzman okullar açabilme yetkisine sahip olan Turizm ve Tanıtma Bakanlığı, 1964-68 yılları arasında kurslar düzenlemiştir. Ancak bu kurslar bir yönetmeliğe değil, 8 Kasım 1925 gün ve 2730 sayılı Kararname'ye dayanmış, hizmetiçi talimat ve genelgelerle gerçekleştirilmiştir.

Turist rehberliği ile ilgili ilk yönetmelik 3 Eylül 1971 gün ve 13945 sayılı Tercüman Rehber Kursları ve Tercüman Rehber Yönetmeliği adıyla yayımlanmıştır. Bu yönetmelik 21 Mart 1974 tarihinde değiştirilerek Profesyonel Turist Rehberliği Yönetmeliği adını almış, 1981 ve 83 yıllarında bazı değişikliklere uğramıştır. 2 Temmuz 1986 tarihinde 19152 sayılı Resmi Gazete'de yayımlanarak yürürlüğe giren Profesyonel Turist Rehberliği Yönetmeliği de bazı değişikliklere uğrayarak 25 Kasım 2005 tarihli ve 16004 sayılı Resmi Gazete'de yeni Profesyonel Turist Rehberliği Yönetmeliği adıyla yayımlanmıştır. Günümüzde Bakanlığın açtığı kursların yanında üniversitelerin iki yıllık ön lisans ve dört yıllık lisans programları ile de turist rehberi yetiştirilmektedir.

Geçen yıllar içerisinde barındırdığı kültürel ve doğal zenginlikler nedeniyle Türkiye için turizmin önemi giderek artmış, ancak turizmde önemli rol oynayan rehberlerle ilgili yasal düzenlemelerde arzulanan sonuca ulaşamamıştır. Rehberliğin gerektirdiği çağdaş yasal düzenlemeleri içeren bir yasa önerisi “6326 Sayılı Turist Rehberliği Meslek Kanunu “07.06. 2012 tarihinde TBMM’de kabul edilmiş, 22.06.2012 tarihli Resmi Gazete’de yayımlanarak yürürlüğe girmiştir.[1]



Günümüzde turistler tur acentalarının sunmuş oldukları paketlerden yararlanmaktadır. Bu paketlerin içlerinin turistin beklentisini karşılamaması, ücretlerinin pahalı olması , müşteri ilişkilerinin verimli olmaması gibi turistlerin memnun olmadığı bir çok ekten olmaktadır.

Yapmış olduğumuz mobil uygulama ile işinde uzman tur rehberleri ile tursitler online bir platformda güvenli ve beklentileri karşılayacak bir şekilde buluşturmaktadır. Hedef kitleye kısa yoldan, kolay ve doğru bir şekilde ulaşılabilir olması, hızlı destek sağlanabilmesi, mobil uygulamaların hedef kitlenin ihtiyaçlarına daha net çözümler sunabilmesi, yer ve zamandan bağımsız uygulamaların kullanılabilir olması ile kullanıcıya daha rahat hizmet sunulabilmesi, kullanıcıların mobil uygulamaları daha çok tercih etmeleri gibi sebeplerden dolayı mobil uygulama konusunda çalışma yapılırken Googlenin geliştiricisi olduğu Dart dilinin Flutter Framework’ü Flutter’ı seçmemizdeki nedenlerin başında öncelikle cross platform oluşuydu yani aynı kodu bir kez yazdıktan sonra aynı ekran çıktısını hem android hem ios hemde web platformları için alınabilecek olmasıdır. Bir diğer neden ise bu dilin arkadaşında Google gibi büyük bir şirketin oluşu ve dilin dünya çapında hızla bilinirliğinin artmasıydı.[2]

### **1.1. Tur Rehberleri Siteleri**

Eskiden her hangi bir yere gezmeye gidileceği zaman önce akraba ziyaretleri yapılırdı. Sonra kulaktan duyma anlatımlar ile bir yerlere gidilir yine aynı şekilde gidelen mekanlar kulaktan duyma, doğru olmayan veya eksik bilgiler dinlenirdi. Bu da gezilen görülen mekanla ilgili verimli bir etkinlik olmamış oluyordu.

Zamanla rehberlik adı altında bir faaliyet alanı ortaya çıktı. Bölge hakkında araştırmalar yapan, doğru ve net bilgilere ulaşarak uzman kişiler bu faaliyet alanında çalışmalara başladılar. Bu çalışma ile turisler gezme aktivitesinden daha çok zevk almaya, doğru bir biçimde gezme-görme aktivitesinin içi dolu hale gelmiştir.

İnternetin 1990 lı yıllarda yaygınlaşarak hayatımızın büyük bir kısmında yer alması sonucunda[3] web siteleri kurulmaya başlanmıştır. Tabi bu web sitelerinin içerisinde turizme yönelik olanları da mevcuttu. Gün geçtikçe site sayısı artmaktadır. Bu da her bir sitede farklı bilgilerin oluşmasına yerli yabancı tüm turistlerin bilgi karmaşasında boğulmasına neden olmaktadır. Bu da hem zaman hem de maddi kayıplara neden olmaktadır.

## **1.2. Mobil Uygulamalar**

1973 yılında Martin Cooper cep telefonunu icat etmiştir. 80'li yılların sonlarına doğru telefon kullanımı iyice yaygınlaşmıştır. Ardından telefon geliştiricileri tarafından yeni bir arayış başlamış ve insanların arama yapma ihtiyacından doğan ilk arama motorları gündeme gelmiştir. Elbette o dönemin telefonları donanım açısından yeterli değildi. Telefon hafızalarının düşük olması ise diğer bir sorundu. Mobil uygulamalar fikir olarak düşünülse de hayatımıza bu yüzden çok geç gelmiştir. Telefon modelleri geliştirildikçe işletim sistemleri de geliştirilmiş ardından uygulamaları yüklü olarak teslim edilen telefon modelleri çıkmıştır. Bunların başında şüphesiz bir zamanların mobil devi olan Nokia gelmektedir. 90'lı yıllara gelindiğinde telefonlar mesajlaşma uygulaması, Snake, Tetris gibi oyunlarla birlikte satılmaya başlanmış. Ancak o zamanlarda yeni bir uygulama indirmek mümkün değildi. İlk işletim sistemleri ise 2000'li yıllarda çıkar. Hepimizin severek kullandığı İOS işletim sistemi 2007 yılında, Android işletim sistemi ise 2008 yılında hayatımıza girmiştir. Beraberinde mobil uygulama mağazaları da geliştirilmeye başlandı ve uygulama geliştiricileri artan talep üzerine sayısız mobil uygulama üretmeye devam etmektedir.[4]

Gezme planı yapan yerli veya yabancı turistler gerek tavsiye üzerine gerek internet arama motorlarından yapmış oldukları araştırmaya neticesinde ulaştıkları sonuç neticesinde veya gerekse herhangi bir tur acentalarının hazırlamış oldukları paketlerden satın alarak sınırlı kapsamda oluşturulan planlar doğrultusunda gezilmektedir. Fakat bu bazen turistlerin beklentisini karşılamamakta bazen de zaman planlamasının acente yetkililerince yapılması, rehber ücretlerin yüksek olması,

rehberlerle yaşanan iletişim problemi gibi birçok problemler ile karşılaşilmektedir. Mobil uygulamaların gün geçtikçe gelişmesi bizleri bu alanda teşvik etmektedir.

### **1.3. İletişim Bilgisi Paylaşımı**

E-posta, cep telefonu numarası gibi iletişim bilgilerinin paylaşımı internetin olmadığı zamanlarda kişisel olarak paylaşıldığından bir zorluk oluşturmaktadır. Zaman içinde internetin yaygınlaşması sonucunda e-posta, cep telefonu numarası gibi iletişim bilgilerinin herhangi bir platformda paylaşmak oldukça riskli hale gelmiştir.

#### **1.3.1.Email**

Elektronik ortamda bu adres, e-mail (e-posta) adresi olarak adlandırılır. Bu bilginin normal şartlarda paylaşımı ve işlenmesi KVKK'ya tabi olduğundan aydınlatma metni ve kullanıcı onayı alındıktan sonra paylaşımına açık hale gelecektir[5]. 3. Kişilerle paylaşılması durumunda kullanıcılar sözleşme dahilinde bilgilendirilecektir.

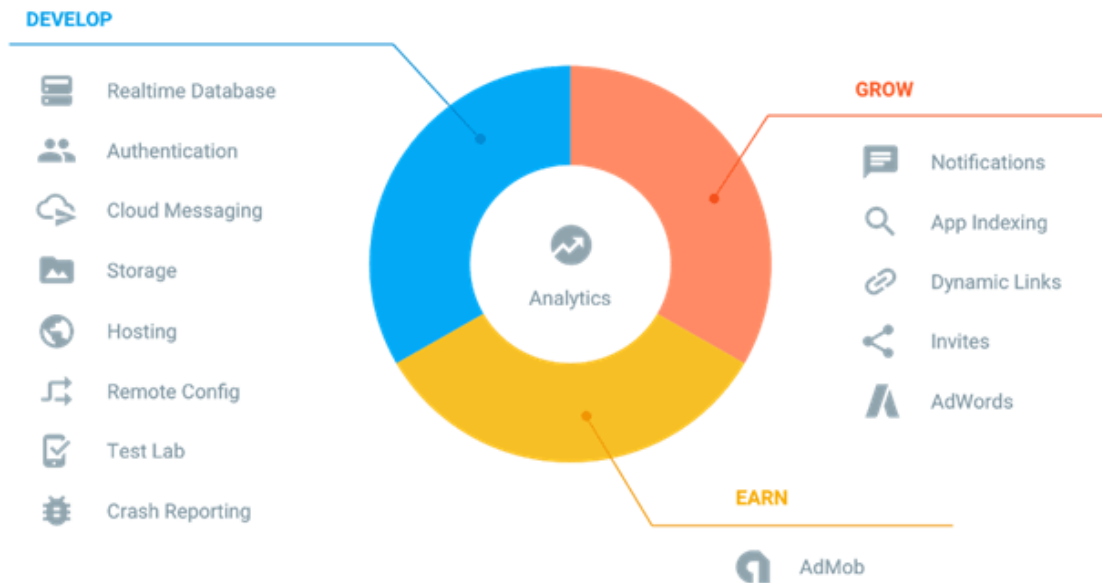
#### **1.3.2.Telefon**

Telefon numarası, bir telefon hattına bağlı sabit hatlı bir telefon abone istasyonuna veya telsiz telefon veya cep telefonu gibi bir kablosuz elektronik telefon cihazına ya da genel veri aktarımı için diğer cihazlara atanan bir dizi rakamdır. Bu bilgi de 1.3.1'deki KVKK'ya aynı şekilde uyularak paylaşılacaktır. [5]

## BÖLÜM 2 VERİ TABANI

Sunucu tarafında programlama sorunları yaşamadan kullanıcılara Android veya iOS uygulamaları sunmak için kaliteli uygulama geliştirme platformunu bulmanız son derece önemlidir.

API'larımızda Googlenin geliştiricisi olduğu Dart dilinin Flutter Framework'ü tercih ettiğimizden Flutter ile entegre bir şekilde çalışan yine Google'un geliştiriliğini üstlendiği Firebase'i tercih ettik. Google tarafından 2014 yılında satın alınan bir Hizmet olarak Backend platformu olan Firebase'in en bilinen özelliği gerçek zamanlı veritabanı işlemleridir ve platform dahilindeki Mart 2020 tarihli sürüm notlarına göre konsolun tamamı Angular kullanılarak yeniden yazılmış, mobil ve web geliştiriciler için back-end hizmeti sunan bir cloud(bulut) teknolojisidir. Firebase geliştiricilere, gerçek zamanlı veritabanı(realtime database), kimlik doğrulama(authentication), depolama(storage), hosting, bildirim gönderme gibi birçok hizmeti sağlıyor. Firebase mobil ve web uygulamaları için bütün veriyi bulutta saklayarak, tek bir platformun üzerinden burada bahsettiğimiz hizmetleri bir yazılım geliştiriciler için sunmaktadır.[6]



Şekil 2.1 Firebase Hizmet Yapısı

## 2.1. FIREBASE

Google Firebase; web ve mobil uygulamalarının server tarafıyla geliştiricinin uğraşmasına gerek kalmadan kullanıcı giriş yetkilendirmeli ve verilerini gerçek zamanlı ve senkron bir şekilde tutulmasını sağlayan bir platformdur. Günümüzde ki projeler tüm markete hitap etmesi açısından iOS , Android ve web platformlarında geliştirilir fakat her platformun kendine ait yazılım dili ve bağlantı şekilleri vardır. Server-Side dediğimiz arkaplanda ki verilerin tutulması ve gerektiği zaman kullanıcıya kullanılması her platformun ortak sorunudur ve Google Firebase bu konuda geliştirilmiş ortak bir çözümdür.



Şekil 2.2 Firebase Logosu

Google Firebase'in başlıca özellikleri :

- Gerçek zamanlı veritabanı
- Kullanıcı giriş yetkilendirmesi
- Depolama
- Makine Öğrenmesi Kitleri
- Performans ve hata test ortamları
- Platformlar arası ortak uygulama analizi
- Ortak fonksiyon

- Toplu bildirim
- Reklam araçları

Bu özelliklerin bir çoğu her platformda tek tek uğraşılması gereken işler olması gerekirken geliştiriciler Google Firebase ile tek seferde ve kolay bir çözüm olanağı bulabiliyor.[7]

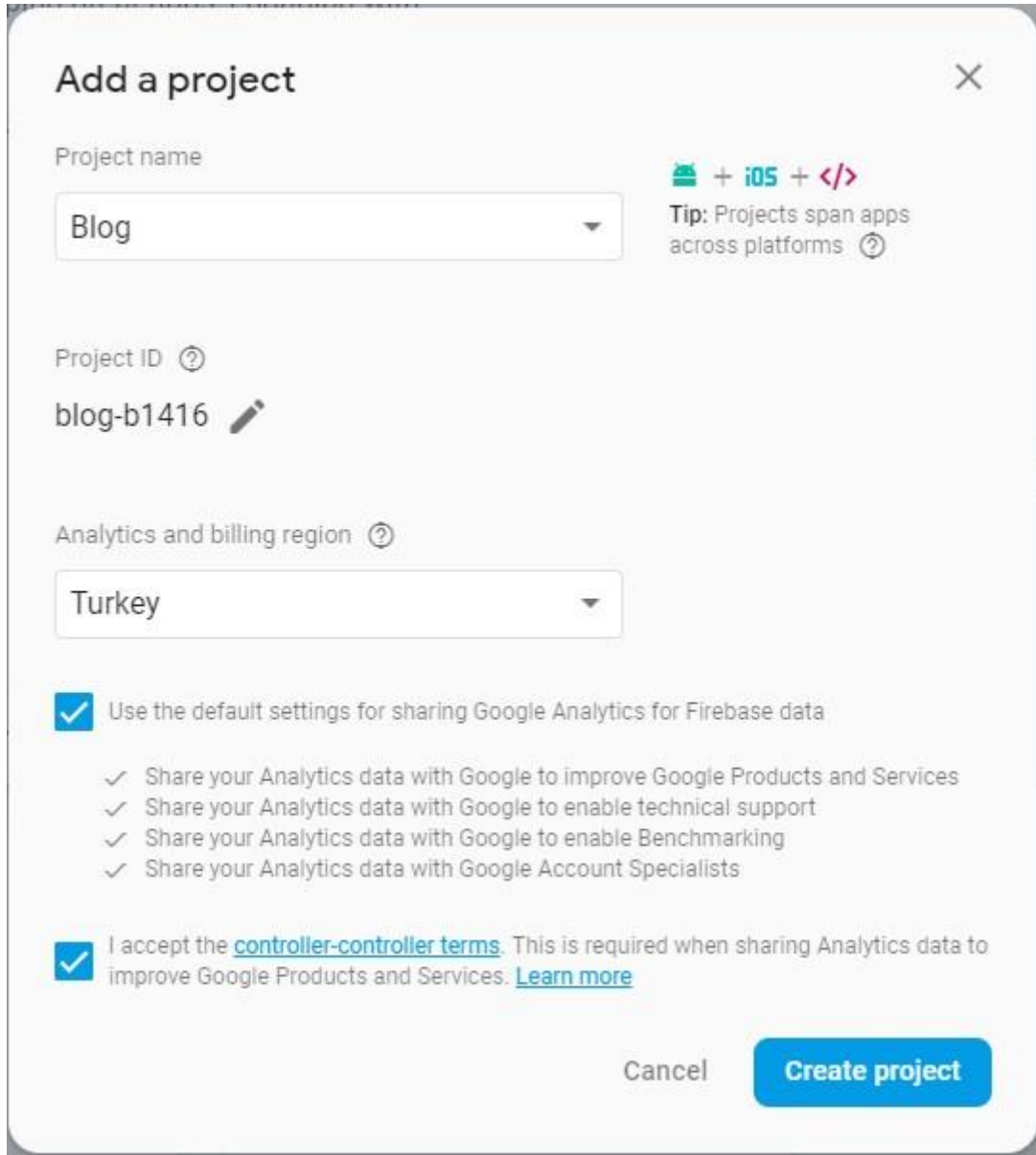
### 2.1.1. FİREBASE HANGİ KİTLEYE HITAP EDİYOR?

Kullanıcı girişlerinin olduğu ve belirli verilerin saklandığı birden fazla platforma (iOS&Android vb.) geliştirilecek bir projeniz varsa Google Firebase size bu konuda oldukça fayda sağlayacaktır. Ayrıca Firebase'in sağladığı araçlar işinizi çok kolaylaştıracağı için tek platformlu uygulamalarınızda da kullanabilirsiniz.[7]



Şekil 2.3 Firebase Ağı




### 2.1.1.GOOGLE FIREBASE NASIL KULLANILIR ?



**Add a project** ×


Project name

Blog

 +  + 

**Tip:** Projects span apps across platforms ?

Project ID ?

blog-b1416 

Analytics and billing region ?

Turkey

☒ Use the default settings for sharing Google Analytics for Firebase data

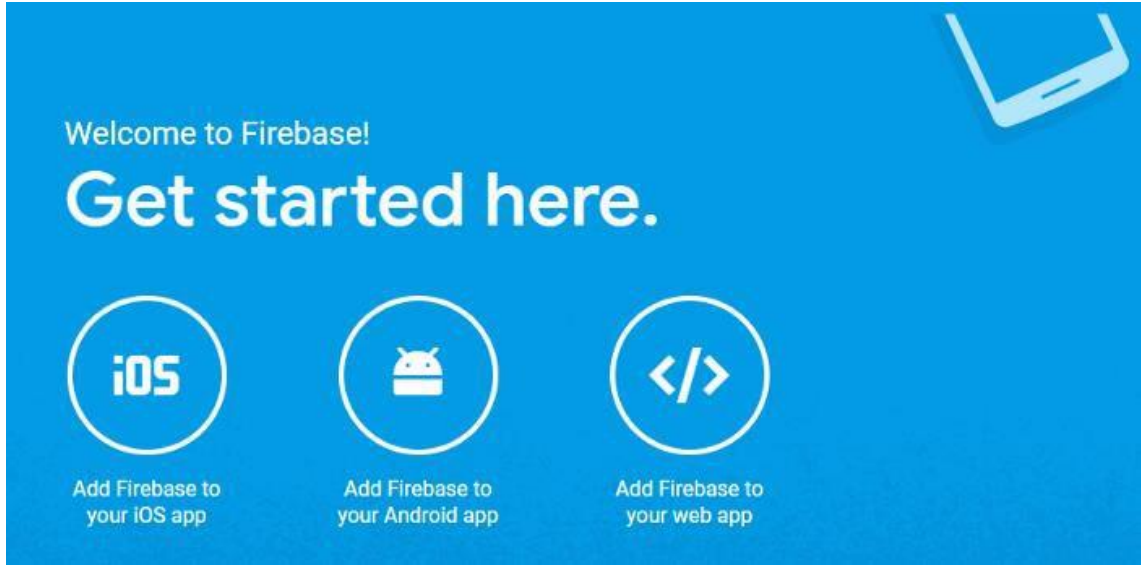
- ✓ Share your Analytics data with Google to improve Google Products and Services
- ✓ Share your Analytics data with Google to enable technical support
- ✓ Share your Analytics data with Google to enable Benchmarking
- ✓ Share your Analytics data with Google Account Specialists

☒ I accept the [controller-controller terms](#). This is required when sharing Analytics data to improve Google Products and Services. [Learn more](#)

Cancel **Create project**

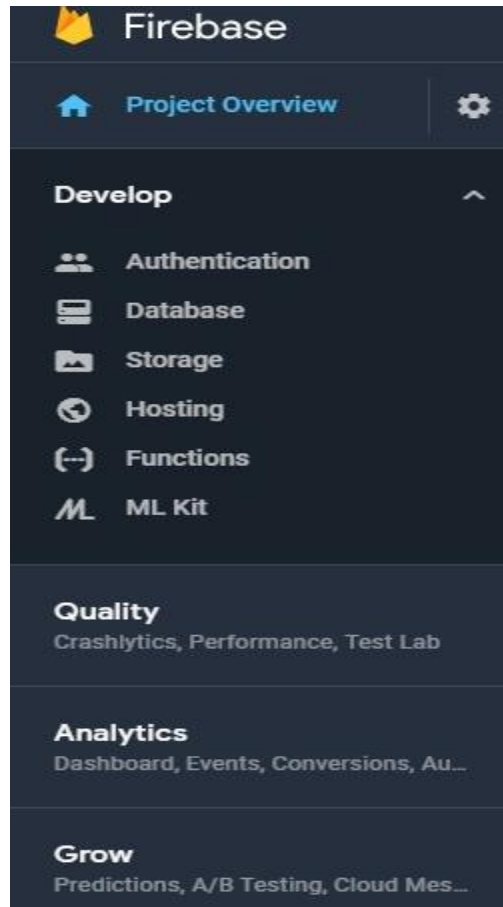
Şekil 2.4 Yeni Proje Oluşturma Ekranı

Google Firebase' i kendi resmi sitesinden kullanmaya başlayabilirsiniz. Gmail hesabınızla kaydolduktan “New Project” butonuna basın. Ardından proje isminizi ve ülkenizi seçtikten sonra “Create Project” butonuna basarak bir proje yaratın.



Şekil 2.5 Giriş Ekranı

Ardından kullanacağınız platformu seçerek yönlendirmeler sayesinde kurulumu yapın.



Şekil 2.6 Özellik Ekranı

Son olarak yapılması gereken yukarıda gördüğünüz Firebase'in istediğiniz özelliğini seçip kullanmaya başlamak.[7]

The New York Times, npr one , HALFBRICK, duolingo,Alibaba.com, venmo,The



Economist,trivago,wattpad,Ctrip,GAMELOFT,lyft dahil olmak üzere dünyanın dört bir yanındaki geliştirme ekipleri, uygulamalarını göndermek için Firebase'I kullanmaktadır.

Gerçek zamanlı veri tabanı,bulut tabanlı NoSql(Not Only Sql) bir veri tabanı sistemi olan Firebase hiç bir sql sorgusuna gerek duymadan json parametreleri ile yöneterek veri depolamanın yanı sıra asenkron çalışması ile veri değişimlerinin anlık olarak takip edilmesine olanak sağlıyor.[8].

## BÖLÜM 3 API ve GÜVENLİK

Veri tabanını, tabloları, oluşturduktan sonra bunları kullanıcı ile buluşturmak için API kullanmamız gerekiyor. Arka tarafta API için Flutter SDK'lar kullanılarak Dart programlama dili kullanmayı tercih ettik. Google tabanlı Flutter ile oluşturmuş olduğumuz projedeki veriler yine Google tabanlı olan Firebase Veritabanı ile saklanacaktır.[9]

### 3.1. Arayüz ve Kod Blokları

Bu bölümde program tasarımını oluşturan kodlar ve arayüzler ele alınacaktır.

#### 3.1.1. SplashScreen Arayüzü

Öncelikle uygulamanın kullanıcıyı karşılama kısmı olan kullanıcı karşılama arayüz sayfası Flutter'da Dart programlama dili kullanılarak oluşturuldu. Kodları aşağıdaki gibidir.



Şekil 3.1 SplashScreen Arayüzü Görseli

```

import 'dart:async';
import 'package:pathfinder/screen/welcome.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

class SplashScreen extends StatefulWidget {
  const SplashScreen({Key? key}) : super(key: key);

  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
    Timer(
      const Duration(seconds: 3),
      () => Navigator.pushReplacement(
        context, MaterialPageRoute(builder: (context) => const Welc
omePage())));
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        height: double.infinity,
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topRight,
            end: Alignment.bottomRight,
            colors: [

```

```

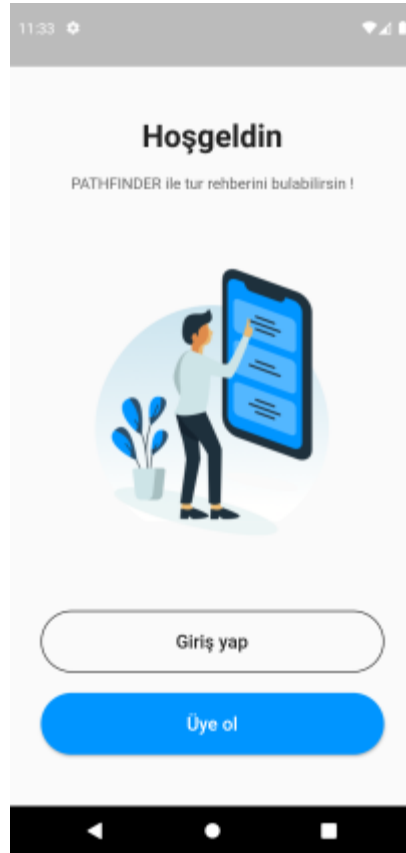
        Colors.white,
        Colors.lightBlueAccent,
    ])),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisAlignment: MainAxisAlignment.spaceAround,
      children: [
        Column(
          children: [
            Image.asset(
              'assets/images/logo.png',
              width: 300.0,
              height: 300.0,
            ),
            const SizedBox(
              height: 50.0,
            ),
            const Text(
              "PATHFINDER",
              textAlign: TextAlign.center,
              style: TextStyle(
                color: Colors.white,
                fontWeight: FontWeight.bold,
                fontSize: 42.0),
            ),
          ],
        ),
        const CircularProgressIndicator(),
      ],
    ),
  ));
}
}

```

Şekil 3.2 SplashScreen Arayüzü Kodları

### 3.1.2. Kullanıcı Karşılama Ekranı Oluşturulması

Uygulama açıldıktan 3 saniye sonra uygulama kullanıcıyı aşağıdaki görseli mevcut olan login sayfasına otomatik yönlendirilir. Yönlendirilen bu sayfada kullanıcıya 2 farklı seçenek sunulur: Üye Ol, Giriş Yap. Görseli ve kodu aşağıdaki gibidir.



Şekil 3.3 Kullanıcı Karşılama Ekranı Görseli

```
import 'package:flutter/material.dart';
import 'package:pathfinder/screen/singup.dart';
import 'login.dart';

class WelcomePage extends StatelessWidget {
  const WelcomePage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```

body: SafeArea(
  child: Container(
    width: double.infinity,
    height: MediaQuery.of(context).size.height,
    padding: const EdgeInsets.symmetric(horizontal: 30, vertical:
50),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceBetween,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[
        Column(
          children: <Widget>[
            const Text(
              "Hoşgeldin",
              style: TextStyle(
                fontWeight: FontWeight.bold,
                fontSize: 30,
              ),
            ),
            const SizedBox(
              height: 20,
            ),
            Text(
              "PATHFINDER ile tur rehberini bulabilirsin !",
              textAlign: TextAlign.center,
              style: TextStyle(
                color: Colors.grey[700],
                fontSize: 15,
              ),
            )
          ],
        ),
        Container(
          height: MediaQuery.of(context).size.height / 3,

```

```

        decoration: const BoxDecoration(
          image: DecorationImage(
            image: AssetImage("assets/images/welcome.png"))
        ),
      ),
      Column(
        children: <Widget>[
          MaterialButton(
            minWidth: double.infinity,
            height: 60,
            onPressed: () {
              Navigator.push(context,
                MaterialPageRoute(builder: (context) => Login
Page()));
            },
            shape: RoundedRectangleBorder(
              side: const BorderSide(color: Colors.black),
              borderRadius: BorderRadius.circular(50)),
            child: const Text(
              "Giriş yap",
              style:
                TextStyle(fontWeight: FontWeight.w600, fontSi
ze: 18),
            ),
          ),
        ],
      ),

      const SizedBox(height: 20),
      MaterialButton(
        minWidth: double.infinity,
        height: 60,
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => SignupPage()));
        },
      ),
    ],
  ),
),
);

```

```

        },
        color: const Color(0xff0095FF),
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(50)),
        child: const Text(
            "Üye ol",
            style: TextStyle(
                color: Colors.white,
                fontWeight: FontWeight.w600,
                fontSize: 18),
        ),
    ),
],
),
],
),
),
),
),
);
}
}

```

Şekil 3.4 Kullanıcı Karşılama Ekranı Kodları

### 3.1.4. Üye Ol Sayfasının Oluşturulması

Kullanıcı uygulamayı ilk kez indirmiş ve üye kaydı oluşturmamış Üye Ol sayfasında kullanıcı hesap oluştururken kullanıcı adı, e-mail adresi, parola bilgilerini ilgili alanlara girmektedir. Tüm bilgileri kullanıcı doğru ve eksiksiz bir şekile doldurulur. Girilen bu bilgiler kayıtOl fonksiyonu ile veritabanının otomatik atamış olduğu kullanıcıID ile birlikte veritabanındaki ilgili alana eklenmektedir. Kullanıcının Üye Ol butonuna tıklamasının ardından kullanıcı Kullanıcı Karşılama Ekranı sayfasına yönlendiriliyor. Sayfanın



arayüz görseli ve arka plan kodları aşağıdaki gibidir.



Şekil 3.5 Üye Ol Sayfası

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:pathfinder/screen/welcome.dart';

// ignore: must_be_immutable
class SignupPage extends StatelessWidget {
  final passTextController = TextEditingController();
  final mailTextController = TextEditingController();
  final userNameTextController = TextEditingController();

  FirebaseAuth auth = FirebaseAuth.instance;

  SignupPage({Key? key}) : super(key: key);
```

```

Future<void> kayitOl(BuildContext context) async {
    await FirebaseAuth.instance
        .createUserWithEmailAndPassword(
            email: mailTextController.text, password: passTextController
r.text)
        .then((kullanici) {
            FirebaseFirestore.instance
                .collection("Kullanicilar")
                .doc(mailTextController.text)
                .set({
                    'kullaniciID': auth.currentUser!.uid,
                    "KullaniciEposta": mailTextController.text,
                    "KullaniciSifre": passTextController.text,
                    "KullaniciAdi": userNameTextController.text,
                }).then((kullanici) {
                    Navigator.pushAndRemoveUntil(
                        context,
                        MaterialPageRoute(builder: (context) => const WelcomePage()
),
                        (route) => false);
                });
            });
    });
}

@override
Widget build(BuildContext context) {
    return Scaffold(
        resizeToAvoidBottomInset: true,
        backgroundColor: Colors.white,
        appBar: AppBar(
            elevation: 0,
            brightness: Brightness.light,
            backgroundColor: Colors.white,

```

```

leading: IconButton(
  onPressed: () {
    Navigator.pop(context);
  },
  icon: const Icon(
    Icons.arrow_back_ios,
    size: 20,
    color: Colors.black,
  ),
),
),
body: SingleChildScrollView(
  child: Container(
    padding: const EdgeInsets.symmetric(horizontal: 40),
    height: MediaQuery.of(context).size.height - 50,
    width: double.infinity,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Column(
          children: <Widget>[
            const Text(
              "Üye ol",
              style: TextStyle(
                fontSize: 30,
                fontWeight: FontWeight.bold,
              ),
            ),
            const SizedBox(
              height: 20,
            ),
            Text(
              "Ücretsiz hesap oluşturun",

```

```

        style: TextStyle(fontSize: 15, color: Colors.grey[700]),
      ),
    ],
  ),
  Column(
    children: <Widget>[
      inputFile(
        label: "Kullanıcı adı",
        controller: userNameTextController),
      inputFile(label: "E-
mail", controller: mailTextController),
      inputFile(
        label: "Parola",
        obscureText: true,
        controller: passTextController),
      inputFile(label: "Parolayı tekrarla ", obscureText: t
true),
    ],
  ),
  Container(
    padding: const EdgeInsets.only(top: 3, left: 3),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(50),
      border: const Border(
        bottom: BorderSide(color: Colors.black),
        top: BorderSide(color: Colors.black),
        left: BorderSide(color: Colors.black),
        right: BorderSide(color: Colors.black),
      )),
    child: MaterialButton(
      minWidth: double.infinity,
      height: 60,
      onPressed: () {
        kayitOl(context);

```

```

        },
        color: const Color(0xff0095FF),
        elevation: 0,
        shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(50),
        ),
        child: const Text(
            "Üye ol",
            style: TextStyle(
                fontWeight: FontWeight.w600,
                fontSize: 18,
                color: Colors.white,
            ),
        ),
    ),
),
Row(
    mainAxisAlignment: MainAxisAlignment.center,
    // ignore: prefer_const_literals_to_create_immutables
    children: <Widget>[
        const Text("Zaten bir hesabın var mı ? "),
        const Text(
            " Giriş yap",
            style: TextStyle(fontWeight: FontWeight.w600, fontS
ize: 18),
        ),
    ],
),
],
),
),
),
);
}

```

```

}

Widget inputFile({label, obscureText = false, controller}) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: <Widget>[
      Text(
        label,
        style: const TextStyle(
          fontSize: 15, fontWeight: FontWeight.w400, color: Colors.bl
ack87),
      ),
      const SizedBox(
        height: 5,
      ),
      TextFormField(
        controller: controller,
        obscureText: obscureText,
        decoration: const InputDecoration(
          contentPadding: EdgeInsets.symmetric(vertical: 0, horizonta
l: 10),
          enabledBorder: OutlineInputBorder(
            borderSide: BorderSide(color: Colors.grey),
          ),
          border:
            OutlineInputBorder(borderSide: BorderSide(color: Colors
.grey))),
      ),
      const SizedBox(
        height: 10,
      )
    ],
  );
}

```

Şekil 3.6 Üye Ol Sayfası Kodları

### 3.1.5. Giriş Yap Sayfasının Oluşturulması

Kullanıcı ilk login safyasından Giriş Yap sayfasına yönlendirilmektedir. Buradaki kodlarda yine aşağıdaki gibidir.



Şekil 3.7 Giriş Yap Sayfası

```
// ignore_for_file: sized_box_for_whitespace
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:pathfinder/screen/home_page.dart';
// ignore: use_key_in_widget_constructors
class LoginPage extends StatelessWidget {
  final passTextController = TextEditingController();
  final mailTextController = TextEditingController();

  girisYap(BuildContext context) async {
    await FirebaseAuth.instance
      .signInWithEmailAndPassword(
```

```

        email: mailTextController.text, password: passTextController.text)

        .then((kullanici) {
            Navigator.pushAndRemoveUntil(
                context,
                MaterialPageRoute(builder: (context) => const AnaSayfa()),
                (route) => false);
        });
    }

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            resizeToAvoidBottomInset: false,
            backgroundColor: Colors.white,
            appBar: AppBar(
                elevation: 0,
                brightness: Brightness.light,
                backgroundColor: Colors.white,
                leading: IconButton(
                    onPressed: () {
                        Navigator.pop(context);
                    },
                    icon: const Icon(
                        Icons.arrow_back_ios,
                        size: 20,
                        color: Colors.black,
                    ),
                ),
            ),
            body: Container(
                height: MediaQuery.of(context).size.height,
                width: double.infinity,
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,

```



```

children: <Widget>[
  Expanded(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: <Widget>[
        Column(
          children: <Widget>[
            const Text(
              "Giriş yap",
              style:
                TextStyle(fontSize: 30, fontWeight: FontWeigh
t.bold),
            ),
            const SizedBox(
              height: 20,
            ),
            Text(
              "Hesabın ile giriş yap",
              style: TextStyle(fontSize: 15, color: Colors.grey
[700]),
            )
          ],
        ),
        Padding(
          padding: const EdgeInsets.symmetric(horizontal: 40),
          child: Column(
            children: <Widget>[
              inputFile(
                label: "E-
mail", controller: mailTextController),
              inputFile(
                label: "Parola",
                obscureText: true,
                controller: passTextController)
            ],

```

```

    ),
  ),
  Padding(
    padding: const EdgeInsets.symmetric(horizontal: 40),
    child: Container(
      padding: const EdgeInsets.only(top: 3, left: 3),
      decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(50),
        border: const Border(
          bottom: BorderSide(color: Colors.black),
          top: BorderSide(color: Colors.black),
          left: BorderSide(color: Colors.black),
          right: BorderSide(color: Colors.black),
        )),
      child: MaterialButton(
        minWidth: double.infinity,
        height: 60,
        onPressed: () {
          girisYap(context);
        },
        color: const Color(0xff0095FF),
        elevation: 0,
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(50),
        ),
        child: const Text(
          "Giriş yap",
          style: TextStyle(
            fontWeight: FontWeight.w600,
            fontSize: 18,
            color: Colors.white,
          ),
        ),
      ),
    ),
  ),
),

```

```

        ),
      ),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,
        // ignore: prefer_const_literals_to_create_immutables
        children: <Widget>[
          const Text("Bir hesabın yok mu ? "),
          const Text(
            " Üye ol",
            style: TextStyle(
              fontWeight: FontWeight.w600,
              fontSize: 18,
            ),
          ),
        ],
      ),
    ],
  ),
  Container(
    padding: const EdgeInsets.only(top: 100),
    height: 200,
    decoration: const BoxDecoration(
      image: DecorationImage(
        image: AssetImage("assets/images/login.png"),
        fit: BoxFit.fitHeight),
    ),
  ),
],
))
],
),
),
);
}
}

Widget inputFile({label, obscureText = false, controller}) {

```

```

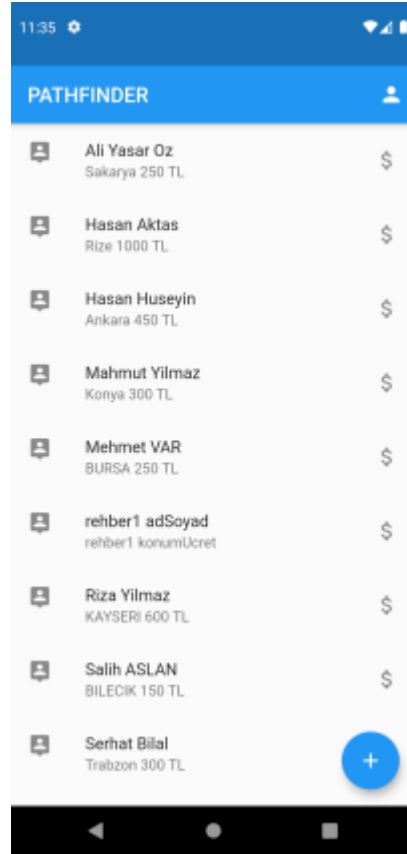
return Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: <Widget>[
    Text(
      label,
      style: const TextStyle(
        fontSize: 15, fontWeight: FontWeight.w400, color: Colors.bl
ack87),
    ),
    const SizedBox(
      height: 5,
    ),
    TextField(
      controller: controller,
      obscureText: obscureText,
      decoration: const InputDecoration(
        contentPadding: EdgeInsets.symmetric(vertical: 0, horizontal: 10),
        enabledBorder: OutlineInputBorder(
          borderSide: BorderSide(color: Colors.grey),
        ),
        border:
          OutlineInputBorder(borderSide: BorderSide(color: Colors
.grey))),
    ),
    const SizedBox(
      height: 10,
    )
  ],
);
}

```

Şekil 3.8 Giriş Yap Sayfası Kodları

### 3.1.5. Ana Sayfasının Oluşturulması

Üye olmuş olan kullanıcı Üye Giriş sayfasından Ana sayfaya yönlendiriliyor. Bu sayfada kullanıcılar verilmiş olan ilanları inceleyebilecekleri gibi sayfanın alt kısmında bulunan mavi ekleme ikonu ile de ilan verebilecekleri sayfaya yönlendirilebilirler. Arayüz görseli ve kodu aşağıdaki gibidir.



Şekil 3.9 Ana Sayfa Arayüzü

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:pathfinder/screen/ilangir.dart';
import 'package:pathfinder/screen/profil.dart';

class AnaSayfa extends StatelessWidget {
  const AnaSayfa({Key? key}) : super(key: key);
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text("PATHFINDER"),
      actions: <Widget>[
        IconButton(
          onPressed: () {
            Navigator.pushReplacement(
              context,
              MaterialPageRoute(
                builder: (context) => const ProfilSayfasi()));
          },
          icon: const Icon(Icons.person))
      ],
    ),
    body: const UserInformation(),
    floatingActionButton: FloatingActionButton(
      child: const Icon(Icons.add),
      onPressed: () {
        Navigator.pushAndRemoveUntil(
          context,
          MaterialPageRoute(builder: (context) => IlanGir()),
          (route) => false);
      },
    ),
  );
}

class UserInformation extends StatefulWidget {
  const UserInformation({Key? key}) : super(key: key);

```

```

@override
_UserInformationState createState() => _UserInformationState();
}

class _UserInformationState extends State<UserInformation> {
  final Stream<QuerySnapshot> _usersStream =
    FirebaseFirestore.instance.collection('Rehberler').snapshots();

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: _usersStream,
      builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
        if (snapshot.hasError) {
          return const Text('Bir şeyler ters gitti');
        }

        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Text("Yukleniyor");
        }

        return ListView(
          children: snapshot.data!.docs.map((DocumentSnapshot document)
{
            Map<String, dynamic> data =
              document.data()! as Map<String, dynamic>;
            return ListTile(
              leading: const Icon(Icons.person_pin_rounded),
              trailing: const Icon(Icons.attach_money),
              title: Text(data['adSoyad']),
              subtitle: Text(data['konumUcret']),
            );
          }).toList(),

```

```

    );
  },
);
}
}

```

Şekil 3.10 Ana Sayfa Kodları

Ana sayfanın tasarımı için yazılmış olan kod blokları Home\_page.dart isimli dosyanın içerisinde saklanır.

### 3.1.6. Ücretsiz İlan Ver Sayfasının Oluşturulması

İlan vermek için ilgili sayfadan yönlendirilen kullanıcılar görseli aşağıda paylaşılan Ücretsiz İlan Ver sayfasına geçiş yaparlar. Bu kısımda yer alan rehberin adı soyadı tur rehberliği yapmış olduğu bölge ve ücreti yer bilgileri kullanıcı tarafından girilir. Bir kullanıcı birden fazla ilan verebileceği için bu kısımda yer alan ad soyad verisini kullanıcı giriş hesabından çekilmedi. Arayüz görseli ve kodu aşağıdaki gibidir.

11:35

Ücretsiz İlan Ver

Ad Soyad

Hilal NARMAN

Konum Ücret

SAKARYA 700

Ücretsiz Rehber İlanları

Gönder

Şekil 3.11 Ücretsiz İlan Ver Sayfasının Arayüzü



```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'package:pathfinder/screen/home_page.dart';

// ignore: use_key_in_widget_constructors
class IlanGir extends StatelessWidget {
  final adSoyadTextController = TextEditingController();
  final konumUcretTextController = TextEditingController();

  ilanGonder(BuildContext context) {
    FirebaseFirestore.instance
      .collection('Rehberler')
      .doc(adSoyadTextController.text)
      .set({
        'adSoyad': adSoyadTextController.text,
        'konumUcret': konumUcretTextController.text
      }).then((value) {
        Navigator.pushAndRemoveUntil(
          context,
          MaterialPageRoute(builder: (_) => const AnaSayfa()),
          (route) => false);
      });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: true,
      backgroundColor: Colors.white,
      appBar: AppBar(
        elevation: 0,
        brightness: Brightness.light,
        backgroundColor: Colors.white,
      ),
      body: SingleChildScrollView(

```

```

child: Container(
  padding: const EdgeInsets.symmetric(horizontal: 40),
  height: MediaQuery.of(context).size.height - 50,
  width: double.infinity,
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: <Widget>[
      Column(
        // ignore: prefer_const_literals_to_create_immutables
        children: <Widget>[
          const Text(
            "Ücretsiz İlan Ver",
            style: TextStyle(
              fontSize: 30,
              fontWeight: FontWeight.bold,
            ),
          ),
          const SizedBox(
            height: 20,
          ),
        ],
      ),
      Column(
        children: <Widget>[
          inputFile(
            label: "Ad Soyad", controller: adSoyadTextControl
ler),
          inputFile(
            label: "Konum Ücret",
            controller: konumUcretTextController),
        ],
      ),
      Row(
        mainAxisAlignment: MainAxisAlignment.center,

```

```

// ignore: prefer_const_literals_to_create_immutables
children: <Widget>[
  const Text(
    "Ücretsiz Rehber İlanları",
    style: TextStyle(fontWeight: FontWeight.w600, fontS
ize: 18),
  ),
],
),
MaterialButton(
  minWidth: double.infinity,
  height: 60,
  onPressed: () {
    ilanGonder(context);
  },
  color: const Color(0xff0095FF),
  elevation: 0,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(50),
  ),
  child: const Text(
    "Gönder",
    style: TextStyle(
      fontWeight: FontWeight.w600,
      fontSize: 18,
      color: Colors.white,
    ),
  ),
),
],
),
),
),
);

```

```

    }
  }
Widget inputFile({label, obscureText = false, controller}) {
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: <Widget>[
      Text(
        label,
        style: const TextStyle(
          fontSize: 15, fontWeight: FontWeight.w400, color: Colors.bl
ack87),
      ),
      const SizedBox(
        height: 5,
      ),
      TextFormField(
        controller: controller,
        obscureText: obscureText,
        decoration: const InputDecoration(
          contentPadding: EdgeInsets.symmetric(vertical: 0, horizontal: 10),
          enabledBorder: OutlineInputBorder(
            borderSide: BorderSide(color: Colors.grey),
          ),
          border:
            OutlineInputBorder(borderSide: BorderSide(color: Colors
.grey))),
      ),
      const SizedBox(
        height: 10,
      )
    ],
  );
}

```

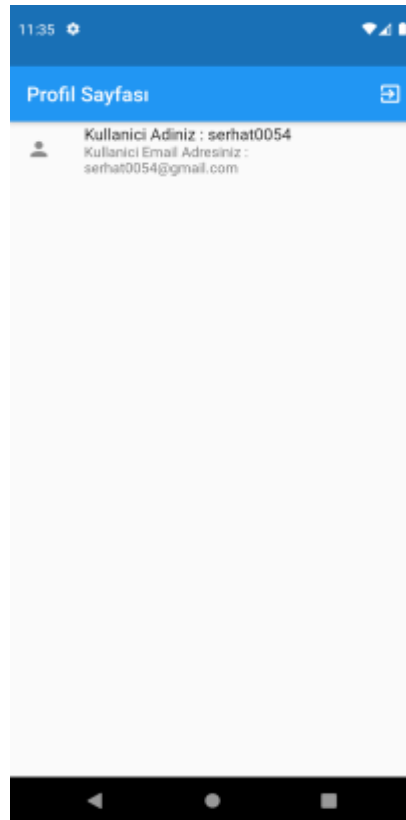
Şekil 3.11 Ücretsiz İlan Ver Sayfasının Kodları

Kullanıcının ücretsiz ilan vermesini sağlayan sayfanın kod blokları `ilangir.dart` isimli dosyanın içerisinde saklanır. Bu kısımda yer alan `ilanGonder` fonksiyonu ile `textBox`'lar içerisinde yazılan ilan bilgilerini Firebase veri tabanına kaydetmiş olunuyor. Bu konu veritabanı kısmında tekrar ele alınacaktır.

### 3.1.6. Profil Sayfasının Oluşturulması

Uygulamaya giriş yapmış olan kullanıcı hesabı ile ilgili kullanıcı adı ve e posta kayıt esnasında girmiş olduğu bilgileri kullanıcıID kullanılarak Firebase veritabanında çekilerek Profil Sayfasında görüntüleyebilecektir. Ayrıca Profil sayfasında AppBar'ın sağ tarafında bulunan icon ile kullanıcı uygulamadan çıkış yaparak, kullanıcı giriş ekranına yönlendirilmektedir.

Sayfanın arayüz görseli ve kod bilgileri aşağıdadır.



Şekil 3.12 Profil Sayfasının Arayüzü

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'login.dart';

class ProfilSayfasi extends StatelessWidget {
  const ProfilSayfasi({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Profil Sayfası"),
        actions: <Widget>[
          IconButton(
            onPressed: () {
              FirebaseAuth.instance.signOut().then((value) =>
                Navigator.pushAndRemoveUntil(
                  context,
                  MaterialPageRoute(builder: (_) => LoginPage()),
                  (route) => false));
            },
            icon: const Icon(Icons.exit_to_app))
        ],
      ),
      // ignore: avoid_unnecessary_containers
      body: Container(
        child: const KullaniciBilgi(),
      ),
    );
  }
}

```

```

class KullaniciBilgi extends StatelessWidget {
  const KullaniciBilgi({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    FirebaseAuth kullanici = FirebaseAuth.instance;

    Query users = FirebaseFirestore.instance
      .collection('Kullanicilar')
      .where('kullaniciID', isEqualTo: kullanici.currentUser!.uid);
    return StreamBuilder<QuerySnapshot>(
      stream: users.snapshots(),
      builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
        if (snapshot.hasError) {
          return const Text('Bir şeyler ters gitti');
        }

        if (snapshot.connectionState == ConnectionState.waiting) {
          return const Text("Yukleniyor");
        }

        return ListView(
          children: snapshot.data!.docs.map((DocumentSnapshot document)
{
            Map<String, dynamic> data =
              document.data()! as Map<String, dynamic>;
            return ListTile(
              leading: const Icon(Icons.person),
              title: Text("Kullanici Adiniz : " + data['KullaniciAdi']),
              subtitle: Text(
                "Kullanici Email Adresiniz : " + data['KullaniciEposta'],
              ),
            );
          },
        );
      },
    );
  }
}

```

```

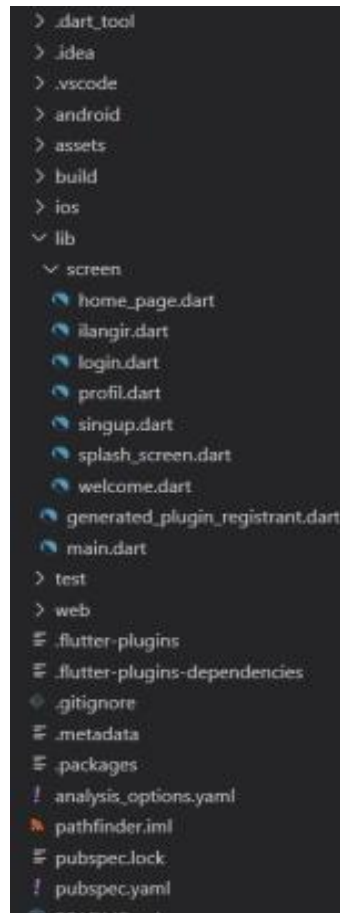
        }).toList(),
    );
  },
);
}
}

```

Şekil 3.13 Profil Sayfasının Kodları

## 3.2. Program Dosya Yerleşim Yapısı ve Bağlantıları

### 3.2.1. Flutter Dosya Yapısı



Şekil 3.14 Flutter Dosya Yapısı

Üst kısımda bulunan resimde oluşturmuş olduğumuz assets/images klasörünün içerisinde uygulamada kullanmış olduğumuz resimler yer almaktadır. Bu dosyaların uzantısı pubspec.yaml kısmından ayrıca belirtilerek uygulamanın resimlere erişim izni



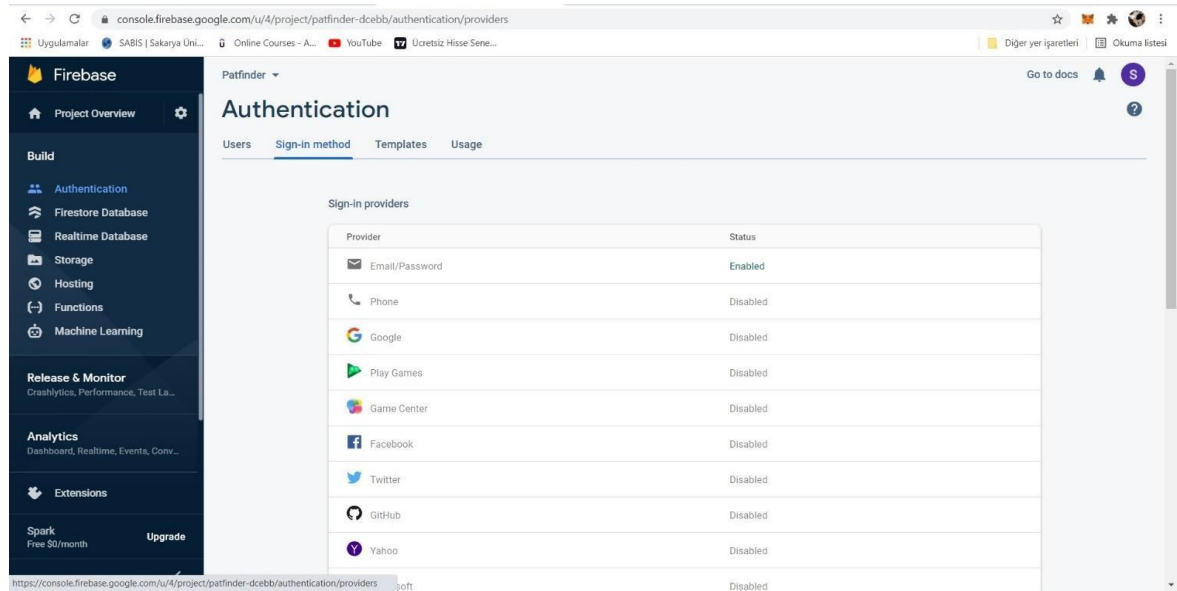
verilmiştir.

Lib klasörü içerisine oluşturmuş olduğumuz screen klasörünün içerisine uygulamada şimdiye kadar oluşturmuş olduğumuz sayfalar yer almaktadır. Bu klasör içerisinde bulunan home\_page, login, singup, splash\_screen, welcome sayfaları bulunmaktadır. Aynı klasör içerisinde yer almakta olan generated\_plugin\_registrant ise uygulamanın Firebase ile bağlantısını yaptıktan sonra otomatik olarak oluşmuştur.

Projenin diğer aşamalarında ilk olarak firebase veritabanına aktarılması daha sonra ise tur rehberi profillerinin oluşturularak sırasıyla home\_page ekranında listelenmesi olacaktır.

### 3.2.2. Firebase Veritabanı Özellikleri

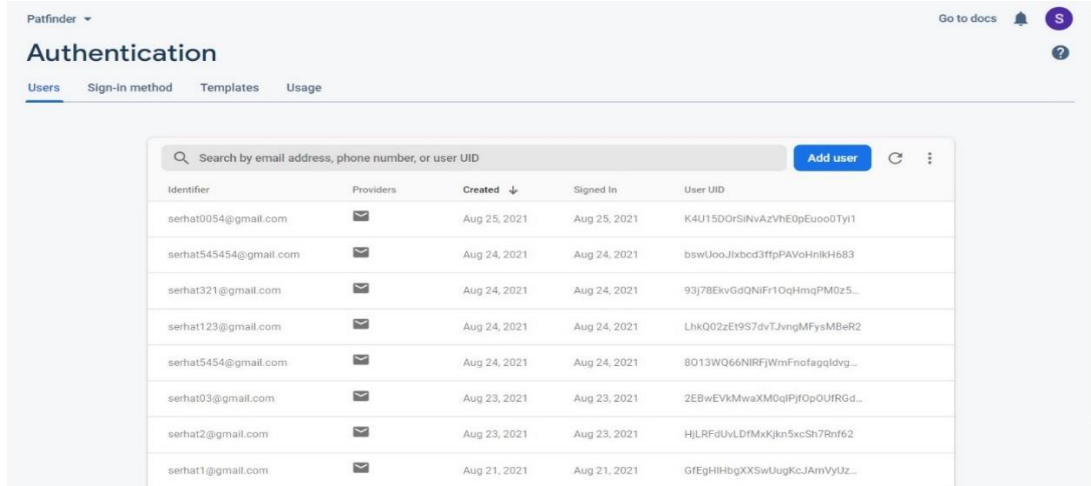
Uygulamaya Üye Ol sayfasından kayıt yaptıran kullanıcının girmiş olduğu bilgilerden Email ve parola bilgileri ile giriş yapabilmesini sağlayan görseldeki gibi bir metod tanımlandı. Görseli ve kodu aşağıdadır.



Şekil 3.15 Firebase Kayıt Yapısı

Tanımlanmış olan metod sonucunda kullanıcının kayıt esnasında girmiş olduğu bilgiler veri tabanında aşağıdaki görselde görüldüğü gibi kullanıcıAdı,

kullaniciEposta,kullaniciSifre ve tanımlanan metod sonucunda kullanıcıya atanan uniq kullanıcıID şeklinde kayıtlar tutulmaktadır.



Patfinder ▾ Go to docs 🔔 S ?

## Authentication

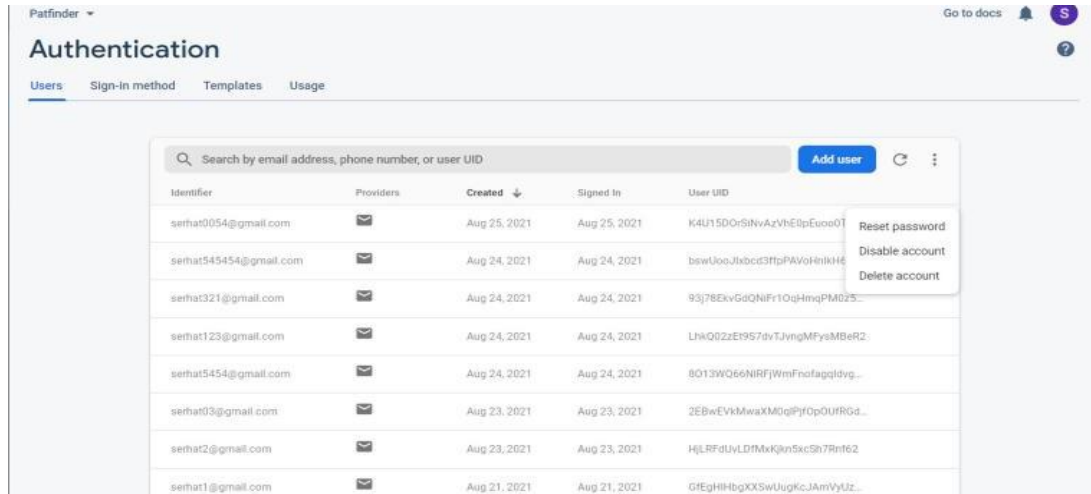
Users Sign-in method Templates Usage

Search by email address, phone number, or user UID Add user ↻ ⋮

Identifier	Providers	Created ↓	Signed in	User UID
serhat0054@gmail.com	✉	Aug 25, 2021	Aug 25, 2021	K4U15DOrSiNvAzVhE0pEuoo0Ty1
serhat545454@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	bswUooJlxbcd3ftpPAVoHnIkH683
serhat321@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	93j78EkvGdQNiFr1OqHmqPM0z5...
serhat123@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	LhkQ02zEt9S7dvTJvngMFysMBer2
serhat5454@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	8O13WQ66NIRFjWmFnofagqldvg...
serhat03@gmail.com	✉	Aug 23, 2021	Aug 23, 2021	2EBwEVkMwaXM0qIPjOpOUIRGd...
serhat2@gmail.com	✉	Aug 23, 2021	Aug 23, 2021	HjLRFdUvLDFmXKjn5xcSh7Rnt62
serhat1@gmail.com	✉	Aug 21, 2021	Aug 21, 2021	GEgHHbqXXSwUugKcJAmVvYUz...

Şekil 3.16 Firebase Kayıt Yapısı

Görselde de görüldüğü gibi kullanıcıların girmiş olduğu bilgileri yanısıra kullanıcının uygulamaya kayıt tarihi, uygulamaya en son girmiş olduğu tarih , veritabanının atamış olduğu ID bilgisi tutulmaktadır. Bu bilgilerin haricinde admin bu kısımdan kullanıcı ekleyebileceği gibi var olan kullanıcıyı engelleyebilme işlemlerini de gerçekleştirebilecektir.



Patfinder ▾ Go to docs 🔔 S ?

## Authentication

Users Sign-in method Templates Usage

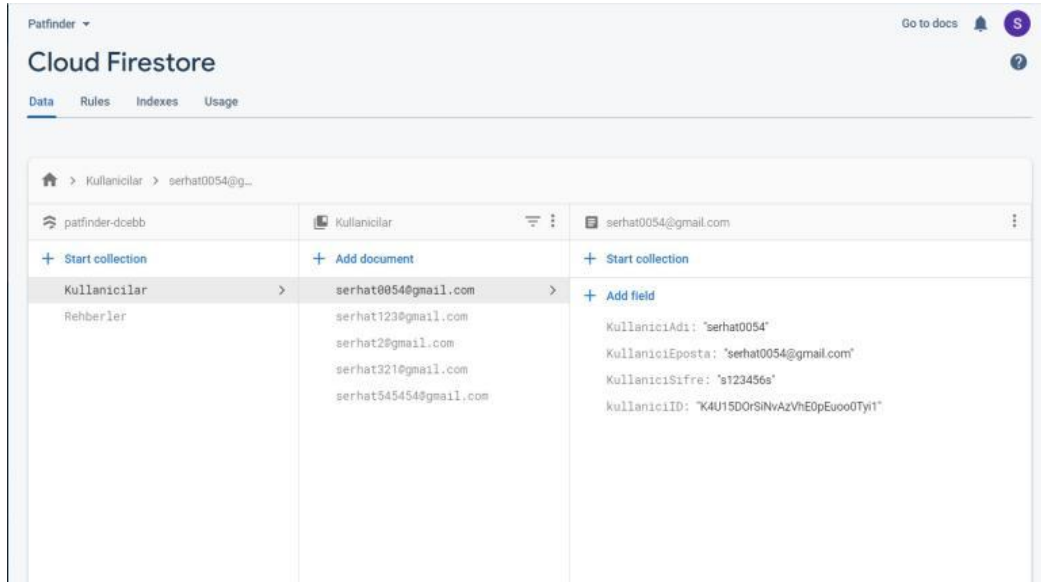
Search by email address, phone number, or user UID Add user ↻ ⋮

Identifier	Providers	Created ↓	Signed in	User UID
serhat0054@gmail.com	✉	Aug 25, 2021	Aug 25, 2021	K4U15DOrSiNvAzVhE0pEuoo0Ty1
serhat545454@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	bswUooJlxbcd3ftpPAVoHnIkH683
serhat321@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	93j78EkvGdQNiFr1OqHmqPM0z5...
serhat123@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	LhkQ02zEt9S7dvTJvngMFysMBer2
serhat5454@gmail.com	✉	Aug 24, 2021	Aug 24, 2021	8O13WQ66NIRFjWmFnofagqldvg...
serhat03@gmail.com	✉	Aug 23, 2021	Aug 23, 2021	2EBwEVkMwaXM0qIPjOpOUIRGd...
serhat2@gmail.com	✉	Aug 23, 2021	Aug 23, 2021	HjLRFdUvLDFmXKjn5xcSh7Rnt62
serhat1@gmail.com	✉	Aug 21, 2021	Aug 21, 2021	GEgHHbqXXSwUugKcJAmVvYUz...

Reset password  
Disable account  
Delete account

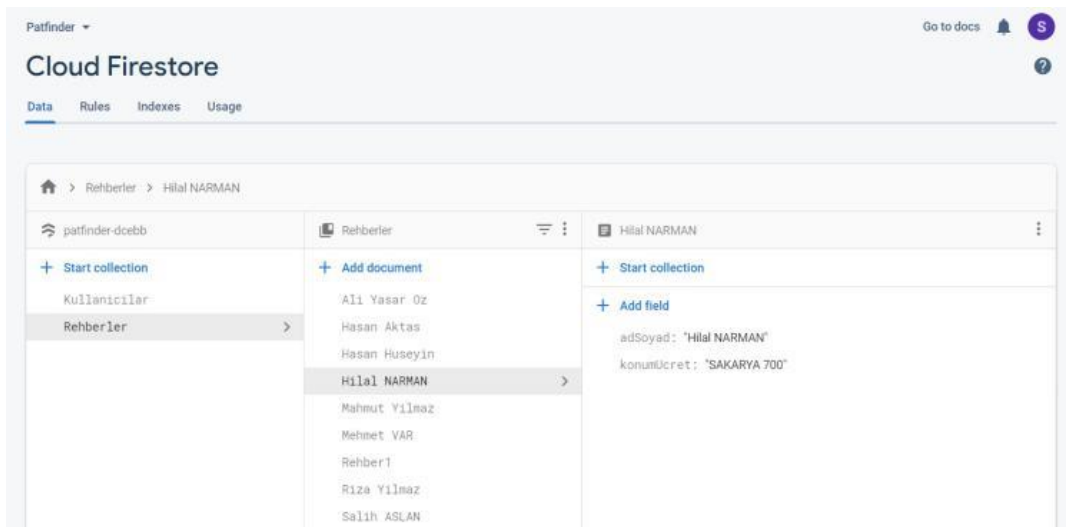
Şekil 3.17 Firebase Kayıt Yapısı

Kayıt yapılan herkes kullanıcı koleksiyonunda yer almaktadır. Koleksiyonun görseli aşağıdadır.



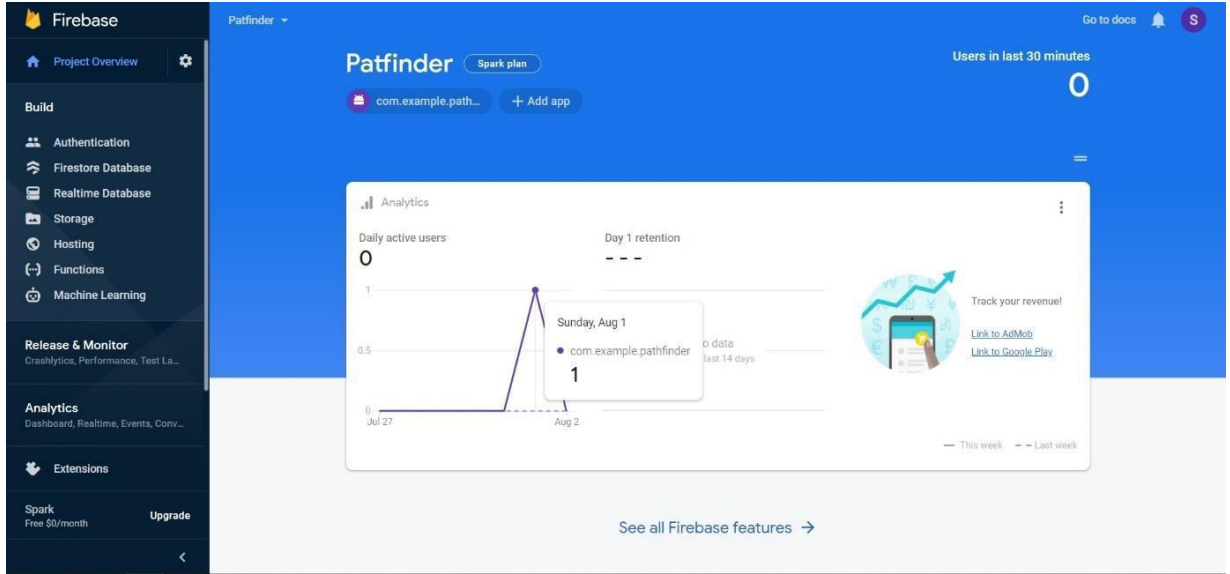
Şekil 3.18 Firebase Kullanıcı Koleksiyon Yapısı

Veri tabanına kayıt olan kullanıcılar Ücretsiz İlan Sayfasından ilan oluşturması sonucunda kullanıcının bu kısımda girmiş olduğu adsoyad ve konum ücreti rehberler koleksiyonunun içerisinde tutulur.



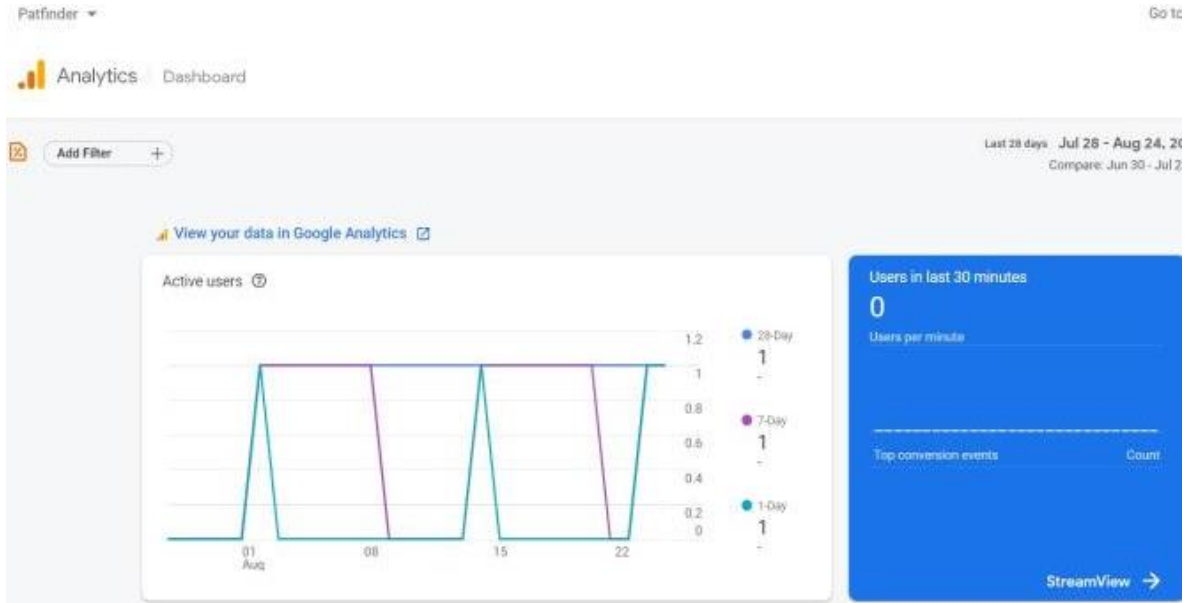
Şekil 3.19 Firebase Rehber Koleksiyon Yapısı

Uygulamanın analizleri Firebase veritabanında tutulmaktadır.



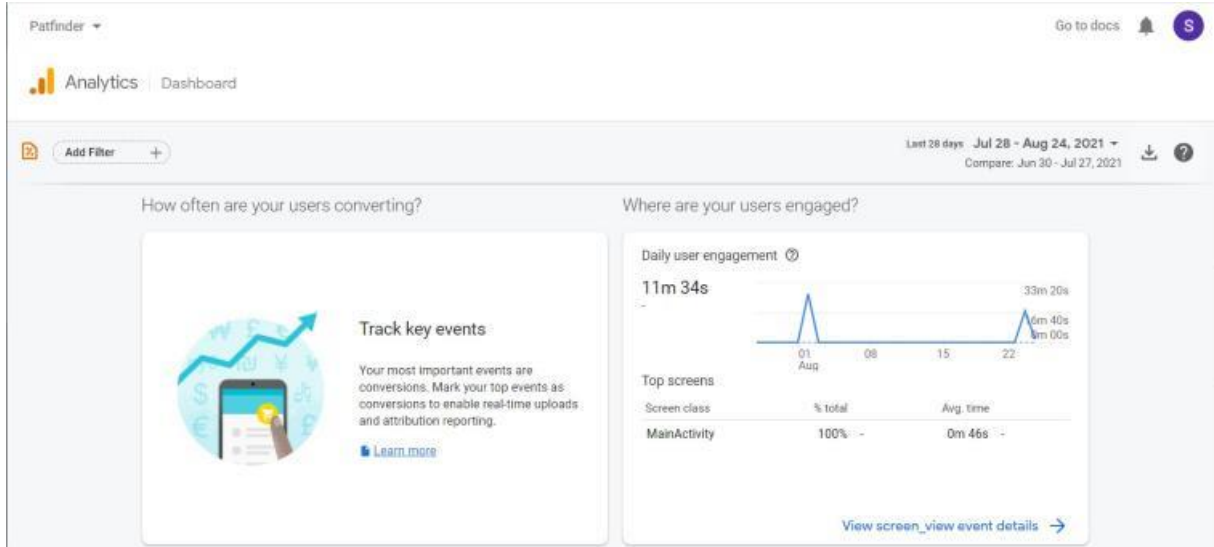
Şekil 3.20 Firebase Analiz Yapısı

1 günlük, 7 günlük ve 28 günlük aktif kullanıcılar, zaman içinde çizilir. Özet değerler (sağda), tarih aralığının son günü itibariyle etkin kullanıcı sayısını gösterir.



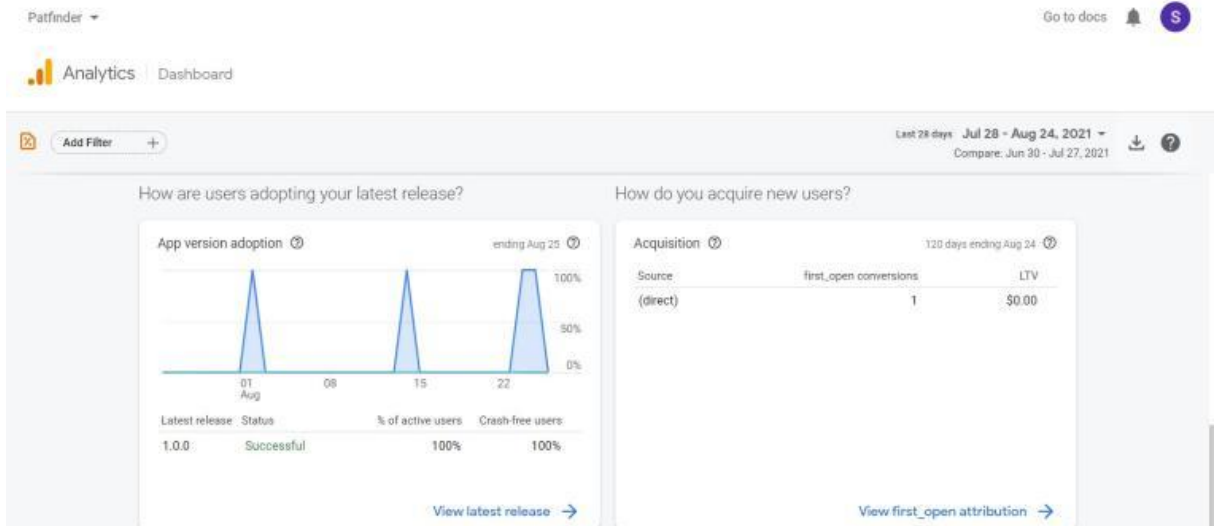
Şekil 3.21 Firebase Analiz Yapısı

Kullanıcıların anlık olarak hangi ekranları kullandıklarının analiz grafiği;

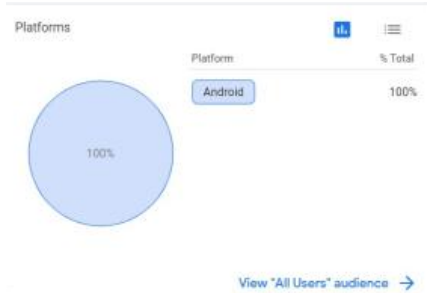


Şekil 3.22 Firebase Analiz Yapısı

Uygulamanın sürümlerinin kullanım analizi;



Uygulamanın kullanıldığı cihazların işletim sistemleri ile ilgili analizler;



## BÖLÜM 4 SONUÇLAR VE ÖNERİLER

Projemiz şu anda canlı ortamda çalışmaktadır ve kaynak kodları tamamen açık şekilde Youtube’da paylaşılmıştır.

Uygulama video linki: [https://youtu.be/A0\\_aJ7o9oxM](https://youtu.be/A0_aJ7o9oxM)

Bu proje, aşağıdaki özelliklerle geliştirilebilir:

- İlan Giriş Sayfası

Rehberin vermiş olduğu ilanlar bölgesel olarak farklı bir olarak belirtilebilir.

İlan bilgilerinin içerisinde harita konum bilgisi yer alabilir.

Rehberlerin vermiş olduğu ilanlar filtrelenerek görüntülenebilir.

- Profil Sayfası

Profil sayfasına profil resimleri eklenebilir.

Rehberlerin vermiş olduğu tüm ilanlar toplu halde görüntülenebilir.

- Satın Alma Sayfası

Kullanıcılar ilanların içerisinde uygun olanları satın alabilmeleri için bir satın alma sayfası oluşturulabilir.

Projenin geleceğinde ise:

- Uygulama playstore ve App store da kullanıcıların hizmetine sunulabilir.
- Sosyal medya platformlarında reklam verilerek kullanıcı sayısı artırılabilir.
- T.C. Kültür ve Turizm Bakanlığı ile anlaşma imzalanarak bakanlığa bağlı olan tüm tur rehberleri uygulamaya dahil edilerek tek bir merkezde tur rehberleri buluşturularak çok daha geniş kapsamlı hizmetler sunulabilir.

## KAYNAKLAR

[1]	Rehberliğin Tarihçesi, <a href="https://iro.org.tr/tr/53-page-rehberligin-tarihcesi.aspx">https://iro.org.tr/tr/53-page-rehberligin-tarihcesi.aspx</a>
[2]	Flutter, <a href="https://flutter.dev/">https://flutter.dev/</a>
[3]	İnternetin tarihi, 31 Mayıs 2021 <a href="https://tr.wikipedia.org/wiki/%C4%B0internetin_tarihi">https://tr.wikipedia.org/wiki/%C4%B0internetin_tarihi</a>
[4]	Telefon: Kim, Ne Zaman İcat Etti? <a href="https://bilimgenc.tubitak.gov.tr/makale/telefon-icadi#:~:text=1973%20y%C4%B1l%C4%B1nda%20Motorola%20%C5%9Firketinde%20%C3%A7al%C4%B1%C5%9Fan,ile%20yaln%C4%B1zca%2035%20dakika%20kullan%C4%B1labiliyordu.">https://bilimgenc.tubitak.gov.tr/makale/telefon-icadi#:~:text=1973%20y%C4%B1l%C4%B1nda%20Motorola%20%C5%9Firketinde%20%C3%A7al%C4%B1%C5%9Fan,ile%20yaln%C4%B1zca%2035%20dakika%20kullan%C4%B1labiliyordu.</a>
[5]	KVKK, Kişisel Verilerin Korunumu Kanunu ve Uygulaması <a href="https://www.kvkk.gov.tr/yayinlar/K%C4%B0%C5%9E%C4%B0SEL%20VER%C4%B0LER%C4%B0N%20KORUNMASI%20KANUNU%20VE%20UYGULAMASI.pdf">https://www.kvkk.gov.tr/yayinlar/K%C4%B0%C5%9E%C4%B0SEL%20VER%C4%B0LER%C4%B0N%20KORUNMASI%20KANUNU%20VE%20UYGULAMASI.pdf</a>
[6]	Web ve Mobil Uygulamalar için Firebase <a href="https://devnot.com/2017/web-ve-mobil-uygulamalar-icin-firebase/">https://devnot.com/2017/web-ve-mobil-uygulamalar-icin-firebase/</a>
[7]	Google Firebase Nedir ?,2018 <a href="https://medium.com/furkanpacikgoz/google-firebase-nedir-ae013e495a74">https://medium.com/furkanpacikgoz/google-firebase-nedir-ae013e495a74</a>
[8]	Firebase <a href="https://firebase.google.com/?hl=tr">https://firebase.google.com/?hl=tr</a>
[9]	Dart Programlama Dili,2019 <a href="https://halilozel1903.medium.com/dart-programlama-dili-eeafb64ad300">https://halilozel1903.medium.com/dart-programlama-dili-eeafb64ad300</a>

## **EKLER**

### **ÖZGEÇMİŞ**

Serhat BİLAL 06.01.1992’de Sakarya’da doğdu. İlk ve Orta eğitimini Pamukova Fatmahanım ilkokulun ’da tamamladı. 2010 lise eğitimini Pamukova Endüstri Meslek Lisesi Bilgisayar Teknolojileri bölümünde tamamladı. 2013 yılında Mustafa Kemal Üniversitesi Harbiye MYO Bilişim Teknolojileri bölümünü bitirdi. Ardından 2016 yılında Anadolu Üniversitesi Kamu Yönetimi bölümünü bitirdi. 2019 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı. Aynı zamanda Eskişehir Anadolu Üniversitesinde Adalet bölümünü okumaktadır. Yazılım stajını 2021 haziran ayında HAVELSAN A.Ş de yapmıştır.

Hilal NARMAN 02.02.1993’de Bayburt’ta doğdu. İlk, orta ve lise eğitimini Sakarya’da tamamladı. 2011 yılında Adapazarı merkezde bulunan Hacı Zehra Akkoç Kız Lisesi’nden mezun oldu.2011 yılında kazandığı Kocaeli Üniversitesi Fen Edebiyat Fakültesi Matematik Bölümünden 2016 yılında mezun olduktan sonra 2017 yılında Anadolu Üniversitesi Açıköğretim Fakültesi Web Tasarımı ve Kodlama bölüme başlayıp 2019 yılında mezun oldu, yine aynı yılda Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümünü kazandı.

2018 yılından itibaren özel bir firmanın Ar-Ge merkezinde görev yapmaktadır.



## BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU: PATHFINDER

ÖĞRENCİLER: B191210309 SERHAT BİLAL – B191210301 HİLAL NARMAN

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
<b>Yazılı Çalışma</b>			
<b>Çalışma klavuza uygun olarak hazırlanmış mı?</b>	x	0-5	
<b>Teknik Yönden</b>			
<b>Problem tanımı yapılmış mı?</b>	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?			
<b>Yapılan işlerin zorluk derecesi?</b>	x	0-25	
<b>Sözlü Sınav</b>			
<b>Yapılan sunum başarılı mı?</b>	x	0-5	
<b>Soruları yanıtlama yetkinliği?</b>	x	0-20	
<b>Devam Durumu</b>			
<b>Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?</b>	x	0-5	
<b>Diğer Maddeler</b>			
<b>Toplam</b>			

DANIŞMAN: PROF. DR. NEJAT YUMUŞAK

DANIŞMAN İMZASI: