

Kafe Otomasyon Uygulaması

Ferhat BULGA, Melek ÖFKELİ, Zehra H. BABACAN
Yazılım Mühendisliği, 3.Sınıf, A/B

Özet

Proje kapsamında kafe tipi işletmelerin, müşterilerinin, garsonlarının ve mutfak elemanlarının kullanacağı Android, iOS işletim tabanlı telefonlar için bir otomasyon uygulaması geliştirilmesi amaçlanmıştır. Bu amaç kapsamında projenin ihtiyaçlarını karşılayacak tümleşik geliştirme ortamları ve yazılım dilleri, benzer ve yakın amaçla üretilmiş daha önceki projeler incelenmiştir.

Uygulamanın geliştirilmesi aşamasında VSCode, AndroidStudio gibi tümleşik geliştirme ortamları yardımıyla Flutter yazılım dili ve FireBase veri tabanı kullanılmıştır.

Yaptığımız projenin 4 farklı kullanıcı tipi bulunmaktadır.

- Müşteri, işletmeye müşteri olarak gelen herkesin işletmenin menüsüne ulaşabileceği, ilgili masaya siparişlerini oluşturabileceği bir ara yüz oluşturulmuştur.
- Garson, işletmede garson görevi ile çalışan kişilerin kullanacağı bu ara yüzde mutfaktan onaylanmış ve servise hazır siparişler görüntülenmekte ve siparişlerin teslimleri belirtilmektedir.
- Kasiyer, işletmenin kasa kısmında ilgili masaların onaylanmış ve teslim edilmiş olan ürünlerinin ödenecek tutarları görüntülenmekte ve ödeme işlemiyle veri tabanından masanın doluluk durumunun değiştirilebileceği bir ara yüz bulunmaktadır.
- Şef, işletmede ürünleri hazırlayan mutfak görevlisini temsil etmektedir. Bu kullanıcı türü için müşterilerin girmiş olduğu siparişlerin onaylanması, hazırlanması ve dolayısıyla veri tabanına işlenmesini sağlayan bir ara yüz bulunmaktadır.

Gerçekleştirdiğimiz bu uygulamanın amacı istenilen işletme tipine entegre edilebilir olmasıdır ancak bu raporda ve sunum için kolaylık oluşturulması için bir örnek üzerinden görselleştirilmiş ve örnek veriler ile sunulmuştur.

Projenin iyileştirilmesi bakımından, oluşturduğumuz bu ara yüzler sonucunda işletmeye ait kullanıcı türleri ile müşteri kullanıcı türünün farklı uygulamalar ile sisteme giriş sağlayabilir olmasının daha kullanıcı dostu bir yapı oluşturacağına karar verilmiştir.

Anahtar Kelimeler: Otomasyon, işletme, kolaylık

1. GİRİŞ

Projenin temel amacı müşterilerin diledikleri şekilde ve zamanda işletmenin menüsüne erişebilmesi ve istediği ürünü sipariş edebilmesi amacıyla ortaya çıkmıştır. Buna bağlı olarak işletme çalışanlarının belirtilen siparişlere yönelik üretim ve onay süreçlerini kolayca görüntüleyebilmeleri gerekmektedir.

Bu bağlamda bir telefon uygulaması geliştirilmiştir. Bu uygulamanın işletmelere göre değişiklik göstermekle beraber temel 4 kullanıcı bulunmaktadır. Her kullanıcının kendi türüne özgü gereklilikleri ve yetkileri bulunmaktadır. Kullanıcı türleri uygulamaya kendi türlerine ait alanlardan giriş ile erişim sağlamaktadır. Bunları açıklamak gerekirse,

- Müşteri kullanıcısının işletmeye ait ürünleri ve masaları görüntülemesi, sistemde herhangi bir ürün giriş bulunmayan masaya dilediği ürünü ekleyerek siparişini oluşturabilmesi beklenmektedir.
- Müşterinin sisteme girmiş olduğu ürünlerin şef (mutfak görevlileri) tarafından işletmenin stok durumuna bağlı onaylanması ve hesaplanmak üzere veri tabanına işlemesi beklenmektedir. Bu durumda sipariş onaylanan masalar dolu kabul edilmektedir. Siparişlerin onaylanmasından sonra hazırlanması durumu da yine aynı kullanıcı tarafından belirtilebilmektedir.
- Bütün bu şef kullanıcı tarafından yapılan girişler sayesinde garson kullanıcısı kendi alanında teslim etmesi gereken hazır siparişleri görüntüleyebilmektedir. Daha sonrasında teslim dilen siparişleri sisteme belirtir ve belirttiği siparişler müşteri tarafından ödemesi alınmak üzere sistemde güncellenir.
- Son olarak kasiyer kullanıcısı bütün güncellemelerden geçmiş, ödenmesi gereken siparişleri görüntüler ve ödemenin gerçekleştirilmesini sistem üzerinden günceller bunun sonucunda ilgili masa farklı müşteriler tarafından sipariş oluşturulabilir forma gelir.

Bütün bu farklı kullanıcı türlerinin gereklilikleri projenin genel yapısını ve beklenen sonuçlarını oluşturmaktadır.

2. KULLANILAN ARAÇLAR

Özet kısmında da belirtildiği gibi uygulamanın geliştirilmesi aşamasında VSCode, AndroidStudio gibi tümleşik geliştirme ortamları yardımıyla Flutter yazılım dili ve Firebase veri tabanı kullanılmıştır. Bu araçları açıklamak gerekirse.

1. Visual Studio Code (VSCode)

Visual Studio Code Microsoft tarafından Windows, Linux ve MacOS için geliştirilen bir kaynak kodu düzenleyicisidir. Hata ayıklama, gömülü Git kontrolü, sözdizimi vurgulama, akıllı kod tamamlama, snippet'ler ve kod yeniden yapılandırma desteği içerir. Bir dizi programlama dili ve belirli bir dilde bulunabilen veya bulunmayacak özellikler kümesini destekler. [1]

2. Android Studio

Android Studio, Android için resmi tümleşik geliştirme ortamı. 16 Mayıs 2013 tarihinde Google I/O etkinliğinde tanıtılmıştır. Android Studio, IntelliJ IDEA'ya dayalı olup Android geliştirme için özel olarak tasarlanmıştır.

Android Studio, Apache lisansı ile lisanslanmıştır ve ücretsiz olarak edinilebilmektedir.

Bu lisans, Apache Yazılım Vakfı tarafından geliştirilmiştir. Ayrıca bu yazılım özgür yazılım lisansları arasında da yer almaktadır. Açık kaynak koduna sahiptir.

Android Studio’da programlama dili olarak Java kullanılmaktadır. Ayrıca Android Studio’nun işletim sistemi Cross-platform şeklindedir. Bu, çoklu platform yazılımları demektir. Böylelikle birden fazla işletim sistemi için yazılımın dağıtımını gerçekleştirilebilmektedir.[2]

3. Firebase

Firebase, gerçek zamanlı, işbirliğine dayalı uygulamaları desteklemek için tasarlanmış bir bulut hizmetidir. Geliştiriciler, paylaşılan bir veri yapısına erişim elde etmek için Firebase kitaplığını uygulamalarına ekler; bu verilerde yaptıkları değişiklikler Firebase bulutuyla ve diğer istemcilerle milisaniyeler içinde otomatik olarak senkronize edilir. Firebase uygulamaları tamamen istemci tarafı kodla yazılabilir, kullanıma hazır gerçek zamanlı olarak güncellenebilir, mevcut hizmetlerle birlikte çalışabilir, otomatik olarak ölçeklenebilir ve güçlü veri güvenliği sağlayabilir.[3]

Firebase veritabanlarında iki tane veri tabanı yapısı vardır. Bunlar,

- Real-Time Database
- Firestore Database

4. 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. [2]

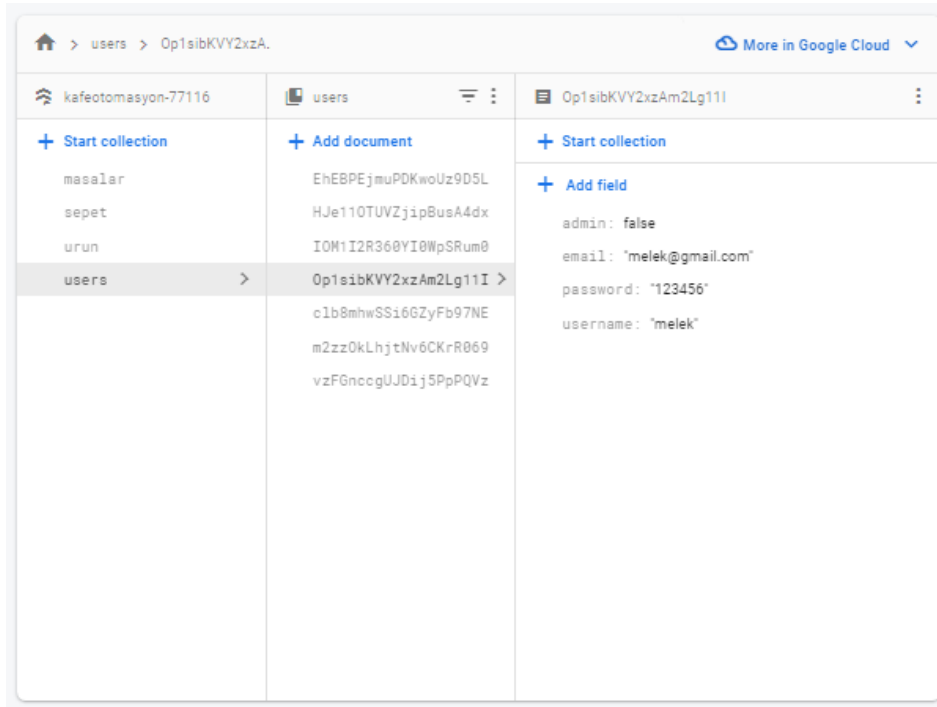
Flutter'ın ana bileşenleri şunlardır:

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

3. VERİTABANI TASARIMI

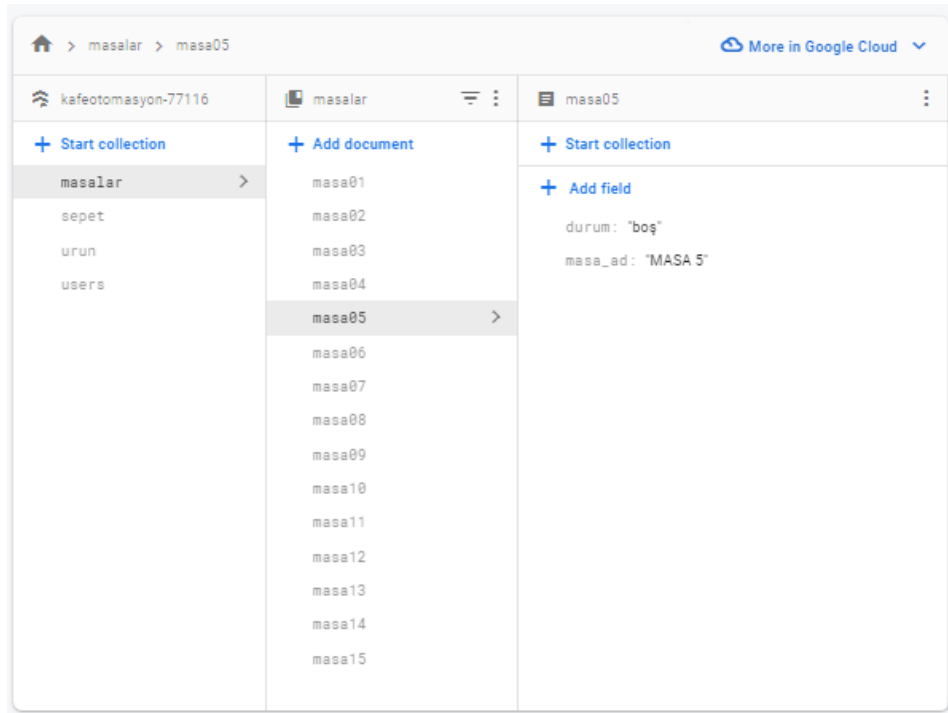
Projemizde veri tabanı olarak Firebase kullanılmıştır. Bunun için KafeOtomasyon adında bir Firebase projesi oluşturulup geliştiricilerin hesapları projeye eklenmiştir. Projemizde sabit ve aktif veriler bulunması dolayısıyla Firestore yapısı kullanılmıştır.

Projenin kapsamında kullanıcıların müşteri olmaması durumunda giriş sağlayabilme için gerekli olan kullanıcı ve şifre bilgileri için ‘users’ isimli bir collection oluşturulmuş ve kayıt olan kullanıcıların bilgilerinin burada tutulması sağlanmıştır. (Fotoğraf1.1)

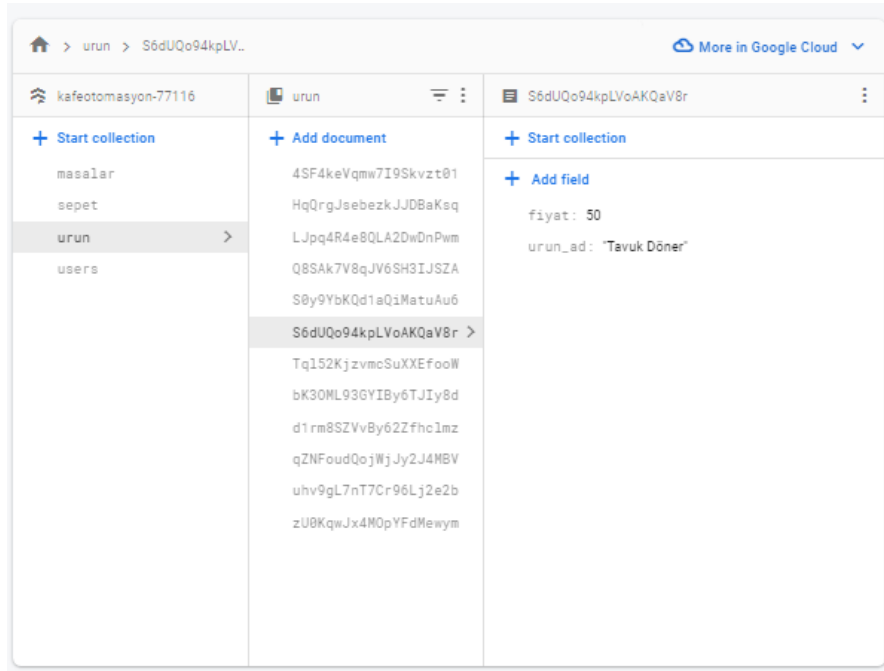


Fotoğraf1.1

Diğer bir yandan işletmeye ait masalar ve ürünler için ‘masalar’ ve ‘urun’ adında collectionlar oluşturup içlerine masa ve ürün bilgileri document olarak eklenmiştir. (Fotoğraf 2.1/2.2) Bu veri alanlarına veri eklenmesi işletmenin talebi üzerine geliştiriciler tarafından gerçekleştirilmektedir.

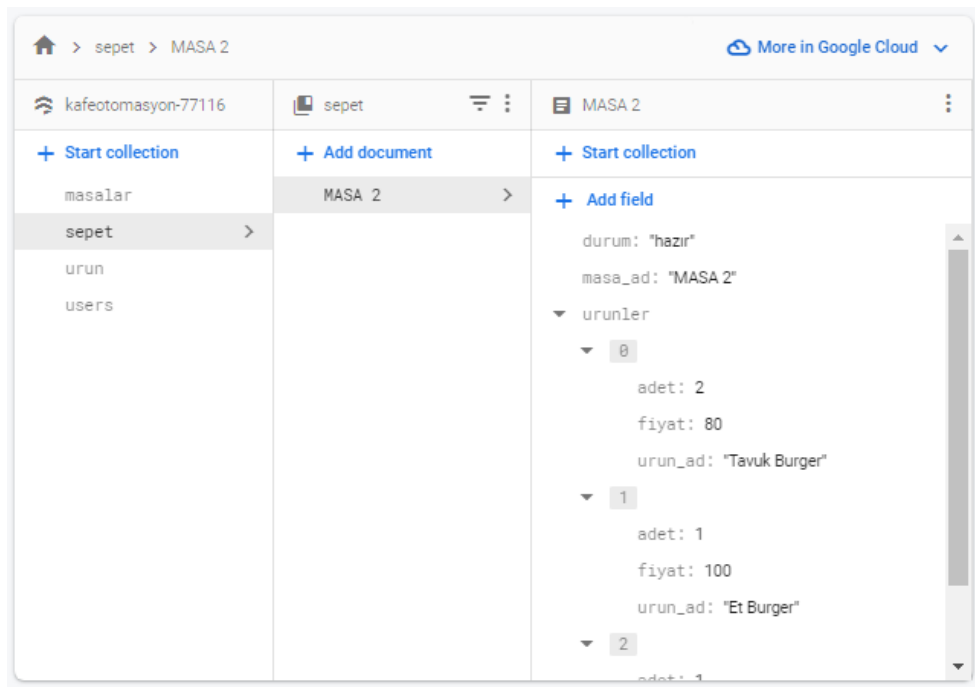


Fotoğraf2.1

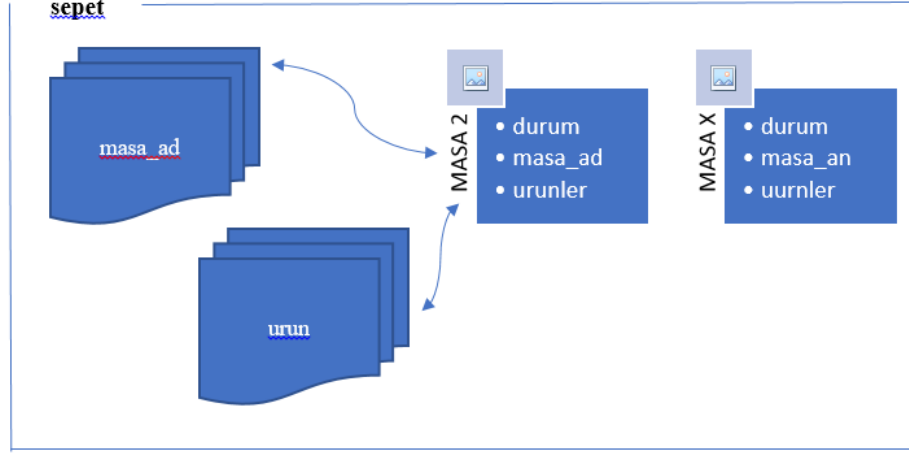


Fotoğraf2.2

Son olarak oluşturduğumuz 'sepet' isimli collection uygulamanın gerekliliklerini yerine getiren aktif olarak güncellenen seçilen masa ve ürünlerin eşleştirildiği alandır. (Fotoğraf3.1) Bu collection alanında documentler masalar collectionunun fieldlarından eklenirken, fieldlar ürünler collectionunun fieldları ile eşleştirilmektedir. (Şekil1.1)



Fotoğraf3.1



Şekil1.1

4. MODÜLLER

Projenin hazırlanması kısmında 'lib' dosyası içerisinde 'model, pages, servisler, widgets' isimlerinde 4 ana klasör ve bunlara bağlı 22 alt sınıf bulunmaktadır. Bu yapının temelinde Flutter dilinin gerekliliklerinden nesneye yönelik programlama kullanılmıştır. Kullandığımız sınıfları açıklamak gerekirse.

lib

- **model** (model sınıfı bilgilerin dinlenmesi ve listelenmesi için kullandığımız sınıfları içerir)
 - **masa_mode.dart** (masa adı gibi masalara ait bilgileri listelediğimiz sınıf)
 - **urun.dart** (ürünlerin özelliklerini listelediğimiz sınıf)
 - **user.dart** (kullanıcıların giriş yaparken kullandığı bilgilerin listelendiği sınıf)
- **pages** (kullanıcıların işlemleri gerçekleştirdiği sınıfları içerir)
 - **garson.dart** (Bu sınıfa personel.dart sınıfından navigatör ile geçiş sağlanır. sef.dart sınıfında hazırlanmış olarak belirtilen siparişlerin görüntülendiği ve teslim edildi bilgisinin belirtildiği fonksiyonları içeren sınıftır.)
 - **kasiyer.dart** (Bu sınıfa personel.dart sınıfından navigatör ile geçiş sağlanır. garson.dart sınıfından teslim edildi olarak belirtilen siparişlerin görüntülendiği ve ödemelerinin gerçekleşmesi durumunda sistemde masanın durumunun sıfırlanmasının yani boş olarak değiştirilmesinin ve sepet collectinonundan masa documentinin silinmesinin sağlandığı fonksiyonları içeren sınıftır.)
 - **login_page.dart** (Giriş sayfasında (HomePace.dart) personel girişinin seçilmesi ile navigatör ile geçilen sınıftır. Bu sınıfta kullanıcıların register_pace.dart sınıfında veri tabanına ekledikleri verilerin doğruluğunun sorgulanması ve buna bağlı personel.dart sayfasına geçiş sağlanmasını oluşturun fonksiyonlar bulunmaktadır.)
 - **menu.dart** (Müşteri girişi sağlandıktan sonra menünün görüntülendiği ve seçilen ürünün sepete eklenmesi metodunu içeren sınıftır. Bu sınıfa HomePace.dart sınıfında bulunan navigatör ile erişim sağlanır.)

- **personel.dart** (Giriş sayfasında (HomePace.dart) personel girişi seçildikten sonra login_pace.dart sayfasına girilen bilgilerin doğru olması durumunda navigatör ile bu sayfaya geçiş sağlanır. Personel türünün seçilmesi ve ilgili sınıflara geçiş bu sınıftan sağlanır.)
- **register_pace.dart** (Bu sınıfa personel.dart sınıfında belirtilen ‘kayıt ol’ kısmında bulunan navigatör ile geçiş sağlanır. İşletmeye kayıt edilmesi gereken kullanıcıların veri tabanına işlenme, kullanıcı adı ve şifrelerinin oluşturulmasını sağlayan sınıftır.)
- **sef.dart** (Bu sınıfa personel.dart sınıfından navigatör ile geçiş sağlanır. Müşterilerin onayladıkları siparişlerin görüntülediği ve siparişlerin hazırlanması durumunda masanın durumunun hazır şeklinde değiştirilmesi ve garson kullanıcısının sayfasına aktarılmasını sağlayan fonksiyonları içeren sınıftır.)
- **HomePage.dart** (Uygulamamızın başlatıldığı ve kullanıcı türlerinin görüntülediği sınıftır.)
- **musteri.dart** (Giriş sayfasında (HomePace.dart) müşteri girişi seçildikten sonra durumunda navigatör ile bu sayfaya geçiş sağlanır.)
- **sepet.dart** (menu.dart sayfasında sağ üstte bulunan icon yardımıyla erişilen bu sayfada müşterilerin seçtikleri ürünlerin görüntülenmesi ve değişiklikler yapabilesi sağlanmaktadır. Bunların ardından sepetin onaylanması durumunda sepet içeriği sef.dart sayfasına aktarılır.)
- **servisler** (model klasöründe listelediğimiz verileri göstermek için kullanılan ve veri tabanındaki yolu oluşturmak veya veriyi çekmek için kullanılacak metotların bulunduğu sınıfları içeren klasördür.)
 - **auth.dart** (Giriş işlemlerinin yapılmasını sorgulayan metotların bulunduğu sınıftır.)
 - **masa_service.dart** (Müşterilerin seçtikleri masaların veri tabanından çağırılması için kullanılan metotların bulunduğu sınıftır.)
 - **siparis_service.dart** (Seçilen masa ve ürünlerin eşleştirilmesi ve veri tabanına aktarılmasını sağlayan metotların bulunduğu sınıftır.)
 - **siparismasaları_service.dart** (Siparişleri Şef tarafından onaylanmış yani durumu ‘dolu’ olarak güncellenecek masaların veri tabanı üzerine oluşturulmasını sağlayan metotların bulunduğu sınıftır.)
 - **urun_service.dart** (Müşterilerin seçtikleri ürünlerin veri tabanından çağırılması için kullanılan metotların bulunduğu sınıftır.)
- **widgets**
 - **my_elevated_button.dart** (Uygulamanın içinde kullandığımız butonların yapısının bulunduğu metot. Bu metot sayesinde uygulamanın için kullandığımız birçok buton için ayrı bir kod alanı ayırmak yerine bu sınıftan nesne üreterek kodumu temiz kılamaya çalıştık.)
 - **firebase_options.dart** (Veri tabanı için gerekli işlemlerin metotlarının bulunduğu sınıf)
 - **list.dart**

Sınıfların yanı sıra sınıflarda oluşturduğumuz metotların çalışabilmesi için projemize eklediğimiz 5 adet paket bulunmaktadır. Bu paketler projemizde ‘pubsec.yaml’ sınıfı içerisine eklenmiştir. Paketleri açıklamak gerekirse,

- **cupertino_icons: ^1.0.2** (Flutter Cupertino widget'ları tarafından kullanılan varsayılan simge varlıkları kümesini içeren bir pakettir.)
- **firebase_core: ^2.4.1** (Birden fazla Firebase uygulamasına bağlanmayı sağlayan Firebase Core API'yi kullanmak için bir Flutter eklentisi.)
- **firebase_auth: ^4.2.4** (Firebase Kimlik Doğrulama API'sini kullanmak için Flutter eklentisi.)
- **cloud_firestore: ^4.3.1** (Cloud Firestore API'yi kullanmak için bir Flutter eklentisi)
- **bordered_text: ^2.0.0** (Flutter tesxt alanlarına çerçeve eklemek için bir Flutter eklentisi)

5. SONUÇ

Hazırlamış olduğumuz bu uygulama genel anlamda ilk adım gereksinimleri karşılamakta olup, tasarımsal ve fonsksiyonel olarak geliştirilebilir olduğuna karar verilmiştir.

KAYNAKÇA

1. 12.09.2022; Visual Studio Code; https://tr.wikipedia.org/wiki/Visual_Studio_Code
2. 25.05.2022; Android Studio; https://tr.wikipedia.org/wiki/Android_Studio
3. 10.11.2022; Firebase; <https://web.archive.org/web/20140424142733/https://www.crunchbase.com/organization/firebase>
4. 15.05.2022; Flutter; <https://tr.wikipedia.org/wiki/Flutter>