

# Fit4Less

Daha Az Kaynakla Daha Çok Fit

Kaan Toraman

211307078

Kocaeli Üniversitesi Bilişim Sistemleri  
Mühendisliği

**Özet—Sporcular için su hatırlatıcı, spor hareketleri ve besin değerleri gibi yararlı bilgileri tek uygulamada toplayan bir ‘fitness uygulaması’ projesi.**

**Abstract- An application project that collects useful information such as water reminders, sports movements and nutritional values for athletes in a single application.**

**Anahtar Kelimeler—spor, egzersiz, kas, besin.**

## I. GİRİŞ

Günümüzde sporcular için birden fazla spor uygulaması bulunmakta ve hepsinin kendine göre amaçları var. Kimisi su içmek için hatırlatma yapıyor, kimisi egzersiz hareketlerini gösteriyor, kimisi besin değerlerini gösteriyor... Bu tip uygulamalar gerçekten kullanışlı ve yararlı olabiliyor ama hepsinin farklı uygulamalar halinde olması, hepsini tek tek indirme yükünü biz kullanıcılara yüklüyor.

Peki bütün bahsettiğim bu uygulamaların tek bir uygulamada toplanması nasıl olurdu? Fit4Less projesi adından da anlaşılabilir gibi daha az ile fit olmayı hedefliyor. Sporcular için gerekli olan birçok özelliği tek bir uygulamada toplayan Fit4Less projesi, sporcular için egzersiz uygulaması kullanımını kolaylaştırmayı hedefliyor.

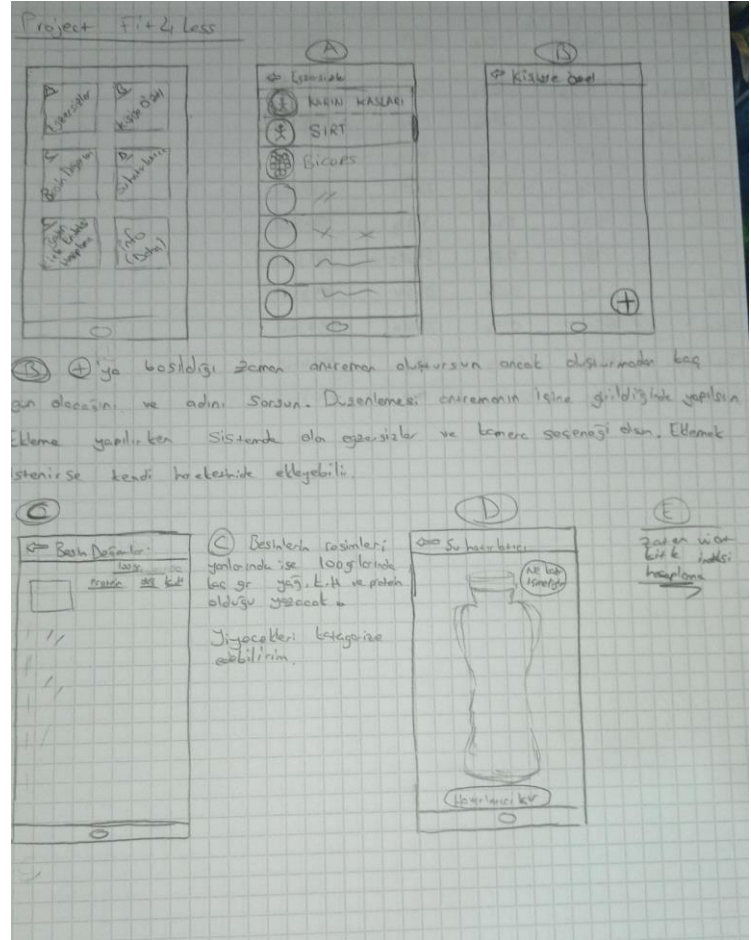
## II. BİLGİ TOPLAMA

Bu kısım proje için oldukça önemli. Projemizin nelere ihtiyacı olduğunu incelediğimiz kısım. Geniş bir uygulama havuzunda araştırma yapmam gerekti. Birçok spor uygulamasını inceledim. Bu incelemeler sonucunda projemde; basit spor hareketleri, vücut kitle indeksi hesap makinesi ve su hatırlatıcısı olması gerektiğine karar verdim.

Uygulamaların tasarımlarına baktığımda ise genellikle açık tonlu renkler kullanıldığını ve basit bir tasarıma sadık kalındığını gözlemledim.

## III. TASARIM

Uygulamayı kodlamaya başlamadan önce basitte olsa bir tasarıma ihtiyacım vardı. Bunun için kaba taslakta olsa bir çizim yaptım (Görsel-1).



Görsel-1

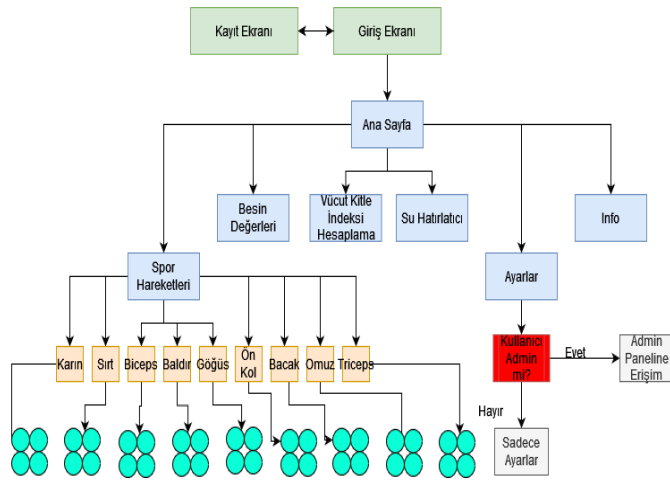
Yapmış olduğum bu tasarım üzerine uygulamamı kurma kararı aldım ve tasarıma mümkün olduğunca sadık kalmaya çalıştım. Buton iconları ve spor hareketlerinin görsellerini kaynakçada belirttiğim sitelerden ücretsiz bir şekilde kullandım. Diğer bütün tasarımlar bana aittir.

## IV. UYGULAMA GELİŞTİRME

Uygulamada Kotlin dilini ve Android Studio'yu kullandım. İlk başlatıldığında bizi giriş ekranı karşılıyor. Kayıtlı bir hesabınız varsa giriş yaparak ana menü ekranına geçiş yapabilirsiniz; kayıtlı bir hesabınız yoksa kayıt olma ekranına gidip kaydınızı yapabilirsiniz. Kayıt olduktan sonra tekrardan giriş ekranına dönüyor ve giriş yapıyorsunuz. Giriş yapıldıktan sonra sizi ana sayfa karşılıyor.

Ana sayfada altı adet sizi farklı sayfalara yönlendiren buton bulunmaktadır. Bu butonlar sırasıyla: besin değerleri sayfası, su hatırlatıcısı, vücut kitle endeksi hesap makinesi, bilgi sayfası, ayarlar sayfası ve egzersizler sayfasına yönlendirmekte.

Egzersizler sayfası kendi içinde dokuz farklı sayfaya ve bu dokuz sayfanın her biri de kendi içinde dört farklı sayfaya açılmakta.



Görsel-2

### A. Manifest Dosyası

Uygulamanın sayfalarından önce manifest dosyama internet kullanımı, titreşim ve alarm için erişim izinleri verdim. Başlangıç etkinliği olarak da giriş sayfasını seçtim.

### B. Gradle Dosyası

Gradle dosyamın içine kullanacağım kütüphaneleri ekledim.(Görsel-3)

```
dependencies {  
  
    implementation 'androidx.core:core-ktx:1.7.0'  
    implementation 'androidx.appcompat:appcompat:1.5.1'  
    implementation 'com.google.android.material:material:1.7.0'  
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
    implementation 'com.google.firebase:firebase-auth-ktx:21.1.0'  
    implementation 'androidx.annotation:annotation:1.3.0'  
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.5.1'  
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.5.1'  
    testImplementation 'junit:junit:4.13.2'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.4'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'  
    //retrofit  
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
    //GSON  
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'  
    //okhttp  
    implementation 'com.squareup.okhttp3:okhttp:4.9.3'  
  
    implementation 'com.google.firebase:firebase-database-ktx:20.1.0'
```

Görsel-3

### C. Giriş ve Kayıt Ekranı

Uygulamada bulunan üyelik sistemi için Google'un hizmeti olan Google Firebase'i kullandım. Android Studio'nun tools kısmından firebase'i seçtikten sonra açılan pencerede Authentication--> authenticate using google [KOTLIN]'i seçtim ve firebase için gerekli kütüphaneleri yükledim. Bu kütüphaneler sayesinde kullanıcıdan kayıt alıp google sunucularında bu bilgileri tutabiliyorum. Bu da giriş ve kayıt işlemlerinin sürekliliğini sağlıyor.

Kayıtlı bir hesap varsa kullanıcıdan alınan e-posta ve şifreyle giriş yapılıyor eğer kayıtlı bir hesap bulunmuyorsa 'kayıt ol' düğmesine bastıktan sonra yeni bir sayfa açılıyor.

Bu sayfada kullanıcıdan e-posta, şifre ve şifrenin yanlış girilmediğine emin olmak için tekrar şifre girişi alınıyor. Bir sıkıntı çıkmadığı sürece başarıyla kayıt sağlanıyor ve tekrardan giriş sayfası açılıyor.

### D. Ana Menü

Giriş yapıldıktan sonra ilk karşılaştığımız sayfa, ana menü. Ana menü, sade ve fazla bir özelliği yok. Ana menünün kod kısmında sadece sayfa geçişleri için kodlar bulunuyor. Bu sayfa köprü görevi görüyor.

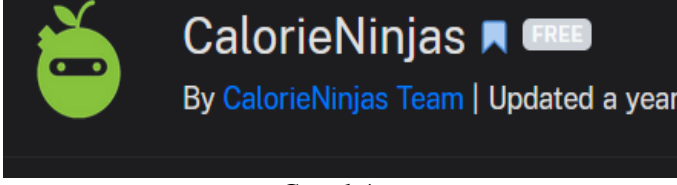
Sayfada altı adet buton var. Bu altı adet butondan her biri diğer sayfalara yönlendiriyor.

### E. Besin Değerleri

Besin değerleri sayfasına girdiğimiz zaman, kullanıcıdan bir yiyecek/ besin ismi girilmesi isteniyor. Girilen ismin kalorisini, yağ değerini, protein değerini, lif değerini, karbonhidrat değerini ekranda yazılı olarak görebiliyorsunuz. Bu değerleri yazdırmak için bilgileri bir API'den çekiyorum.

Projede isterler gereği iki API kullanmam gerekiyordu. Birinci API'yi burada kullandım. API kullanabilmek için

retrofit, gson ve okhttp kütüphanelerini gradle dosyasına ekledim. Okhttp üzerine kurulu bir API kullanımı yaptım. Bunun en büyük sebebi RapidApi sitesinde Kotlin için direkt okhttp'a bağlı kodun bulunmasıydı. Besin değerlerini internetten çekmek için kullandığım API: Ninjas Team'in yapmış olduğu "CalorieNinja API" (Görsel-4)



Görsel-4

API'den dönen değerleri almak için besin değerleri kod sayfasının en altında iki adet veri sınıfı (data class) tanımladım. Birinci veri sınıfı bütün değerleri içeren bir adet matris (array); ikinci dizi matris içinde bulunacak besin değerlerini içeriyor.

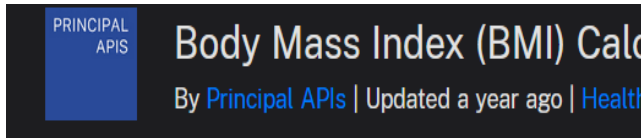
### F. Vücut Kitle İndeksi Hesap Makinesi

Vücut kitle indeksi hesap makinesi sayfasına girdiğimizde, kullanıcıdan kilo ve boy değerlerinin girilmesi isteniyor. Girilen bu değerler sonucunda, sayısal bir değer ve "kilolusun", "obezsin", "zayıfsın" gibi yazı değerleri ekrana geliyor.

Normalde vücut kitle indeksini hesaplamak için bir formül var:  $\text{kilo}/(\text{boy uzunluğu})^2$ .

Formül olmasına rağmen projemde iki adet API kullanmam gerekiyordu. Bu yüzden ikinci API'yi de burada kullandım.

Vücut kitle indeksini hesaplaması için kullandığım API, Principal Apis'in yapmış olduğu "Body Mass Index (BMI) Calculator" adlı API'sini kullandım (Görsel-5).



Görsel-5

API'den toplanan değerleri almak için, kod sayfasında bir adet veri sınıfı (data class) oluşturdum. Veri sınıfı kullanıcıdan alınan kilo ve boy uzunluğu girdileriyle hesaplanan 'BMI' değerini geri döndürüyor.

Bu dönen sayısal değerle yeni bir fonksiyon oluşturdum. Bu fonksiyon 'BMI' değerine göre kilolusun, obezsin, zayıfsın gibi yazılı değerleri döndürüyor.

Fonksiyonun döndürdüğü yazı değeri ve 'BMI' sayısal değeri ekrana yazdırılıyor.

### G. Bilgi Sayfası

Bilgi sayfasına girdiğimizde, geliştirici olan bana ulaşılabilmesi için github hesabımın bağlı olduğu bir text ve linkedin hesabımın bağlı olduğu bir text bulunuyor. Aynı zamanda en altta sürüm bilgileri yazıyor.

Tasarım açısından özenmediğim tek sayfa burası oldu. Kod kısmında da sadece link açmak için Kotlin kodu var.



Görsel-6

### H. Su Hatırlatıcısı

Su hatırlatıcısı sayfasına girdiğimizde, karşımıza bir su şişesi görseli ve altında dört adet buton bulunuyor. Bu butonlar ne kadar zaman sonra hatırlatma yapılacağını belirliyor. İlk buton 30 dakika sonra, 2. buton bir saat sonra, 3.buton iki saat sonra hatırlatma yapmak için ayarlıyor. 4. Buton ise hatırlatıcı oluşturulduysa onu iptal ediyor. Hatırlatma eylemi için bir tane broadcast receiver sınıfı oluşturdum (Görsel-7).

```
package com.example.fit4lesfirebase

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import androidx.core.app.NotificationCompat
import androidx.core.app.NotificationManagerCompat

class ReminderBroadcast:BroadcastReceiver(){
    override fun onReceive(context: Context?, intent: Intent?) {
        val builder= NotificationCompat.Builder(context!!, channelId: "notifyWaterReminder")
            .setSmallIcon(R.drawable.bildirim)
            .setContentTitle("Fit4Less")
            .setContentText("It's time to drink water!!!")
            .setPriority(NotificationCompat.PRIORITY_DEFAULT)

        val notificationManager= NotificationManagerCompat.from(context!!);
        notificationManager.notify( id: 200,builder.build());
    }
}
```

Görsel-7

Bu broadcast sınıfı sayesinde su hatırlatıcısı sınıfında, java.util.Calendar dan türemiş olan android.icu.util.Calendar aracılığıyla 30dakika , 1 saat veya 2 saat sonrasına alarm kurma fonksiyonlarını yazdım.

### İ. Ayarlar Sayfası

Ayarlar sayfasına girdiğimiz zaman bizi iki kısım karşılıyor. İlk kısım bir scrool view, ikinci kısım ise üzerinde panel yazan bir buton. İlk kısmın içinde kullanıcıların e-postalarını ve şifrelerini değiştirmesi için bir bölüm, en altında ise kullanılan hesabı silmek için bir buton var.

Bunlara hesap açmış olum uygulamayı kullanan herkes erişebiliyor. Burdaki tüm işlemleri de giriş ve kayıt kısmında belirmiş olduğum google firebase ile yapıyorum. İkinci kısım olan panel yazan butona ise herkes erişemiyor.

Kodlarda belirtmiş olduğum iki admin hesap sadece bu bölüme erişebiliyor. Bu panel, admin paneli. Uygulamamda admin gerektirecek bir durum yok. Lakin proje isterlerim arasında olduğu için ekledim. Admin panelinde, istediğimiz e-postayla yeni bir kayıt oluşturabiliyoruz. Aynı zamanda şifresini değiştirmesi için bir hesap belirtip, o hesaba şifre değiştirme maili de gönderebiliyoruz. Bu duruma örnek vermek gerekirse, benim deneme@deneme . com adresiyle fit4les uygulamasında bir hesabım var ve adminler giriş yapma etkinliğinden şüphelendi. Beni uyarmak için e-posta adresime şifre değişime e-postası yollayabiliyorlar.

## J. Egzersiz Sayfası

Egzersizler sayfası projemde tasarımına en çok dikkat ettiğim yerlerden biri oldu. Projede yapım aşamalarını sıralayacak olursam, kotlin kodu olarakta tasarımsal anlamda en son geliştirdiğim yer burasıydı. Nedeni ise bu sayfadan dallanan birden fazla alt sayfa var.

Sayfaya ilk girdiğimiz zaman bizi 9 adet buton karşılıyor. Bu butonlar sporcuların çalıştırdıkları kas bölgelerini temsil ediyor. Hangi butona basarsak, o kas bölgesi için yapabileceğim antrenmanlar karşımıza çıkıyor. Uygulamada toplamda 36 adet egzersiz bulunmakta. Her egzersizin kendi sayfasında, egzersizin nasıl yapıldığına dair görsel ve nasıl yapılacağına dair veya o egzersizle ilgili tüyolar bulunuyor. Burada ki bilgiler tüm uygulamada olduğu gibi ingilizce. Neden ingilizce olduğuna tartışma bölümünde değindim.

Egzersizler kısmında kotlin kodu anlamında önemli satır fazla yok. Genel sayfa geçişleri ve buton/text atamaları var.

Ama xml olarak önemliydi. Nedeni ise bu sayfaların kullanıcıya egzersiz bilgilerini doğru vermesi ve bunu yaparken gözü yormaması gerekmesiydi.

## V. UYGULAMAYI TEST ETME

Aslen bu kısım geliştirme kısmıyla sürekli olarak entegre haldeydi. Bu süreç benim için en sıkıntılı süreçti. Bunun

nedeni kullanmış olduğum emulator, uygulama test süreci boyunca sürekli çöktü. Her yaptığım değişiklikten sonra yedek aldım. Bu test süreçlerini benim için daha katlanılabilir bir hal almasını sağladı. Uygulamamı en son haline getirdiğim zaman .apk dosyası haline getirip iki adet cihazda test ettim ve şu anki durumunda bir sorunla karşılaşmadım .

## VI. TARTIŞMA

Geliştirdiğim uygulamanın amacına hizmet ettiğini düşünüyorum. Çünkü planlamış olduğum gibi sporcuların sık sık kullandığı uygulamalarda ki özellikleri tek bir uygulamada toplamayı başardım. Bunu yaparken uygulama dilini İngilizce olarak seçtim. Bunun nedeni ise uygulamayı geliştirdiğim ilk aşamada Google Play üzerinde yayınlamayı planlıyor olmamdı. Geliştirme süreci boyunca bu fikrimden vaz geçtim. Çünkü bu projem, ilk android studio projemdi.

## KAYNAKÇA

- Uygulamada kullanılan iconlar için[1]
- Besin değerlerini çekmek için kullandığım API[2]
- Vücut kitle indeksi hesaplamak için kullandığım API[3]
- Uygulamadaki egzersiz görselleri için[4]

[1] <https://www.flaticon.com/icons> [02.12.22]

[2] <https://rapidapi.com/calorieninjas/api/calorieninjas/> [04.12.22]

[3] <https://rapidapi.com/principalapis/api/body-mass-index-bmi-calculator/> [04.12.22]

[4] <https://www.pexels.com/> [04.12.22]

[5] <https://www.muscleandstrength.com/> [04.12.22]

[6] <https://www.lakewashingtonpt.com/> [04.12.22]

[7] <https://medium.com/hardwareandro/android-kotlin-ve-retrofit-ile-veri-%C3%A7ekip-listeleme-nas%C4%B1l-yap%C4%B1%C4%B1r-1864d3668e1b> [04.12.22]

[8] <https://www.verywellfit.com/> [04.12.22]

[9] <https://www.healthline.com/> [04.12.22]

[10] <https://www.masterclass.com/articles/dumbbell-side-bend-exercise-guide> [04.12.22]

[11] <https://firebase.google.com/docs/auth> [04.12.22]