

# AKILLI REKLAM YÖNETİM SİSTEMİ

Emre Altay, Batuhan Subaşı

## Bilgisayar Mühendisliği Bölümü

### Kocaeli Üniversitesi

emrealltay@gmail.com, batuhansubasi@hotmail.com

## Özet

Bu projede bizden android uygulama ve bulut bilişim teknolojilerinin kullanılması istenmektedir. Projede akıllı reklam yönetim sistemi yapmanız istenmektedir. Android uygulama geliştirmeniz beklenmektedir. Uygulama geliştirilirken firebase kısıtlaması getirilmektedir. Basit bir arayüzle veri eklemesi yapılmalı ve eklenen veriler api ile haberleşme sağlanarak android studio ile işlem yapılması istenmektedir. Basit bir android arayüzü hazırlayarak işlem yapılacaktır.

Bu proje, Android uygulama ve bulut bilişim teknolojilerinin kullanılmasını amaçlamaktadır.

Projede akıllı reklam yönetim sistemi yapmanız istenmektedir. Android uygulama geliştirmeniz beklenmektedir.

Android uygulamada bulunması beklenen istekler:

1-Log in ekranı: Kullanıcı, kullanıcı adı ve şifresi ile giriş yapabilmeli, şifre değiştirebilmeli veya kayıt olabilmeli.

2-Kullanıcının konum bilgileri GPS üzerinden alınacaktır ve elle de giriş yapılabilecektir.

3-Kullanıcı, lokasyon bilgileri alınarak kullanıcının belirlediği mesafedeki mağazalardan istediği kategoriye ait kampanyaları bildirim olarak ekranında görebilmelidir.

4-Kullanıcı bilgileri telefonda tutulmayacak bulut üzerinden doğrulama (log-in) yapılacaktır.

5-Kullanıcı istediği durumlara göre arama işlemlerini kategoriye (yiyecek, giyim, elektronik eşya, kozmetik vb.) ve mağaza adına göre yapabilecek.

6-Android telefonu olmayan öğrenciler uygulamayı emulator üzerinde gerçekleyebilirler.

### Bulut Platformunda

1-Veri tabanı tutulacak. Tablomuzda FirmaID, FirmaAdı, FirmaLokasyon, KampanyaIcerik ve KampanyaSuresi bilgileri tutulacaktır.

2-Bulut platformu üzerinde geliştireceğiniz API ile mobil uygulamamızın haberleşmesi gerekmektedir.

3-API'nizin ara yüzünden kampanya girişi yapılabilecek (Veri tabanına el ile kampanya girişi yapılmayacaktır).

## 1. Giriş

Akıllı reklam sistemi uygulamasında firebase ile haberleşmesini sağlama için java masaüstü uygulaması geliştirerek işlemlerimizi yaptık. Firebase'in bize sağladığı api ile android kısmında veriler arası bağlantı kurarak projenin son haline ulaştırdık. Ancak proje geliştirilirken bazı durumları göz önünde bulundurulması gereklidir. Notification adımları çok dikkatli yapılmalıdır. Gps ve elle giriş işlemleri de ayrıyetten çok ayrıntılıdır. Mobil uygulama için hazırlanan bu özel yazılım, açık kaynak olması için geliştiricilere herhangi bir teknolojik ilerlemeyi tanıtmaya ve dahil etme fırsatı verir. Android'i tablet veya telefonlara kurarak, kullanıcılar, telefon için interaktif ve etkili bir uygulama ve işletim yazılımı başlatabilir. Bu uygulama android mantığını ve api ile haberleşme mantığını çok iyi kurmanızı sağlar ve bu alanda çok katkı sağlamaktadır

## 2. Temel Bilgiler

### Java

İlk bilgisayarlar ortaya çıktığında, şimdi kullandığımız FORTRAN, COBOL, Pascal, C/C++, Java gibi üst düzey diller yoktu. Farklı makinelerin donanım yapıları farklı olduğu için, farklı marka ve model bilgisayarların makine dilleri (makinelerin anladığı tek dil) birbirlerinden farklı idi. O nedenle, ilk programcılar, bilgisayara, ancak kullandıkları makinenin *makine dili* ile iş yaptırabiliyorlardı. Makine dilini öğrenmek zor olmak yanında, bir makine için öğrenilen *makine dili*, farklı marka ve modeller için geçerli olmuyordu. Bu zorluğu aşmak için önce *assembler* dili ve hemen arkasından *üst düzey diller* ortaya çıktı. Üst düzey diller, programcılara büyük rahatlık getirdi. Çünkü, programcı, işletim sistemini ve makineyi düşünmeden, kaynak programını istediği dille yazabiliyordu. Örneğin, C ile yazılan bir kaynak programı, uygun derleyici ile her işletim sisteminde ve her makinede derlenip çalıştırılabiliyordu. [Tabii, C derleyicisi yazan farklı şirketlerin standarttan ufak sapmalarını göz ardı ediyoruz.]

Şimdi adına prosedürel diller denilen FORTRAN, COBOL, Pascal, Modula, C gibi üst düzey diller uzun yıllar programcılara başarıyla hizmet ettiler; halen de etmektedirler. Ama ortada bir problem vardı. Belirli bir işletim sistemi altında çalışan belirli tür bir makinede derlenen bir kaynak program, ancak o işletim sistemi altında çalışan o tür makinelerde koşturulabiliyordu. İşletim sistemi ve/veya makine türü değişince; yani platform değişince, program orada koşamıyordu; yeni platforma uygun bir derleyici ile yeniden derlenmesi gerekiyordu. Buna, kısaca platform bağımlılık diyoruz. Platform bağımlılıktan kastettiğimiz şey, bir kaynak programın belirli bir işletim sistemi altında çalışan belirli bir derleyici ile derlendiği ve ancak belirli bilgisayar türlerinde koşturulabildiği gerçeğidir. Örneğin, *Windows* işletim sistemi altında çalışan bir PC de derlenen bir bilgisayar programını *Macintosh* ya da *Linux* işletim sistemi altında çalışan bir makinede koşturmak mümkün değildir.

Piyasaya farklı işletim sistemlerinin ve çok sayıda farklı donanımın çıkması, bilgisayar ağlarının gelişmesi, bir ağa bağlı makinelerin farklı işletim sistemleri altında çalışan farklı marka ve model bilgisayarlardan oluşuyor olması problemi büyüttü.

Problemin çözümü, platformdan bağımsız çalışabilecek bir dilin yaratılmasını gerektiriyordu. Bu iş, *Sun* firmasının geliştirdiği *Java* dili ile çözüldü. Gerçi *Sun* firması, ortaya çıkan bu büyük problemi çözmek için değil, daha basit bir amaçla yola çıktı. Elektrikli ev aletlerinin kolay kullanılmasını sağlayacak bir dil geliştirmek istiyordu. Kısa zamanda, bu işin ancak ortak bir platformda yapılabileceğini anladı. *Sun*'ın programcısı *James Gosling*, 1995 yılında her platformda çalışan *Java* dilini yarattı.

*Gosling* basit ama harika bir buluş yaptı. Farklı işletim sistemleri ve farklı donanımlarda kurulabilen ortak bir sanal makine tasarladı. Adına JVM (Java Virtual Machine) denilen bu sanal makine ücretsiz dağıtıldı. JVM her platforma kolayca kurulabilir. *Java* dili ile yazılan kaynak programlar, *java* derleyicisi ile JVM de çalışabilen bir tür makine diline dönüştürülür. Adına *java byte kod* denilen bu program JVM tarafından yorumlanır ve çalıştırılır. *Java byte kod* bir bakıma, prosedürel dillerde yazılıp derlenmiş kaynak programların işlevini yapar.

*Java* uygulamalarını çalıştırmak isteyen herkes, kendi makinesine JVM 'i kurabilir. Bunun için JRE (Java Runtime Environment) adlı programı internetten indirip yüklemesi yeterlidir. JRE bilgisayara bir kez yüklenir. Ondan sonra *Java* uygulamalarının hepsi bu makinede çalışabilir. Makinede bir *java* uygulaması çalışacağı zaman, JRE kendiliğinden JVM sanal makinesini oluşturur. JVM gerektiğinde çalışan bir programdır; her program gibi, işi bitince ana bellekten silinir. Dolayısıyla

makineye bir zarar vermesi söz konusu değildir. Günümüzde, *java* uygulamaları ile karşılaşan tarayıcıların çoğu, JRE'yi otomatik olarak indirip yükleme yeteneğine sahiptir.

*Java* basit, modern, nesne yönelimli, tip-korumalı ve C ile C++ dillerinin iyi özelliklerini almış bir dildir. Fazla olarak, her platformda çalışabilme yeteneğine sahiptir. Bu yetenek, onu yalnız bilgisayarlarda değil, internet uygulamalarına, cep telefonlarına, oyun makinelerine, ev aletlerine kadar geniş bir yelpazede uygulama alanı bulmasını sağladı. Bu nedenle, *Java*'yı hem programlama dili, hem de bir ortam olarak düşünebiliriz. Bu ortamda işletim sistemi, ağlar, internet programlama, veritabanı ve bütün orta katman (middleware) teknolojileri yer alır.

Özetle, *Java* basit bir dildir. Kolay öğrenilir. Genel amaçlı bir dilin bütün hünerlerine sahiptir. C, C++ dillerinin iyi özelliklerini almıştır. Kullanıcı dostu ve hızlı bir uygulama geliştirme aracıdır.

## Android Studio

Android Studio ise IntelliJ alt yapısı kullanarak *Java* nesne tabanlı dili ile işlem yapmaktadır. Android Studio Google tarafından önerilmekteydi. Ancak Google kendi mobil programlama dili olan Go dilini piyasaya sürdü. Ama Android Studio'nun kullanılması hala devam etmektedir. Tercih edilmesinin birçok sebebi vardır. Esnek Otomatik zengin tabanlı bir yapı sisteminin olması, içinde barındırdığı hızlı emülatörü, hazır şablonlar ve gitHub entegrasyonu, kapsamlı test araçları ile uygulamanızı farklı sürümlerde test etme imkanı sağlaması ayırt edici ve tercih edilme sebeplerindendir.

## Node JS

Frontend dillerinden en önemlilerinden bir tanesi Javascript'tir. Web tarayıcılarının etkin bir şekilde kullanılması için geliştirilmiştir. Kullanıcıyla etkileşimin artırılması, iş yükünün sunuculardan istemcilere taşınması, daha görsel animasyonlu web siteleri/ web uygulamaları geliştirilmesi gibi kullanım alanları vardır. 1995 yıllarında çıktı. *Java* diliyle isim benzerliği dışında hiçbir alakası yoktur. Javascript standartları Ecma International firması tarafından ECMAScript standartlarıyla belirlenir. Günümüzde kullanılan ECMAScript standardı 5.1 sürümüdür. Ve bu standart Javascript 1.8.1 ile desteklenmektedir. Ancak şu aralar ECMAScript 6 standardı üzerinde çalışmalar devam etmektedir.

Javascript prototip bazı nesne yönelimli programlama, fonksiyonel programlama, imparatif programlama paradigmalarını destekler. Genellikle tarayıcılarda

kullanılır. Google'ın Chrome tarayıcısını ve bu tarayıcı içerisindeki V8 Javascript motorunu geliştirmesiyle Javascript performansında ve gelişimde büyük yükselişler oldu. Bu Javascript motorunun gücü sayesinde daha önce yapılamayan birçok uygulama yapıldı. Ve ilk defa Javascript frontend'den çıkarak Backend tarafına geçiş yaptı ve Nodejs doğdu.

NodeJs 2009 yılında Joyent firmasında çalışan Ryan Dahl tarafından geliştirilen Javascript Çalışma Ortamıdır.(Javascript Runtime Environment). Joyent firmasının desteği ve V8 motorunun gücü ile Dahl Javascript'i Backend kısmına taşıdı.

Javascript'in backend'e(server side) taşınması ile developerlar tek dil bilerek hem frontend hem backend kod yazabilir hale geldiler. Frontend de kullanılan javascript kütüphanelerinden bazıları backend tarafında kullanılabilir hale geldi.

Javascript'in doğal Non Block mimarisi sayesinde backend tarafında ölçeklenebilir uygulamalar daha kolay bir şekilde yazılabilir hale geldi. Nodejs'in ölçeklenebilir uygulamalardaki başarısı sayesinde kullanıcı sayısı milyonlarla ifade edilen büyük siteler nodejs kullanmaya başladı. Üstelik bunu single thread olarak yapabiliyor.

Peki bunu nasıl başarıyor? Nodejs olay bazlı(event-driven) Non-Blocking yapısını Event loop denilen bir mimariyle sağlıyor. (İlerde detayına gireceğiz.) Bu mimariyle tek bir thread(single thread) ile asenkron çalışabilmektedir.

Bunların dışında Nodejs'in en büyük avantajlarından bir taneside modüler yapısı ve bu yapıyı destekleyen paket yönetim sistemi: npm(nodejs package manager). Nodejs npm ve npm yi destekleyen topluluk sayesinde hemen hemen her işimizi paketler sayesinde halledebiliyoruz. Şuan npm üzerindeki paket sayısı yaklaşık 250000 civarında. E okadar paket içerisinde mutlaka işimize yarayan bir paket vardır. Tabi bu kadar paket Nodejs'in arkasında çok büyük bir topluluğun olduğunda gösteriyor. Bu kadar büyük bir topluluğun içerisinde sorunları çözmekte kolay.

## Firestore

Google tarafından yeni özelliklerin eklenmesiyle gelişen Firestore, bütün bu ihtiyaçları karşılayabilme iddiasında ücretsiz kullanım da sunan bir platform. Herhangi bir platformda uygulama geliştirmeye bir sebeple başlayıp daha sonra bir kontrol paneli ve her durumda ulaşılabilir kullanıcı veri deposuna ihtiyacınız olduğunu hemen hissetmişsinizdir. Günümüzde uygulamalar platform farketmeksizin aynı veriye her cihazdan erişmek istiyor. Uygulaması, birçok kullanıcı tarafından yüklenen geliştiricilerin de kayıt - oturum bilgilerini tutma, uygulamaların kullanım verilerini analiz etme, yeni duyurular yapmak için aynı zamanda kullanıcıya bildirim

gönderme, uygulamayı test etme gibi işlemleri rahatlıkla yönetebileceği bir yönetim paneli gerekiyor. İşte Google tarafından yeni özelliklerin eklenmesiyle sürekli kendini geliştiren **Firestore**, bütün bu ihtiyaçları karşılayabilmek için uygulama geliştiricilerine ücretsiz kullanım da sunan bir platformdur.

## 3. Geliştirilen Mimari

Proje geliştirilmeye başlamadan önce projenin tasarımı aşamasında hangi dil ile geliştirilmesi gerektiğini kararlaştırdık. Ve java ile Firestore'e veri ekleme işlemi için geliştirmeye kararlaştırdık. Önemli olan isterlerin tamamen karşılanmasıydı. Arayüz tasarımı için tasarlanacak arayüz üzerinden textField okuma işlemleri gerçekleştirdik. Sonra okunan bu verileri Firestore ekleme işlemi gerçekleştirdik. III. dönem Nesneye Yönelik Programlama dersinde gördüğümüz class yapıları, Programlama derslerinde gördüğümüz temel algoritmalar bu projenin mimarisinin gelişmesinde rol oynadı.

### 3.1 Kullanılan Veri Yapıları

Proje java nesne tabanlı olarak geliştirilmesi kararlaştırılmıştı. Firestore'e veri ekleme işlemleri java ile yapılmaktadır. Ve masaüstü uygulaması ile gerçekleştirilmiştir. Android kısmında geliştirilmesi istenen birkaç tane ister bulunmaktadır. Kullanıcı, kullanıcı adı ve şifresi ile giriş yapabilmeli, şifre değiştirebilmeli veya kayıt olabilmelidir.. Bu isterleri sağlamak için Firestore içerisinde bulunan authentication ile login ekranı girişi sağlanmaktadır. Kullanıcı konum bilgisi iki farklı şekilde alınmaktadır. El ile latitude ve longitude girişi sağlanmalıdır. Veya Gps üzerinden konum alma işlemleri yapılmalıdır. Biz bu işlemleri BroadcastReceiver ile gerçekleştirdik. Daha önceden kullandığımız için bildiğimiz bir konuydu. Ve çok zorlanmadan işlemlerimizi gerçekleştirdik. Kullanıcı, kullanıcı adı ve şifresi ile giriş yapabilmeli, şifre değiştirebilmeli veya kayıt olabilmeli. Ve projede en önemli kısım olan filtreleme işlemidir. Bu işlem için Firestore ile verileri çekerek bir list içerisine yükleme işlemi gerçekleştirdik. Buradan işlemlerimize başladık. Dört tip filtreleme işlemi mevcut. Tarihe göre, firma adına, eşik değerine ve içeriğe göre filtreleme işlemi gerçekleştirdik.

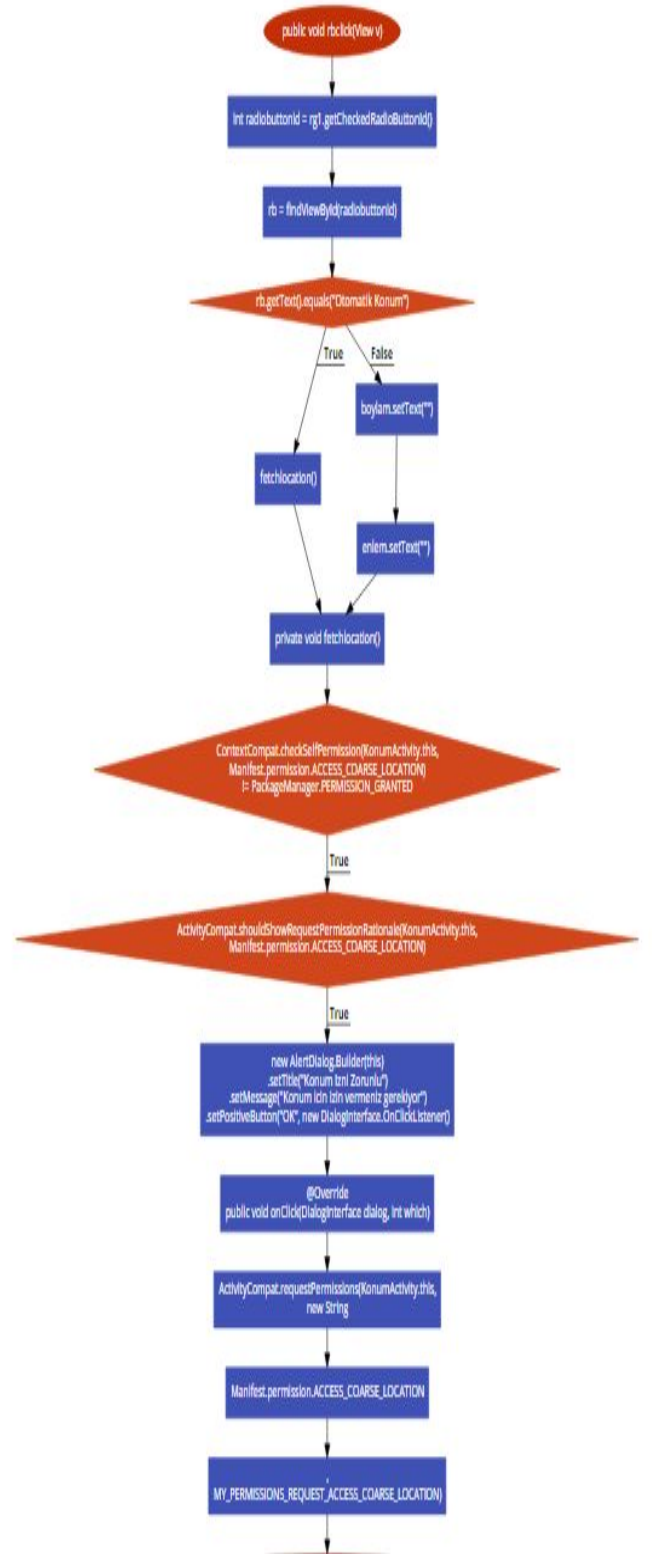
### 3.2 Projenin Yapımı Sırasında Karşılaşılan Sorunlar

Proje tasarım aşamasında oldukça problemle karşılaştık. Proje pdf dosyasını anlamak biraz zamanımızı aldı. İsterleri adım adım yapacağımız için ne istendiğini anlamalıydık. Ve api kısmı için Firebase gibi bulutta api sağlayan bir teknoloji kullanımı kısıtlanmamaktaydı. Bizde bu sayede bulutta api oluşturduk. Ve işlemlerimizi o sayede gerçekledik. Sadece verilere ulaşırken ve java üzerinde veri eklerken sorunlar ile karşılaştık. Veri ekleme örnekleri fazla olmasına karşın çok karmaşık çözümlere ulaşmaktaydı. Bizde Google üzerinde birçok araştırma yaparak sorunlara çözüm üretmeye çalıştık. Ancak java üzerindeki hatalarımıza çözüm ürettikten sonra Firebase database’inde problemler almaktaydık. Veri ekleme işlemimiz çok zamanımızı aldı. Veri çekme işlemi de bir o kadar zamanımızı aldı. Verileri çekerken bazı izinler verilmesi gerekmektedir. Bi böyle bir izin vermediğimiz için yaklaşık 2 saatimizi bu problem için harcadık. Gerekli izinleri verdikten sonra projeye verileri çektik. Ve artık Filtreleme için veri çekmeye başlayacaktık. Ancak daha önce Gps ve el ile konum girişi alınmalıydı. İlk defa kullanacağımız için burada da biraz zaman harcadık. Ve çözüme ulaşmamış internet bağlantımız çözmemizle başladı. Bulunduğumuz konumu Gps ile almak için internet bağlantımızı kontrol ettik. Ve çözüme kavuşturduk. Konum bilgisini Gps ile aldıktan sonraki adımımız filtreleme işlemleri idi. Dört tip filtreleme vardı. Tarih filtrelemesi arka planda yapılmaktadır. Diğer filtreleme işlemleri firma adı, firma içeriği ve eşik değeri kullanıcıdan alınıp gerekli matematiksel işlemler ile çözüme kavuşturuldu.

### 3.3 Yazılım Geliştirme İçin Harcanan Süreler

Proje anlatımı yapıldığı gün neler yapabiliriz hangi kütüphaneleri kullanabilir araştırmaya başladık. Benzer yapılmış uygulamaları tarayıp nasıl bir yol izlemeliyiz buna karar verip proje için çalışmaya başladık ve toplam 20 günde uygulama son halini aldı. Kişi başı yaklaşık çalışma saatleri: Emre Altay 45 saat, Batuhan Subaşı 45 saat.

### 4 Akış Şeması



## 4. Kullanıcı Kataloğu

### 4.1 Giriş ekranı

AkıllıReklamYonetimSistemi

Giris Ekranı

Kullanıcı Adı

Sifre

GİRİS YAP

KAYIT OLMAK İÇİN

SİFRE DEĞİSTİRMEK İÇİN

AkıllıReklamYonetimSistemi

Kayıt Olma Ekranı

Kullanıcı Adı

Sifre

Tekrar Giriniz

KAYIT OL

AkıllıReklamYonetimSistemi

Sifre Degistirme Ekranı

Kullanıcı Adı

Eski Sifre

Yeni Sifre

ŞİFRE DEĞİŞTİR

### 4.2 Uygulama görüntüsü

AkıllıReklamYonetimSistemi

Konum Bilgileri

☒ Otomatik Konum

☐ Manuel Konum

Boylam (Longitude): 29.225817822201872

Enlem (Latitude): 40.95495495495496

Uzaklık (Metre Cinsinden):

Mağaza Adı (ZORUNLU DEĞİL):

Mağaza Türü (ZORUNLU DEĞİL):

Otomobil Alım Satı..

Yemek

Kıyafet

KAYDET

**AkıllıReklamYonetimSistemi**

Konum Bilgileri

☒ Otomatik Konum  
☐ Manuel Konum

Boylam (Longitude): 29.225817822201872

Enlem (Latitude): 40.95495495495496

Uzaklık (Metre Cinsinden): \_\_\_\_\_

Mağaza Adı (ZORUNLU DEĞİL): \_\_\_\_\_

Mağaza Türü (ZORUNLU DEĞİL): Otomobil Alım Satı..  
Yemek  
Kıyafet

**KAYDET**

#### 4.3 Bildirim

00:39 Mayıs 28, Sal

Wi-Fi Konum Titreşim Ekran döndürme Bluetooth

Otomatik

Firma İsmi: yerlerkarisik 00:39  
Kamp.Turu: Yemek Kamp.İcerik: yerlerkari..

Yalnızca acil aramalar. 29.225817822201872 **TEMİZLE**

Boylam (Longitude): \_\_\_\_\_

Enlem (Latitude): 40.95495495495496

Uzaklık (Metre Cinsinden): 500000000

Mağaza Adı (ZORUNLU DEĞİL): \_\_\_\_\_

Mağaza Türü (ZORUNLU DEĞİL): Yemek

**KAYDET**

## 5. Sonuçlar

Akıllı reklam yönetim sistemi uygulaması günümüzdeki teknolojiler kullanılarak bizlere ileriye yönelik büyük bir adım attırılmıştır. Bulut ortamıyla çalışma fırsatı sağlayan bu uygulama, api haberleşme aracını otomatik olarak vererek büyük bir yükten bizler korumaktadır. Firebase gibi bir teknoloji günümüzde çok popüler bir bulut ortamıdır. Bulut ortamı içerisinde authentication ve database işlemleri ile bu uygulamayı kolayca gerçekleyebildik.

## 6. Kaynakça

[1]geodatasource.com.(n.d)."SampleCodes (Java)".[Online].Adres:<https://www.geodatasource.com/developers/java>. [Ulaşım Tarihi: Mayıs 25,2018].

[2]chantisandroid.blogspot.com."GetCurrent LocationExample".[Online]. Adres:<https://chantisandroid.blogspot.com/2017/06/get-current-location-example-in-android.html>. [Ulaşım Tarihi: Mayıs 25,2019].

[3]stackoverflow.com.(n.d)."Calculating distance between two points, using latitude longitude?".[Online]. Adres:[https://stackoverflow.com/questions/3694380/calculating-distance-between-two-points-using-latitude-longitude\\_](https://stackoverflow.com/questions/3694380/calculating-distance-between-two-points-using-latitude-longitude_)[Ulaşım Tarihi: Mayıs 25,2019].

