

AKILLI REKLAM YÖNETİM SİSTEMİ

Mehmet Oğuz Aktaş (150201170), Mert Var (150201222)

oguzaktass@gmail.com, var_mert@hotmail.com

Kocaeli Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Özet

Bu projede, Android mobil uygulaması üzerinden bulut bilişim teknolojisini kullanarak; authentication, GPS tracking ile lokasyon tespit etmek ve gerçek zamanlı veritabanı işlemlerinin yapılması amaçlanmaktadır. Projede, Google Cloud Firebase, Amazon Web Services ve Microsoft Azure gibi bulut platformlarını araştırarak başladık. GPS işlemleri, bildirim ve authentication işlemlerinde kullanımını daha kolay bulduğumuz için Google Cloud ile Firebase'i kullanmayı tercih ettik. Öncelikle Firebase platformu üzerinden proje oluşturarak, bulut platformunda oluşturduğumuz projeyi, Android uygulamasına bağladık. Daha sonra MainActivity classı içinde Firebase authentication ve veritabanına erişim sağlayarak Giriş Yap, Şifremi Unuttum ve Kayıt Ol işlemlerinin yapılmasını sağladık. Kullanıcı girişi yapıldıktan sonra Dashboard activity classı içinde yine Firebase authentication ile veritabanı ve Google Map API'ye erişim sağlayarak Şifremi Değiştir, Reklam Ekle, Reklam Ara ve Çıkış Yap butonları ile ayrı Activity classları oluşturup bu işlemlerin yapılabilmesini sağladık. Projenin en önemli özelliği olan Reklam Ara seçeneği için FindAdvertisement activity classını tanımladık, bu class içinde getLocation(), getData(), getInfo(), findAd(), getNotification() metotlarını yazdık. Arama sonuçlarına göre bulunan reklamları ListAdvertisement activity classı ile listeleyp, her arama sonucu için bildirim gösterilmesini sağladık.

1. Giriş

Bu projede; bulut bilişim teknolojisi kullanılarak Android mobil uygulama ile istenen özelliklerde authentication işleminin yapılması, bulut tabanlı

veritabanına bağlanılarak veritabanı işlemlerinin yapılması, GPS tracking ile lokasyona, mesafeye ve diğer arama seçeneklerine göre arama yapılması, bulunan sonuçlar için mobil uygulamada bildirim gösterilmesi istenmektedir. Ayrıca mobil uygulamada kullanıcının istediği durumlara göre (kategoriye, mağaza adına göre) arama yapılabilmesi de istenmektedir.

Projeye Google Firebase cloud üzerinden proje oluşturarak başladık. Projeyi oluşturduktan sonra generate edilen google-services.json dosyasını projede istenen dizine ekledik. Firebase authentication, veritabanı ve bildirim işlemleri için kullanacağımız kütüphaneleri Gradle ile import ettikten sonra, Android uygulama ile Firebase cloud bağlantısını 2 farklı şekilde sağladık; ilk olarak Android üzerinden Database activity classı ile gerekli kütüphaneleri import edip Firebase'e bağlanarak veritabanı işlemlerini yaptık fakat daha sonra Android üzerinden yapılması istenmediği için C# form arayüzü ile (istenen API özelliklerini karşılayan) yeni bir uygulama yazdık. Bu uygulamada Firebase veritabanı URL'sini ve Database secret keyini kullandık. Uygulamada giriş yapma, kayıt olma, şifremi unuttum, giriş yaptıktan sonra üye panelinin açılması, Firebase'e bağlanarak veritabanına kayıt ekleme, reklam arama (bu activity içinde GPS üzerinden lokasyon bulma işlemini de yaptık), bulunan reklamları listeleyp bildirim gösterme işlemlerinin her biri için ayrı activity classları yazdık.

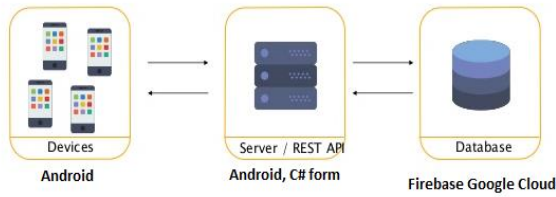
2. Temel Bilgiler

Uygulamayı geliştirirken Firebase Authentication, Firebase Core, Firebase Database, Google Play Services Map, Firebase Messaging ve Firebase UI Database kütüphaneleri ile, Java EE Development Kit 8u202 sürümüyle Android Studio 3.4'ü, IntelliJ IDEA 2019.1.1 ve Visual Studio 2017 IDE'lerini

kullandık. Android uygulamada ListView ve Snackbars gibi çeşitli tasarım componentleri ile GPS üzerinden mevcut lokasyonu tespit etmek için Google Maps kütüphanelerini ve Google Maps API'yi kullandık.

3. Yazılım Mimarisi

Proje; Android uygulaması dışında Firebase ile Android bağlantısını sağlayıp veritabanı işlemlerini yapmak amaçlı yazdığımız istenen API özelliklerini karşılayan bir C# uygulamasından oluşuyor.



Şekil 1: Genel sistem mimarisi

Android uygulamada; MainActivity, SignUp, ForgotPassword, Dashboard, Database, MapActivity, FindAdvertisement, ListAdvertisement activity classları ile Reklam, ReklamListAdapter ve SendNotification Java classları bulunuyor. Firebase özelliklerini kullanabilmek için Google Play Services, Firebase Core, Firebase Authentication, Firebase Database, GeoFire ile Firebase UI Database kütüphanelerini import ettik.

Sırasıyla kullanıcı girişi, kayıt ol ve şifremi unuttum seçenekleri için yazdığımız MainActivity, SignUp ve ForgotPassword activity classları içinde kullandığımız metotlardan bazıları şunlar;

- loginUser(): String tipinde email ve şifreyi parametre olarak alan bu metot ile FirebaseAuth classını kullanarak Firebase Authentication bağlantısı kurarak girilen kullanıcı adı ve şifre doğru ise giriş işleminin yapıp DashBoard activity'sinin başlatılmasını, eğer 6 karakterden küçük bir şifre girilmişse Snackbar componenti ile hata mesajının gösterilmesini sağladık.
- resetPassword(): ForgotPassword activity'si içinde, String email parametresi alan yazdığımız bu metot ile FirebaseAuth hazır classını kullanarak Firebase veritabanı bağlantısı sağlayıp, girilen email adresine kullanıcı şifresinin gönderilmesini sağladık.

- signUpUser(): Yine aynı şekilde FirebaseAuth hazır classı ile Firebase bağlantısı kurup girilen email ve password ile Firebase'de yeni kullanıcı oluşturulmasını bu metot ile sağladık.

Kullanıcı girişi yapıldıktan sonra açılan DashBoard activity classı ve o class içinden butonlar ile erişilebilen Database, MapActivity, FindAdvertisement, ListAdvertisement activity classları içinde kullandığımız önemli metotlar şunlardır;

- logoutUser(): DashBoard classında kullanıcının çıkışı yapmasını sağlamak için bu metodu yazdık. Çıkış Yap butonuna basıldığında oluşturduğumuz FirebaseAuth'u null yaparak ve ana ekrana dönerek çıkışı yapılmasını sağladık.
- ChangePassword(): Arayüzdeki EditText ile girilecek String tipindeki yeni şifreyi parametre olarak alan bu metot ile kullanıcının şifresini değiştirebilmesini sağladık.
- addAdvertisement(): Database activity classı içindeki bu metot ile firma adı, latitude, longitude, kampanya içerik ve kampanya süresi özelliklerinin kullanıcıdan alınarak, Firebase'in DatabaseReference ve FirebaseDatabase özelliklerini import edip veritabanına kayıt ekleme işleminin yapılabilmesini sağladık. Unique bir FirmaID özelliği oluşturabilmek için DatabaseReference classının push() ve getKey() hazır metotlarını kullandık.
- onMapReady(): MapActivity classı içinde Google Maps özelliklerini ve diğer hazır metotları kullanarak mevcut lokasyonun GPS üzerinden bulunup gösterilmesini ve Geri Dön butonuna basıldığında açılan DashBoard'da bulunan latitude ve longitude değerlerinin tutulmasını sağladık. MapActivity classında GPS işlemleri için GoogleMap, GoogleApiClient, Location, LocationRequest kütüphaneleri ile hazır classlarını kullandık.
- getLocation(): FindAdvertisement classında latitude ve longitude değerleri için kullanıcı tarafından elle girme veya GPS üzerinden otomatik bulma seçenekleriyle latitude ve longitude değerlerinin alınmasını sağladık.
- getData(): Bu metot ile Firebase veritabanı bağlantısı sağlayıp tüm reklamların çekilmesini ve oluşturduğumuz Reklam classı tipindeki ArrayList'e kaydedilmesini sağladık.
- getNotification(): Bildirim göstermek için Notification ve NotificationManager gibi hazır

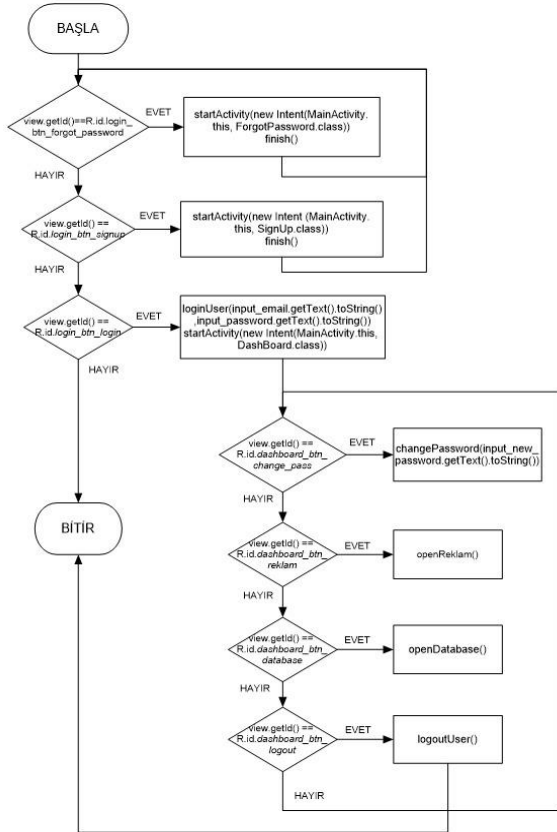
classları kullanarak her bulunan arama sonucu için firma adı ve kampanya içeriğinin bulunduğu bir bildirim gösterilmesini sağladık.

- getInfo(): MapActivity ile GPS üzerinden lokasyon bulunduğunda ve reklam listeleme ekranından reklam arama ekranına geri dönüldüğünde latitude ve longitude özelliklerinin classlar arasında aktarılması için bu metodu kullandık. Özellikleri aktarabilmek için getIntent() ile getExtras() hazır metotlarını kullandık.

Bunların dışında her activity’de onCreate(), onClick() ve diğer birçok hazır metot kullandık. onClick() metotları ile Geri Dön butonuna basıldığında önceki ekrana dönülmesini sağlamak gibi birçok tıklama özelliğinin yapılmasını sağladık.

Reklam, ReklamListAdapter ve SendNotification Java classları ile de sırasıyla reklam özellikleri ile get set metotlarının tanımlanması, arama sonuçlarında bulunan reklamların düzenli biçimde listelenmesinin sağlanması, bildirim kanalının oluşturulup NotificationManager ile kullanılması işlemlerinin yapılmasını sağladık.

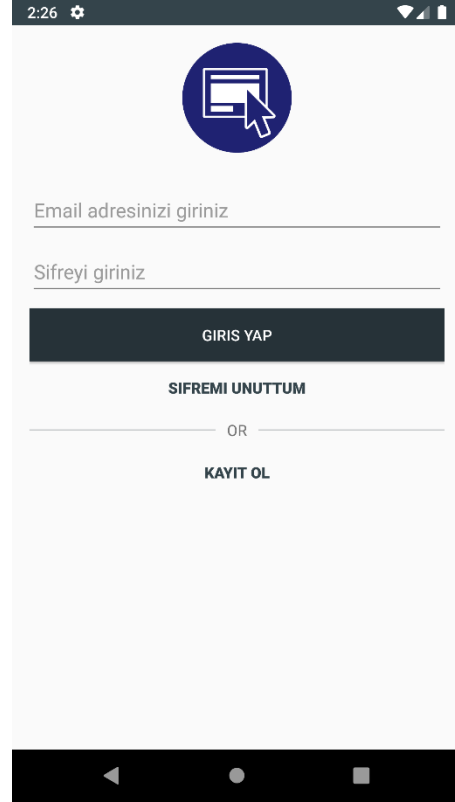
4. Akış Şeması



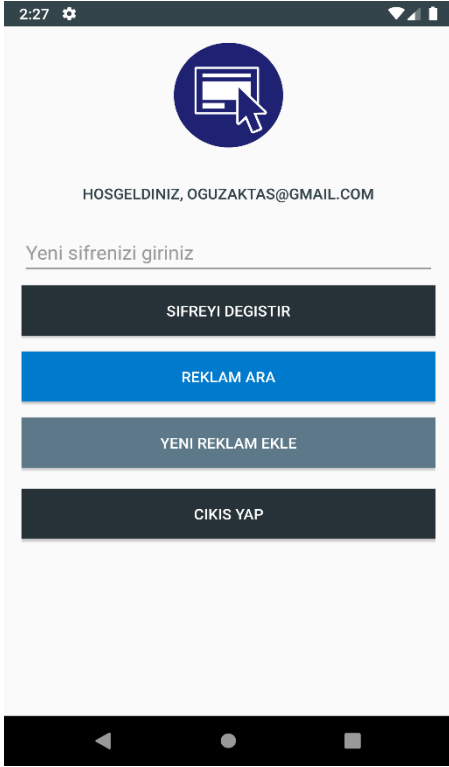
Şekil 2: Akış şeması

5. Kullanıcı Kataloğu

Uygulamayı çalıştırınca ilk olarak email ve şifre ile giriş yapma ekranı karşımıza geliyor. Buradan şifremi unuttum ekranına ve kayıt ol ekranına geçiş yapılabilir.

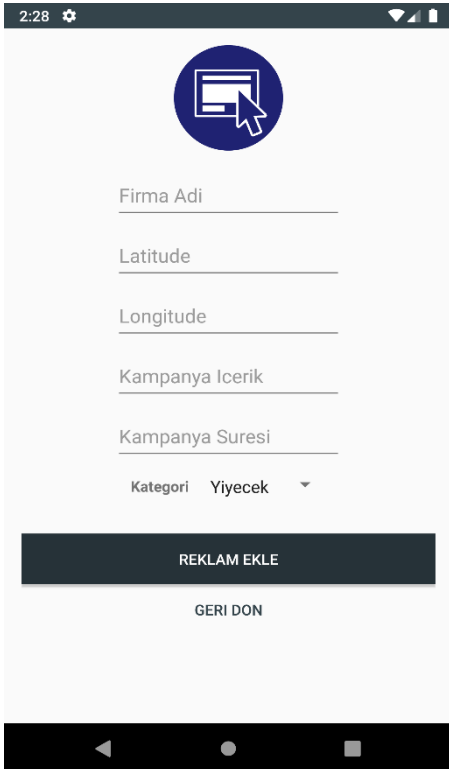


Şekil 3: Programın ilk çalıştırılması ve giriş yapma
Giriş yapıldıktan sonra kullanıcı paneli açılır.



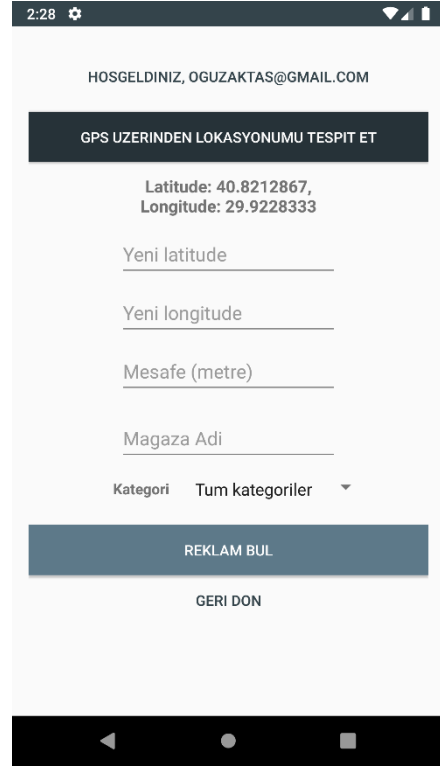
Şekil 4: Kullanıcı paneli seçenekleri

Kullanıcı panelinde yeni reklam ekle seçeneği ile Firebase bulut üzerinden reklam eklenebilir;



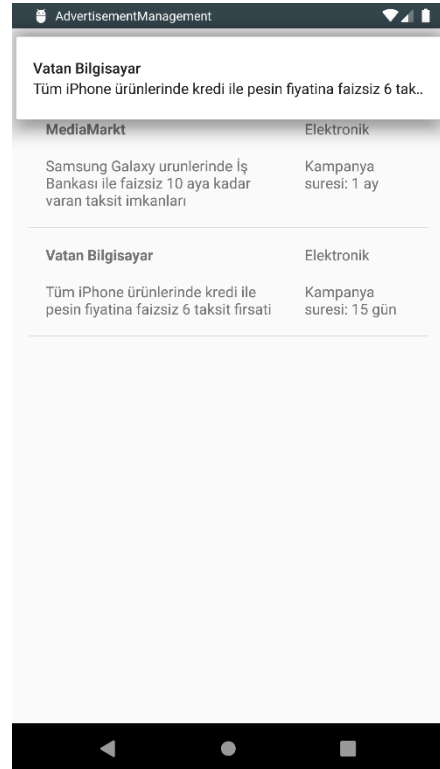
Şekil 5: Firebase bağlantısı ile reklam ekleme

Reklam arama panelinden çeşitli seçeneklere göre arama işlemi yapılabilir;



Şekil 6: Reklam arama

Arama sonucuna göre bulunan her bir reklam için bildirim oluşturularak liste şeklinde gösterilir;



Şekil 7: Arama sonucu

6. Sonuçlar

Bu proje ile bulut tabanlı bir sistem üzerinden Android uygulama ile haberleşmenin nasıl yapıldığını, bulut platformundan authentication işlemleri ve veritabanı işlemlerinin nasıl yapılacağını, GPS üzerinden lokasyonu tespit etme ve diğer Google Maps API özelliklerini, Android’de nasıl bildirim gösterileceğini ve Snackbar gibi componentlerin nasıl kullanılacağını, latitude ve longitude değerlerini kullanarak uzaklık hesaplama ve diğer lokasyon işlemlerinin nasıl yapıldığını öğrendik. Ayrıca Firebase dışında AWS ve Microsoft Azure’yi de araştırarak bulut platformlarının genel olarak nasıl çalıştığı hakkında bir fikir sahibi olduk.

7. Kaynakça

- [1] <https://stackoverflow.com/>
- [2] <https://firebase.google.com/docs>
- [3] <https://developer.android.com/docs>
- [4] <https://developer.android.com/studio/write/firebase> (Erişim tarihi: 23.05.2019)
- [5] <https://cloud.google.com/solutions/mobile/mobile-firebase-app-engine-flexible> (Erişim tarihi: 23.05.2019)
- [6] <https://developers.google.com/android/guides/client-auth?authuser=0> (Erişim tarihi: 23.05.2019)
- [7] <https://www.youtube.com/watch?v=9ARoMRd1kXo> (Erişim tarihi: 23.05.2019)
- [8] <https://www.youtube.com/watch?v=4kk-dYWVNsc> (Erişim tarihi: 24.05.2019)
- [9] <https://code.luasoftware.com/tutorials/android/android-get-current-location/> (Erişim tarihi: 24.05.2019)
- [10] <https://developer.android.com/reference/android/support/design/widget/Snackbar> (Erişim tarihi: 24.05.2019)
- [11] <https://www.youtube.com/playlist?list=PLzV8uW>

[UcseN91J9bLuVLnlmWlE6wIWTiY](#) (Erişim tarihi: 25.05.2019)

[12] <https://www.zoftino.com/current-location-and-nearby-places-android-example> (Erişim tarihi: 25.05.2019)

[13] <https://www.sanfoundry.com/java-android-program-calculate-distance-two-points-using-latitude-longitude/> (Erişim tarihi: 25.05.2019)

[14] <https://www.latlong.net/> (Erişim tarihi: 25.05.2019)

[15] <https://firebaseopensource.com/projects/firebase/firebaseui-android/> (Erişim tarihi: 26.05.2019)

[16] <https://developer.android.com/reference/android/app/NotificationManager> (Erişim tarihi: 26.05.2019)