AKILLI REKLAM YÖNETİM SİSTEMİ

Muhammed Emre KARA, Muhammed Ali Dilekçi

Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi

mailemrek@gmail.com madilekci@gmail.com

1. Özet

Bu projede konum tabanlı girilen reklamların yönetilmesi ve kullanıcıya ulaştırılması hedeflenmiştir. Android OS platformu için geliştirilen proje JAVA diliyle geliştirilmiş ve Firebase Cloud sistemi kullanılmıştır.

2. Giriş

Mağazaların girişi yapılırken **Admin** arayüzü kullanılır. mağazaya ait ad, konum, kampanya süresi, kategori(AVM, YEMEK, KOZMETİK, GİYİM, EĞLENCE, KUAFÖR, OTOPARK) bilgileri kullanıcıdan alınır. Kullanıcı aynı zamanda mağaza konumu olarak anlık GPS konumunu kullanabilmektedir. Daha sonra oluşturlan mağaza bilgisi Firebase Database'de depolanır. Normal kullanıcı arayüzünde ise var olan kampanyaları **tümü, kategoriye göre, isme göre, hem kategori hem isme göre** olarak 4 farklı filtre ile listeleyebilmektedir. Listelenen kampanyalar eğer **mesafe** belirtilmemişse tüm uzaklıklar için eğer belirlenmişse belirlenen eşik değeri ve daha altı için listelenir. Kullanıcının yaptığı aramaya uygun mesafede hiç mağaza bulunulmaması durumunda kullanıcı eğer hareket ederek mağazaların belirtilen mesafe değeri kadar yakınına gelirse anlık bildirimle uyarılacaktır. Bunun dışındaki aramalar yine konum bilgsinin el ile girilmesi ile de listelenebilmektedir.

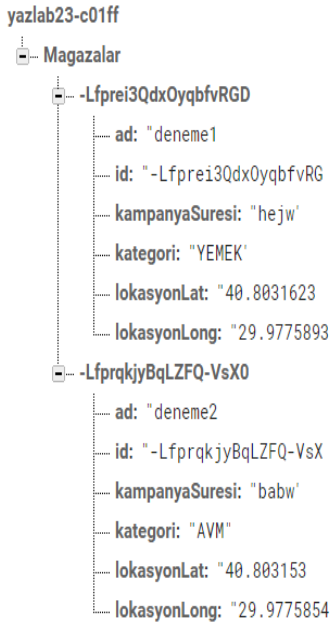
3. Temel Bilgiler

Projeyi gerçekleştirirken kullandığımız teknolojiler aşağıda verilmiştir.

Firebase Database : Firebase gerçek zamanlı veri tabanı,bulut tabanlı NoSql(Not Only Sql) bir veri tabanı sistemidir.Hiç bir sql sorgusuna gerek duymadan json parametreleri ile yönetebilirler. Veri depolamanın yanı sıra asenkron çalışması ile veri değişimlerinin anlık olarak takip edilmesine olanak sağlar.

Firebase Authentication: Firebase Auth, kullanıcıları sadece müşteri tarafı kodunu kullanarak doğrulayabilen bir hizmettir. Sosyal giriş sağlayıcılarını Facebook, GitHub, Twitter ve Google (ve Google Play Games) destekliyor. Ek olarak, geliştiricilerin Firebase'de depolanan e-posta ve şifre girişi ile kullanıcı kimlik doğrulamasını etkinleştirebilecekleri bir kullanıcı yönetim sistemi içerir.

Firebase Database'in Yapısı:



Firabase Authentication'un yapısı:

Kayıt olan kullanıcılara bir **userID** ataması yaparak verilen giriş yöntemine(bu projede yalnızca email ve şifre ile kayıt kullanılmıştır) bir kayıt oluşturur. Proje yöneticisi dahil hiçkimsenin şifreyi açıkça görme yetkisine sahip değildir. Yalnızca doğrulama için saklanır.

Bu modül Şifreyi Değiştirme, Şifremi Unuttum, Email doğrulama gibi servisleri de içerir.

4. Proje İçeriği

Magaza.class:

Kaydı oluşturulan mağazaların ram'de saklanması ve Firebase Database'e yazarken kolaylık sağlanması açısından oluşturulmuş bir sınıftır. Dökümanda belirtilene ek olarak mağaza ID'sini içerir bu ise Database'e yazılan key'in tutulabilmesi yani gerçek konumuna ulaşılabilmesi için eklenmiştir.

```
String id;
String ad;
String kategori;
String lokasyonLong;
String lokasyonLat;
String kampanyaSuresi;

public Magaza(String id, String ad, String
kategori, String lokasyonLong, String
lokasyonLat, String kampanyaSuresi) {
    this.id = id;
    this.ad = ad;
    this.kategori = kategori;
    this.lokasyonLong = lokasyonLong;
    this.lokasyonLat = lokasyonLat;
    this.kampanyaSuresi = kampanyaSuresi;
}
```

MainActivity.java

Sadece Fragmentlerin implementasyonu ve **Anasayfa.java** 'a yönlendirmek için kullanılır. Android Tarafından zorunlu tutulur.

```
implements
AuthenticationFragment.OnFragmentInteracti
onListener,
SifreyiDegistirFragment.OnFragmentInteract
ionListener {


AuthenticationFragment auth = new
AuthenticationFragment();
        setFragment(auth);

    }
```

AuthenticationFragment.class:

Bu yapı bir Fragment yapısıdır.

Fragment: API Level 11+ sonrasında geliştiricilere sunulan Fragment'lar bir Activity içerisinde çalışan bir user interface'i ya da bir process'i temsil eden küçük activity parçaları olarak tanımlanabilir. Fragmentlar, Activity sınıflarına göre daha performanslı ve daha kullanışlı olarak karşımıza çıkmaktadır .

Kullanıcıdan mail ve şifre alır.

Kayıt Ol butonuna basılması durumunda kriterlere uygunsa yeni bir kullanıcı oluşturur. **FireBaseAuthentication** modülünün **createUserWithEmailAndPassword** metodunu kullanır.

```
mAuth.createUserWithEmailAndPassword(email
, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void
onComplete(@NonNull Task<AuthResult> task)
{
  if (task.isSuccessful()){

Log.d(TAG, "createUserWithEmail:success");

Toast.makeText(getContext(), "Giriş
Yapılıyor Lütfen Bekleyiniz.",
Toast.LENGTH_LONG).show();

signIn(email,password);
                }
```

Giriş Yap butonuna basılması durumunda kriterlere uygunsa **AnasayfaActivity** başlatılır. **FireBaseAuthentication** modülünün **signInWithEmailAndPassword** metodunu kullanır..

```
mAuth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
@Override public void onComplete(@NonNull
Task<AuthResult> task) {
if (task.isSuccessful()) {
                    Log.d(TAG,
"signInWithEmail:success");
Intent adminActivity = new
Intent(getContext(), AdminActivity.class);

startActivity(adminActivity);
```

AdminActivity.class:

Bu Activity yalnızca admin kullanıcısı giriş yaptığında açılır. Mağzaların eklendiği yönetim arayüzüdür.

Mağazaların girişi için kullanıcıdan. mağazaya ait ad, konum, kampanya süresi, kategori (AVM, YEMEK, KOZMETİK, GİYİM, EĞLENCE, KUAFÖR, OTOPARK) bilgileri kullanıcıdan alınır. Kullanıcı aynı zamanda mağaza konumu olarak anlık GPS konumunu kullanabilmektedir. Daha sonra oluşturulan mağaza bilgisi Firebase Database'de "**Mağazalar**" referansında depolanır.

Lokasyon İşlemleri :

```
locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE)
;
```

locationManager anlık konumu alabilmek için konum seervisini aktif hale getirir.

```
locationManager.requestLocationUpdates(Loc
ationManager.GPS_PROVIDER, 1000, (float)
0.2, this);
```

OnLocationChange metodunu tetiklemek için istenen süre ve hassaslığı ayarlar

```
this.konumLocation =
locationManager.getLastKnownLocation(Locat
ionManager.NETWORK_PROVIDER);
```

Anlık konumu GPS'den alır ve saklar.

Mağaza Kayıt İşlemleri :

```
DatabaseReference Magazalar =
db.getReference().child("Magazalar");

id = Magazalar.push().getKey();
Magaza magaza = new Magaza(id, ad,
kategori, lokasyonLong, lokasyonLat,
kampanyaSuresi);

DatabaseReference magazaYaz =
Magazalar.child(id);
magazaYaz.setValue(magaza);

Toast.makeText(this, "Mağaza Başarıyla
Eklendi", Toast.LENGTH_LONG).show();
```

SifreyiDegistirFragment :

Kullanıcıdan email ve şifre bilgilerini alır ve doğrulama için **FireBaseAuthor**'a yollar. Kullanıcının kaydı varsa şifreyi değiştirmesi için yeni bir Text alanı açarak şifreyi 2 kez doğrulamayla alır. Kriterlere Uyması durumunda kullanıcının şifresini günceller ve kullanıcıyı **AuthenticationFragment** 'a yönlendirir

```
signIn(email, sifre);

!(editTextSifreyiDegistirFragmentSifre.get
Text().toString().equals(editTextSifreyiDe
gistirFragmentSifreDogrula.getText().

user.updatePassword(editTextSifreyiDegisti
rFragmentSifreDogrula.getText().toString()
)
                .addOnCompleteListener(new
OnCompleteListener<Void>() {
                    @Override
                    public void
onComplete(@NonNull Task<Void> task) {
                        if
(task.isSuccessful()) {
                            Log.d(TAG,
"User password updated.");

Toast.makeText(getContext(),"Şifre
Başarıyla
Güncellendi!",Toast.LENGTH_LONG).show();

setFragment(new AuthenticationFragment());
                        }
else {
String Hata = "" + task.getException();

String[] kelime = null;
kelime = Hata.split(":");
                            if
(kelime[1].equals(" The given password is
invalid. [ Password should be at least 6
characters ]")) {

Toast.makeText(getContext(), "Şifre En Az
6 Haneli Olmalı.",
Toast.LENGTH_LONG).show();
                            }
                        }
                    }
                });
    }

}
```

AnasayfaActivity :

Uygulumamızın ana activitysi burasıdır kullanıcı arayüzü tamamen buradadır.

Kullanıcıya istediği kategori isim ve lokasyonda istediği uzaklıktaki tüm mağzalar bilgileri ile beraber canlı olarak listelenir.

Kullanıcının berlilediği eşik değerine sonradan giren mağazalar kullanıcıya bildirim olarak gösterilir.

Filtreleme fonksiyonları:

```
private void getData00(final String
lokasyonLongEditTextM, final String
lokasyonLatEditTextM)
```

Tüm kategori ve isimlere göre mağazaları listeler.

```
private void getData01(final String
lokasyonLongEditTextM, final String
lokasyonLatEditTextM, final String
magazaAd)
```

Tüm kategorilere, verilen isime göre mağazaları listeler.

```
private void getData10(final String
lokasyonLongEditTextM, final String
lokasyonLatEditTextM, final String
kategoriM)
```

Tüm isimlere, verilen kategoriye göre mağazaları listeler.

```
private void getData11(final String
lokasyonLongEditTextM, final String
lokasyonLatEditTextM, final String
kategoriM, final String magazaAd)
```

Verilen isim ve verilen kategorideki mağazaları listeler.

```
String magazaAd =
editTextActivityAnasayfaAd.getText().toStr
ing();
if (kategoriSpinner.equals("HEPSİ") &&
magazaAd.equals("")) {
    tableLayoutAnasayfa.removeAllViews();
    initBaslangic();
    getData00(lokasyonLongEditText,
lokasyonLatEditText);
} else if (kategoriSpinner.equals("HEPSİ")
&& !(magazaAd.equals(""))) {
    tableLayoutAnasayfa.removeAllViews();
    initBaslangic();
    getData01(lokasyonLongEditText,
lokasyonLatEditText, magazaAd);
} else if
(!(kategoriSpinner.equals("HEPSİ")) &&
magazaAd.equals("")) {
    tableLayoutAnasayfa.removeAllViews();
    initBaslangic();
    getData10(lokasyonLongEditText,
lokasyonLatEditText, kategoriSpinner);
} else if
(!(kategoriSpinner.equals("HEPSİ")) &&
!(magazaAd.equals(""))) {
    tableLayoutAnasayfa.removeAllViews();
    initBaslangic();
```

AnasayfaActivity(2) :

Bildirim :

```
NotificationManager notificationManager =
(NotificationManager)
getSystemService(Context.NOTIFICATION_SERV
ICE);
NotificationCompat.Builder
notificationBuilder = new
NotificationCompat.Builder(this,NOTIFICATI
ON_CHANNEL_ID);
notificationManager.notify(new
Random().nextInt(),notificationBuilder.bui
ld() );
```

Mesafe Hesaplama :

```
public void
distanceBetweenCoordinates(Double long1,
Double lat1, Double long2, Double lat2) {
    Location loc1 = new Location("LOC1");
    Location loc2 = new Location("LOC2");

    loc1.setLongitude(long1);
    loc1.setLatitude(lat1);
    loc2.setLongitude(long2);
    loc2.setLatitude(lat2);
```

Tablolama :

```
TableRow tbrow = new TableRow(this);
TextView t0v = new TextView(this);
t0v.setText("      " + ad);
t0v.setTextColor(Color.BLACK);
t0v.setGravity(Gravity.LEFT);
t0v.setTextSize(15);
tbrow.addView(t0v);
TextView t1v = new TextView(this);
t1v.setText("      " + mesafe);
t1v.setTextColor(Color.BLACK);
t1v.setGravity(Gravity.LEFT);
t1v.setTextSize(15);
tbrow.addView(t1v);
TextView t2v = new TextView(this);
t2v.setText("      " + kategori);
t2v.setTextColor(Color.BLACK);
t2v.setGravity(Gravity.LEFT);
t2v.setTextSize(15);
tbrow.addView(t2v);
tableLayoutAnasayfa.addView(tbrow);
```

Kaynakça

1) https://www.mobilhanem.com/android-firebase-authentication-kimlik-dogrulama/
2) https://www.youtube.com/watch?v=_siNdB28ySM
3) https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-401/firebase-realtime-database
4) https://firebase.google.com/docs/android/setup

Akış Diagramı