

PROJE RAPORU

Proje Adı: Acil Yardım Hizmetleriyle İletişimi Kolaylaştıracak Mobil Uygulamanın Tasarlanması ve Geliştirilmesi (Conserva)

İÇİNDEKİLER

1. Giriş	2
1.1 Projenin Amacı	2
1.2 Android Platformu	3
1.3 Javascript	3
1.4 Firebase.....	3
1.5 Google Maps.....	3
2. Yöntem.....	4
2.1 Yapım Basamakları	4
a) Mobil Uygulama	4
b) Web Uygulaması.....	4
3. Bulgular ve Gerçekleştirme	4
3.1 Mobil Uygulama	4
3.1.1 Uygulamanın Tasarlanması	4
3.1.2 Haritaların Kurulumu ve Sistem İzinlerinin Alınması	4
3.1.3 Veritabanının Tanıtılması ve Yazma Fonksiyonunun Eklenmesi.....	5
3.1.4 SMS ile Gönderim Fonksiyonunun Eklenmesi.....	5
3.2 Web Uygulaması	6
3.2.1 Uygulamanın Tasarlanması	6
3.2.2 Haritaların Kurulumu	6
3.2.3 Veritabanının Tanıtılması ve Konumun Alınması	6
3.2.4 Harita Üzerine Pin Eklenmesi ve Konumun Açık Adrese Çevrilmesi.....	6
3.3 Veritabanı	6
3.4 Proje Kodlarının Açıklanması.....	6
3.4.1 Android Uygulaması.....	6
a) Manifest Dosyası.....	6
b) MainActivity.java.....	7
c) SesTuslari.java	9
3.4.2 Web Uygulaması	10
a) Firebase ve Maps API Kurulumu.....	10
b) Alınan Konum Bilgisinin Haritada İşaretlenmesi.....	11
3.5 Proje Ekran Görüntüleri	12
4. Sonuçlar ve Tartışma.....	14
5. Öneriler.....	14
6. Kaynakça.....	15

1. Giriş

Güvenlik; toplum yaşamında yasal düzenin aksamadan yürütülmesi, kişilerin korkusuzca yaşayabilmesi durumudur (TDK). Varoluştan bu yana, insanların ve diğer canlıların hayatlarındaki temel ihtiyaçlardan biridir. Amerikalı psikolog Abraham Maslow tarafından 1943 yılında yayımlanmış bir çalışmaya göre (Şekil-1), bireyin kendini gerçekleştirebilme yetisini kazanmadan önce karşılaması gereken birtakım ihtiyaçlar vardır. Bunlardan en önemlisi fizyolojik ihtiyaçlar (solunum, yemek vs.) olurken; ikinci sırada güvenlik ihtiyacı gelmektedir. Bir kategorideki ihtiyaçlar tam olarak gidermeden bir üst düzeydeki ihtiyaç kategorisine geçilemez; dolayısıyla birey, özünde barındırdığı potansiyeli gerçekleştiremez. (Çakmak, 2018)

Güvenlik ihtiyacını karşılamak üzere; insanlar, geçmişten günümüze çeşitli çözümler aramışlardır. Bu arayışın getirdiği sonuçlardan birisi de, insanların telefon aracılığı ile yardım isteyebileceği acil yardım hatları olmuştur. Ülkemizde bu hatların kullanımı 1986 yılında belirli bölgelerde hizmete alınan 077 Hızır Acil Servis ile başlamıştır ve günümüze kadar pek çok hat kullanıma alınmıştır. Günümüzde önemli numaraları 112 çatısı altında birleştirme çalışmaları sürüyor olsa da, henüz Türkiye genelinde yayılmamıştır.

Acil yardım hatlarının yararlı olması tartışmasız bir konu olsa da, birtakım problemleri de beraberlerinde getirmişlerdir. Bu problemlere örnek olarak, sık kullanılan acil yardım numaralarının tek numarada birleştirilmediğinden karıştırılması ve insanların yardım isteyememesi; ciddi durumlarda, adres tarifi yapılabilinecek fırsat bulunamadığında veya açık adres bilinemediğinde acil yardımın ulaşamaması gösterilebilir.

Bu problemlerin farkına varan ülkelerden biri olan Türkiye’de ise, Sağlık Bakanlığı tarafından bir acil yardım uygulaması geliştirilmiş ve mobil platformlarda kullanıma sunulmuştur. İşlevi konum bilginizi alıp sağlık ekiplerinin cihazlarına taşımak olan bu küçük boyutlu uygulama, dünyadaki diğer örneklerine göre gayet basit ve efektif bir çözüm sunmaktadır. Ancak, sadece tıbbi yardım ihtiyacı durumunda kullanılabilir.

Projemizde; Sağlık Bakanlığının geliştirmiş olduğu mobil uygulamaya ek özellikler içeren, daha kapsamlı ve daha efektif çözümler üretmek üzere yola çıkıldı. Maliyet ve taşıma sırasında yaşanacak problemlerden dolayı ekstra bir cihaz tasarımı yapmak yerine, hali hazırda 5.57 milyar kullanıcısı olan mobil platform tercih edildi (Statista’nın verilerine göre). Yardım çağrısı, sadece tıbbi yardımdan ibaret tutulmayıp; polis ve itfaiye gibi aciliyet içeren ihtiyaçların da karşılanabilmesi için daha kapsamlı hale getirildi. Bunların yanında, internet bağlantısının kurulamadığı konumlarda da kullanım için kısa mesaj gibi alternatiflerin düşünülmesinin yanında, zor koşullarda kullanım için ses açma-kapama tuşları ile işlem yapılabilmesi için çözüm üretildi. Proje adı (*Conserva*, Latince), “güvende tutmak” anlamına gelmektedir.

1.1. Projenin Amacı

Projemizin amacı; acil durumlarda insanların yardım istemesi ve yardımın ulaşması süreçlerini en hızlı ve en kolay şekilde çözmektir. Ayrıca, insanların güven ihtiyaçlarını ve isteklerini karşılamak, kendilerini gerçekleştirmelerine destek olmaktır.



Şekil 1 - Maslow'un Gereksinimler Hiyerarşisi

1.2. Android Platformu

Projede, ihtiyaç anında yardım çağrılabilmesi için en maliyetsiz ve yaygın çözümün mobil platform olduğunu gözlemledik. Bu yüzden mobil işletim sistemleri içerisinde en yaygın kullanımı oranına sahip Android Platformu'nu seçtik. Android Platformu, 2018'in ikinci çeyreği itibarıyla %88'lik bir pazar payına sahipken; ona en yakın platform olan iOS ise, %11.9 ile ikinci sıradadır.

Android, Google tarafından 2008 yılında tanıtılmıştır. Mobil platform için geliştirilmiş; Linux tabanlı, açık kaynak kodlu bir işletim sistemidir.

Android platformu üzerine mobil uygulama geliştirebilmek için, Google tarafından sağlanan Android SDK (Software Development Kit) ve onun entegre edilebileceği bir geliştirme ortamı gerekmektedir. Google, 2013 yılı itibarıyla Çek Cumhuriyeti merkezli yazılım şirketi JetBrains tarafından geliştirilen IntelliJ IDEA adlı geliştirme ortamı üzerine inşa ettiği Android Studio'yu tanıtmıştır. Bugün, Android için en tercih edilen geliştirme ortamlarından birisi olmuştur.

Prototipimizin geliştirilmesinde Android Studio kullanılmıştır. Java ile yazılmıştır.

1.3. Javascript

Javascript, genellikle web üzerinde kullanılan dinamik bir programlama dilidir. İstemci tarafında işlenir. Brendan Eich tarafından 1995 yılında LiveScript adıyla kullanıma sunulmuştur. Java dili ile isim benzerliği dışında ortak bir yanı yoktur.

Projemizde, web üzerinde stabil çalışan bir dil olduğu için, yardım ekipleri için geliştirilecek web uygulamasında kullanılmasına karar verilmiştir. Böylelikle yardım ekipleri, merkezlerindeyken bilgisayar ekranlarından; yolculuk anında mobil cihazlardan yardım çağrısı alabilir, alınmış olan yardım çağrılarını görüp detaylarına erişebilir.

1.4. Firebase

Firebase, Google tarafından geliştiricilerin kullanımına sunulmuş bir arka uç servsidir. Hosting (barındırma), saklama alanı, veritabanı gibi hizmetler sunar. Yapılan işlemler kullanıcı tarafından görülmez. Prototipimizin veritabanı, Firebase üzerinde barındırılmaktadır.

1.5. Google Maps

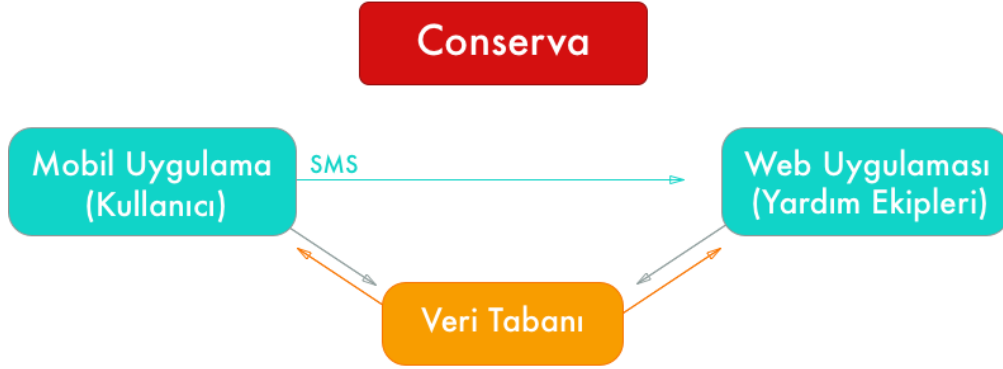
Google Maps, Google tarafından hizmete sunulmuş çevrimiçi harita servsidir. Ayrıca geliştiricilerin de kullanabilmesi için API desteği sunmaktadır.

Prototipimizde; konumların işaretlenmesi Google Maps üzerinden yapılmaktadır.

2. Yöntem

Projenin yapımı iki temel basamakta planlanmıştır. İlk aşama, yardım isteyecek kullanıcı tarafından kullanılacak Android uygulamasının kodlanması ve veritabanına bağlanıp konum bilgilerinin yazılmasıdır. İkinci aşama ise, ilgili ekiplere bilgi sağlayacak olan web uygulamasının kodlanması ve veritabanından alınan konum bilgilerinin harita üzerinde işaretlenmesidir. İnternet bağlantısının bulunmadığı durumlarda ise, konum bilgisi SMS (kısa mesaj) yolu ile ekiplere bildirilecektir. Böylelikle, telefonun sinyal alabildiği her yerde yardım çağrısı yapılabilecektir.

2.1. Yapım Basamakları



Şekil 2 - İlişki Tablosu

a) Mobil Uygulama

- a.1. Uygulamanın tasarlanması
- a.2. Haritaların kurulumu ve sistem izinlerinin alınması
- a.3. Veritabanının tanıtılması ve yazma fonksiyonunun eklenmesi
- a.4. SMS ile gönderim fonksiyonunun eklenmesi

b) Web Uygulaması

- b.1. Uygulamanın tasarlanması
- b.2. Haritaların kurulumu
- b.3. Veritabanının tanıtılması ve konumun alınması
- b.4. Harita üzerine pin eklenmesi ve konumun açık adrese çevrilmesi

3. Bulgular ve Gerçekleştirme

3.1. Mobil Uygulama

3.1.1. Uygulamanın tasarlanması

Mobil uygulama tasarımı, kullanıcının panik durumunda şaşırmadan kullanabileceği kadar basit ve aradığı bilgiyi bulabileceği kadar da kapsamlı tutulmaya çalışıldı.

3.1.2. Haritaların kurulumu ve sistem izinlerinin alınması

Google'ın sağladığı API kodu alındı ve uygulamaya eklendi

Android sistemi üzerinde kullanıcı bilgilerine erişebilmek için manifest.xml dosyası üzerinden gerekli izinler alınmalıdır. Aksi halde, bilgiler istendiğinde uygulama çalışmayı durdurur.

android.permission.ACCESS_FINE_LOCATION
android.permission.ACCESS_COARSE_LOCATION izinleri alındı.

android.permission.SEND_SMS izni alındı.

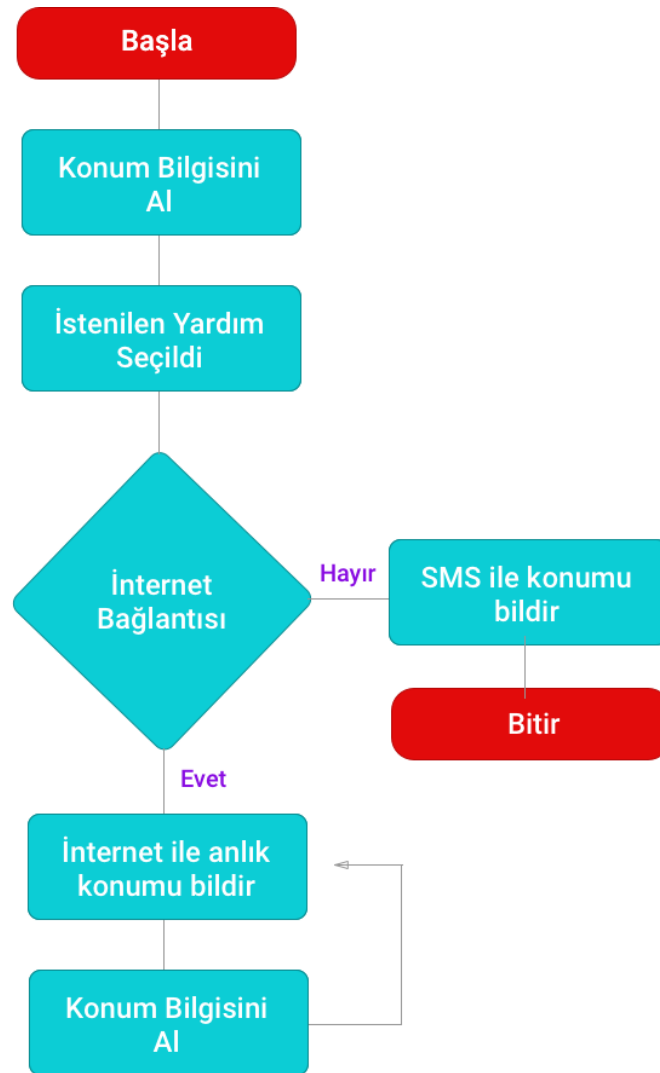
android.permission.ACCESS_WIFI_STATE
android.permission.ACCESS_NETWORK_STATE
android.permission.INTERNET izinleri alındı.

3.1.3. Veritabanının tanıtılması ve yazma fonksiyonunun eklenmesi

Veritabanının uygulamaya tanıtılması ve üzerinde değişiklik yapabilmek için gerekli parçalar yazıldı.

3.1.4. SMS ile gönderim fonksiyonunun eklenmesi

SMS ile gönderim için gerekli parçalar yazıldı ve sınılandı.



Şekil 3 - Mobil Uygulama Algoritması

3.2. Web Uygulaması

3.2.1. Uygulamanın tasarlanması

Yardım ekiplerinin kolayca yardım çağrısı detaylarını görebilmeleri için olabildiğince sade tasarım tercih edildi

3.2.2. Haritaların kurulumu

Google'ın sağladığı API kodu alındı ve uygulamaya eklendi

3.2.3. Veritabanının tanıtılması ve konumun alınması

Veritabanının uygulamaya tanıtılması ve bilgilerin görülebilmesi ve değiştirilebilmesi için gerekli parçalar yazıldı.

3.2.4. Harita üzerine pin eklenmesi ve konumun açık adrese çevrilmesi

Alınan konumu belli etmek üzere görüntülenecek olan harita pini eklendi. Konum bilgisi, aynı zamanda açık adres bilgisine çevrilerek ekrana yazılması sağlandı.

3.3. Veritabanı

Prototipte kullanılan gerçek zamanlı veritabanı, Google'ın Firebase adı altında geliştiricilere sunulan bir hizmet ile sağlanmaktadır. Mobil cihazdan alınan veri, web uygulaması tarafından okunmaktadır. Böylelikle cihazın hareketleri takip edilmektedir. Verinin gönderilip alınması arasındaki süre sistem yoğunluğuna göre değişiyor olsa da kullanımı etkilemeyecektir.

3.4. Proje Kodlarının Açıklanması

Projenin temel özelliklerinin çalışma prensibi verilmiştir. Tasarımsal öğelere girilmemiştir. Projenin nasıl çalıştığı özetle anlatılmıştır.

3.4.1. Android Uygulaması

a) Manifest Dosyası

Her Android uygulaması bir manifest dosyası ile çalışır. Uygulamanın adından, kullandığı servislere ve aldığı izinlere kadar tüm bilgiler bu dosya içerisinde tanımlanır, XML formatında tutulur. APK (Android uygulama kurulum dosyası) içerisinde kriptolu halde saklansa da basit bir kaç adım ile salt hale döndürülebilir.

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Kod a.1 - Alınan İzinler

Kod a.1 üzerinde alınan izinler gösterilmiştir. Cihaz konumu almak, sms göndermek, internet bağlantısını sınamak gibi işlemler için bu izinler gerekmektedir. Bu işlemler izinsiz yapılırsa uygulama çökecektir.

ACCESS_FINE_LOCATION - Kesin konuma erişim izni
ACCESS_COARSE_LOCATION - Yaklaşık konuma erişim izni
SEND_SMS - SMS gönderme izni
ACCESS_WIFI_STATE - Wifi bağlantı durumuna erişme izni
ACCESS_NETWORK_STATE - İnternet bağlantı durumuna erişme izni
INTERNET - İnternet bağlantısı izni

```
</activity>
<receiver android:name="sesTuslari" >
    <intent-filter>
        <action android:name="android.media.VOLUME_CHANGED_ACTION" />
    </intent-filter>
</receiver>
```

Kod a.2 - Receiver Tanımı

Kod a.2 üzerinde bazı tanımlamalar gösterilmiştir. Uygulama açık değilken ses tuşları üzerindeki aksiyonu alabilmek için bir receiver tanımlanmıştır ve “sesTuslari” isimli Java sınıfına bağlanmıştır. Böylelikle uygulama kapalıyken ses tuşları ile aktif edebilmek için gerekli düzenleme yapılmıştır.

b) MainActivity.java

```
public class MainActivity extends FragmentActivity implements OnMapReadyCallback, LocationListener{

    private GoogleMap harita;
    private LocationManager locationManager;
    Button yardım;
    DatabaseReference konumGir;
    Location konum;
    SmsManager smsgonder;
    Geocoder geocoder;
    List<Address> addresses;
    String adres;
    //tanımlamalar

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        yardım = findViewById(R.id.yardım);
        locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        konumGir = database.getReference("loc");
        smsgonder = SmsManager.getDefault();
        geocoder = new Geocoder(this, Locale.getDefault());
        //tanımlamalar

        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
            ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) { return;}
        //Kötüye kullanımın önüne geçmek üzere kütüphaneleri kullanmak için izinlerin alındığı sağlanır, otomatik eklenir

        konum = locationManager.getLastKnownLocation(locationManager.NETWORK_PROVIDER);
        //son bilinen konum alınır
        onLocationChanged(konum);
        //metod bu konum ile çalıştırılır
    }
}
```

Kod b.1 - Tanımlamalar

MainActivity bir Java sınıfıdır. Manifest dosyasında, uygulamamız açıldığında çalışacak olan ana sınıf olarak tanımlanmıştır.

Kod b.1’de görüldüğü üzere mobil uygulama içerisinde kullanılacak olan objeler, sınıflar tanımlanmış ve isimlendirilmiştir. “LocationListener” sınıfının implement edilmesiyle, internet bağlantısına gerek olmadan konum bilgisi alınabilinecektir.

“konum = locationManager.getLastKnownLocation(locationManager.NETWORK_PROVIDER);” satırı ile son bilinen konum değeri, “konum” değişkenine tanımlanır. “onLocationChanged” (implement edilen LocationListener sınıfı ile birlikte gelen bir metoddur) fonksiyonunda çalıştırılır.

Konum bilgisinin alınmasından önce kötüye kullanıma karşı olarak, kullanıcı izni alınıp alınmadığı kontrol edilir. İzinler alınmamış ise kodlar çalışmaz.

```
@Override
public void onLocationChanged(Location location) {
    harita.setOnMyLocationChangeListener(new GoogleMap.OnMyLocationChangeListener() {
        LatLng loc = new LatLng(konum.getLatitude(), konum.getLongitude());

        try {
            addresses = geocoder.getFromLocation(konum.getLatitude(), konum.getLongitude(), 1);
        } catch (IOException e) {
            e.printStackTrace();
        }
        adres = addresses.get(0).getAddressLine(0);
        //Geocoder ile; konum bilgisi değiştiğinde, adres bilgisi de o konuma göre güncellenir.
        harita.moveCamera(CameraUpdateFactory.newLatLngZoom(loc, 15f));
        //konum değiştikçe haritada ortalılır.

        yardım.setOnClickListener(new View.OnClickListener() { //butona basıldığında
            @Override
            public void onClick(View view) {
                konumPaylas(String.valueOf(location.getLatitude()), String.valueOf(location.getLongitude()));
                //konum paylaşımı yapacak metod çalıştırılır.
            }
        })
    })
}
```

Kod b.2 - Konum Alınması

Kod b.2’de görülen kodlar, konum her güncellendiğinde tekrarlanır. Konumun açık adresi çıkartılır, ilgili değişkene tanımlanır (internet bağlantısı var ise). Harita üzerinde bulunduğu yere doğru ölçek küçültülür ve ortalılır. Kullanıcı tam konumunu görebilir.

Yardım butonuna basıldığında “konumPaylas” metodu, konum (enlem, boylam) bilgileri ile çalıştırılır.

```
private void konumPaylas(String latitude, String longitude) {
    if(baglantiDurumu){
        konumGir.child("latlng").setValue(String.valueOf(latitude+"*"+longitude)); //bağlantı varsa tanımlı sunucuya konum bilgisi gönderilir.
    }else{
        msgonder.sendTextMessage(getResources().getString(R.string.acilDurumNumarası),
        null, getResources().getString(R.string.mesaj) + "\n" + adres + "\n" + latitude + "*" + longitude, null, null);
        //bağlantı yoksa acil durum mesajı ile adres bilgisi ve konum bilgisi sms gönderilir
    }
}
```

Kod b.3 - Konum Paylaşımı

Kod b.3'te görüldüğü üzere “konumPaylas” adında bir metod oluşturulmuş ve içerisine enlem ve boylam değerleri almıştır. Metod çalıştırıldığında bağlantı durumu kontrol edilir. Bu, “baglantiDurumu” isimli Boolean değişkeni ile sağlanır. Bağlantı olduğu durumlarda true; olmadığında false değerini döndürür.

Bağlantı olması durumunda “enlem*boylam” formatında veritabanına gönderilir.

Bağlantı bulunmuyor ise, acil durum numarasına; önceden belirlenen mesaj, açık adres, enlem-boylam bilgileri gönderilir.

“R.string.acilDurumNumarasi” ve “R.string.mesaj” değişkenleri Strings.xml dosyası içerisinde tanımlıdır.

```
public boolean baglantiDurumu() {
    Runtime runtime = Runtime.getRuntime();
    try {
        Process process = runtime.exec("/system/bin/ping -c 1 8.8.8.8"); //Google DNS sunucuları pinglenir
        int deger = process.waitFor();
        return (deger == 0); //iletişim kurulursa true değeri döner
    }
    catch (IOException e) { e.printStackTrace(); }
    catch (InterruptedException e) { e.printStackTrace(); }

    return false; //hata alınırsa false değeri döner
}
```

Kod b.4 - Internet Bağlantısı Kontrolü

Kod b.4'de internet bağlantısının kontrolünün sağlandığı kod parçası verilmiştir. Burada runtime oluşturularak bir sistem komutu çalıştırılmıştır.

“/system/bin/ping -c 1 8.8.8.8” Komutunun anlamı, ping komutu kaynağından, belirli bir sayıda (-c), belirtilen sayı kere (1), belirtilen adresin (8.8.8.8) ping gönderilmesidir. Pingleme bir tür paket alışverişidir. Çok küçük boyutlarındaki bu paketleri başarıyla gönderip cevap almak, internet bağlantınızın kurulu olduğuna kanıt gösterilebilir. Belirtilen 8.8.8.8 adresi Google’ın DNS sunucularına aittir. IPv4 formatında bir ip adresidir. Hızlı geri dönüt alma olanağından dolayı tercih edilmiştir.

c) sesTuslari.java

```
public class sesTuslari extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals("android.media.VOLUME_CHANGED_ACTION")) {

            Intent i = new Intent();
            i.setClass(context, MapsActivity.class);
            i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            context.startActivity(i);
        }
    }
}
```

Kod c.1 - Arkaplanda Ses Tuşları Kontrolü

Kod c.1’de, acil durumda uygulama açık olmasa dahi ses tuşlarıyla açabilmeyi sağlayan kodlar verilmiştir. Manifest dosyasında tanımlanan receiver bu sınıfa bağlanmıştır ve ses tuşlarındaki her aksiyon, bu kodları tekrar çalıştırır.

d) strings.xml

```
<resources>
    <string name="ambulans">112</string>
    <string name="polis">155</string>
    <string name="itfaiye">110</string>
    <string name="mesaj">Acil bir durumla karşı karşıyayım. Adres:</string>
</resources>
```

Kod d.1 - Tanımlı Metinler

Kod d.1’de görülen tanımlar Strings.xml dosyasına aittir. Uygulama içerisine kullanılacak uzun metinler burada tanımlanarak, üzerlerinde değişiklik yapılması daha kolaylaştırılmıştır. Burada tanımlı olan metinlerin, farklı dillerdeki karşılıkları da eklenerek uygulamada birden fazla dil kullanılabilir. Bu diller, sistem diline bağlı olarak değişir. Bu özellik ile başka ülkelerde kullanılan acil durum numaraları da eklenebilir.

3.4.2. Web Uygulaması

a) Firebase ve Maps API kurulumu

```
<script src="https://www.gstatic.com/firebasejs/5.4.0/firebase.js"></script>
<script src="https://www.gstatic.com/firebasejs/5.4.0/firebase-database.js"></script>
```

Kod a.1 - Firebase Kütüphanelerinin Tanıtılması

```
var config = {
  apiKey: "AIzaSyAocMbZ-XS60PhwapNdAl2**",
  authDomain: "nine11-**.firebaseapp.com",
  databaseURL: "https://nine11-**.firebaseio.com",
  projectId: "nine11-**",
  storageBucket: "nine11-**.appspot.com",
  messagingSenderId: "46049**"
};

firebase.initializeApp(config);
```

Kod a.2 - Firebase Kimliğinin Girilmesi

Kod a.1 ve a.2’de görüldüğü üzere, Firebase API kurulumu için gerekli Javascript kütüphaneleri eklenmiş ve kimlik bilgileri girilmiştir. Bu bilgiler Google tarafından kullanılır ve her proje için eşsiz bir kimlik bilgisi ile karışıklığın önüne geçmek amaçlanır.

```
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyA76-DhBgB8n-**A&callback=initMap">
</script>
```

Kod a.3 - Maps Kimliğinin Girilmesi

Kod a.3 ile, Google Maps kullanımı için üretilmiş olan eşsiz kimlik girilerek uygulamamızda kullanılması sağlanılmıştır.

b) Alınan Konum Bilgisinin Haritada İşaretlenmesi

```
function initMap() { //google haritalar eklendi

    var geocoder = new google.maps.Geocoder; //koordinat- açık adres çevirmeye yarayan metod
    var infowindow = new google.maps.InfoWindow; //adres yazımı için bilgi balonu
    var marker = new google.maps.Marker({ //pin tanımı
        position: loc,
        title:"Tehlikede!"
    });
};
```

Kod b.1 - Gerekli Metodlar

Kod b.1, alınan koordinat bilgilerinin açık adrese dönüştürülmesi, harita üzerinde gösterilmesi ve işaretlenmesi için Google Maps kütüphaneleri ile gelen bazı metodların tanımını içerir.

Geocoder: Konuma ait koordinatları açık adrese çevirir.

Infowindow: Harita üzerinde, adresin ve koordinatın yazılacağı pencere.

Marker: Konum işaretlemek için pinler eklenmesine olanak tanır.

Uzak veritabanının üzerindeki verinin değişmesiyle uygulanan satırlardır. Konum değiştiğinde “location” değişkenini günceller.

```
var locationCh = firebase.database().ref('loc');
locationCh.on('child_changed', function(data) { //konum değiştiğinde koordinatları alır

var location = data.val() //koordinatlar tanımlanır
```

Kod b.2 - Uzak Veritabanının Okunması

Kod b.2, veritabanındaki konum bilgisi her değiştiğinde bilgi tekrar alınarak güncellenmesi işlevini görür.

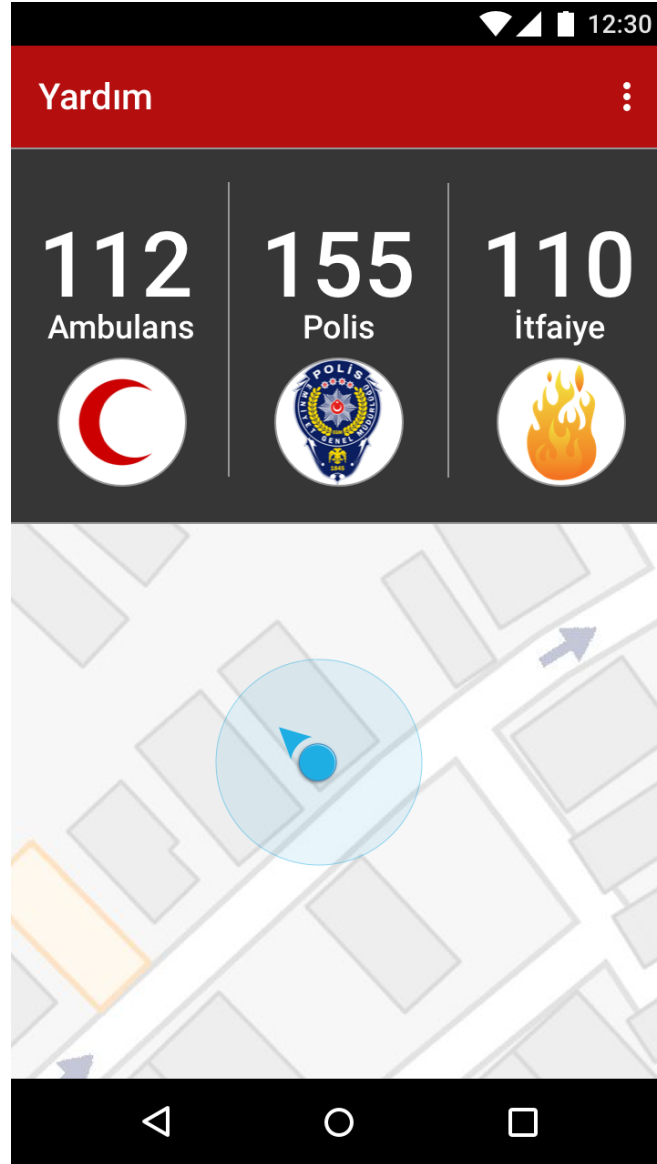
```
if(loc.lat != 'b'){
    var latlng = new google.maps.LatLng(loc.lat, loc.lng);
    marker.setPosition(latlng); //pin ile işaretlenir
    marker.setMap(map); //harita güncellenir
    geocoder.geocode({'location': latlng}, function(results, status) {
        if (status === 'OK') {
            if (results[0]) {
                infowindow.setContent(results[0].formatted_address); //adres yazılır.
                infowindow.open(map, marker);
            }
        }
    });
}
}else{
    marker.setMap(null); //pin kaldırılır
}
```

Kod b.3 - Alınan Konumun İşaretlenmesi

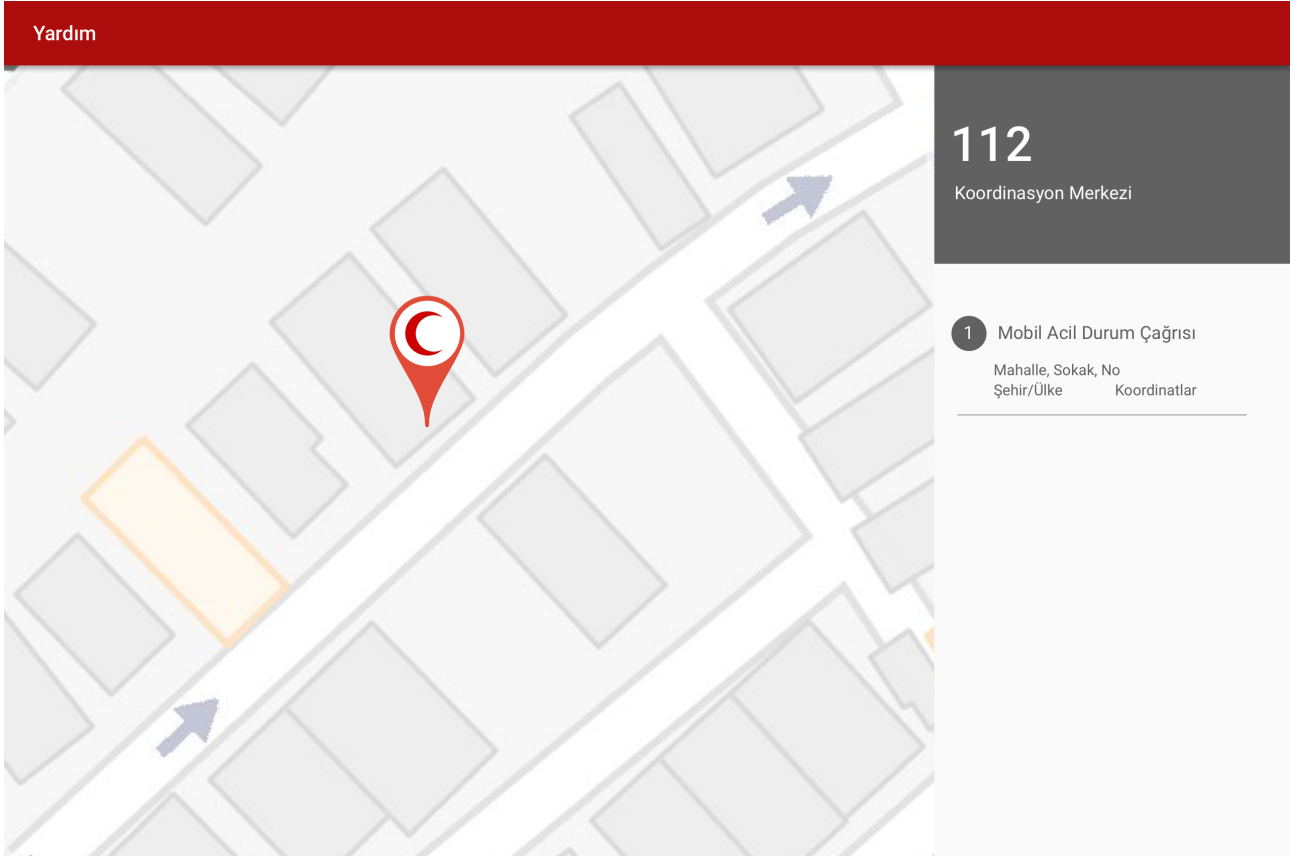
Kod b.3 ile, alınan deęerler harita üzerinde iřaretlenir, koordinatın aık adresi de ekrana yazılır. Bylelikle konum bilgisi acil yardım ekiplerine ulařır.

Eęer konum deęeri “b” (boř anlamına gelir, acil durum aęrısı iptal edilince bu deęer gnderilir. aęrı iptal olur) olduęunda iřaretlenmiř pin kaldırılır.

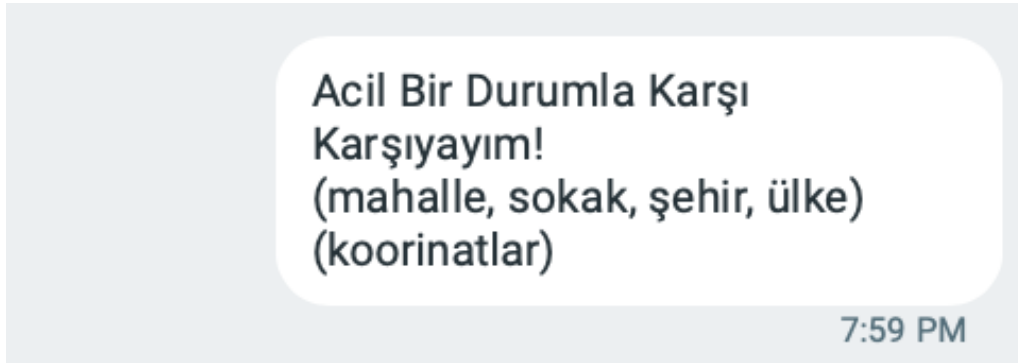
3.5. Proje Ekran Grntleri



Ekran Grnts 1 - Mobil Uygulama Tasarımı



Ekran Görüntüsü 2 - Web Uygulaması Tasarımı



Ekran Görüntüsü 3 - Gönderilen Mesai

4. Sonular ve Tartışma

Proje kapsamında tasarlanan sistem prototiplenmiş olup, yapılan denemelerde olumlu sonuç alınmıştır. Geline nokta, geliştirilen uygulamalar ile anlık konum takibi yapılabilmektedir. Yardıma ihtiyacı olan birey, mobil uygulamaya girerek ihtiyaç duyduğu uygun yardımcı seçip yardımcı çağrısı oluşturabilmektedir. Uygulamaya girilemeyecek ciddi durumlarda ise, üç defa ses açma-kapatma butonlarına basıldığında uygulama otomatik aktif olup yardımcı çağrısı yapmaya başlamaktadır. Süreden tasarruf amacıyla herhangi bir doğrulama beklenmemesi uygun görüldüğünden, sadece yardımcı çağrısının başladığına dair sesli uyarı verilmektedir. Yanlışlıkla çağrı yaptığını farkedene birey, yaptığı hatalı çağrıyı iptal edebilmektedir.

Yardıma çağrısı başladığında ilk olarak cihazın koordinat bilgileri alınmaktadır ve konum değıştikçe bilgiler güncellenmektedir. Alınan konum bilgisi, internet bağlantısı ile (varsa) acil yardımcı ekiplerine harita üzerinde bildirilmektedir. İnternet bağlantısının bulunmadığı fakat operatör kapsama alanı içerisinde bulunulan durumlarda ise koordinat bilgisi, kısa mesaj yolu ile bildirilmektedir. Böylelikle insanlara yardımcı ulaştırılabilecek alan çok büyük ölçüde genişletilmiştir.

Yardıma ekipleri; kendileri için hazırlanmış web uygulamasına girerek oluşturulan yardımcı çağrılarını görüp, detaylarına erişebilmektedir. Böylelikle olay yerinde yönlendirilmesi gereken personeli uygun olarak yönlendirebilmektedir. Ulaşımda olan ekipler ise, mobil cihazlar ile konum bilgisini görebilmektedir.

Geliştirilen bu sistem ile, acil yardıma ihtiyacı olan birey;

- Acil yardımcı hat numaralarını bilinmediğı, olay heyecanı ile unuttuğı veya arayacak fırsat bulamadığı,
- Bulunduğı adresi bilmediğı, olay heyecanı ile unuttuğı veya söyleyecek fırsat bulamadığı,
- Telefon ekranına ulaşmak için yeterli süresi olmadığı, ulaşamadığı,
- İnternet bağlantısının bulunmadığı

durumlarda acil yardımcı isteğini çok daha kolay bir biçimde bildirebilecektir.

Projemiz ile, acil durumlarda bireylerin yardımcı istemesi ve yardımcı ulaşması süreleri daha hızlı ve kolay biçimde çözülmeye çalışılmış; böylelikle bireylerin, acil durumdan daha az zamanda, daha az zararla kurtulması istenmiştir. İnsan yaşamının gerektirdiğı güven duygusunun perçinlenip, bireyin kendini gerçekleştirme sürecine katkı sağlanmıştır.

5. Öneriler

Kullanıcı uygulaması tasarımı tekrar gözden geçirilebilir. Daha sade bir tasarım eklenebilir. Eklenecek ayarlar menüsü ile kullanıcının seçimine bırakılabilir.

Sadece Android Platformu üzerinde çalışan mobil uygulama; cross-platform, işletim sisteminden bağımsız bir uygulama olarak tekrar geliştirilerek kullanıcı kitlesi önemli oranda büyütülebilir.

Arka uçta kullanılan Google'ın Firebase hizmeti yerine, Noje.js dili ile daha işlevsel bir sistem yazılabilir. Böylelikle üçüncü parti yazılım kaynaklı güvenlik sorunları azaltılabilir.

İşaretlemede kullanılan Google'ın Maps hizmeti yerine, açık kaynak lisanslı haritalar kullanılarak üçüncü parti yazılım kaynaklı güvenlik sorunları azaltılabilir.

Uygulama bir çok farklı dile çevrilerek hedef kitlesi önemli oranda büyütülebilir.

6. Kaynakça

Çakmak, H. (2018). Abraham Maslow'un "Kendini Gerçekleştirme" Kavramının Aziz Nesefi'nin "İnsan-ı Kâmil" Kavramıyla Karşılaştırılması. (Onaylanmış Yüksek Lisans Tezi). Eskişehir Osmangazi Üniversitesi/Sosyal Bilimler Fakültesi, Eskişehir.

Google Developers (2 Ağustos 2018). <receiver>, Erişim Tarihi: 4.11.18, Erişim Adresi: <https://developer.android.com/guide/topics/manifest/receiver-element>

Google Developers (18 Eylül 2018). Receive Location Updates, Erişim Tarihi: 4.10.18, Erişim Adresi: <https://developer.android.com/training/location/receive-location-updates>

Google Firebase (16 Eylül 2018). Add Firebase to your Javascript Project, Erişim Tarihi: 5.10.18, Erişim Adresi: <https://firebase.google.com/docs/web/setup>

Karnik, G. E. ve Mallavarapu J. S. (2001). Method and Apparatus for Servicing Emergency Calls from a Data Network, US 6,744,856 B2

Maslow, A. H. (1943). A Theory of Human Motivation. Psychological Review, 50, 370-396.

Narman, A. (2016). Android Studio İle Programlama, İstanbul: Kodlab Yayıncılık

Sofuoğlu T. Acil Ambulans Hekimleri Tarihçesi, Erişim Tarihi: 2.10.18 Erişim Adresi: <http://www.aahd.org.tr/tarihce/>

Statista (2016). Number of smartphone users worldwide from 2014 to 2020 (in billions), Erişim Tarihi: 11 Ocak 2018, Erişim Adresi: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide>

Statista (2017). Mobile operating systems' market share worldwide from January 2012 to December 2017, Erişim Tarihi: 11 Ocak 2018, Erişim Adresi: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>