



**T.C.  
DÜZCE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ**

**ANABİLİM DALI**

**MOBİL GPS VERİSİ İLE GELİŞMİŞ TAKİP VE YÖNLENDİRME  
SİSTEMİ**

**YÜKSEK LİSANS TEZİ**

**ARİF CEYLAN**

**HAZİRAN 2016**

**DÜZCE**

## KABUL VE ONAY BELGESİ

Arif CEYLAN tarafından hazırlanan Mobil GPS Verisi ile Gelişmiş Takip ve Yönlendirme Sistemi isimli lisansüstü tez çalışması, Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 25/05/2016 tarih ve 2016/467 sayılı kararı ile oluşturulan jüri tarafından Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Üye  
(Tez Danışmanı)  
Yrd. Doç. Dr. Ali ÇALHAN  
Düzce Üniversitesi

Üye  
Doç. Dr. İbrahim ŞAHİN  
Düzce Üniversitesi

Üye  
Yrd. Doç. Dr. Alper KARAHAN  
Kocaeli Üniversitesi

Tezin Savunulduğu Tarih : 03.06.2016

### ONAY

Bu tez ile Düzce Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Arif CEYLAN'ın Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans derecesini almasını onamıştır.

Doç. Dr. Resul KARA  
Fen Bilimleri Enstitüsü Müdürü

## BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

03 Haziran 2016

İMZA



*Sevgili Aileme*

## **TEŐEKKÖR**

Yüksek lisans öğrenimim ve bu tezin hazırlanmasında süresince gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Yrd. Doç. Dr. Ali ÇALHAN'a en içten dileklerle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

**03 Haziran 2016**

**Arif CEYLAN**

# İÇİNDEKİLER

Sayfa

TEŞEKKÜR.....	i
İÇİNDEKİLER.....	ii
ŞEKİL LİSTESİ.....	v
ÇİZELGE LİSTESİ.....	vii
SİMGELER VE KISALTMALAR.....	viii
ÖZET: .....	1
ABSTRACT .....	2
EXTENDED ABSTRACT.....	3
1. GİRİŞ.....	5
1.1 AMAÇ VE KAPSAM .....	5
2. GENEL BİLGİLER.....	8
2.1 MOBİL KONUMLANDIRMA SİSTEMLERİ.....	8
2.1.1 Ağ temelli Konum Sistemi .....	8
2.1.2 GPS (Küresel Konumlama Sistemi).....	9
2.1.3 Ağ Temelli Konum Sistemi ve GPS Karşılaştırması.....	9
2.2 ÇEVİRİMİÇİ HARİTA HİZMETLERİ.....	10
2.2.1 OpenStreetMap.....	10
2.2.1.1 OpenLayers .....	11
2.2.1.2 Open Source Route Machine (OSRM) .....	16
2.2.2 Google Haritalar .....	20
2.2.2.1 Google Haritaların Web Sayfasında Kullanımı.....	20
2.2.2.2 İşaretçi (Marker) Ekleme.....	23
2.2.2.3 İşaretçiye Özel Resim Ekleme.....	24

2.2.2.4 İşaretçilerde Bilgi Kutuları.....	24
2.2.2.5 Yönlendirme UPA'sı .....	26
<b>2.2.3 Yandex Haritalar .....</b>	<b>26</b>
2.2.3.1 Yandex Haritaların Web Sayfalarında Kullanımı .....	26
2.2.3.2 İşaretçi Ekleme .....	27
2.2.3.2 Çoklu Çizgi Çizimi .....	28
<b>2.3 MOBİL PROGRAMLAMA.....</b>	<b>29</b>
<b>2.3.1 Native Programlama .....</b>	<b>29</b>
<b>2.3.2 HTML5 Web Uygulamaları .....</b>	<b>30</b>
<b>2.3.3 Melez (Hybrid) Programlama .....</b>	<b>30</b>
2.3.3.1 Avantajları .....	30
2.3.3.2 Mobil Melez Çatılar (Mobile Hybrid Frameworks) .....	31
<b>2.4 ANLIK VERİ İLETİMİ İÇİN KULLANILAN YÖNTEMLER.....</b>	<b>38</b>
<b>2.4.1 Web üzerinde veri iletişimi ve kullanılan yöntemler.....</b>	<b>39</b>
2.4.1.1 Düzenli HTTP İsteği .....	39
2.4.1.2 Ajax HTTP Polling.....	39
2.4.1.3 Ajax HTTP Long Polling .....	40
2.4.1.4 WebSocket .....	41
<b>2.5 HTML5 GEOLOCATION .....</b>	<b>43</b>
<b>2.6 BİLGİ GÜVENLİĞİ VE KRİPTOGRAFİ.....</b>	<b>44</b>
2.6.1 Anahtarsız Şifreleme .....	45
2.6.2 Gizli Anahtarlı Şifreleme .....	45
2.6.3 Açık Anahtarlı Şifre .....	45
<b>2.7 PRESSY DÜĞMESİ .....</b>	<b>45</b>
<b>3. MATERYAL VE YÖNTEM.....</b>	<b>47</b>
<b>3.1 Sunucu Uygulaması.....</b>	<b>47</b>

<b>3.2 Emniyet Birimi Uygulaması .....</b>	<b>48</b>
<b>3.2.1 Yöntem I .....</b>	<b>49</b>
<b>3.2.2 Yöntem II .....</b>	<b>50</b>
<b>3.3 Kişisel Uygulama .....</b>	<b>51</b>
<b>3.3 Emniyet Ekibi Mobil Uygulama .....</b>	<b>53</b>
<b>3.4. Verilerin işlenmesi.....</b>	<b>53</b>
<b>4. BULGULAR .....</b>	<b>54</b>
<b>5. SONUÇLAR VE ÖNERİLER .....</b>	<b>56</b>
<b>6. KAYNAKLAR.....</b>	<b>58</b>
<b>ÖZGEÇMİŞ.....</b>	<b>61</b>



## ŞEKİL LİSTESİ

	<b><u>Sayfa No</u></b>
Şekil 1.1. Uygulama görüntüsü	7
Şekil 2.1. Baz istasyonu sistemi	8
Şekil 2.2. Google harita ve OpenStreetMap karşılaştırması	11
Şekil 2.3 OpenLayers çizgi çizme	15
Şekil 2.4. OSRM iki nokta arasındaki yol verileri	19
Şekil 2.5. JSON verinin parsellenmiş hali.	19
Şekil 2.6. Google Haritalar UPA anahtar alımı.	20
Şekil 2.7. Google Haritalar UPA anahtar için yeni proje oluşturma	21
Şekil 2.8. Google Haritalar UPA anahtar üretim ekranı	22
Şekil 2.9. Google Haritalar alınan anahtar ekranı	22
Şekil 2.10. İşaret kutusu örneği.	25
Şekil 2.11. Yandex harita işaretçi ve açıklama kutusu örneği	28
Şekil 2.12. Yandex haritalar çoklu çizgi çizimi örneği	29
Şekil 2.13. Kurulum sonrası oluşan klasör ve dosyalar	32
Şekil 2.14. Tarayıcı üzerinde çalışan uygulamanın görüntüsü	33
Şekil 2.15. Ionic Framework dokümantasyonu.	34
Şekil 2.16. Ionic örnek proje oluşturma	35
Şekil 2.17. ngcordova.com sitesi ve eklentiler	36
Şekil 2.18. Onsen Ui Framework	37
Şekil 2.19. Kendo UI	38
Şekil 2.20. Klasik HTTP teknolojisi	39
Şekil 2.21. Ajax HTTP Polling.	40
Şekil 2.22. Ajax Long Polling	41
Şekil 2.23. WebSocket çalışma mantığı	42
Şekil 2.24. Pressy düğmesi	46
Şekil 3.1. Sunucu uygulaması şeması	47
Şekil 3.2. Emniyet birimi uygulaması görseli	49
Şekil 3.3. Kuş bakışı mesafe hesabı	50
Şekil 3.4. İkinci yöntemin mesafe hesabı	51

Şekil 3.5. Kişisel uygulama ekranı

52

Şekil 4.1. Geçmiş olay arama ekranı

55



## ÇİZELGE LİSTESİ

	<b><u>Sayfa No</u></b>
Çizelge 2.1. HTML5 Geolocation UPA'sı ile alabileceğimiz veriler	44
Çizelge 4.1. MySql veri tabanı geçmiş kayıtlar tablosu alan bilgileri	53



## SİMGELER VE KISALTMALAR

$\Phi$ :	Enlem
$\Lambda$ :	Boylam
d:	İki nokta arasındaki uzaklık
r:	Kürenin Yarıçapı
DDOS	Dağıtılmış Reddetme Servis Saldırısı (Distributed Denial of Service Attack)
GPS	Küresel Konumlama Sistemi (Global Positioning System)
GSM	Mobil İletişim İçin Küresel Sistem (Global System for Mobile Communications)
HTML	Zengin Metin İşaretleme Dili (HyperText Markup Language)
HTTP	Zengin Metin Transfer Protokolü (HyperText Transfer Protocol)
JSON	JavaScript Nesne Gösterimi (JavaScript Object Notation)
TCP	Geçiş Kontrol Protokolü (Transmission Control Protocol)
UPA	Uygulama Programlama Arayüzü (Application Programming Interface)
XSS	Çapraz Site Senaryosu (Cross Site Scripting)

## ÖZET

# MOBİL GPS VERİSİ İLE GELİŞMİŞ TAKİP VE YÖNLENDİRME SİSTEMİ

Arif CEYLAN

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik, Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı  
Yüksek Lisans Tezi

Danışman: Yrd. Doç. Dr. Ali ÇALHAN

Haziran 2016, 73 sayfa

Günümüzde, hemen herkesin sahip olduğu akıllı telefonlar oldukça farklı amaçlar için kullanılmaktadır. Bu tez çalışmasında, akıllı telefonların güvenlik sistemleriyle bütünleşeceği bir sistemin tasarlanması amaçlanmakta ve ayrıca, akıllı telefonların konum özelliği kullanılarak konum takibi, yönlendirme ve veri işleme gerçekleştirilmektedir. Bunun yanı sıra güvenlik sistemi üzerine model uygulama geliştirilmiş olsa da bu sistem; sağlık, iş takibi, müşteri takibi gibi birçok sistemde de kullanılabileceği düşünülmektedir.

Tez çalışmamızda mobil konumlandırma sistemleri, çevrimiçi harita sistemleri, mobil programlama yöntemleri, avantaj ve dezavantajları, anlık veri iletimi için kullanılan yöntemler detaylıca incelenmekte ve uygulanan model ve yazılım anlatılmaktadır.

Yeni teknolojilerin kullanıldığı bu çalışmada minimum sistem gereksinimlerinin en aza indirilmesi, hız ve kullanım açısından da üst düzey verimin alınması amaçlanmaktadır.

**Anahtar Kelimeler:** Anlık veri iletimi, Çevrimiçi harita sistemleri, Gerçek zamanlı takip, Html5, Mobil takip ve güvenlik sistemi, WebSocket

## ABSTRACT

# ADVANCED TRACKING AND GUIDANCE SYSTEM WITH MOBILE GPS DATA

Arif CEYLAN

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-  
Electronical and Computer Engineering

Master of Science Thesis

Supervisor: Assist. Prof. Dr. Ali ÇALHAN

June 2016, 73 pages

Nowadays, smart mobile phones are used for quite different purposes by almost everyone. In this thesis, the aim is to design a system that becomes smart mobile phones integrated with security systems and also, location tracking, redirecting and data processing are realized with using the property of location in smart mobile phones. Besides, however model application has been developed on security system, this system can be used for several systems like health, job follow-up, and customer follow-up.

In our thesis, mobile positioning systems, online map systems, mobile programming methods, their advantages and disadvantages, used instant data transmission methods are being examined in details and also developed model with software is being explained.

Some new technologies are being used in this thesis; the aim is getting minimum system requirements to the lowest degree and getting maximum efficiency in terms of speed and utilization.

**Keywords:** HTML5, Mobil tracking and security system, Online map systems, Real-time communication, Real-time tracking, WebSocket

# **EXTENDED ABSTRACT**

## **ADVANCED TRACKING AND GUIDANCE SYSTEM WITH MOBILE GPS DATA**

Arif CEYLAN

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-  
Electronical and Computer Engineering

Master of Science Thesis

Supervisor: Assist. Prof. Dr. Ali ÇALHAN

June 2016, 73 pages

### **1. INTRODUCTION:**

Personal security and emergency systems are necessities of increasing importance day by day and needed of the individual's daily life as becoming an indispensable hardware and software integrity.

Today, security systems are being used to set up on address description and make phone calls, and long after the intervention can be made in case of danger. In our thesis, it aimed to develop a new personal security system that can instantly start tracking events. Thus, a person who is in danger or needs emergency aid can send personal information from the phone to the security unit with location and start to track.

Security unit will see on the map the location of the signal to the people on the closest team practice; will direct the scene, instant tracking of endangered individuals can made over the internet.

### **2. MATERIAL AND METHODS:**

It is aimed to keep system requirements and application speed in a minimum level in our thesis. Accordingly, WebSocket is used for instant transmission system. WebSocket is started to develop in 2011 and still developed, web browsers using the HTTP port on a protocol that provides bidirectional data transmission works to be realized on TCP connection. All applications in our thesis are using this protocol to realize instant data

transmission. In addition, security research made in data transmission is included to the thesis.

The hybrid application programming methods that are qualified to work on any platform has been developed to support the wishes of the individual. In this application, Pressy button is used to send instant request in a secret way. To show individual requests on the map online mapping system of free, open source is used that is open to use OpenStreetMap. OpenLayers JavaScript library is used as the coding system. In addition, all data is stored in the MySQL database and it can be accessed at any time.

### **3. RESULTS AND DISCUSSIONS:**

Developed server application is installed on a virtual computer rented from the internet and makes it run. The developed team application is set up on four phones located to various locations. The request was sent from two mobile phones with developed personal applications. Also, finding the closest team, sending request, and tracking processes are observed in tests.

### **4. CONCLUSION AND OUTLOOK:**

We believe that the developed application brings the system speeds and minimum requirements for high-level applications using personal safety with new technologies and can be used in many fields.

We think that this application can be used for country wide applications or companies in similar tracking systems likewise 112 emergency, 155 police emergency and 110 fire brigade.



# 1. GİRİŞ

## 1.1 AMAÇ VE KAPSAM

Kişisel güvenlik ve acil yardım sistemleri, günümüzde gün geçtikçe önemi ve ihtiyacı artan bir gereklilik olup, bireyin günlük yaşamında vazgeçilmez bir donanım ve yazılım bütünlüğü haline gelmiştir. Günümüzde güvenlik sistemleri, telefonla arama yapıp yer tarifi üzerine kurulmakta ve tehlike durumundan çok sonra müdahale yapılabilmektedir.

Şu an Türkiye’de veya çoğu ülkede herhangi bir güvenlik veya yardım ihtiyacı durumunda telefonla arama yapıp yer tarifi ve sorun bilgisi verilmekte, yer tarifi yaparken de zorlanılabilmektedir. Geçmiş yıllarda arama kurtarma ekiplerinden yardım talep edildiği halde yerinin bulunamamasından kaynaklı olarak can kaybı yaşadığı görülmektedir. Teknolojinin bu noktada kullanılabileceğini düşünürsek mobil telefonlardaki GPS verisi sayesinde yer bildirimini yapmak mümkün olacak, belki de kısa sürede bulunabilecektir. Başka bir örnek vermek gerekirse kaçırılan bir insan, kaçırılan kişilerin yanında acil yardım telefon numaralarını arayamayacak, bilgi veremeyecektir. Oysaki telefondan kısa yol bildiriminiyle yetkili birimlerden istekte bulunması mümkün olacaktır. GPS verisinin sürekliliği sayesinde de konumu takip edilebilecek, gereken yardım sağlanabilecektir.

Cep telefonu üreticileri ve uygulama geliştiriciler artık kişisel güvenliğini sağlama amaçlı uygulamalar geliştirmektedirler. Örneğin bazı telefon üreticileri acil yardım hatlarına arama yapmak için kısa yol tuşu ataması yapılmasına olanak sağlamaktadır. 155 gibi bir telefon numarasına kısa yol tuşu atamasıyla telefon kilidi açılmadan kısa yol tuşuyla doğrudan arama yapılarak ortam sesi dinlenebilmekte, mobil ağ sistemleri sayesinde yer tespiti de yapılabilmektedir. Fakat bu tarz bir durumda telefonun belirli bir süre açık kalması gerekmekte, anında yer tespiti yapılamamaktadır. Bunun haricinde bazı uygulama geliştiriciler ve firmalar bu tarz durumlarda yardım isteyen bireyin bir yakınına konum gönderilmesini sağlayan mobil uygulamalar geliştirmişlerdir. Fakat bu uygulamalarda da mesaja geç bakılma ihtimali olmakta ve anlık takip, güvenlik güçlerini arama gibi işlemler bulunmamaktadır [1]. Geliştirilen bazı uygulamalar da konumun bireyin bir yakınına veya emniyet ekibine mesaj olarak gönderilmesi şeklinde

tasarlanmaktadır. Anlık veri iletimi kullanılmamakta ya da eski teknolojilerden yararlanılmakta, belirli zaman aralıklarıyla konum gönderilmektedir [2, 3, 4].

Çalışmamızda, anında olay takibini başlatabilecek yeni bir kişisel güvenlik sisteminin geliştirilmesi amaçlanmaktadır. Bu sayede tehlikede olan veya acil yardım ihtiyacı bulunan bir birey, telefonundan güvenlik birimlerine kişisel bilgilerini ve konumunu gönderebilecek, takibi başlatılabilecektir. Güvenlik birimi, harita üzerinde sinyal gönderen bireyin ve tüm ekiplerin konumunu görecek, uygulama akıllı bir sistemle en yakın ekibi olay yerine yönlendirecek, tehlikedeki bireyin anlık takibi internet üzerinden yapılabilecektir.

Tezimizin temelini OpenStreetMap harita projesi, bilgi güvenliği, WebSocket, Mobil Programlama ve veri işleme oluşturmaktadır. OpenStreetMap açık kaynaklı ve ücretsiz bir dünya haritası projesi olup dünyanın her yerinden gönüllü olarak katılımcılar sayesinde geliştirilmektedir [5]. Bu proje sayesinde masaüstü, web ve mobil uygulamalar yapılabilmektedir. Çalışmamızda bu uygulama web tarayıcı üzerinde gerçekleştirilmektedir. Web tarayıcılar üzerinde harita görüntüleme, işaretçi ekleme, yol çizdirme gibi işlemler için OpenLayers JavaScript kütüphanesi kullanılmaktadır [6]. Harita üzerinde mesafe hesaplama ve en yakın birimlerin görevlendirilmesi işlemlerinde Open Source Routing Machine (Açık Kaynak Yönlendirme Makinesi) kullanılmaktadır. Open Source Routing Machine iki konum arasındaki noktaları enlem-boylam şeklinde veren bir sunucudur [7]. Noktalar verildikten sonra noktaları Google'ın Polyline algoritmasını kullanarak şifrelemekte ve çıktısını vermektedir [8].

Tezimizde birey ve güvenlik birimleri arasında anlık veri aktarma işlemleri için WebSocket kullanılmaktadır. WebSocket, tarayıcılar üzerinde kullanılan bir soket iletişim teknolojisidir. Bağlantı sağlandıktan sonra iki yönlü dinleme sağlanıp anlık bilgi akışı yapılabilmektedir. Bunun yanında Nodejs web uygulamaları geliştirmek için kullanılan bir platformdur. Socket.io kütüphanesi ile tarayıcı üzerinde soket bağlantı-iletim işlemleri kolay bir şekilde yapılabilmektedir. Mobil telefon üzerinden de web tarayıcısı ile soket bağlantısı işlemi yapılabilmektedir.

Tasarlanan güvenlik ve acil yardım sisteminde öncelikle akıllı telefon üzerinden gönderilen Küresel Konumlama Sistemi (GPS) bilgileri sunucuya yollanmakta, gelen

bilgi anlık olarak OpenStreetMap harita üzerinde görüntülenmektedir. Harita üzerinde anlık olarak görüntülenen birey için güvenlik birimlerinden en yakın olan veya olanların uzaklığı hesaplanmakta, istek gönderen bireye OpenStreetMap üzerinde OpenLayers kütüphanesi kullanılarak yol çizilmektedir. Gelen bilgiler sürekli takip edilmekte ve sisteme kayıt edilmektedir. Şekil 1’de geliştirilen uygulamanın görüntüsü verilmektedir.



**Şekil 1.1.** Uygulama görüntüsü.

Tezimizde önce, yapılan araştırmalar sonucunda edinilen bilgiler, teknolojiler ve kullanımları aktarılmakta, daha sonra da uygulama hakkında bilgi verilmektedir.

## 2. GENEL BİLGİLER

### 2.1 MOBİL KONUMLANDIRMA SİSTEMLERİ

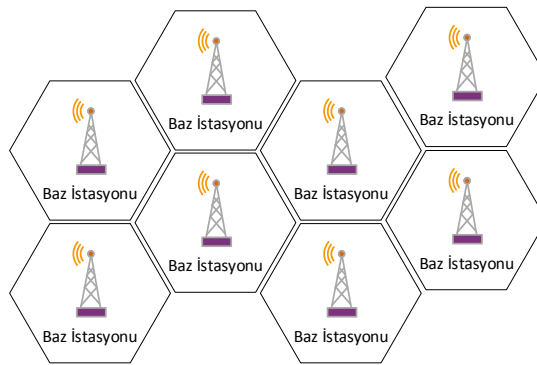
Mobil cihazlarda, Mobil İletişim İçin Küresel Sistem (GSM) modülü varsa, baz istasyonlarına bağlı olarak ağ temelli konum bilgisi; GPS modülü varsa, uydu sistemini kullanarak konum verisi üretilebilmektedir. Bu kavramlar aşağıda açıklanmaktadır.

#### 2.1.1 Ağ Temelli Konum Sistemi

Ağ temelli konum sistemi, baz istasyonlarını kullanarak konumu bulan bir sistemdir. Baz istasyonu radyo frekanslarını kullanarak, telefon ve istasyon arasında, atmosfer içerisinde sinyal gönderimi ve alımı yapmaktadır. Sinyal gönderimi ve alımı işlemleri için iki anten kullanılmaktadır [9].

Baz istasyonları ile akıllı telefon arasındaki sinyalin gidiş-geliş süresi, antenin başlangıç ve bitiş açısı gibi faktörlerle hesaplamalar yapılarak ağ temelli konum bulunmaktadır. Bulunulan bölgede baz istasyonlarının sayısının çokluğuna göre doğruluk oranı artmaktadır. Şehir içi alanlar gibi baz istasyonlarının fazla olduğu bölgelerde daha az sapma veren bu sistem baz istasyonlarının az olduğu bölgelerde yüksek sapmalı sonuçlar verebilmektedir [10].

Mobil telefonda ağ temelli konum bilgisine ulaşmak için telefonda GSM aktif olarak kullanılmalıdır. Telefonla konuşma yapılsa da yapılmısa da ağ temelli konum sistemi ile konum bulunması mümkün olmaktadır. Şekil 2.1’de örnek baz istasyonu sistemi gösterilmektedir.



Şekil 2.1. Baz istasyonu sistemi.

### **2.1.2 GPS (Küresel Konumlama Sistemi)**

GPS (Global Positioning System-Küresel Konumlama Sistemi) uzay tabanlı bir uydu sistemini kullanarak engelsiz bir görüş hattında kesin konum belirlemeye yarayan teknolojinin adıdır. GPS alıcıları, GPS sinyallerini almaya yarayan cihazlardır. GPS alıcılar sayesinde 4 veya daha fazla uydudan frekans bilgileri alınıp yorumlanarak kesin konuma ulaşılabilmektedir. Amerika Birleşik Devletleri Savunma Bakanlığı'na ait bu sistemdeki uydular yörüngelerinde sürekli olarak ilerleyerek radyo sinyallerini yaymaktadır [11]. Günümüzde akıllı telefonların hemen hepsinde GPS alıcı özelliği bulunmaktadır.

### **2.1.3 Ağ Temelli Konum Sistemi ve GPS Karşılaştırması**

Gelen konum verisi ile gerçek konum arasında sapmalar olabilmektedir. Ağ temelli konum sisteminde çevredeki baz istasyonlarının sayısı yüksekse düşük sapma (40 metre sapma), düşükse de yüksek sapma olabilmektedir (1-2 km sapma). GPS ise ağ temelli konumlandırmaya göre çok daha düşük sapmalar ile gerçek konumu verebilmektedir. Akıllı telefonlarda yapılan uygulamalarda ağ temelli konum sistemi verisine ve GPS verisine ulaşılabilmekte, verinin hangi kaynaktan geldiği öğrenilebilmektedir. Bu da bize verinin güvenilirliğini göstermektedir.

GPS sisteminin dezavantajı, alıcının açık havada bulunmasının gerekmesidir. Kapalı ortamlarda, çok sık ağaçlıklı bölgelerde sorun yaşanmaktadır.

## **2.2 ÇEVİRİMİÇİ HARİTA HİZMETLERİ**

İnternette uygulamalarda kullanılmaya üzere birçok çevrimiçi harita hizmeti bulunmaktadır. Tezimizde bu haritalardan bazıları incelenmekte, kullanımı anlatılmaktadır. Bu haritalardan en çok kullanılanlarına Google Haritalar, OpenStreetMap, Yandex Haritalar, Bing Haritalar ve Yahoo Harita örnekleri verilebilir.

Bazı haritalar ücretli olarak kullanıma sunulmakta, bazıları belirli bir limitten sonra ücret talep etmektedir.

Çevrimiçi haritaların çeşitli görünümüleri bulunmaktadır. Harita, Uydu görünümü şeklinde ve çeşitli coğrafi amaçlara yönelik görünümleer bulunmaktadır. Çoğu çevrimiçi harita sistemi, harita görünümü ve uydu görünümünü desteklemektedir.

### **2.2.1 OpenStreetMap**

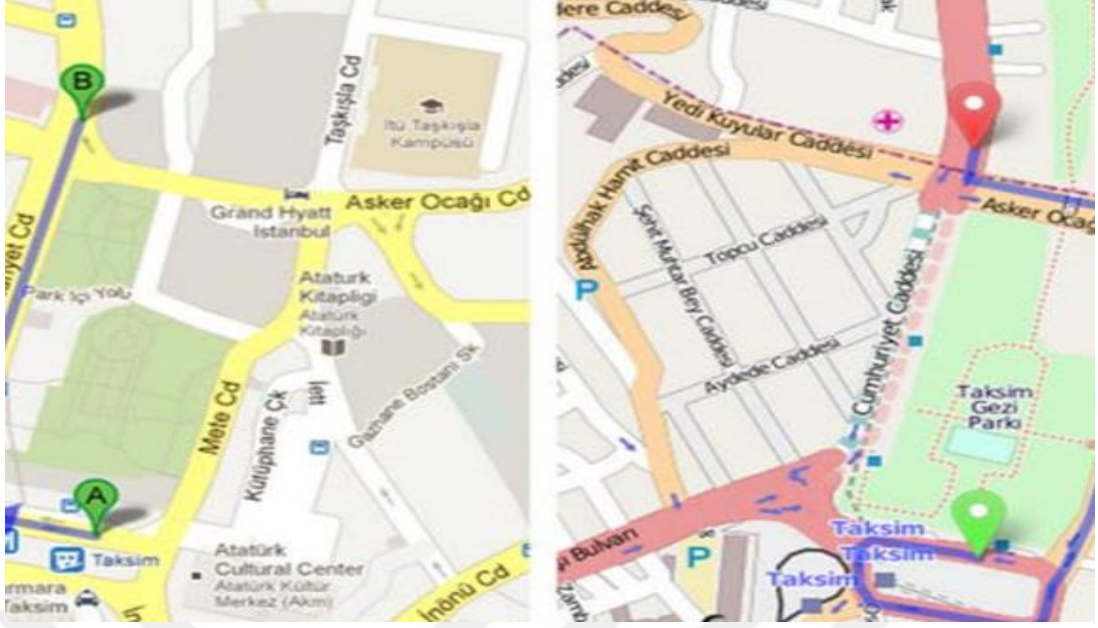
OpenStreetMap (kısaca OSM) özgür yazılım şartları altında oluşturulan açık kaynaklı bir dünya çapında harita oluşturma projesidir. GPS alıcılarıyla ve diğer kamu malı kaynaklardan toplanan bilgiler ile oluşturulur.

OpenStreetMap açık kaynaklı olan ve dünyanın her yerinden gönüllü olarak katılımcıların oluşturdukları bir dünya haritası projesidir. OpenStreetMap tıpkı Vikipedi (ki Vikipedi'den esinlenmiştir) gibi kullanıcıların içeriklerini oluşturduğu bir yapıya sahiptir. Haritalar kullanıcıların sunmuş olduğu GPS izleri, uydu fotoğrafları ve yerel yönetimler tarafından sağlanan çeşitli veriler doğrultusunda oluşturulmaktadır. Yapılan değişiklikler anında yayınlanır.

OpenStreetMap'te bütün içerik Open Database License lisansı ile korunmaktadır. Katkıları üreticilerine ait olmaya devam etmekle birlikte, CopyLeft lisansı içeriğin serbestçe dağıtımına olanak sağlamaktadır. Tamamen bedava ve tamamen reklamsızdır. Avrupa'da, ABD'de ve Güney Amerika'da birçok şirket ve yerel yönetim tarafından desteklenen bir projedir. Fakat Türkiye'den henüz herhangi bir yerel yönetiminden destek olmamıştır. Dünya üzerinde pek çok ülkeden gönüllülerin katılımıyla büyümeye devam eden OpenStreetMap, bugün pek çok şehrin özgür haritasını barındırmaktadır.

Son güncellemesi yönlendirme imkânları sunmaktadır ve 03.08.2015 itibariyle gönüllü kullanıcı sayısı; 2.204.822 kişidir. Kullanıcılar genellikle haritalama işleri ile uğraşan harita mühendisleri, teknikerleri ve teknisyenlerinden oluşmaktadır [12].

Şekil 2.2'de 29 Ocak 2013'teki sorgulama sonucunda Google Haritalar ile OpenStreetMap haritalarının karşılaştırılması görülmektedir.



Şekil 2.2. Google Harita ve OpenStreetMap karşılaştırması [13].

OpenStreetMap kullanıcılar tarafından düzenlendiği için bazı yenilikler de kısa sürede yapılabilmektedir. Şekil 2.2'deki örnekte Taksim'deki yol kapatıldığında Google haritalarda hala açık olarak gözükmeye rağmen OpenStreetMap'te kullanıcılar tarafından içeriğin güncellenip doğru yolu gösterdiği görülmektedir [13].

OpenStreetMap birçok alanda kullanılmakta ve birçok programlama dili ile çalışmaktadır. Webdeki ara yüzü için OpenLayers JavaScript kütüphanesi kullanılmaktadır [5].

### 2.2.1.1 OpenLayers

OpenLayers JavaScript kütüphanesi ile OpenStreetMap web üzerinde gösterilebilmekte, işaretler eklenebilmekte, üzerinde çizimler yapılabilmekte ve konumlandırma işlemleri de yapılabilmektedir. Ayrıca HTML-JavaScript dillerini kullandığı için web üzerinde yapılabilecek birçok işlem de gerçekleştirilebilmektedir [6].

Openlayers kullanımı aşağıda örnek kodlarla anlatılmaktadır.

```
<!doctype html>  
<html lang="en">  
<head>
```

```
<link rel="stylesheet" href="http://OpenLayers.org/en/v3.10.1/css/ol.css"
type="text/CSS">
<style>
  .map {
    height: 400px;
    width: 100%;
  }
</style>
<script src="http://openlayers.org/en/v3.10.1/build/ol.js"
type="text/JavaScript"></script>
<title>OpenLayers 3 örneği</title>
</head>
<body>
  <h2>Harita Örneği</h2>
  <div id="map" class="map"></div>
  <script type="text/JavaScript">
    var map = new ol.Map({
      target: 'map',
      layers: [
        new ol.layer.Tile({
          source: new ol.source.MapQuest({layer: 'sat'})
        })
      ],
      view: new ol.View({
        center: ol.proj.fromLonLat([37.41, 8.82]),
        zoom: 4
      })
    });
  </script>
</body>
</html>
```

Yukarıda OpenStreetMap'i web sayfasında gösterebilmek için örnek kod gösterilmektedir. Bunun açıklamasına bakacak olursak;

Öncelikle standart HTML kodları eklenmektedir.



```
<script src="http://OpenLayers.org/en/v3.10.1/build/ol.js"
type="text/JavaScript"></script>
```

Üstteki kodla OpenLayers JavaScript kütüphanesi sayfaya dahil edilmektedir.

Sayfa içerisinde;

```
<div id="map" class="map"></div>
```

Satırı ile bir id parametresi ve class parametresi map olan bir div HTML etiketi eklenmektedir. Harita bu nesnenin içerisinde gözükmektedir.

```
<script type="text/JavaScript">
    var map = new ol.Map({
        target: 'map',
        layers: [
            new ol.layer.Tile({
                source: new ol.source.MapQuest({layer: 'sat'})
            })
        ],
        view: new ol.View({
            center: ol.proj.fromLonLat([37.41, 8.82]),
            zoom: 4
        })
    });
</script>
```

Haritayı JavaScript ile göstermek için yukarıdaki kodlar kullanılmaktadır. Gösterim olarak uydu katmanı kullanılmakta, 37.41 enlem ve 8.82 boylamı merkez olarak alınmaktadır. Haritaya da 4 kat yakınlaştırma işlemi uygulanmaktadır [14]. Yolları gösteren harita istenirse sat yazan yer osm olarak değiştirilmelidir.

İşaretçi eklemek için aşağıdaki kodlar kullanılmaktadır.

```
var markers = new OpenLayers.Layer.Markers("Markers");
map.addLayer(markers);
```

```
markers.addMarker(new OpenLayers.Marker(lonLat));
```

Tezimizdeki uygulamalarda harita üzerinde çizgiler çizme ihtiyacı doğmaktadır. Bunun için örnek kod aşağıdaki kodlarda gösterilmektedir.

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Çizgi Çizdirme</title>
  <style>
    #mapdiv {
      height: 500px;
    }
  </style>
</head>
<body>
  <div id="mapdiv"></div>
  <script type="text/JavaScript" src="https://code.jquery.com/jquery-2.1.3.min.js"></script>
  <script src="http://www.OpenLayers.org/api/OpenLayers.js"></script>
  <script type="text/JavaScript">
    map = new OpenLayers.Map("mapdiv");
    var mapnik = new OpenLayers.Layer.OSM();
    map.addLayer(mapnik);
    epsg4326 = new OpenLayers.Projection("EPSG:4326");
    projectTo = map.getProjectionObject();
    var lonLat = new OpenLayers.LonLat( 31.156335
,40.844283).transform(epsg4326, projectTo);
    var zoom=18;
    map.setCenter (lonLat, zoom);
    var vectorLayer = new OpenLayers.Layer.Vector("Overlay");
    var lineStyle = OpenLayers.Util.applyDefaults(lineStyle,
OpenLayers.Feature.Vector.style['default']);
    lineStyle.strokeColor = "blue";
```

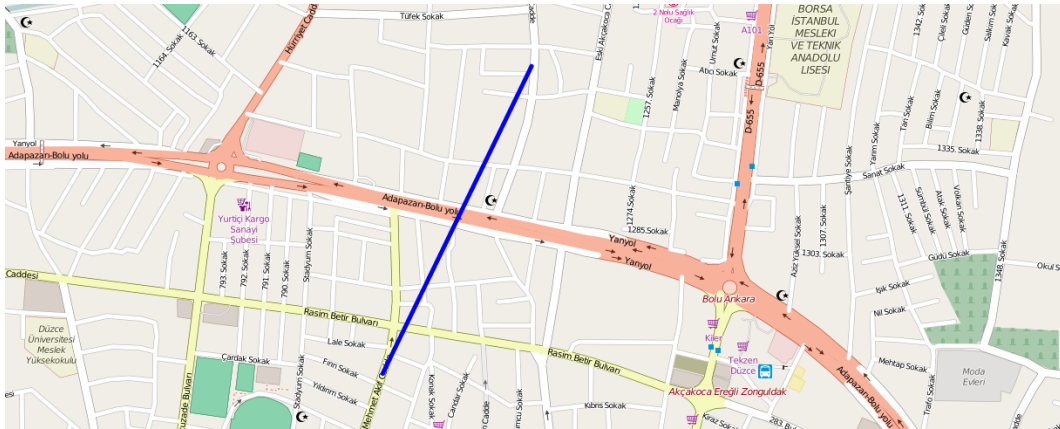
```

lineStyle.strokeWidth = 5;
var points = [
    new OpenLayers.Geometry.Point( 31.156335, 40.844283
).transform(epsg4326, projectTo),
    new OpenLayers.Geometry.Point( 31.16, 40.85 ).transform(epsg4326,
projectTo)
];
var feature = new OpenLayers.Feature.Vector(
    new OpenLayers.Geometry.LineString(points)
);
feature.style=lineStyle;
vectorLayer.addFeatures(feature);
map.addLayer(vectorLayer);
</script>
</body>
</html>

```

Yukarıdaki kod örneğinde standart HTML ve OpenLayers JavaScript kodlarıyla sayfa ve harita oluşturulmaktadır. Points dizi değişkeni ile çizilecek konumların noktaları tanımlanmakta, linestyle değişkeni ile çizgiye stil verilmiş ve feature değişkenine bu biçim uygulanmaktadır.

Şekil 2.3'te OpenLayers ile çizgi çizme örneğinin sonucu görülmektedir.



Şekil 2.3 OpenLayers çizgi çizme.

### 2.2.1.2 Open Source Route Machine (OSRM)

Open Source Route Machine iki nokta arasındaki yol bilgilerini veri olarak sunan açık kaynak bir yazılımdır. Girilen iki nokta arasındaki yolların noktalarını, cadde isimlerini şifrelenmiş bir veri olarak bize sunmaktadır [7]. Bu şifreleme işleminde Google'ın Polyline algoritmasını kullanmaktadır [8]. OSRM birçok ticari ve kurumsal projede kullanılmaktadır.

OSRM Linux işletim sisteminin birçok sürümüne kurulabilmektedir. Örneğimizde Ubuntu 14.04 sunucuya kurulmuştur. Sistem gereksinimi olarak minimum tek çekirdekli bir işlemciye ve 4 Gigabyte ana belleği olan bir sanal veya gerçek makineye ihtiyaç duymaktadır. Ayrıca işletim sisteminde takas (swap) alanı ayrılması önerilmektedir. İşletim sistemi, ana belleğin yetersiz geldiği durumlarda takas alanını kullanmaktadır.

Linux'da komutları çalıştırabileceğimiz terminalde yazacağımız komutlarda yönetici yetkisi istediğinden komut başlarında "sudo" kullanılmıştır.

Öncelikle işletim sistemimizdeki güncelleştirmeleri yapmak için aşağıdaki komutların girilmesi gerekmektedir.

```
>> sudo apt-get update
>> sudo apt-get upgrade
>> sudo apt-get install unattended-upgrades
```

Güncellemeler yapıldıktan sonra osrm adlı bir klasör oluşturulup giriş yapılabilir.

```
>> mkdir osrm
>> cd osrm
```

Bu işlemten sonra kullanacağımız haritanın verisinin indirilmesi gerekmektedir. <http://www.openstreetmap.org> adresinden harita üzerinde istediğimiz bölümleri seçip indirebileceğimiz gibi GeoFabric [15] veya Mapzen [16] servislerinin sitelerinden de indirilebilir. İndirilen dosyayı FTP aracıyla sunucuya atılabilmekte veya komut satırından indirme yapılabilir.

```
>> wget -O map.osm dosyaadresi
```

Üstteki komutla indirilecek olan dosyanın adı ve uzantısı map.osm olmaktadır.

OSRM'yi işletim sistemimizde derlenmelidir. Bunun için öncelikle derleme için gerekli olan uygulamalarının (bağımlılıklar) yüklenmesi sağlanmalıdır. Bu işlemler için aşağıda verilen komutlar sırasıyla çalıştırılmalıdır.

```
>> sudo apt-get install build-essential git cmake
```

```
>> sudo apt-get install libboost-all-dev libtbb-dev liblua5.2-dev libluabind-dev libstxxl-  
dev libxml2 libxml2-dev libosmpbf-dev libbz2-dev libprotobuf-dev
```

Sonrasında kaynak kodlar indirilip derlenmelidir. OSRM'nin kaynak kodları yine bir özgür yazılım projesi olan GitHub [17] üzerinde bulunmakta ve kullanıma sunulmaktadır.

Aşağıdaki komutla OSRM projesinin kodları bilgisayara kopyalanabilmektedir.

```
>> git clone https://github.com/Project-OSRM/osrm-backend.git
```

Kopyalanan klasöre aşağıdaki komutlarla giriş yapıp, derleyip kurulması işlemi gösterilmektedir.

```
>> cd osrm-backend
```

```
>> mkdir build
```

```
>> cd build
```

```
>> cmake ..
```

```
>> sudo make install
```

Bu işlemlerden sonra kurulum gerçekleşmektedir. map.osm dosyası içerisindekileri bir klasöre çıkarmak ve kullanıma hazır hale getirmek için aşağıdaki komutun çalıştırılması gerekmektedir.

```
>> osrm-extract map.osm
```

Seyahat süresi hesapları için aşağıdaki komut çalıştırılabilir.

```
>> osrm-prepare map.osrm
```

Yönlendirme hizmetinin başlatılması için aşağıdaki komut kullanılmaktadır.

```
>> osrm-routed map.osrm
```

Üstteki komuttan sonra sunucu çalışır hale gelmektedir. Adrese Zengin Metin Transfer Protokolü (HTTP) isteği göndermek için;

```
http://ipadresi:5000
```

adresini kullanılmaktadır. İki nokta arasındaki yol bilgilerini vermesi için adrese iki noktanın enlem ve boylam koordinatlarını GET metoduyla gönderilmesi gerekmektedir [18].

OSRM'nin kullanımı için Düzce ilindeki iki nokta arasındaki bilgileri alabiliriz. Bu örnek için adres olarak;

```
http://ipno/viaroute?loc=40.86878,31.191711&loc=40.8391691,31.1547847&instructions=true
```

kullanılabilmektedir.

Bu adrese HTTP isteği gönderildiğinde Şekil 2.4'teki çıktı elde edilmektedir.

```

{"status":200,"hint_data":{"locations":
["0f9YC9L_WASAAAAAAAAAFYAAACJBAAAngAAAMqZjwGC9H8CbawAAHCjwL9tsBGwABAQ","pgXrDNp0LQ9AOF4AAQAAAF
EAAAAAdAAAAAAAAAFUEVgP1_PwHbawAACMowBJLYdsBAQABAQ"],"checksum":569422412},"route_name":["Kuzey
Çevre Yolu","Zaid Ül Kevseri Bulvarı"],"found_alternative":false,"route_summary":
{"total_distance":9051,"total_time":611,"end_point":"Resmi
Otopark","start_point":"","route_instructions":[[{"10","","",1001,0,116,"1000m","E",103,1,"W",283},
["1","Kuzey Çevre Yolu",1185,27,67,"1184m","E",110,1,"W",290],[{"11-1","Zaid Ül Kevseri
Bulvarı",3019,40,162,"3018m","SW",204,1,"NE",24],[{"1","","",31,58,2,"31m","SW",235,1,"NE",55},
["1","Zaid Ül Kevseri Bulvarı",342,59,27,"342m","SW",235,1,"NE",55],[{"1","Zaid Kevseri
Bulvarı",518,62,28,"518m","SW",233,1,"NE",53],[{"7","Zaid Kevseri
Bulvarı",659,68,37,"658m","SW",231,1,"NE",51],[{"1","","",44,73,3,"43m","SW",233,1,"NE",53},
["1","Zahid Kevseri Bulvarı",263,74,21,"263m","SW",233,1,"NE",53],[{"11-1","Bolu
Caddesi",775,80,57,"775m","W",291,1,"E",111],[{"1","İstanbul
Caddesi",691,100,48,"690m","W",286,1,"E",106],[{"11-3","","",261,130,20,"260m","S",182,1,"N",2},
["1","Mimar Sinan Caddesi",15,135,3,"14m","E",90,1,"W",270},
["7","","",146,136,11,"146m","N",349,1,"S",169],[{"3","Resmi
Otopark",19,140,0,"19m","E",101,1,"W",281],[{"15","","",0,141,0,"0m","N",0,1,"N",0}],
"via_points":
[[40.870768,31.192799],[40.839203,31.154507]],
"via_indices":[0,141],"status_message":"Found route
between
points","route_geometry": "_vp}lA}lznz@xBkTnG}eBrFch@hIqa@~J{WjKuOzK}GbrkFfXaYrKeZrKi[d@{Pe@uO}AqN
eLea@o@{IZ_KrAoIxEGQI{IqDoIiSs_@eLmW}PaYcEsQkF{P{E}MrZadBd]qfBtm@y~Bvf@a_BxLu[h1@clArc@_A~Z}k@nK
_S|@cBnp@ydAlGmHvHdBxpAdn@nF`Ct`@dzv}pYvrAl}AndjEt{CfkDfp@z|@ru@bfAj{@prAnz@nwAz_@bs@lVjg@bErIjdA
zhCfhAdnCxUrk@roGrnObI|Qn`Aj{Bh^jr@jPn[tJpSv`@`t@pi@xaA`\\fi@jd@ju@c@nt@dVbe@zg@dbAvOh\\~i@njArc
BtqDzMxXxn@jraAd^dw@tHxOLzBl@lBdAtAgGb\\}Fn]}Gj^eA~LmK~aAu@`HqIh]mIhVoExMiFhP_Mh\\mGnR{AtF}Pfp@eE~
IgdTiwKf[oGrWwEdXsJdt@wDpYuAvJg@rKe@xKwA~YqA~Zm@nFgAfJcAnIgaAlJmCbT_AnHyA~JyEf\\kFjVoMde@kBbHiFdTa
BzFaKze@w@zIoA`XwE`fAw@jMoCrEIpEtAfErAfA~A~bBOnS`@`XdE`ZpDj_A`IvFnBA}IwKjBa}cDsKcA_YmCd@mL"}

```

Şekil 2.4. OSRM iki nokta arasındaki yol verileri.

JavaScript Nesne Gösterimi (JSON) verisi olarak gelen çıktı düzenlendiğinde Şekil 2.5 elde edilmektedir.

```

"via_points": [
  [
    [
      40.870768,
      31.192799
    ],
    [
      40.839203,
      31.154507
    ]
  ],
  "via_indices": [
    0,
    141
  ],
  "status_message": "Found route between points",
  "route_geometry": "_vp}lA}lznz@xBkTnG}eBrFch@hIqa@~J{WjKuOzK}GbrkFfXaYrKeZr
Ki[d@{Pe@uO}AqNeLea@o@{IZ_KrAoIxEGQI{IqDoIiSs_@eLmW}PaYcEsQkF{P{E}MrZadBd]
qfBtm@y~Bvf@a_BxLu[h1@clArc@_A~Z}k@nK_S|@cBnp@ydAlGmHvHdBxpAdn@nF`Ct`@dzv
}pYvrAl}AndjEt{CfkDfp@z|@ru@bfAj{@prAnz@nwAz_@bs@lVjg@bErIjdAzhCfhAdnCxUrk
@roGrnObI|Qn`Aj{Bh^jr@jPn[tJpSv`@`t@pi@xaA`\\fi@jd@ju@c@nt@dVbe@zg@dbAvOh
\\~i@njArcBtqDzMxXxn@jraAd^dw@tHxOLzBl@lBdAtAgGb\\}Fn]}Gj^eA~LmK~aAu@`HqIh]
mIhVoExMiFhP_Mh\\mGnR{AtF}Pfp@eE~IgdTiwKf[oGrWwEdXsJdt@wDpYuAvJg@rKe@xKwA~
YqA~Zm@nFgAfJcAnIgaAlJmCbT_AnHyA~JyEf\\kFjVoMde@kBbHiFdTaBzFaKze@w@zIoA`XwE
`fAw@jMoCrEIpEtAfErAfA~A~bBOnS`@`XdE`ZpDj_A`IvFnBA}IwKjBa}cDsKcA_YmCd@mL"

```

Şekil 2.5. JSON verinin parsellenmiş hali.

Buradaki `route_geometry` bilgisi Google'ın Polyline algoritması ile şifrelenmiş veri topluluğudur. Polyline algoritmasını tersine çözdüğümüzde bize iki nokta arasındaki yolun yönlendirme noktaları bilgisini enlem boylam bilgileriyle birlikte vermektedir. Harita üzerinde bu noktalar arasına çizim yapıldığında iki nokta arasındaki yol bilgisine ulaşılabilmektedir.

## 2.2.2 Google Haritalar

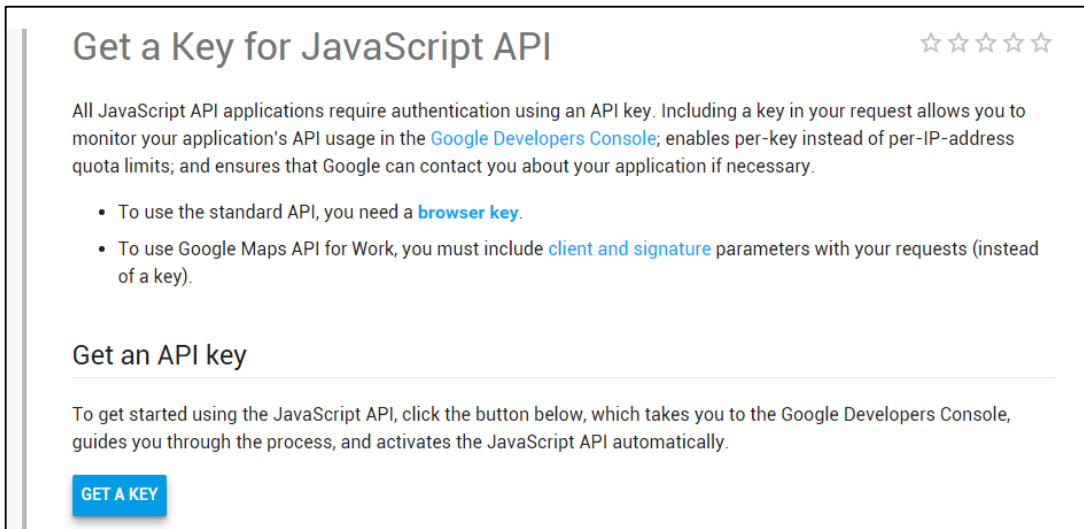
Google'ın sunduğu çevrimiçi harita hizmetidir. Web projelerinde en çok kullanılan harita hizmetlerindedir. Mobil, web, web servislerinde kullanım için UPA desteği sunmaktadır.

<https://developers.google.com/maps> adresinden UPA'lara ulaşılabilmektedir.

Google haritalar belirli bir limite kadar ücretsiz kullanılabilir. Limit aşıldığında Google ekstra ücret talep etmektedir [19].

### 2.2.2.1 Google Haritaların Web Sayfasında Kullanımı

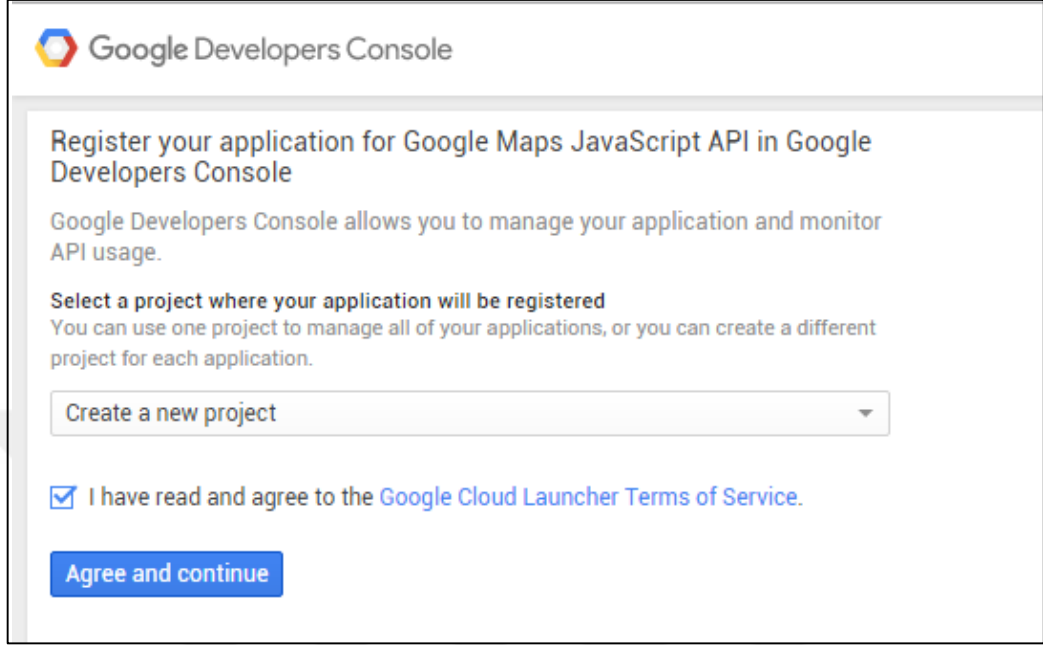
Google haritaları kullanmak için öncelikle bir UPA anahtarı (key) alınması gerekmektedir. Geliştirici sayfasındaki "Get a Key" düğmesi tıklanarak anahtar alınabilmektedir. Şekil 2.6'da anahtar alımı ekranı görülmektedir.



Şekil 2.6. Google Haritalar UPA anahtar alımı.



“Get a key” düğmesine tıklandıktan sonra gelen sayfada lisans sözleşmesi kabul edilip yeni bir proje oluşturulacağı belirtilmelidir. Şekil 2.7’de anahtar için yeni proje oluşturma ekranı verilmektedir.



Google Developers Console

Register your application for Google Maps JavaScript API in Google Developers Console

Google Developers Console allows you to manage your application and monitor API usage.

Select a project where your application will be registered  
You can use one project to manage all of your applications, or you can create a different project for each application.

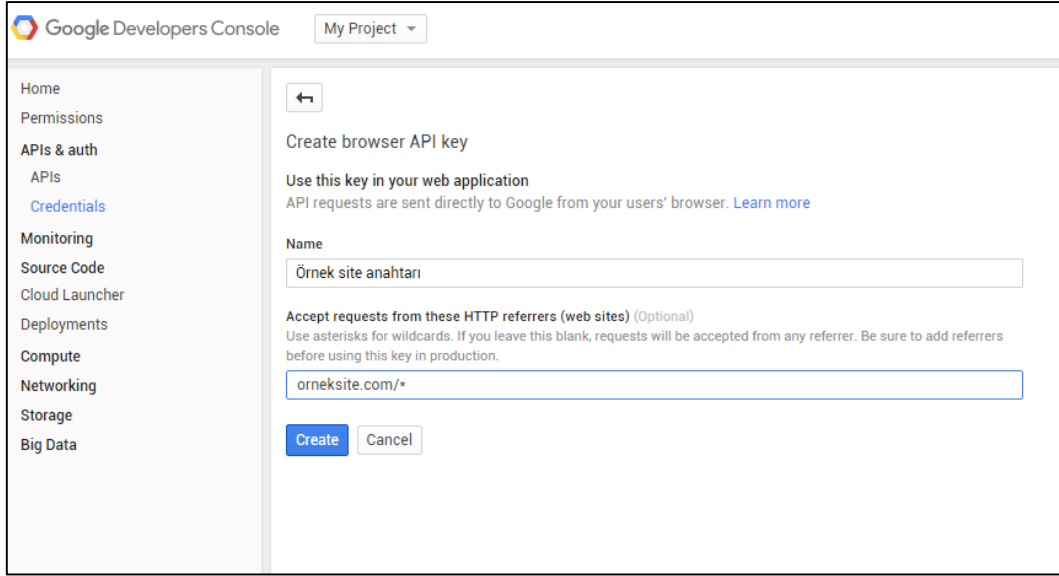
Create a new project

I have read and agree to the [Google Cloud Launcher Terms of Service.](#)

Agree and continue

Şekil 2.7. Google Haritalar UPA anahtar için yeni proje oluşturma.

“Agree and continue” (kabul ve devam et) düğmesi tıklandığında gelen sayfada site adı ve adresi girilmesi gerekmektedir. Sitedeki tüm sayfalarda kullanılacaksa sonuna \* eklenebilmektedir. Şekil 2.8’de UPA anahtarı üretme ekranı görülmektedir.



Şekil 2.8. Google Haritalar UPA anahtar üretim ekranı.

“Create” (oluştur) düğmesi tıklandıktan sonra Şekil 2.9’da gösterilmekte olan gerekli UPA anahtarı sunulmaktadır.



Şekil 2.9. Google Haritalar alınan anahtar ekranı.

Haritayı web sayfası içerisinde göstermek için temel komutların örneği aşağıdadır. HTML kodlarının içerisinde JavaScript kodları yazılmıştır.

```
<!doctype html>
<html>
<head>
  <title>Örnek Harita</title>
  <meta name="viewport" content="initial-scale=1.0">
  <meta charset="utf-8">
</style>
  html, body {
```

```
        height: 100%;
        margin: 0;
        padding: 0;
    }
    #map {
        height: 100%;
    }
</style>
</head>
<body>
    <div id="map"></div>
    <script>
        var map;
        function initMap() {
            map = new google.maps.Map(document.getElementById('map'), {
                center: {lat: -34.397, lng: 150.644},
                zoom: 8
            });
        }
    </script>
    <script
src="https://maps.googleapis.com/maps/api/js?key=UPAANAHTARINIZ&callb
ack=initMap"
        async defer></script>
</body>
</html>
```

Üstteki kodda UPAANAHTARINIZ yazan bölüme alınan UPA numarası yazılmalıdır. <div id="map"></div> ile haritanın gösterileceği HTML etiketi açılmakta, initMap metoduyla da harita çağırılmaktadır. Initmap metodu içerisinde map adlı bir sınıf oluşturulmakta, center ve zoom özelliklerine değer girilmektedir. Center ile haritanın merkezi enlem ve boylam olarak, zoom ile de yakınlaştırma derecesi belirtilmektedir. Google harita hizmetinin kullanımı için gerekli olan JavaScript kütüphanesi de script etiketi ile çağırılmaktadır.

Haritalar üzerine işaretçi, çizgiler, poligon, çember tarzı çizimler yapılabilir.

#### 2.2.2.2 İşaretçi(marker) Ekleme

Oluşturulan haritaya Google.maps.Marker sınıfından türetme işlemi yapılarak yeni bir işaretçi eklenebilir.

```
function initMap() {  
    var myLatLng = { lat: -25.363, lng: 131.044 };  
    var map = new google.maps.Map(document.getElementById('map'), {  
        zoom: 4,  
        center: myLatLng  
    });  
  
    var marker = new google.maps.Marker({  
        position: myLatLng,  
        map: map,  
        title: 'Merhaba Dünya'  
    });  
}
```

Üstteki örnekte marker adlı bir değişken google.maps.Marker sınıfından türetilip pozisyon, harita ve başlık bilgisi girilerek haritaya eklenmiştir.

### 2.2.2.3 İşaretçiye Özel Resim Ekleme

İşaretçilere özel resimler eklenebilmektedir. Uygulamamızda her bir bireyin resmini işaretçi olarak belirlediğimizde daha kullanışlı olacaktır.

```
var image = 'images/people.png';  
var beachMarker = new google.maps.Marker({  
    position: { lat: -33.890, lng: 151.274 },  
    map: map,  
    icon: image  
});
```

İşaretçiye bir resim eklenmesi örneği için kullanılan kod yukarıdadır.

### 2.2.2.4 İşaretçilerde Bilgi Kutuları

İşaretçilere bilgi kutuları eklenebilmekte, işaretçilerle ilgili bilgilere ulaşılabilir. Şekil 2.10'da işaret kutusu örneği gösterilmektedir.



Şekil 2.10. İşaret kutusu örneği.

Örnek JavaScript kodu:

```
function initMap() {  
    var uluru = { lat: -25.363, lng: 131.044 };  
    var map = new google.maps.Map(document.getElementById('map'), {  
        zoom: 4,  
        center: uluru  
    });  
    var contentString = '<div id="content">' +  
        '<div id="siteNotice">' +  
        '</div>' +  
        '<h1 id="firstHeading" class="firstHeading">İstek</h1>' +  
        '<div id="bodyContent">' +  
        '<p>Arif Ceylan adlı kişiden istek</p>' +  
        '</div>' +  
        '</div>';  
  
    var infowindow = new google.maps.InfoWindow({  
        content: contentString,  
        maxWidth: 200  
    });  
  
    var marker = new google.maps.Marker({  
        position: uluru,  
        map: map,  
        title: 'AC (Arif C)'  
    });  
    marker.addListener('click', function () {  
        infowindow.open(map, marker);  
    });  
}
```

### 2.2.2.5 Yönlendirme UPA'sı

Yönlendirme UPA'sı ile birden fazla konum arasındaki yol tarifi, mesafesi, süresi ile ilgili bilgilere ulaşılabilmektedir. Bunun için de Google Developers Console sitesinden UPA için anahtar almak gerekmektedir.

### 2.2.3 Yandex Haritalar

Yandex Haritalar da kendi harita sisteminin web sayfaları üzerinde kullanımına izin veren UPA geliştirmekte ve sunmaktadır.

#### 2.2.3.1 Yandex Haritaların Web Sayfalarında Kullanımı

Yandex harita UPA'larını kullanabilmek için öncelikle script dosyasının web sayfasına eklenmesi gerekmektedir.

```
<script src="http://api-maps.yandex.ru/2.1/?lang=en_US"
type="text/JavaScript"></script>
```

Üstteki kodlarla bu işlem gerçekleştirilebilmektedir.

Sayfa içerisine de harita gösterilecek alanın eklenmesi gerekmektedir.

```
<div id="harita" style="width: 600px; height: 400px"></div>
```

Body etiketleri içerisinde id özelliği harita olan, genişliği 600 pixel, yüksekliği 400 pixel bir harita alanı eklenmektedir.

Haritayı başlatmak için gerekli JavaScript kodlarının yazılması gerekmektedir.

```
<script type="text/JavaScript">
  ymaps.ready(init);
  var harita;
  function init(){
    harita = new ymaps.Map("harita", {
      center: [40.83, 31.16],
      zoom: 7
    });
  }
</script> [20]
```

Haritayı başlatmak için kullandığımız yukarıdaki koda öncelikle harita adlı bir değişken oluşturulup init fonksiyonu içerisinde tanımlaması yapıp özelliklerinin verilmesi (center ve zoom) işlemi gerçekleştirilmektedir. init fonksiyonu da ymaps.ready içerisinde çağrılmaktadır. Harita merkezi olarak center içerisinde enlem, boylam bilgileri girilmekte ve zoom ile de yakınlaştırma değeri verilmektedir.

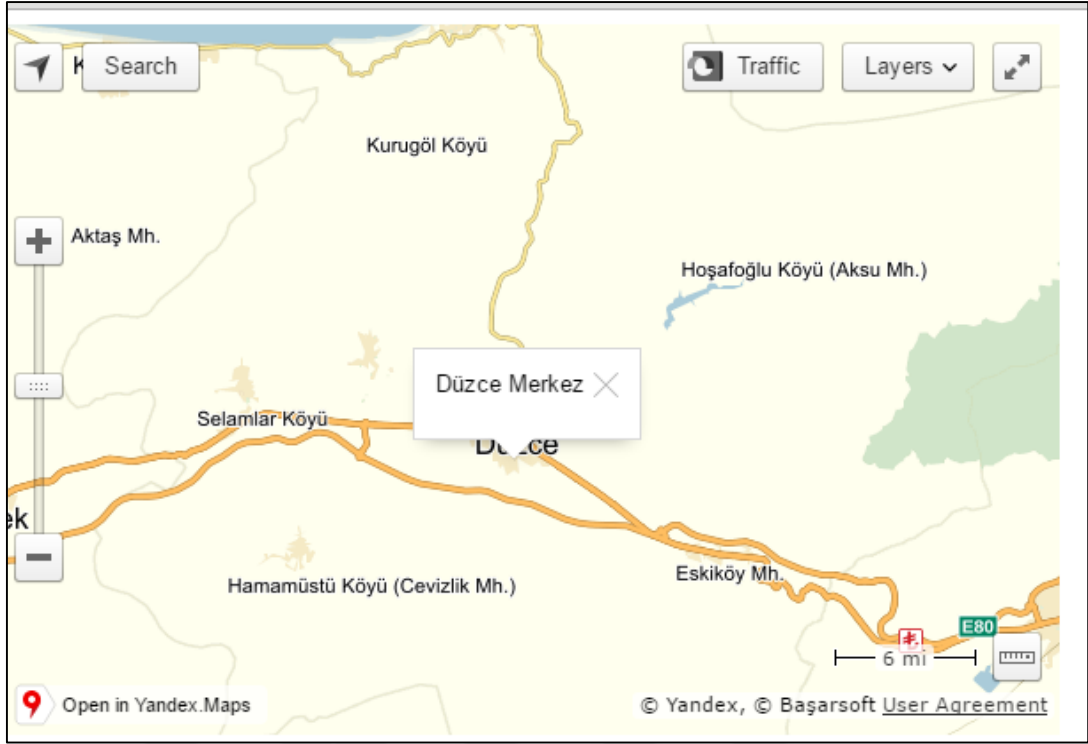
### 2.2.3.2 İşaretçi Ekleme

Harita üzerinde işaretçi ekleme için aşağıdaki JavaScript kodları kullanılabilir.

```
<script type="text/JavaScript">
  ymaps.ready(init);
  var harita,
      isaretci;

  function init(){
    harita = new ymaps.Map("harita", {
      center: [40.83, 31.16],
      zoom: 7
    });
    isaretci = new ymaps.Placemark([40.83, 31.16], {
      hintContent: 'Düzce',
      balloonContent: 'Düzce Merkez'
    });
    harita.geoObjects.add(myPlacemark);
  }
</script> [20]
```

Bu kodlarda Yandex Haritaların Placemark özelliği kullanılarak haritaya işaretçi eklemesi yapılmıştır. Örnekte Düzce ilinin enlem ve boylamı kullanılmış olup başlık olarak da Düzce Merkez içeriği gösterilmiştir. Şekil 2.11’de Yandex Haritalar’da kullanılan işaretçi ve açıklama kutusu örneği gösterilmektedir.



Şekil 2.11. Yandex Harita işaretçi ve açıklama kutusu örneği.

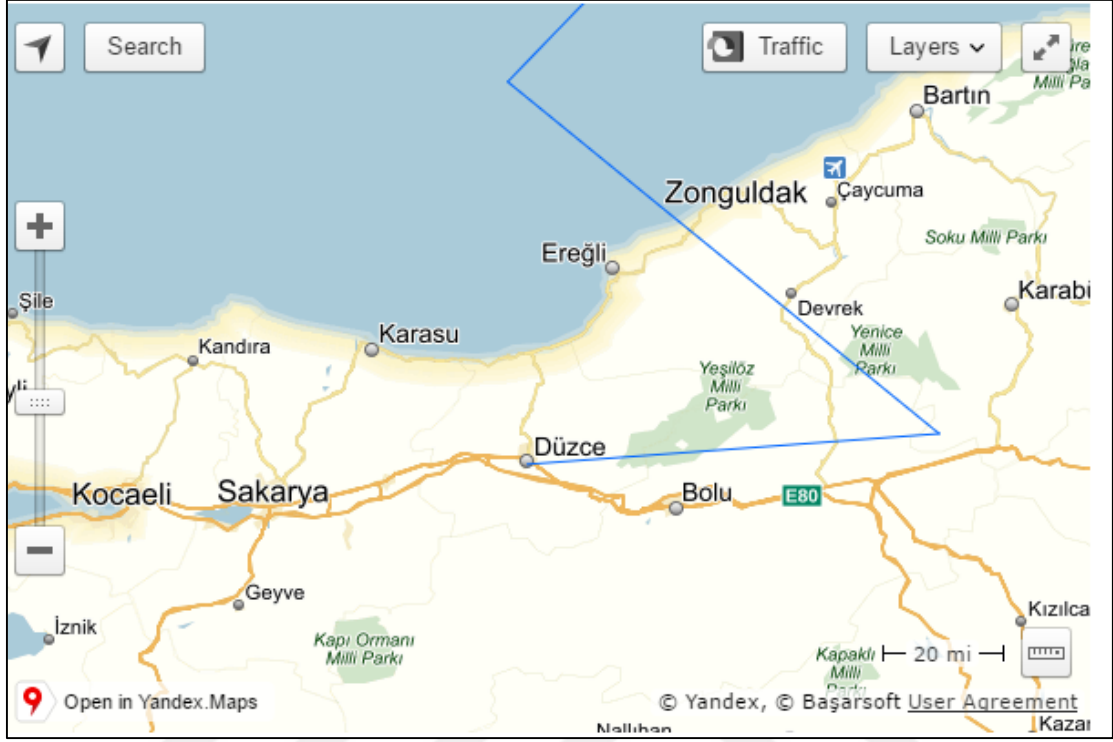
### 2.2.3.2 Çoklu Çizgi Çizimi

Yandex Harita UPA'ları ile çoklu çizgi çizimi örneği için örnek kodlar aşağıda gösterilmektedir.

```
<script type="text/JavaScript">
  ymaps.ready(init);
  var harita;
  function init(){
    harita = new ymaps.Map("map", {
      center: [40.83, 31.16],
      zoom: 7
    });
    var coklucizgi = new ymaps.Polyline(
      [[40.83, 31.16],[40.90, 32.40],[41.70, 31.10],[42.70, 32.40]]
    );
    harita.geoObjects.add(coklucizgi);
  }
</script> [21]
```



Üstteki koda çoklucizgi adlı PolyLine nesnesi oluşturup, enlem ve boylam bilgileri girilmektedir. Haritaya eklenen bu çoklucizgi nesnesi Şekil 2.12'deki görüntüyü oluşturmaktadır.



Şekil 2.12. Yandex Haritalar çoklu çizgi çizimi örneği.

## 2.3 MOBİL PROGRAMLAMA

Günümüzde akıllı telefonların yaygın bir şekilde kullanıldığı göz önünde bulundurulduğunda mobil programlamanın çok önemli bir konumda olduğu anlaşılmaktadır. Mobil programlama telefonun işletim sisteminin desteklediği dil ile Native olarak veya melez programlama dediğimiz aracı yazılım sayesinde yapılabilmektedir.

### 2.3.1 Native Programlama

Native Programlama telefonun desteklediği diller ile telefonun işletim sistemine özel geliştirilen programlama yöntemidir. Örneğin Android işletim sistemli telefonlarda Java dili desteklenmekte, Java ile Android telefonlara program yazılabilmektedir. Iphone marka telefonlarda IOS işletim sistemi bulunmakta, C dilinden türetilen Objective C ile IOS işletim sistemine özel programlar yazılabilmektedir.

### 2.3.2 HTML5 Web Uygulamaları

Web tarayıcısı üzerinden yönetilen uygulamalardır. Bu tarz uygulamalar cihazdan bağımsızdır. İnternet olan her telefonda kullanılabilir. Fakat cihazın sunduğu donanımsal imkânlar bu uygulama yönteminde kullanılamamaktadır ve çevrimdışı çalışmamaktadır.

### 2.3.3 Melez (Hybrid) Programlama

Hemen hemen tüm telefonlar web tarayıcı bileşenine sahiptir. Yazdığımız program içerisine web tarayıcı bileşeni gömülebilmektedir. Günümüz telefonlarının hemen hemen hepsi HTML5 desteği sunmaktadır. HTML5 ile güncel yeni etiketler kullanılabilir, veri kaydetme olanağı sunmakta, konum alma sistemlerini kullanmaktadır. Melez programlamanın temelinde uygulama içerisine gömülen bir web tarayıcısı bileşeninde web sayfası gösterilmesi ve bütün işlemlerin bu web sayfasında yapılması mantığı vardır. Melez Programlama Native ve HTML5 Web uygulaması yönteminin avantajlarını kullanmakta, iki yöntemi de kapsamaktadır. Ayrıca telefonun donanımsal özellikleri için kodlamalar (fonksiyonlar oluşturulup) yapıp, bu tarayıcı üzerinden o kodlamalar çağırılabilir. [22]

Oluşturulan bazı kütüphanelerde her telefona özel fonksiyonlar üretilmekte, kullanımımıza sunulmaktadır. Bunlara örnek olarak Phonegap, Apache Cordova, Ibm Bluemix gösterilebilir.

#### 2.3.3.1 Avantajları

Melez programlamanın avantajları aşağıda listelenmiştir.

- Hemen hemen her akıllı telefonda çalışabilmektedir.
- Tek bir programlama diliyle uygulama geliştirilebilmekte, tüm telefonlarda Native olarak kullanılan programlama dilini öğrenme gerekliliğini ortadan kaldırmaktadır.
- Native ve HTML5 uygulama geliştirme yöntemlerinin avantajlarını kullanmaktadır.
- Çevrimiçi ve çevrimdışı olarak çalışabilmektedir.
- HTML5 ve CSS3 yöntemlerini kullanma imkânı sağlamaktadır.

- Geliştirilen kütüphane sayısı gün geçtikçe artmaktadır.

### 2.3.3.2 Mobil Melez Çatılar (Mobile Hybrid Frameworks)

Mobil Melez Programlama üzerine hazır çatılar bulunmakta, sürekli olarak yenileri çıkmakta ve geliştirilmektedir. Kullanılan bazı çatılara örnek olarak aşağıdakileri verebiliriz. Bu çatılardan Ionic Framework tezimizdeki mobil uygulamada kullanılmış olup detaylı bir şekilde anlatılmaktadır.

**Apache Cordova:** Mobil ortamda melez ve native uygulamalar geliştirmek için kullanılan bir platformdur. İçerisindeki eklentilerle aygıtın özelliklerini kullanmaya imkân sunmaktadır. Birçok çatı Apache Cordova'nın eklentilerini kullanmaktadır. Ücretsiz ve açık kaynak kodlu olarak sunulmaktadır. Son sürümünde Nodejs ve npm-Node Paket Yöneticisi (Node Packet Manager) uygulamasını kullanarak komut satırı üzerinden kurulum, yeni proje oluşturma, eklenti kurulumu gibi imkânlar sağlamaktadır. Windows ve Linux işletim sistemlerinde kullanılabilir [23].

Apache Cordova Kurulumu için Npm ile Apache Cordova kurulumu için aşağıdaki satırın komut satırına girilmesi gerekmektedir.

```
>> npm install -g cordova [24]
```

Apache Cordova Yeni Proje Oluşturmak için aşağıdaki komutun komut satırına girilmesi gerekmektedir.

```
>> cordova create MyApp [24]
```

Bu komut girildiğinde MyApp adlı bir klasör oluşturmakta ve temel klasör ve dosyaları içerisine atmaktadır. Şekil 2.13'de kurulum sonrası oluşan klasör ve dosyalar görülmektedir.

Name	Date modified	Type
hooks	6.11.2015 19:05	File folder
platforms	6.11.2015 19:05	File folder
plugins	6.11.2015 19:05	File folder
www	6.11.2015 19:05	File folder
config.xml	6.11.2015 19:05	XML File

**Şekil 2.13.** Kurulum sonrası oluşan klasör ve dosyalar.

Platforms klasöründe platformlar için gerekli dosyalar bulunmaktadır. Platformlara örnek olarak web tarayıcısı, Android, IOS, Blackberry, Windows mobile verilebilir [25].

Plugins klasörüne eklentiler yüklenebilmektedir. Kamera, bildirim, titreşim gibi telefona ait özellikleri kullanabilmek için eklentiler bulunmaktadır.

www klasöründe HTML5 web dosyaları bulunmaktadır.

Apache Cordova Projesine platform ekleme ve çalıştırmak için;

```
>> cordova platform add platformadi
```

Üstteki komut ile projeye platform eklemesi yapılabilmektedir.

Eklenebilecek platformlar;

- Android
- Blackberry10
- Browser
- Ios
- Ubuntu
- Windows
- Wp8 [26].

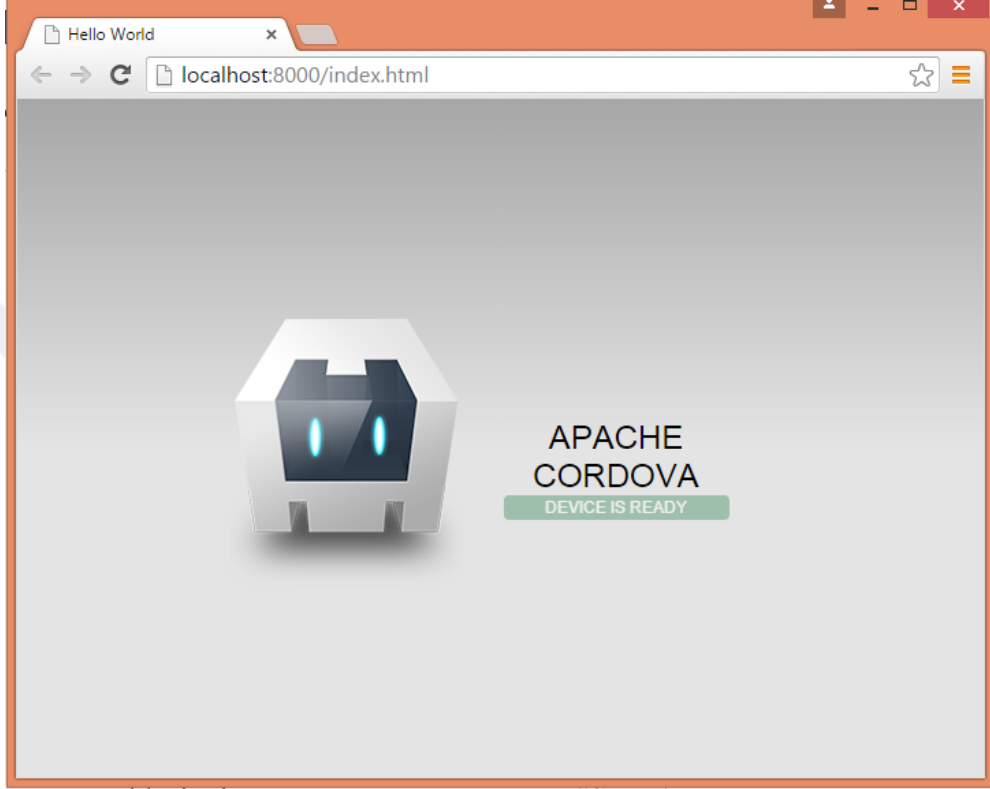
Örneğin web tarayıcısı için platform eklemek istenirse;

```
>> cordova platform add browser
```

komutunun yazılması gerekmektedir. Tarayıcı üzerinde uygulamayı çağırmak için;

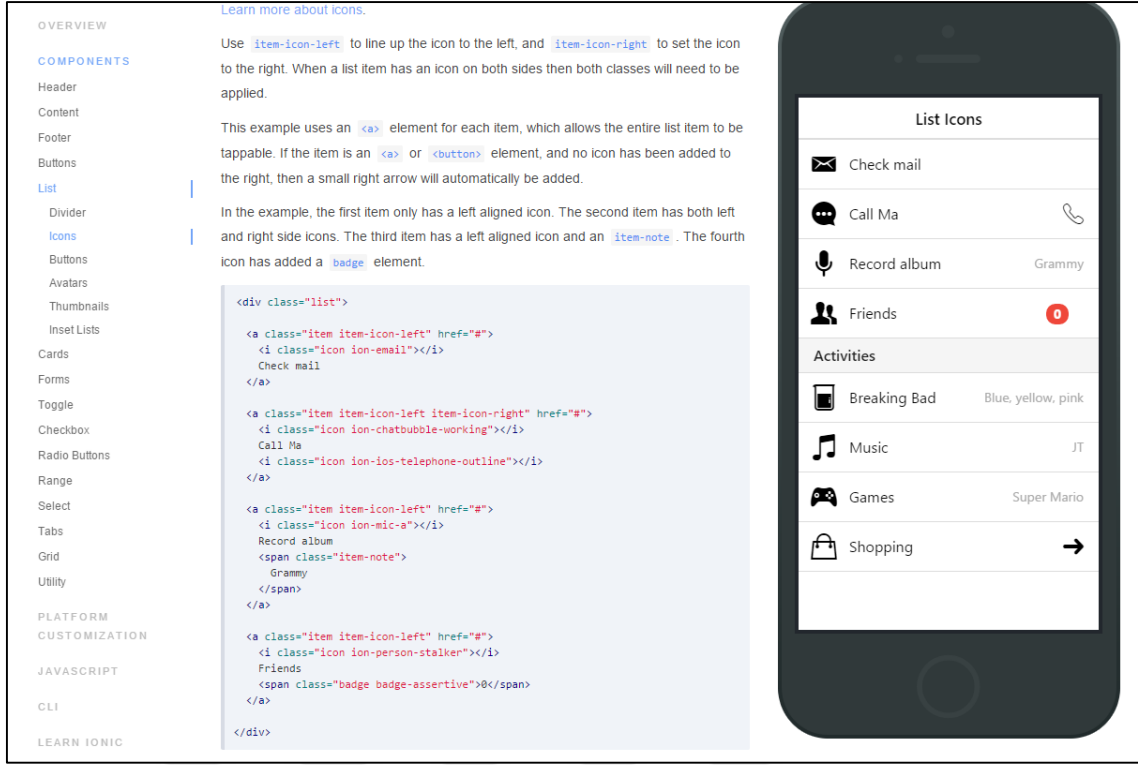
```
>> cordova run browser
```

komutu kullanılmaktadır [24]. Komut çalıştırıldıktan sonra gelen uygulama görüntüsü Şekil 2.14'te görülmektedir.



**Şekil 2.14.** Tarayıcı üzerinde çalışan uygulamanın görüntüsü.

**Ionic Framework:** En popüler olan açık kaynak çatılardandır. Google tarafından geliştirilen Angularjs JavaScript kütüphanesini ve Apache Cordova alt yapısını kullanmaktadır. Angularjs ile MVC (Model-View-Controller) yapısı kullanılarak JavaScript uygulamaları yazılmaktadır. Ayrıca HTML5, CSS, JavaScript ile kodlanan hazır bileşenler sunmaktadır [27]. Şekil 2.15'de Ionic Framework'ün web sitesindeki dokümantasyon ekranı gösterilmektedir.



Şekil 2.15. Ionic Framework dokümantasyonu.

Tezimizde kişisel uygulama olarak Ionic Framework kullanılmaktadır. Ionic Framework Windows, Mac OSX, Linux işletim sistemlerinde çalışabilmektedir. Ionic Frameworkü kullanmak için Nodejs, Apache Cordova ve Ionic komut satır araçlarının kurulması gerekmektedir. Npm (Node Package Manager) ile Ionic Framework kurulumu için işletim sisteminde komut satırına aşağıdaki kod girilmelidir.

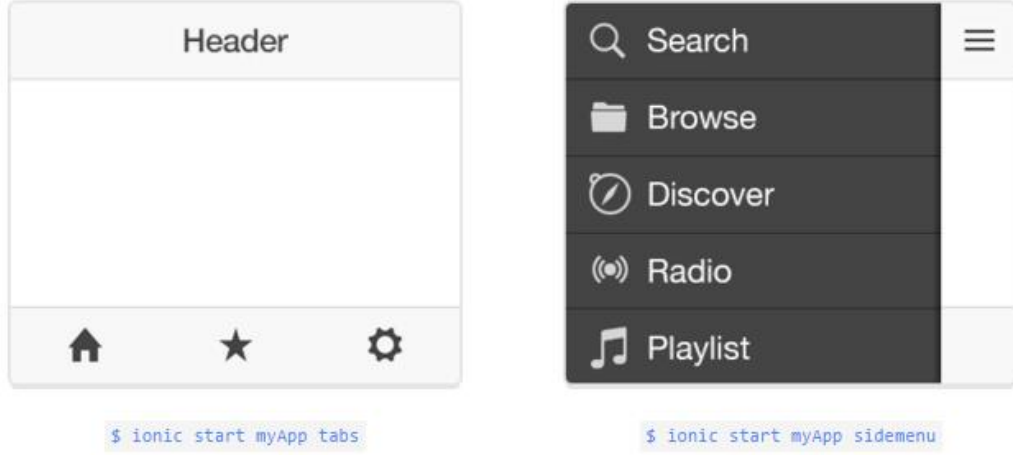
```
>> npm install -g cordova ionic [28]
```

Bu komuttan sonra Ionic Framework kurulumu gerçekleşecektir. Yeni bir proje oluşturmak için ionic start komutu çalıştırılmalıdır. Oluşturulacak proje boş proje olacağı gibi ionic framework içerisindeki hazır projelerden de oluşturulabilir.

Boş bir proje oluşturmak için komut satırına:

```
>> ionic start projeAdi blank [28]
```

komutu girilmesi gerekmektedir. Örnek sekme menüli uygulaması ve yan menüli proje oluşturmak için Şekil 2.16 temel alınabilir. Şekil 2.16’da oluşturulacak uygulamanın ön izlemesi ve komutları verilmektedir.



**Şekil 2.16.** Ionic örnek proje oluşturma.

Proje hangi platformlar için kullanılacaksa o platformların eklenmesi gerekmektedir. Platform eklemek için kullanılacak komut;

```
>> ionic platform add platformadi [28]
```

şeklindedir. Buradaki platformadi ios, Android, browser, windows olabilir.

Oluşturulan proje geliştirildikten sonra derlemek için;

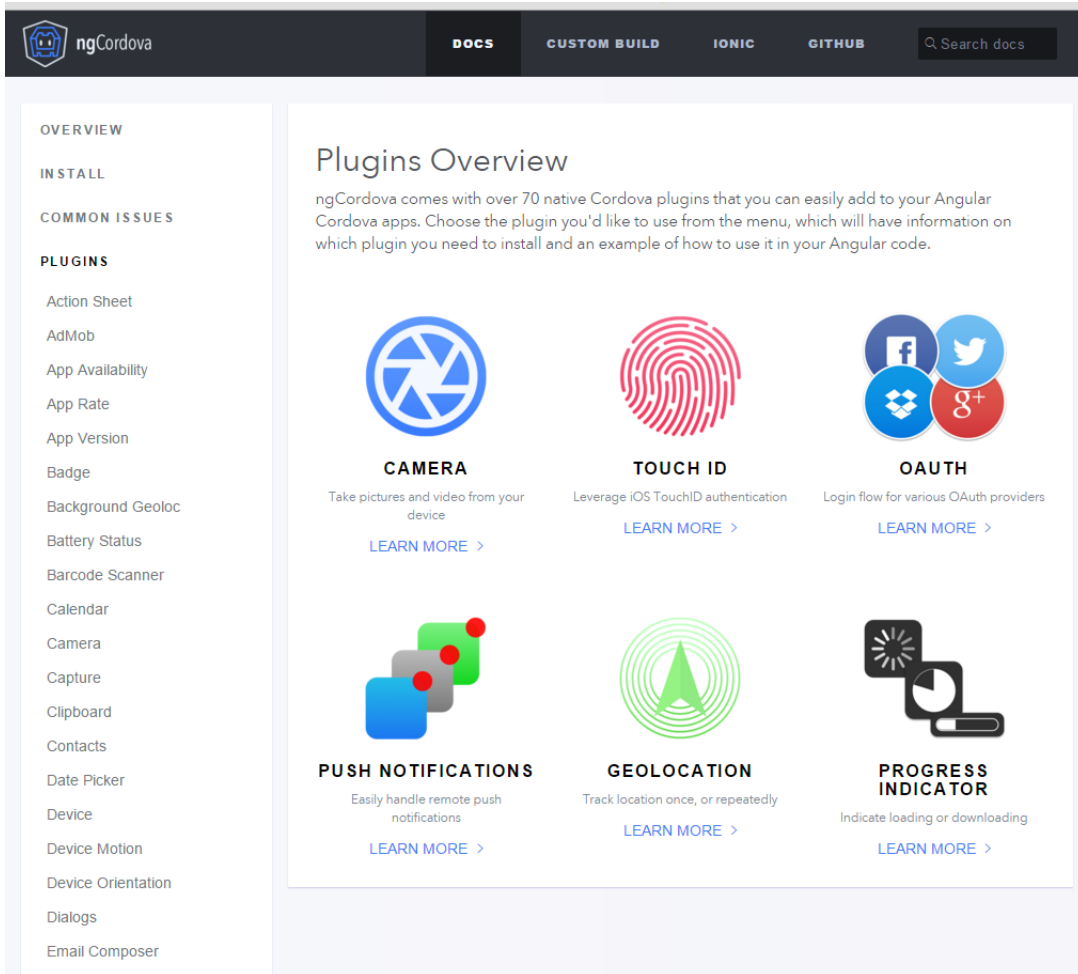
```
>> ionic build platformadi [28]
```

komutu kullanılmalıdır. Platform için gerekli SDK yüklü olduğu takdirde derleme işlemi yapılacak ve program mobil cihaza kurulabilir hale gelebilecektir. Emulätör üzerinde çalıştırmak için de;

```
>> ionic emulate platformadi [28]
```

komutu kullanılabilir. Ionic Framework içerisinde yazılacak programda kullanılmak üzere birçok bileşen bulunmaktadır. ionicframework.com sitesinde bu bileşenlere ulaşılabilir.

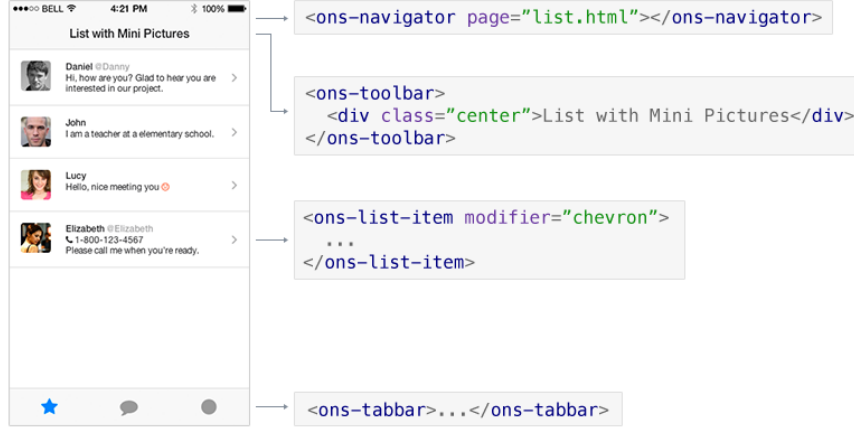
Ayrıca akıllı telefonun donanımsal özelliklerini kullanmamız gereken durumlar için geliştirilmiş eklentiler kullanılabilir. Bu özelliklere kamera, GPS, sensör, bildirim sistemi kullanımı örnek olarak verilebilir. Bu eklentiler yazılımcı tarafından yazılabileceği gibi internetten açık kaynak desteğiyle yazılmış olan eklentiler kullanılabilir. ngcordova.com sitesinde bu eklentiler listelenmiş, kullanım örnekleri verilmiştir. Şekil 2.17’de ngcordova.com sitesindeki eklenti listesi gösterilmektedir [29].



Şekil 2.17. ngcordova.com sitesi ve eklentiler.

**Onsen Ui:** Açık kaynak, Apache lisansına sahip yeni bir çatıdır. Yeni olmasına rağmen gelecek vaat etmektedir. JavaScript kütüphanesi olarak JQuery kullanılabilir. HTML, CSS, JavaScript yazarak uygulama tasarlanabilmekte, Apache Cordova/Phonegap ile de native hale getirilebilmektedir. Şekil 2.18’de Onsen Ui Framework’ün örnek kod ve ekranı gösterilmektedir.



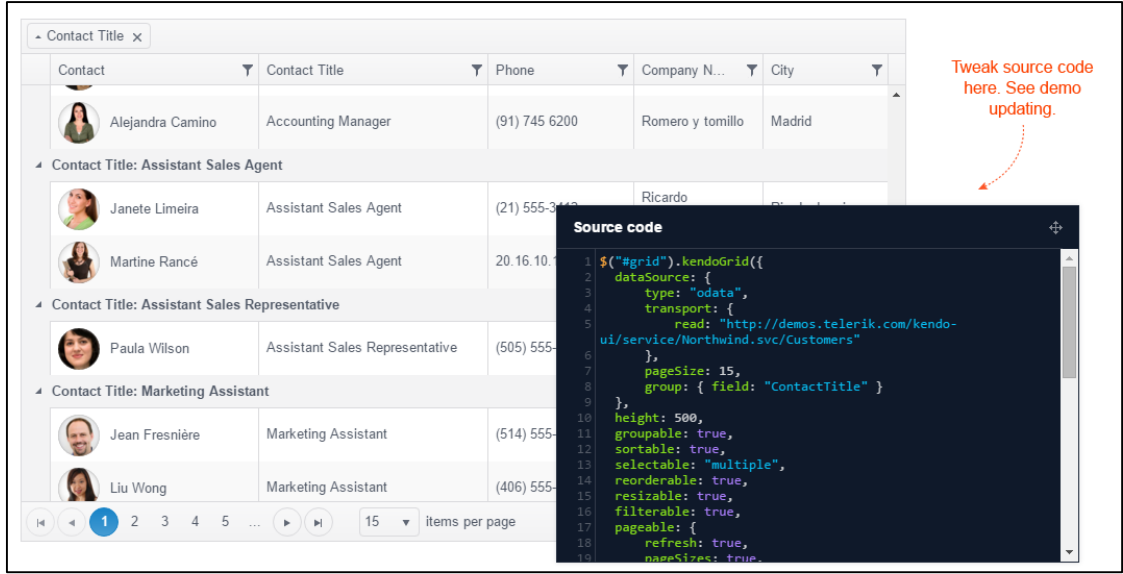


**Şekil 2.18.** Onsen Ui Framework.

**Intel XDK:** Intel'in geliştirdiği Intel XDK içerisinde melez uygulamalar geliştirebilmekte, kendi emulatöründe çalıştırabilir, hata ayıklama yapabilmektedir. Kendi içerisinde yine Intel tarafından geliştirilen App framework JavaScript kütüphanesini kullanmaktadır.

**Sencha Touch:** Extjs JavaScript kütüphanesini kullanan, ücretli bir framework tür. Ücretli olarak destek de vermektedirler.

**Kendo UI:** Telerik firmasının geliştirmiş olduğu web ve mobil uygulamalar geliştirmek için kullanılan çatıdır. JavaScript kütüphanesi olarak JQuery kullanmakta, içerisinde 70'den fazla JQuery bileşenini sunmaktadır. Şekil 2.19'da Kendo UI için örnek kod ve ekran görüntüsü gösterilmektedir.



Şekil 2.19. Kendo UI.

**Framework7:** Framework7 ücretsiz ve açık kaynak kodlu çatılardandır. Kolay kullanım, görsel tema ve bileşenler gibi avantajlar sunmaktadır.

**Jquery Mobile:** Jquery JavaScript kütüphanesinin mobil sürümüne uyarlanmış halidir.

**Mobile Angular UI:** Bootstrap CSS çatısını ve Angular JavaScript kütüphanesini kullanmaktadır.

## 2.4 ANLIK VERİ İLETİMİ İÇİN KULLANILAN YÖNTEMLER

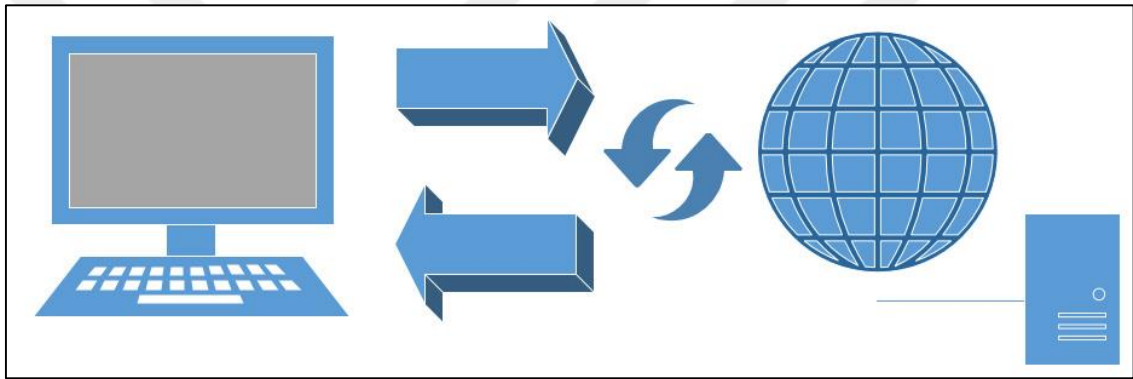
Sunucu ve istemci arasında çift yönlü, sürekli veri iletimi için çeşitli yöntemler kullanılmakta ve geliştirilmektedir.

HTTP, internette bağlandığımız web sayfalarının kodlarını aktarmak için kullanılan bir protokoldür. Web tarayıcısına bir web sayfası adresi girdiğimizde, bu protokol alt seviyedeki protokolleri bu adresin nereden, nasıl isteneceğini ve nasıl aktarılacağını söylemektedir [30]. Kullanılan yöntemler bu protokol üzerinde çalışmaktadır.

## 2.4.1 Web üzerinde veri iletişimi ve kullanılan yöntemler

### 2.4.1.1 Düzenli Http İsteği

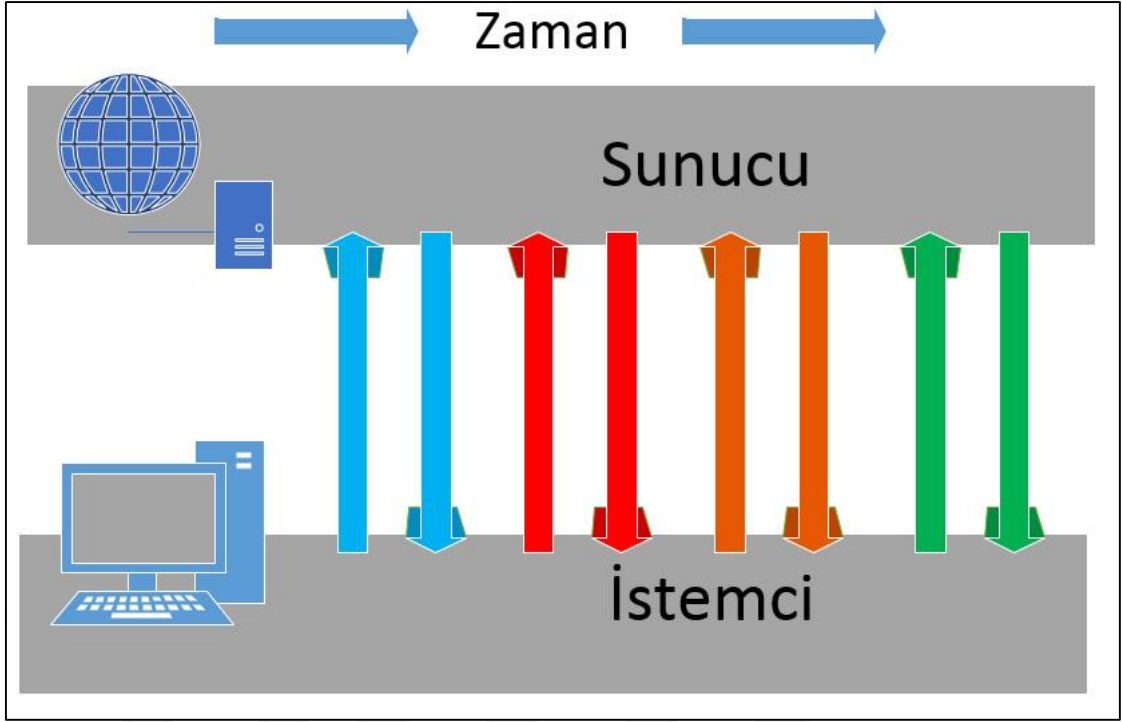
Web istemci ile web sunucular arasında HTTP isteği ile veri alışverişi şu şekilde çalışmaktadır. İstemci, sunucuya bir istek yollar, sunucudan istenen sayfa HTML sayfası ise sunucu sayfayı diskten okuyup başka işlem yapmadan istemciye gönderir. Eğer istenen sayfa PHP, ASP.NET gibi işletilebilir kod içeriyorsa web sunucusu, işletim görevini uygulama sunucusuna gönderir. Uygulama sunucusunda kodlar çalıştırılır ve sonucu HTML veya istenen formatta istemciye geri gönderir. İstemci bu şekilde sunucudan veri almaktadır. Yeni bir bilgi almak için tekrar istek yollamak zorundadır [31]. Şekil 2.20’de Klasik HTTP teknolojisi şekli görülmektedir.



Şekil 2.20. Klasik HTTP teknolojisi.

### 2.4.1.2 Ajax HTTP Polling

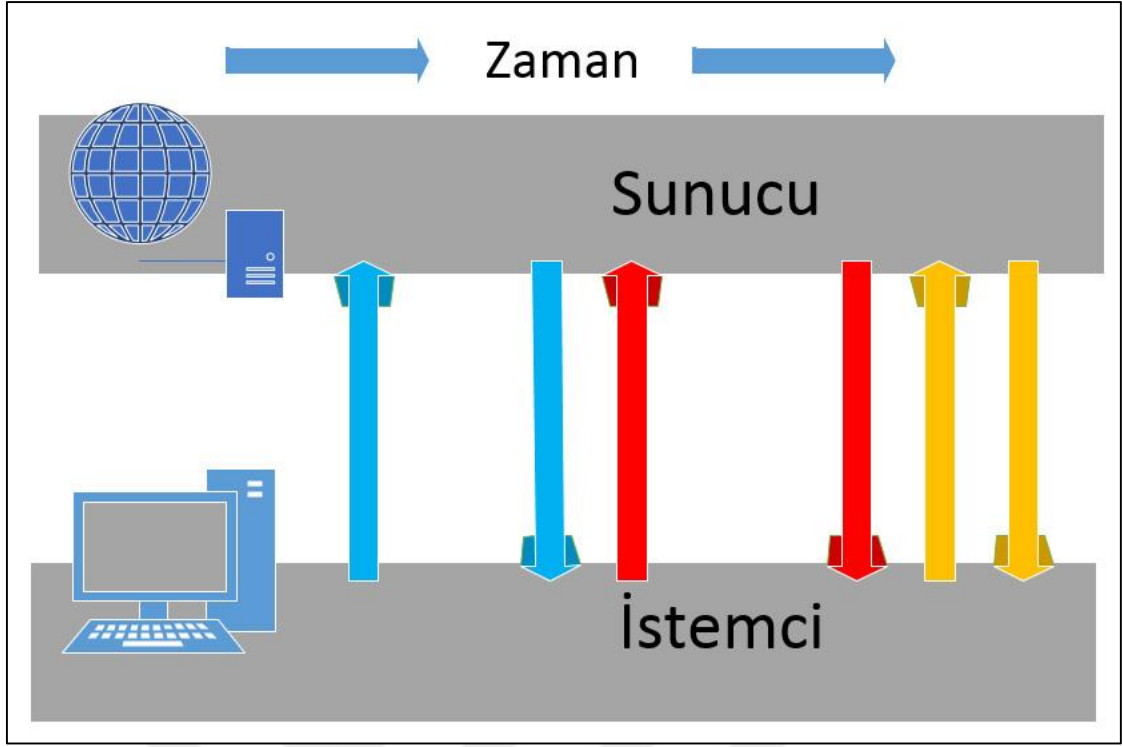
Ajax, web tarayıcı üzerinde, xml httprequest objesini kullanarak açık olan web sayfasını yeniden yüklemeyen web sunucusuna istek göndermek için kullanılan bir yöntemdir. Ajax HTTP Polling, JavaScript kullanarak klasik http isteğini düzenli aralıklarla sunucuya yollayıp cevaplar alan yöntemdir. Örneğin her saniye web sunucuya istek yollanılmakta ve sonrasında web sunucusundan cevap alınmaktadır. Anlık bilgi alışverişi için kullanılan yöntemlerdendir fakat istemciden sunucuya sürekli istekler yollandığı için sunucuyu gereksiz yere meşgul etmektedir [32]. Şekil 2.21’de Ajax HTTP Polling’in çalışma sistemi görülmektedir.



Şekil 2.21.3 Ajax HTTP Polling.

#### 2.4.1.3 Ajax HTTP Long Polling

Bu yöntemle sunucuya başlangıçta bir istek gönderilmekte, sunucuda bir değişiklik olana kadar sonsuz döngü çalıştırılmakta, değişiklik olduğunda istemciye veri gönderilmektedir. Veri geldikten sonra da tekrardan yeni bir değişiklik olana kadar döngü çalıştırılmaktadır. Anlık veri iletimi işlemi Ajax Long Polling ile bu şekilde yapılmaktadır [32]. Şekil 2.22'de Ajax HTTP Long Polling'in çalışma sistemi görülmektedir.



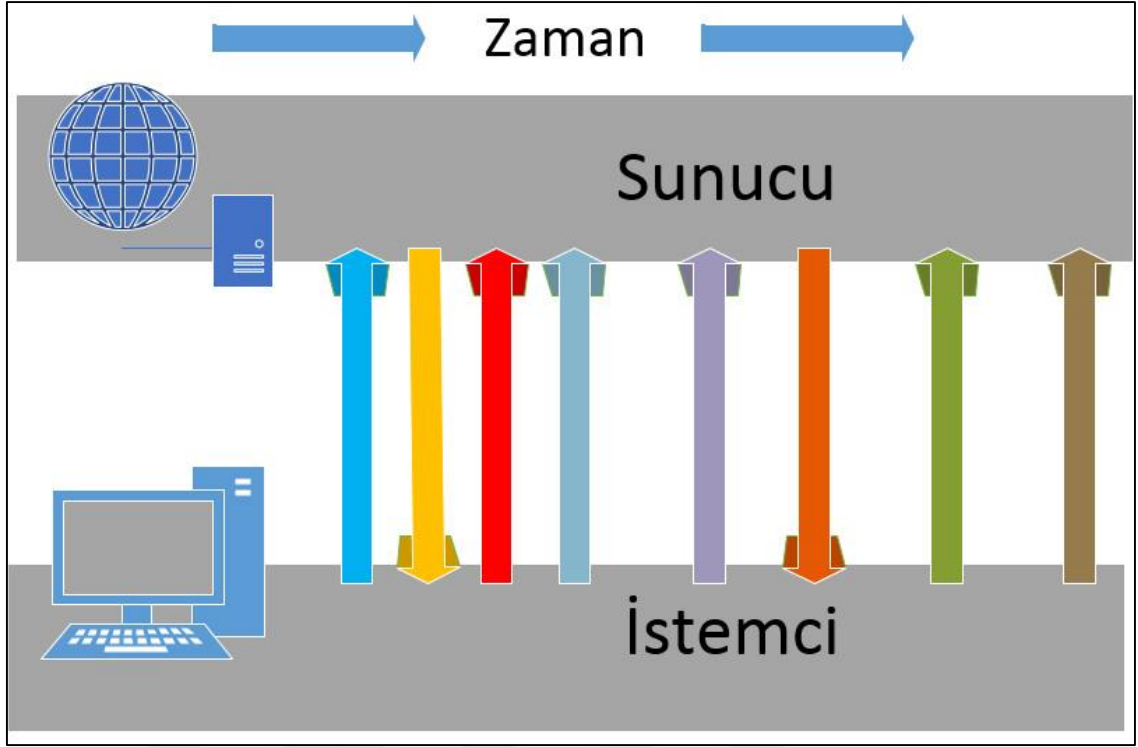
Şekil 2.22. Ajax Long Polling.

#### 2.4.1.4 WebSocket

2011 yılına kadar gerçek zamanlı, çift yönlü veri iletimi için bir protokol bulunmamaktaydı. 2011 yılında geliştirilen, WebSocket ile HTTP portu üzerinden Geçiş Kontrol Protokolü (TCP) bağlantısı ile çift yönlü anlık veri iletimi yapılabilmektedir. WebSocket HTML5 standartlarının bir parçasıdır.

WebSocket öncesinde gerçek zamanlı veri iletimi için HTTP Polling, Long Polling gibi çeşitli yöntemler kullanılmış olup hız olarak WebSocket'e göre daha yavaş olduğu, kaynakları çok tükettiği görülmüştür. Özellikle küçük verilerin iletiminde büyük etki sağlamaktadır [33, 34].

WebSocket başlangıçta sunucuya bir HTTP isteği göndermekte ve web sunucusu isteği alıp bir TCP bağlantı oluşturmaktadır. Bağlantı oluştuktan sonra çift yönlü veri iletimi işlemi yeni bir HTTP isteği göndermeden yapılabilmektedir. Sunucu istediği zaman istemciye veri gönderebilmekte, istemci de istediği zaman sunucuya veri gönderebilmektedir. Şekil 2.23'te WebSocket çalışma mantığı gösterilmektedir.



Şekil 2.23. WebSocket çalışma mantığı.

Kullanılan yöntemlerde veri iletimi için HTTP isteği gönderilmektedir. HTTP isteğinde sunucuya gönderilmek istenen verinin dışında birçok veri gönderilmektedir.

Bu verilerden bazıları aşağıda listelenmektedir.

- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- Expect
- From
- Host
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Proxy-Authorization

- Range
- Referer
- TE
- User-Agent [35].

Anlık veri iletimi için kullanılan bu yöntemlerde HTTP isteğindeki bu alanların her seferinde gönderilmesi gereksiz bir durumdur ve fazla trafik oluşturmaktadır. WebSocket ile sadece ilk istekte HTTP bilgisi gönderilmekte, bağlantı işleminden sonraki işlemlerde veri iletimi için bu veriler gönderilmemektedir [34].

WebSocket son yıllarda önemi artan bir teknoloji olup, bağlantı-iletim problemlerini büyük ölçüde çözmektedir. Fakat WebSocket geliştirilmekte olan bir teknoloji olduğu için güvenlik anlamında bazı sıkıntıları yaratabilmektedir. Şu anki durumunda iyi bir modelleme ve web hizmetlerinin düzgün bir şekilde kullanılmasıyla güvenlik riski en aza indirilebilmektedir. WebSockette ilk HTTP istek ve cevabı işleminden sonra çift yönlü bağlantı sağlanmaktadır. Bu bağlantı bir adres ve port numarası üzerinden yapılmaktadır. Bu adres ve port numarası üzerinden kötü niyetli şahısların verilere ulaşabilmesi mümkün olmaktadır. Ayrıca XSS, DDOS gibi ataklar da sistemi tehdit etmekte ve bu tarz ataklarla sistem yavaşlatılabilmektedir. Bu tarz tehditleri en aza indirmek için ip kontrolü, web servisler ile kimlik onaylama sistemi, veri şifreleme gibi yöntemlerin kullanılarak risk en aza indirilebilmektedir. Ayrıca web tarayıcısı geliştiricilerinin WebSocket güvenliği için de zamanla güncellemeler yapmaları önemli gözükmemektedir [34].

## **2.5 HTML5 GEOLOCATION**

HTML5 ile tarayıcı üzerinden konum alınabilmektedir. Cihazın veya telefonun konum özelliğinin açık olması ve gerekli izinlerin verilmesi halinde GPS veya network verisi alınabilmektedir. Bu özellik üzerinden alabileceğimiz veriler Tablo 1'de gösterilmektedir.

**Çizelge 2.1.** HTML5 Geolocation UPA'sı ile alabileceğimiz veriler.

Özellik	Döndürülen Değer
coords.latitude	Bir ondalık sayı olarak enlem
coords.longitude	Bir ondalık sayı olarak boylam
coords.accuracy	Pozisyonun doğruluğu
coords.altitude	Deniz seviyesinden yükseklik
coords.altitudeAccuracy	Deniz seviyesinden yükseklik doğruluğu
coords.heading	Kuzeyden saat yönünde derece bilgisi
coords.speed	Metre/saniye cinsinden hız
timestamp	Yanıtın tarih ve zamanı

Geolocation UPA'sı aşağıdaki mobil işletim sistemlerinde çalışmaktadır.

- Amazon Fire OS
- Android
- BlackBerry 10
- Firefox OS
- iOS
- Tizen
- Windows Phone 7 and 8
- Windows [36].

## 2.6 BİLGİ GÜVENLİĞİ VE KRİPTOGRAFİ

Önemli verilerin iletiminde bilginin güvenliği, sızdırılmaması önemli etkenlerden bir tanesidir. Bilginin yanlış kişilerin eline ulaşması sorun yaratmaktadır. Kriptografi,



gizlilik, kimlik denetimi, bütünlük gibi bilgi güvenliği kavramlarını sağlamak için çalışan matematiksel yöntemler bütünüdür [37].

Bilgi güvenliğini sağlamak için veriler şifrelenmekte, gerekirse karşı tarafta çözülmektedir. Giriş yapılan metin, gerekiyorsa bir anahtarla birlikte şifrelenir ve şifrelenmiş metni çıktı olarak geri döndürür.

Şifreleme işlemlerinde çeşitli yöntemler kullanılmaktadır.

### **2.6.1 Anahtarsız Şifreleme**

Veriyi şifrelemede anahtar kullanılmayan bir yöntemdir. Verinin her şifrelenmesinde aynı şifrelenmiş metin sonucuna ulaşılmaktadır. Bu algoritmalar genelde bir sistemde yalnız olarak kullanılmazlar. Örnek olarak MD5, SHA-1 algoritmaları anahtarsız şifreleme yöntemini kullanmaktadır [38].

### **2.6.2 Gizli Anahtarlı Şifreleme**

Simetrik şifreleme olarak da adlandırılan bu yöntemde anahtar gizlidir. Şifreleme yapılırken de şifrelenmiş metin çözülürken de aynı anahtar kullanılır. Önemli olan anahtarın karşı tarafa güvenli bir şekilde iletilmesidir. Eğer düzgün şekilde iletilirse açık anahtarlı şifreleme yöntemine göre daha güvenlidir. Örnek olarak DES ve AES algoritmaları gizli anahtarlı şifreleme yöntemini kullanmaktadır [39].

### **2.6.3 Açık Anahtarlı Şifre**

Şifre ve deşifre yöntemleri için farklı anahtarların kullanıldığı yöntemdir. Anahtarların biri ile şifreleme yapılır, diğeri ile çözme işlemi gerçekleştirilir. Diffie-Hellman, RSA, ElGamal, Paillier algoritmaları açık anahtarlı şifrelemeye örnek olarak gösterilebilir [40].

## **2.7 PRESSY DÜĞMESİ**

Tezimizdeki uygulamada kullanılacak nitelikte olan Pressy düğmesi mobil telefonlarda kulaklık girişine takılıp düğme ile çeşitli işlevleri yerine getirme görevini yürütmektedir. Şu an sadece Android işletim sistemli telefonlarda çalışmaktadır. Acil durumlarda kolay ve hızlı bir şekilde ulaşılacak olan bu düğmeye çift tıklama, 3

defa peş peşe tıklama, basılı tutma gibi olaylarda yer durumu bildirimi yapılması mümkün olmaktadır [41].



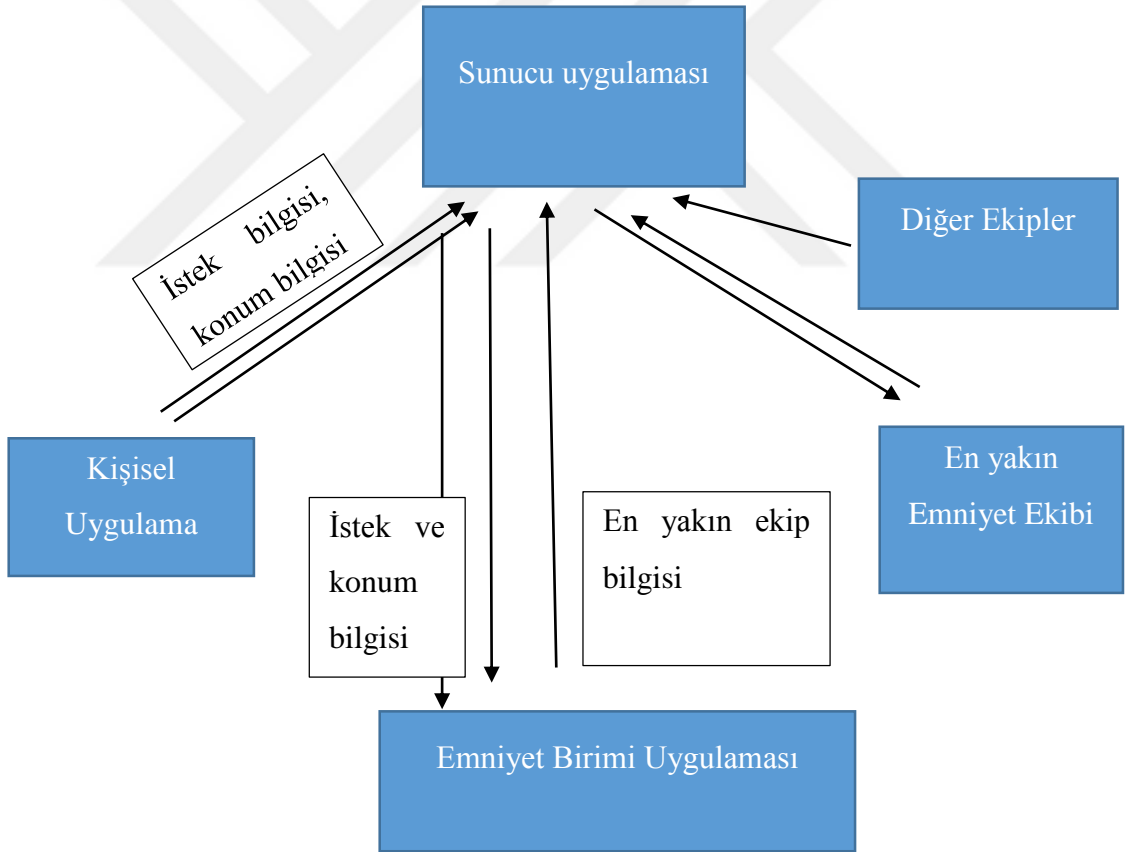
**Şekil 2.24.** Pressy düğmesi.

### 3. MATERYAL VE YÖNTEM

Tezimizde geliştirilen uygulama örneği genel olarak dört bölümden oluşmaktadır. 1. Bölüm veri alışverişinin sağlanacağı sunucu uygulaması, 2. bölüm takiplerin yapılacağı emniyet biriminin uygulaması, 3. bölüm istekte bulunacakların kullanacağı mobil yazılım uygulaması, 4. bölüm emniyet ekiplerinde bulunacak mobil uygulamadır.

#### 3.1 Sunucu Uygulaması

Tüm veri akışının sağlanacağı uygulama bu bölümde tasarlanmıştır. Veri iletişimi için Nodejs-socket.io ikilisi kullanılmış olup, şifreleme yöntemi olarak DES kullanılmaktadır. Sunucu uygulamasının tasarımı Şekil 3.1’de gösterilmektedir.



Şekil 3.1. Sunucu uygulaması şeması.

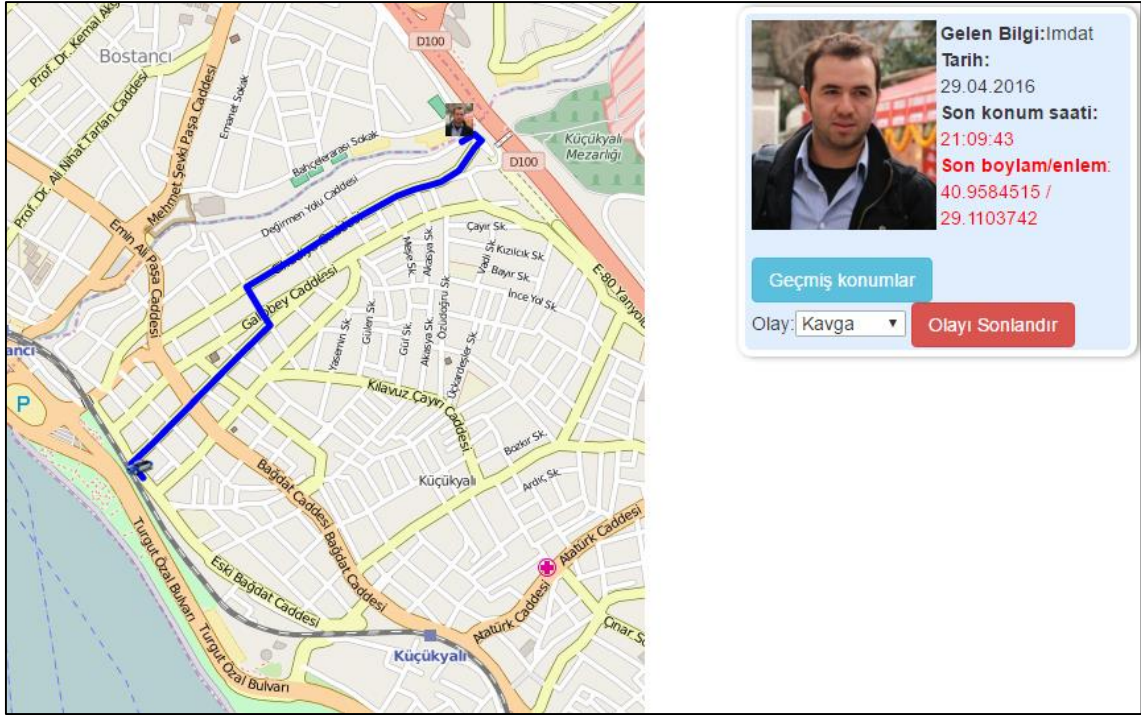
Buradan anlaşılacağı gibi, tüm veri akışının sunucu uygulaması üzerinden gerçekleştiği görülmektedir. Sunucu uygulaması; kişisel uygulamadan, ekiplerden

gelen konum ve istek bilgilerini alıp emniyet birimi uygulamasına göndermektedir. Emniyet birimi uygulamasındaki OpenStreetMap harita üzerinde tüm konumlar ve nesnelere (istekler, ekipler) gözükmektedir.

Sunucu uygulaması için internetten sanal bilgisayar (VPS) kiralanmış olup, kurulumu gerçekleştirilmiştir.

### **3.2 Emniyet Birimi Uygulaması**

Emniyet birimi uygulaması bir web uygulamasından oluşmakta ve web tarayıcısı üzerinde çalışmaktadır. OpenStreetMap harita sistemi, OpenLayers kütüphanesi kullanılarak web sayfasında görüntülenmektedir. Bu uygulama aynı zamanda sunucu ile sürekli bağlantı sağlamaktadır. Çalışmamızı oluşturan tüm mobil uygulamalar sunucu üzerinden bu uygulamaya bağlantı sağlamaktadır. Öncelikle Emniyet ekiplerinden GPS bilgileri bu uygulamaya gelmekte, harita üzerinde otomatik olarak görüntülenmektedir. Aynı zamanda WebSocket teknolojisi ile anlık takipleri de yapılmaktadır. Herhangi bir bireyden yardım isteği geldiğinde harita üzerine otomatik yerleştirme yapılmaktadır. Ekiplerin konumları bir dizi değişkeni içerisinde tutulmakta, bir istek geldiğinde görev verilmemiş ekipler arasından sıralama algoritması ile en yakın ekip saptanmaktadır. Aynı zamanda olay sonlandığında olayın nasıl sonlandırıldığına dair bir bilgi ile veri tabanına kayıt yapılabilmektedir. Bu veriler daha sonra işleme amaçlı kullanılmaktadır. Şekil 3.2’de Emniyet Birimi Uygulamasına istek geldiğinde gelen ekran gözükmektedir.



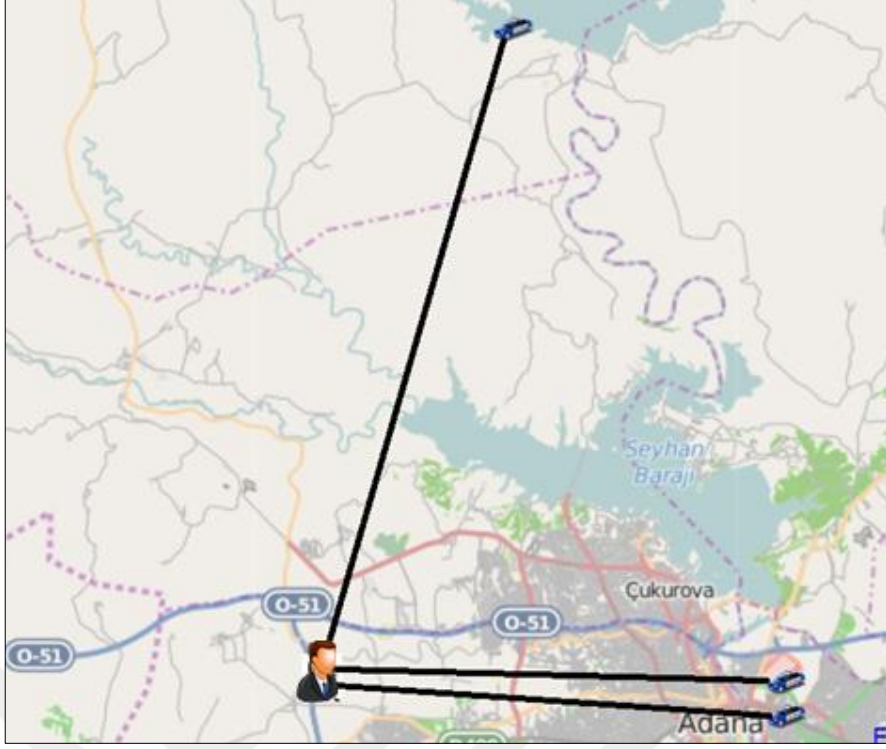
Şekil 3.2. Emniyet birimi uygulaması görseli.

Gelen istek ekranda gözükmekte, en yakın ekibin yönlendirilmesi yapılmaktadır. Anlık olarak da takibi yapılmakta, geçmiş konumlarına ulaşmak istendiğinde Geçmiş konumlar düğmesine tıklanarak, geçmiş konumlar görülebilmektedir. Ayrıca olay bittiğinde olayı sonlandır düğmesiyle olay sonlandırılabilenekte, veri tabanına kaydedilmektedir.

Uygulamada birden fazla ekip olabilmektedir. Yönlendirilecek ekibin en yakındaki ekip olması önem arz etmektedir. Çalışmamızda ekiplerle istek arasındaki mesafeyi bulmak için iki yöntem kullanılmaktadır.

### 3.2.1 Yöntem I

İlk yöntem harita üzerindeki yollar dikkate alınmadan yapılacak kuş bakışı mesafe hesabıdır. Şekil 3.3’de bu yöntem gösterilmektedir.



**Şekil 3.3** Kuş bakışı mesafe hesabı.

İki koordinat arasındaki mesafeyi bu yöntemle bulmak için Denklem 3.1’de gösterilen Haversine formülüne [42] göre JavaScript fonksiyonu yazılmıştır.

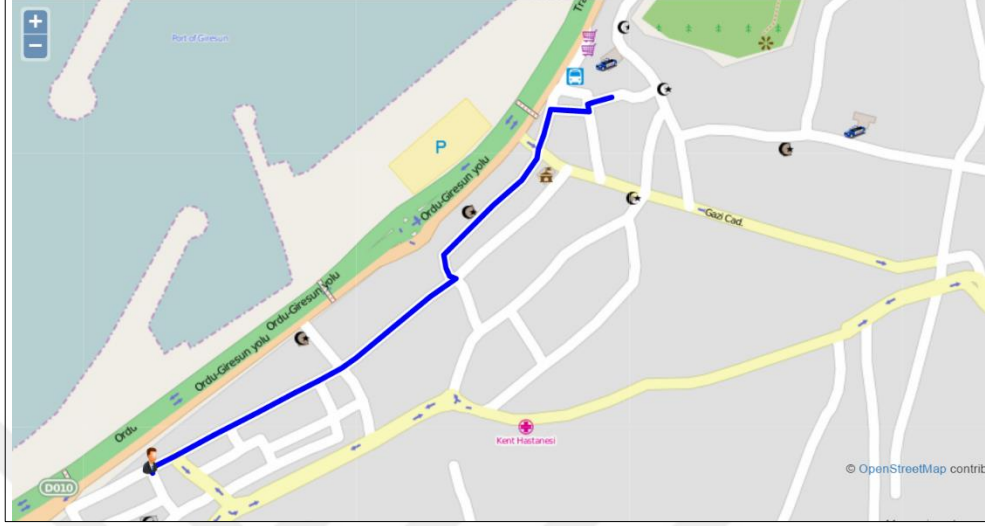
$$d = 2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\Phi_2 - \Phi_1}{2} \right) + \cos(\Phi_1) \cos(\Phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (3.1)$$

Formülde  $\Phi$ ; enlem,  $\lambda$ ; boylam,  $r$ ; kürenin yarıçapı,  $d$ ; iki nokta arasındaki uzaklık bilgilerini göstermektedir. Bu formülde dünyayı küre şeklinde düşünerek iki nokta arasındaki uzaklık hesaplanmaktadır. Mesafeler kısa olduğu için Pisagor teoremine göre de yakınlık hesabı yapılabilir fakat uygulama uzun mesafeli isteklerle de çalışacaksa bu yöntem daha uygun olmaktadır.

### 3.2.2 Yöntem II

İkinci yöntem, harita üzerindeki hedefe gidilebilecek yollar hesaba katılarak yapılan mesafe hesabıdır. Emniyet ekiplerinin giriş yasağı olan yolları da kullanabilme yetkisi olduğundan dolayı iki yöntem de kullanılabilir. Uygulamayı kullanan kişi ayrıca istediği ekibi yönlendirebilecektir. Emniyet birimi uygulamasında bir isteğe birden fazla

ekip yönlendirmesi de yapılabilmekte, anlık olarak bulunduğu konumlar harita üzerinden görüntülenmektedir. Şekil 3.4'te ikinci yöntemin uygulama ekranı gösterilmektedir.



Şekil 3.4. İkinci yöntemin mesafe hesabı.

### 3.3 Kişisel Uygulama

Şekil 3.5'de kişisel uygulama ekranı gösterilmektedir. Bu uygulamada öncelikle sahte isteklerin önlenmesi için SMS ile kayıt gerektirmesi ayrıca vatandaşlık numarası ile de onaylanması uygun görülmektedir. Acil yardım düğmesine tıklanıp yardım istenmesi her zaman mümkün olmayabileceğinden, Pressy düğmesi ile de yardım isteğinde bulunabilmektedir. Şekil 3.5'te kişisel uygulama ekranı görülmektedir.



Şekil 3.5. Kişisel uygulama ekranı.

Kullanıcı acil istekte bulunacağı zaman önceden tanımlanmış bir mesajı veya yeni yazacağı bir mesajı yazıp bir buton ile sunucuya bilgi gönderebilmektedir. Ayrıca kullanıcının bazı acil durumlarda telefonunu açıp bu işlemi gerçekleştirmesi zor olacağından genel kısımlarda açıklanmış olan Pressy Düğmesi de bu amaçla kullanılabilir. Kişisel bilgiler ile birlikte konum bilgisi de gönderilmektedir. İstek gerçekleştirildikten sonra program sürekli koordinatı kontrol etmekte, koordinat değişikliklerinde sunucuya bilgi göndermektedir. Bu işlem WebSocket teknolojisi ile anlık olarak gerçekleştirilmektedir. İstek uygulaması iki farklı yöntemle geliştirilmiştir.

Birincisi doğal (Native) yöntem, yani mobil telefonun işletim sistemine göre desteklediği dil ile geliştirilen yöntemdir. Örneğin Android işletim sistemli telefonlar için Java, IOS işletim sistemli telefonlar da Objective C dilini desteklemektedir. Örnekte aynı amacı gerçekleştiren iki ayrı uygulama yazılmıştır. Android işletim sistemli telefona WebSocket istemci kütüphanesi kullanılarak Native uygulama ve birçok mobil platformda çalışması amaçlandığı için bir de melez uygulama yazılmıştır. Melez uygulama örneğinde melez teknolojiler için en iyi çatlardan olan Apache Cordova kütüphanesi (eski adı phonegap) kullanılmıştır. Veri iletişimi için de WebSocket teknolojisini kullanmada kolaylık sağlayan socket.io kütüphanesi kullanılmıştır.



Bu uygulama ayrıca Native yöntemle de yazılmış olup, WebSocket teknolojisiyle bağlantıyı sağlamak için Github web sitesi içerisinde yazılmış bir eklenti kullanılmıştır [43].

### **3.3 Emniyet Ekibi Mobil Uygulama**

Emniyet ekibindeki mobil uygulama ekibin konumunu emniyet uygulamasına göndermektedir. Sürekli gönderim yapılıp sistemi meşgul etmemesi için de mobil uygulama içerisinde konum değişikliği kontrolü yapılmakta, eğer konum değişikliği gerçekleşmişse sunucuya bilgi gönderilmektedir. Websocket Arduino gibi gömülü sistemlerde de kullanılabilen bir teknoloji olup, bu uygulamanın Arduino gibi gömülü sistemlerde de yapılabileceği düşünülmektedir.

### **3.4. Verilerin İşlenmesi**

Uygulamadaki konum, istek, sonuç bilgileri ve olayın sonlandığındaki bilgiler veri tabanına kaydedilmektedir. Bu veriler istatistik amaçlı işlenmektedir. Hangi ilde veya bölgede ne kadar olay olduğu gözükmektedir.

Bunun için tezimizde MySql veri tabanı ve tablosu oluşturulmuş olup, içerisine kişisel bilgiler, istek enlem ve boylamı kaydedilmektedir. Aynı zamanda konum değişikliklerinin hepsi için yeni kayıt oluşturulmaktadır.

## 4. BULGULAR

Hazırlanan sunucu uygulaması internetten kiralanan sanal bir bilgisayara kurulmuş ve çalıştırılmıştır. Deneme açısından 4 telefona ekip uygulaması kurulmuş, çeşitli konumlara yerleştirilmiş, yine mobil telefona yüklenen 2 kişisel uygulama içerisinde istek gönderilmiş, en yakın ekibin bulunduğu ve istek yollandığı, takibinin yapıldığı gözlemlenmiştir. İsteklerle ilgili işlem sonuçlandırıldığında bilgiler Emniyet uygulamasında sonlandırılmış, veri tabanına kaydetme işlemi gerçekleştirilmiştir. Veri tabanına kayıt işlemi için aşağıdaki veri tabanı tablosu oluşturulmuştur.

**Çizelge 4.1.** MySql Veri Tabanı Geçmiş Kayıtlar Tablosu Alan Bilgileri.

Alan Adı	Veri Türü	Girilen Veri İçeriği
id	Otomatik artan integer sayı tipi	Otomatik artan sayı
lat	double	Enlem
lon	double	Boylam
tur	varchar(250)	Olay türü
not	text	Olayla ilgili varsa ek bilgi
telefonNo	varchar(11)	Telefon Numarası
tcKimlik	varchar(11)	T.C. Kimlik Numarası
zaman	datetime	Zaman (Tarih-Saat-Cinsinden)

Uygulama içerisinde bu tabloya girilen veriler istatistik amaçlı gösterilebilmekte, istenildiği zaman çeşitli amaçlar için kullanılabilir. Ayrıca istek gönderen bireylerin ve ekiplerin geçmiş bütün konumları da veri tabanında tutulmaktadır. Şekil 4.1’de geçmiş olay arama-listeleme ekranı görülmektedir.



### Geçmiş Olay Arama Ekranı

1. Tarih 01.09.2015  
2. Tarih 07.04.2016

### Sonuçlar

	<b>Olay:</b> Kaçırılma <b>Tarih:</b> 15.02.2016 <b>Not:</b> Ekipler zamanında ulaştı
	<b>Olay:</b> Gasp <b>Tarih:</b> 10.02.2016 <b>Not:</b> Ekipler zamanında ulaştı
	<b>Olay:</b> Gasp <b>Tarih:</b> 21.11.2015 <b>Not:</b> Ekipler zamanında ulaştı

Toplam 3 kayıt listelendi

Şekil 4.1. Geçmiş olay arama ekranı.

## 5. SONUÇLAR VE ÖNERİLER

Bu çalışmada yeni teknolojilerin güvenlik sisteminde kullanımının modellenmesi ve ayrıca sistem gereksinimleri ile uygulama hızının minimum seviyede tutulması amaçlanmıştır. Bu doğrultuda anlık iletim sistemi için WebSocket kullanılmıştır. WebSocket 2011 yılında geliştirilmeye başlanıp, halen geliştirilen, web tarayıcısı üzerinde HTTP portunu kullanarak TCP bağlantısı gerçekleştirerek çalışan çift yönlü veri iletimini sağlayan bir protokoldür. Çalışmamızdaki tüm uygulamalar bu protokolü kullanarak anlık veri iletimini gerçekleştirmektedir. Ayrıca veri iletimindeki güvenlik araştırmaları yapılmış, çalışmamıza dahil edilmiştir.

Bireylerin yardım istekleri için her platformda çalışabilecek nitelikte olan melez programlama yöntemiyle uygulama geliştirilmiştir. Bu uygulamada anlık isteği gizli bir şekilde gönderilebilmesi açısından Pressy düğmesinden faydalanılmıştır. Birey isteklerinin harita üzerinde gösterilmesi için çevrimiçi harita sistemlerinden, ücretsiz ve açık kaynak kullanımına açık olan OpenStreetMap, kodlama sistemi olarak da OpenLayers JavaScript kütüphanesi kullanılmıştır. Ayrıca verilerin tümü MySql veri tabanına kaydedilmekte, istenilen zamanda ulaşılabilmektedir.

Yeni teknolojilerle kişisel güvenliği, sistem hızını ve minimum gereksinimi üst düzeylere çıkaran bu uygulamanın birçok sahada kullanılabileceğini düşünmekteyiz. 112 Acil yardım, 155 Polis imdat, 110 itfaiye gibi ülke çaplı uygulamalarda veya özel şirketlerde buna benzer takip sistemlerinde kullanılabileceği düşüncesindeyiz.

Çalışmanın devamında, çalışmada acil yardım isteğinin güvenilirliğini anlayabilme ve durumun gidişatını öğrenme amaçlı, istek sonrasında telefonda anlık ses-görüntü iletimi yapılabilir. Ayrıca Mernis gibi sistemlerle kimlik kontrolü yapılabilir, güvenilirlik artırılabilir.

Sunucu maliyetini azaltma amaçlı, yeni bir teknoloji olan ve web tarayıcısı üzerinden uçtan uca bağlantı gerçekleştirilmesini sağlayan WebRTC teknolojisi de kullanılabilir.

Ekip yönlendirme sistemi daha akıllı bir sistem haline getirilebilir. Her ne kadar en yakın ekip seçilip yönlendirilse de bu görevli olmayan ekipler arasından

gerçekleştirilmektedir. Bir isteğe yönlendirilen ekip, daha sonraki isteklerde, görevli olduğu için göz ardı edilmektedir. Bu sebepten ötürü, başka istek geldiğinde, uzaktaki ekip, görevli olan ekibe yakın bir noktaya görevlendirilebilmektedir. Yeni bir istek geldiğinde görevli ekipler de devreye katılıp görev değişimi yapılabilir.



## 5. KAYNAKLAR

- [1] Anonim, <http://www.aljazeera.com.tr/al-jazeera-ozel/cebinizdeki-panik-butonu> (Eriřim Tarihi: 13 Nisan **2015**).
- [2] Chand D., Nayak S., Bhat S.K., Parikh S., Singh Y., Kamath A.A., A mobile application for Women's Safety: WoSApp, *TENCON 2015 - 2015 IEEE Region 10 Conference*, Macao-India, (**2015**) 1-5.
- [3] Anonim, <https://play.google.com/store/apps/details?id=com.startv.gumrah&hl=tr> (Eriřim Tarihi: 25 Ocak **2016**).
- [4] Anonim, <https://play.google.com/store/apps/details?id=com.hd.panikbutonu&hl=tr> (Eriřim Tarihi: 27 Mart **2015**).
- [5] Anonim, <http://www.openstreetmap.org> (Eriřim Tarihi: 27 Mayıs **2015**).
- [6] Anonim, <http://openlayers.org> (Eriřim Tarihi: 19 Nisan **2015**).
- [7] Anonim, <http://project-osrm.org> (Eriřim Tarihi: 13 Haziran **2015**).
- [8] Anonim, <https://developers.google.com/maps/documentation/utilities/polylinealgorithm> (Eriřim Tarihi: 20 Haziran **2015**).
- [9] Anonim, [https://tr.wikipedia.org/wiki/Baz\\_istasyonu](https://tr.wikipedia.org/wiki/Baz_istasyonu) (Eriřim Tarihi: 16 Ocak **2016**)
- [10] Nakıp İ., Mobil takip sistemi ve optimizasyonu, *Yüksek Lisans Tezi*, ISTANBUL: BAHÇEŞEHİR ÜNİVERSİTESİ, (**2010**).
- [11] Anonim, <https://tr.wikipedia.org/wiki/GPS> (Eriřim Tarihi: 20 Şubat **2016**)
- [12] Anonim, <https://tr.wikipedia.org/wiki/OpenStreetMap> (Eriřim Tarihi: 27 Mayıs **2015**)
- [13] Ünen H. C., Güngör O. , Yılmaz O. M., Özgür harita: OpenStreetMap , TMMOB Coğrafi Bilgi Sistemleri Kongresi, Ankara-Türkiye, (**2013**).
- [14] Anonim, <http://openlayers.org/en/v3.10.1/doc/quickstart.html> (Eriřim Tarihi: 20 Şubat **2016**).
- [15] Anonim, <http://download.geofabrik.de> (Eriřim Tarihi: 20 Ağustos **2015**).

- [16] Anonim, <https://mapzen.com/data/metro-extracts> (Erişim Tarihi: 20 Ağustos **2015**).
- [17] Anonim, <http://www.github.com> (Erişim Tarihi: 10 Ocak **2015**).
- [18] Anonim, <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-osrm-server-on-ubuntu-14-04> (Erişim Tarihi: 5 Mart **2016**).
- [19] Anonim, <https://developers.google.com/maps> (Erişim Tarihi: 3 Ocak **2016**).
- [20] Anonim, <https://tech.yandex.com.tr/maps/doc/jsapi/2.1/quick-start/tasks/quick-start-docpage/> (Erişim Tarihi: 27 Şubat **2016**).
- [21] Anonim, <https://tech.yandex.com.tr/maps/jsbox/2.1/polyline> (Erişim Tarihi: 10 Eylül **2015**).
- [22] Tian L., Du H., Tang L., Xu Y., The discussion of cross-platform mobile application based on Phonegap, *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on.*, Beijing-China (**2013**) 652-655.
- [23] Anonim, <https://cordova.apache.org/> (Erişim Tarihi: 10 Şubat **2016**).
- [24] Anonim, <https://cordova.apache.org/docs/en/latest/guide/cli/index.html> (Erişim Tarihi: 10 Şubat **2016**).
- [25] Anonim, <https://cordova.apache.org/docs/en/latest/reference/cordova-cli/index.html#cordova-platform-command> (Erişim Tarihi: 20 Nisan **2016**).
- [26] Anonim, <https://cordova.apache.org/docs/en/latest/guide/support/index.html> (Erişim Tarihi: 24 Şubat **2016**).
- [27] Anonim, <http://ionicframework.com> (Erişim Tarihi: 20 Nisan **2016**).
- [28] Anonim, <http://ionicframework.com/getting-started/> (Erişim Tarihi: 25 Nisan **2016**).
- [29] Anonim, <http://ngcordova.com/docs/plugins/> (Erişim Tarihi: 25 Nisan **2016**).
- [30] Harman G., Halaç A., *MCSE (Microsoft Certified Systems Engineer) Sertifika Sınavlarına Temel Hazırlık Kılavuzu*, Editör: Bayraktar M., KOYUNCU S., Shark Bilişim Yayınları, (**2005**) 85.
- [31] Baransel C., Mumcuoğlu A., *Web Tabanlı, Üç Katmanlı Yazılım Mimarileri*, 1. dü., SAS Bilişim Yayınları, (**2012**) 30.

- [32] Zhang L., Shen X., Research and development of real-time monitoring system based on WebSocket technology, *Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on*, Shengyang-China, (2013) 1955-1958.
- [33] Nickerson B. G., Pimentel V., Communicating and Displaying Real-Time Data with WebSocket, *IEEE Internet Computing*, (2012) 45-53.
- [34] Erkkilä J. P., Websocket security analysis, *Aalto University School of Science* , (2012) 2-3.
- [35] Anonim, [https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_header\\_fields](https://en.wikipedia.org/wiki/List_of_HTTP_header_fields). (Erişim Tarihi: 10 Ocak 2016).
- [36] Anonim, <https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-geolocation/index.html>. (Erişim Tarihi: 12 Ocak 2016).
- [37] Anonim, <https://tr.wikipedia.org/wiki/Kriptografi> (Erişim Tarihi: 12 Mart 2016).
- [38] Anonim, <http://bidb.itu.edu.tr/seyirdefteri/blog/2013/09/07/%C5%9Fifreleme-y%C3%B6ntemleri> (Erişim Tarihi: 18 Mart 2016).
- [39] Anonim, [https://tr.wikipedia.org/wiki/Gizli\\_anahtar%C4%B1\\_%C5%9Fifreleme](https://tr.wikipedia.org/wiki/Gizli_anahtar%C4%B1_%C5%9Fifreleme) (Erişim Tarihi: 18 Mart 2016).
- [40] Anonim, [https://tr.wikipedia.org/wiki/A%C3%A7%C4%B1k\\_anahtar%C4%B1\\_%C5%9Fifreleme](https://tr.wikipedia.org/wiki/A%C3%A7%C4%B1k_anahtar%C4%B1_%C5%9Fifreleme) (Erişim Tarihi: 18 Mart 2016).
- [41] Anonim, <http://get.pressybutton.com/> (Erişim Tarihi: 1 Mayıs 2016).
- [42] Anonim, [https://en.wikipedia.org/wiki/Haversine\\_formula](https://en.wikipedia.org/wiki/Haversine_formula) (Erişim Tarihi: 13 Eylül 2015).
- [43] Anonim, <https://github.com/nkzawa/socket.io-client.java> (Erişim Tarihi: 20 Eylül 2015).



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : Ceylan, Arif

Uyruğu : T.C.

Doğum tarihi ve yeri : 24.05.1985 GİRESUN

Telefon : 05393745562

Faks :

E-posta : arifceylan@gmail.com

### Eğitim

<i>Derece</i>	<i>Eğitim Birimi</i>	<i>Mezuniyet tarihi</i>
Lisans	Ondokuz Mayıs Üniversitesi / Bilgisayar ve Öğretim. Tkn. Eğ.	2006
Lise	Giresun Atatürk Anadolu Lisesi	2002

### İş Deneyimi

<i>Yıl</i>	<i>Yer</i>	<i>Görev</i>
2006-2010	Gümüşhane Kanuni Ticaret M.L.	Bilişim Tkn Öğr.
2010-2016	Kadıköy Bostancı Halk Eğitimi Mrk.	Bilişim Tkn Öğr.

### Yayımlar

1. Arif Ceylan, Ali Çalhan Açık Kaynak Harita Ve Mobil Yazılım İle İnsan Güvenliği  
Ulusal Mühendislik Araştırmaları Sempozyumu (UMAS'2015) Sf.48 2015