

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



**ONLİNE SİPARİŞ TAKİP OTOMASYONLARI TASARIMI**

**Ferhat KARATAŞ**  
**YÜKSEK LİSANS TEZİ**  
**Danışman: Dr. Öğr. Üyesi Sevcan Aytaç Korkmaz**  
**AĞUSTOS-2019**

T.C.  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

ONLİNE SİPARİŞ TAKİP OTOMASYONLARI TASARIMI

YÜKSEK LİSANS TEZİ

Ferhat KARATAŞ

171111101

Tezin Enstitüye Verildiği Tarih : 12 Haziran 2019

Tezin Savunulduğu Tarih : 19 Temmuz 2019

Tez Danışmanı: Dr. Öğr. Üyesi Sevcan Aytaç Korkmaz (F.Ü.)

Diğer Jüri Üyeleri: Doç. Dr. Semra Türkoğlu (F.Ü.)

Doç. Dr. Muhammed Fatih Talu (İ.Ü.)

HAZİRAN-2019

## İÇİNDEKİLER

<b>İÇİNDEKİLER</b> .....	<b>II</b>
<b>ÖZET</b> .....	<b>IV</b>
<b>ABSTRACT</b> .....	<b>V</b>
<b>ŞEKİLLER LİSTESİ</b> .....	<b>VI</b>
<b>SEMBOLLER LİSTESİ</b> .....	<b>VIII</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
1.1. İlgili Çalışmalar .....	4
1.2. Genel Bilgiler.....	19
1.3. Bulut Bilişimin Faydaları.....	21
<b>2. MATERYAL ve METOD</b> .....	<b>23</b>
2.1. ASP.NET .....	23
2.2. Microsoft SQL Server .....	24
2.3. Veritabanı Tasarımı .....	25
2.4. İş akışı .....	26
<b>3. ONLINE SİPARİŞ TAKİP OTOMASYONU</b> .....	<b>28</b>
3.1. Web uygulaması.....	28
3.1.1. Bugün.....	28
3.1.2. Müşteriler.....	29
3.1.3. Alınacak .....	30
3.1.4. İşlemde.....	31
3.1.5. Teslimat.....	32
3.1.6. Veresiyeler .....	33
3.1.7. Sipariş Sıralama .....	34
3.1.8. Hazırlar .....	35
3.1.9. İptaller .....	36
3.1.10. Bekleyen .....	37
3.1.11. Not.....	38
3.1.12. SMS.....	39
3.1.13. Mesajlar.....	40

3.2.	Android Uygulaması .....	40
3.3.	POS uygulaması .....	43
<b>4.</b>	<b>ENTEGRASYONLAR.....</b>	<b>44</b>
4.1	Bluetooth ile çalışan mobil mini yazıcı .....	44
4.2.	Bluetooth ile çalışan mobil barkod yazıcı .....	44
4.3.	SMS API entegrasyonu .....	45
<b>5.</b>	<b>BULUT HİZMET TÜRLERİ .....</b>	<b>47</b>
<b>6.</b>	<b>SONUÇLAR ve TARTIŞMA .....</b>	<b>49</b>
6.1.	Bu Çalışmadaki Yenilik .....	49
<b>KAYNAKLAR.....</b>		<b>51</b>
<b>ÖZGEÇMİŞ .....</b>		<b>.52</b>

## ÖZET

Teknolojinin girdiđi hemen her yerde zamandan ve maliyetten tasarruf söz konusudur. Saha sipariř takip otomasyonları da bunun örneklerinden sadece birisidir. Firma olarak, örneđin sahadaki 10 ayrı mobil temsilcinizin nerede olduđunu, hangi sipariřlerle ilgilenmesi gerektiđini, koordinasyonunu, yük dađıtımını yapabilmeniz oldukça zor bir süreçtir. Bu tür dađıtık yapılarda kullanılması gereken en iyi çözümlerden birisi de merkezileřtirilmiř yazılım projeleri kullanmaktır. İřte bulut biliřimin gücü de bu tür uygulamalarda kendini göstermektedir. Uygulamayı kullanan binlerce istemci olmasına rađmen, merkezde birkaç server tüm yükü üstlenir ve firma sahibine gerekli rapor, denetim, yönetim imkanı sunar. Tabi en önemli avantaj da maliyet konusudur. Örneđin, yazılım üzerinde yapılacak otomatik (Google Direction API) güzergah belirleme iřlemi, sahadaki çalıřanların gereksiz dolařmasının önüne geçer.

**Anahtar Kelimeler :** Sipariř takip otomasyonu, bulut biliřim, android uygulamaları

## ABSTRACT

### ONLINE ORDER TRACKING AUTOMATIONS DESIGN

Technology saves both time and money. Order tracking automation is just one example of this. For example, you can not handle issues if you don't have any software to track drivers. One of the best solutions for this type of distributed architecture is to use centralized software projects. The power of cloud computing also manifests itself in such applications. Although there are thousands of clients using the application, a few servers in the center can manage all the things and the company going to has the necessary reporting, audit and management facilities. Of course, the most important advantage is the cost. For example, setting a route on the software will prevent the unnecessary circulation of employees on the field.

**Keywords:** Order tracking automation, cloud computing, android applications

## ŞEKİLLER LİSTESİ

Şekil 1.1 MDD uygulamasında izlenen yeniden yapılanma operasyonu. ....	6
Şekil 1.2 SAAS üzerinde konumlandırılan CRM projesi akış şeması .....	8
Şekil 1.3 Restoran sipariş sistemi mimarisi .....	10
Şekil 1.4 A-GPS sisteminin mimarisi .....	13
Şekil 1.5 Google Geocoding API ile alınan enlem/boylam bilgisinin haritada görünüşü .....	14
Şekil 1.6 Google Direction API iki nokta arasındaki en optimal rotanın belirlenmesi.....	15
Şekil 1.7 Google API'lerinin istatistiklerini izlediğimiz bölüm .....	16
Şekil 1.8 Dünyada kullanılan mobil işletim sistemleri.....	17
Şekil 1.9 REST servislerinin performans analizi .....	18
Şekil 1.10 SOAP servislerinin performans analizi .....	19
Şekil 1.11. Dünyada 1 dakikada neler oluyor .....	20
Şekil 1.12. Bulut bilişimin avantajları .....	21
Şekil 2.1. Veritabanı tasarımı.....	25
Şekil 2.2. Sipariş takip otomasyon mimarisi.....	27
Şekil 3.1 - Bugün.....	29
Şekil 3.2 - Müşteriler.....	30
Şekil 3.3 – Alınacak .....	31
Şekil 3.4 – İşlemdede .....	32
Şekil 3.5 – Teslimat.....	33
Şekil 3.6 - Veresiye .....	34
Şekil 3.7 - Sipariş Sıralama .....	35
Şekil 3.8 - Hazır .....	36
Şekil 3.9 - İptal .....	37
Şekil 3.10 - Bekleyen .....	38
Şekil 3.11 - Not .....	38
Şekil 3.12 – SMS.....	39
Şekil 3.13 - Mesaj.....	40
Şekil 3.14. Android Uygulaması .....	41
Şekil 3.15. Sipariş Detayları.....	42

<b>Şekil 3.16.</b> Uygulama menüsü.....	42
<b>Şekil 3.17.</b> POS Uygulaması .....	43
<b>Şekil 4.1.</b> Bluetooth yazıcı .....	44
<b>Şekil 4.2</b> Barkod yazıcı .....	45
<b>Şekil 4.3</b> SMS API entegrasyonu .....	45
<b>Şekil 5.1</b> Bulut hizmet türleri .....	47





## KISALTMALAR LİSTESİ

<b>SOA</b>	: Service Oriented Architecture
<b>Exe</b>	: Executable
<b>USB</b>	: Universal Serial Bus
<b>PC</b>	: Personal Computer
<b>URL</b>	: Uniform Resource Locator
<b>CRM</b>	: Customer Relationship Management
<b>AJAX</b>	: Asynchronous JavaScript And XML
<b>SMS</b>	: Short Message Service
<b>API</b>	: Application Programming Interface
<b>BTK</b>	: Bilgi Teknolojileri ve İletişim Kurumu
<b>SQL</b>	: Structured Query Language
<b>IDE</b>	: Integrated Development Environment
<b>POS</b>	: Point Of Sale
<b>SOAP</b>	: Simple Object Access Protocol
<b>REST</b>	: Representational State Transfer
<b>GPS</b>	: Global Positioning System
<b>JSON</b>	: JavaScript Object Notation
<b>GPRS</b>	: General Packet Radio Service
<b>UI</b>	: User Interface
<b>UX</b>	: User Experience
<b>M2M</b>	: Machine To Machine

## 1. GİRİŞ

Teknoloji ve özellikle yazılımın hayatımızı kolaylaştırdığını söylemek artık o kadar zor değil. Hayatımızın neredeyse tüm safhasında artık yazılımlar var. Bireyin kendi özel ihtiyaçlarını karşılaması her ne kadar teknolojiden bağımsız mümkün ise de toplumların birlikte yaşabilmelerini sağlayacak yegane unsur artık yazılım altyapılarından geçmektedir. E-reçete ile Türkiye'nin her yerinden ilaç alabilmek, istediğiniz hastaneden provizyon ile hizmete ulaşmak mümkündür. E-devlet uygulamaları ile artık muhtarlık, nüfus idareleri tarihe karışmıştır. Çok daha hızlı ve düşük maliyetlerle bu hizmetlere ulaşılabilir.

Franchising yapısıyla büyük firmalar (McDonald's, Burger King vs.) Türkiye'nin her noktasındaki girişimcilere markasını kullandırmakta, logo, reklam ve hatta işletme kurulumuna kadar anahtar teslim hizmet vermektedirler. Dolayısıyla girişimci, bu tür bir işletme açtığına, hangi yazılımı kullanacağını, stoklarını nasıl takip edeceğini, faturasını nasıl keseceğini bilir. İşletme yönetimindeki bu ferahlık, girişimcinin sadece işine odaklanmasını sağlar. Tabi bu da beraberinde hem karlılığı hem de başarıyı getirmektedir.

Fakat, franchising yapısında olmayan işletmelerin (kuru temizleme, halı yıkama firmaları vs) yazılım altyapılarını sağlaması mümkün değildir. Bu tezde, bu tür işletmelerin alt yapılarını sağlayabilmesi için bir yazılım geliştirilmiştir.

Halı yıkama firmaları, Türkiye'de kapıda ürün teslimi yapan en önemli sektörler arasında yer alır. Şimdiye kadarki çözümleri defter ve kalemin ötesine geçmemiş, "en iyi altyapıyı kullanıyorum" diyenlerin ise vardığı son nokta Excel'de kayıt tutmak olmuştur.

El yordamıyla kendine çözüm üretmeye çalışan bu sektörün mevcut iş akışını ve oluşan problemleri inceleyelim:

1. Halısını yıkatmak isteyen müşteri, halı yıkama firmasını aradığında;

a. Müşteriyi bir deftere kaydeder. Müşterinin sonraki zamanlarda aramalarında bu kayda erişme zorluğu olduğundan her aramada, ad soyad, telefon ve adres bilgisi almak zorundadır. İnternet erişimi olmadığı için konum bilgisi (enlem/boylam) alma imkanı olmamaktadır. Konum bilgisi, cep telefonu uygulamalarında nokta atışı adrese götürmeye yarar.

b. Müşterinin geçmiş yıllardaki sipariş hareketlerini borç/alacak durumlarını takip edemez. Bu “şirket hafızası”nın olmadığı anlamına gelir. Daha önceden borç takan bir müşteriye farkında olmadan tekrar hizmet götürür.

c. Telefon ve ad soyad konusunda mevcut veritabanını kontrol edemediği için aynı kişileri bir kaç kez kaydeder. Çok fazla çift kayıt oluşur.

d. Müşterilerin telefonlarının yer aldığı bir veritabanına sahip olunmadığı için, önemli günlerde ve kampanyalarda müşteriye SMS ile ulaşmak mümkün olmaz.

2. Firma yetkilisi, gelen siparişleri sisteme kaydeder;

a. Şöför, elindeki 20 adresin fabrikaya olan uzaklığını bilemez. Tüm siparişleri masaya yatırır, mahalleye göre sıralayarak kendine bir rota oluşturmaya çalışır. Amaç, zaman ve yakıttan tasarruftur.

b. Halıları toplamaya çıktığında gün içinde ofise gelen yeni siparişleri bilemez. Firma sahibi, günün ortasında, şöför arar ve trafikteyken yeni adresleri yazdırmaya çalışır. Bu, hem trafikte hayati tehlike, hem de gürültüde tam anlaşılmayan sokak ve daire numaralarına sebep olur.

3. Şöför, müşteriden halılarını alır;

a. Halıları müşteriden teslim aldığıında, karşılığında bir makbuz yazar ve teslim eder. Bu makbuz 2 nüshadır. 1 nüshası ofise döner, 1 nüshası müşteriye teslim edilir. Çoğu kez okunaklı olmaz.

4. Yikanan halılar teslimata gider;

a. Teslimat esnasında kağıt kalem kullanarak yazılan makbuzlarda, çizme, yeni değer girme, müşteriden 5-10 TL fazla alma çok sık rastlanan bir durumdur. Alt nüshasında değişiklik yapmadan, sadece üst nüshada çizgi çekerek çok rahat 5-10 TL fazla tahsil edilebilmektedir.

5. Halıların tamamı teslim edilip tahsilat yapıldıktan sonra gün sonu raporu almak başlı başına bir problemdir. Şöför, tüm tahsilatlarını belgeleyebilmek için tüm makbuzları beraberinde getirip hesaplamak zorundadır.

Şimdi de bu proje ile birlikte, yukarıda belirttiğimiz safhalarda oluşan problemlere getirdiğimiz çözümleri inceleyelim:

1. Halısını yıkatmak isteyen müşteri, halı yıkama firmasını aradığında;

a. Telefonlara bađladığımız CallerID cihazı sayesinde arayan kişinin tüm bilgileri bilgisayar ekranına gelir. Daha önceden kayıtlı bir müşteri ise, tekrardan ad soyad, adres, telefon bilgisi almasına gerek kalmaz. Bu, sipariş alma süresini en az 10 kat hızlandırır.

Uygulama web tabanlı ve Google API'leriyle entegre çalıştığı için, adres bilgisi anında enlem/boylam değerlerine dönüştürülerek konum bilgisi alınır. Böylece, işe yeni giren bir şöför bile müşterinin adresine nokta atışı ulaşır.

b. Bilgisayar ekranına, müşterinin geçmişte yaptığı tüm sipariş hareketleriyle birlikte, kendisi hakkında alınan müspet/menfi tüm bilgiler yer alır. Dolayısıyla firma, daha önceden kazık yediğı bir müşteri ile tekrar çalışmak zorunda kalmaz.

c. Arayan kişinin telefon ya da ad soyad bilgisi girildiğinde, program bu kayıtları veritabanında sorgulayarak çift kayıtların önüne geçer. Böylece bir telefon numarasının bir kaç kez kaydedilmesi imkansız hale gelir.

d. Firma, bir bakışta binlerce müşterisinin bilgisine ulaşabilir. İstediğı anda tek tuşta müşterilerine SMS atabilir, kampanyalardan haberdar edebilir.

2. Firma yetkilisi, gelen siparişleri sisteme kaydeder;

a. Bu siparişler Google Direction API'si yardımı ile tek tuşta otomatik olarak fabrikaya olan mesafesine göre sıraya dizilir. Böylece işe başlayan yeni şöför hangi adrese önce gideceğini tasarlamak zorunda kalmaz. Bu otomatik rota sistemi firmaya hem zaman hem de yakıt tasarrufu sağlar.

b. Gün ortasında gelen siparişleri firma çalışanı sisteme girdiğinde bu sipariş otomatik olarak şöförün cep telefonunda listelenir. Böylece yeni oluşan siparişler için şöför aranmaz. Bu hem yazım hatalarının hem de trafikte not alma tehlikesinin önüne geçer.

3. Şöför, müşteriden halılarını alır;

a. Halıları aldığı anda, bu bilgileri cep telefonundaki uygulamaya girer. Uygulama, otomatik olarak şöförün belinde taşıdığı kablosuz yazıcıdan fiş basar. Ek olarak müşteriye “halılarının alındığına” dair otomatik bir SMS atar. Basılan fişin detaylarını merkez anında ekranda gördüğü için 2. nüshayı ofise getirmeye gerek kalmaz.

4. Yıkanan halılar teslimata gider;

a. Teslimat esnasında müşteriye otomatik bir SMS göndererek sipariş bilgilerinin doğrulanmasını sağlar. Bu metod, kağıt kalem göre yazma süresini en aza indirdiğı

gibi, otomatik SMS bilgilendirmesi ile de müşterinin çapraz kontrol yapmasını sağlar. Böylece şoför, fabrikanın ve müşterinin bilgisi olmadan fazla tutar tahsil edemez.

5. Gün sonunda şoförün yaptığı tüm teslimatlar, firmanın kullandığı web uygulamasına anında yansır. Çünkü şoförün kullandığı cep uygulaması, merkezdeki web uygulaması ile entegre çalışır. Dolayısıyla kuruluşuna kadar hesaplanmış bir rapor hazır beklemektedir.

### **1.1. İlgili Çalışmalar**

Bir makalade, eski sistemlerde yer alan masaüstü CRM uygulamasının, artan ihtiyaçlar doğrultusunda bir mobil uygulamaya dönüşümü anlatılmıştır [1].

Artık mobil telefonlar, en az masaüstü bilgisayarlar kadar uygulamaları çalıştırabilecek güce erişmiştir. Fakat, yazılım geliştirme süreçlerinde hala masaüstü uygulamalar daha zengin durumdadır. Büyük işletmeler, artık uygulamalarını mobil uyumlu hale getirmektedirler. Bu noktada önümüze çıkan bir sorun da şu ki, eski uygulamaların mobil platformlara olan adaptasyonunun oldukça zor olduğudur. Bu makalede, eski sistemlerin mobil uyumlu haline getirilmesi sürecinde takip edilecek en iyi çözümlerden birinin MDD (Model Driven Development) Model Odaklı Geliştirme ve MDA (Model Driven Architecture) Model Odaklı Mimari olduğudur. Bu makalede, bir CRM (Customer Relationship Management) Müşteri İlişkileri Yönetimi projesinin masaüstünden mobil platforma nasıl aktarıldığı incelenmiştir. Bu proje, açık kaynak kodlu Eclipse, EMF (Eclipse Modeling Framework), EMP (Eclipse Modeling Project) ve Android platformu üzerinde konumlandırılmıştır [1].

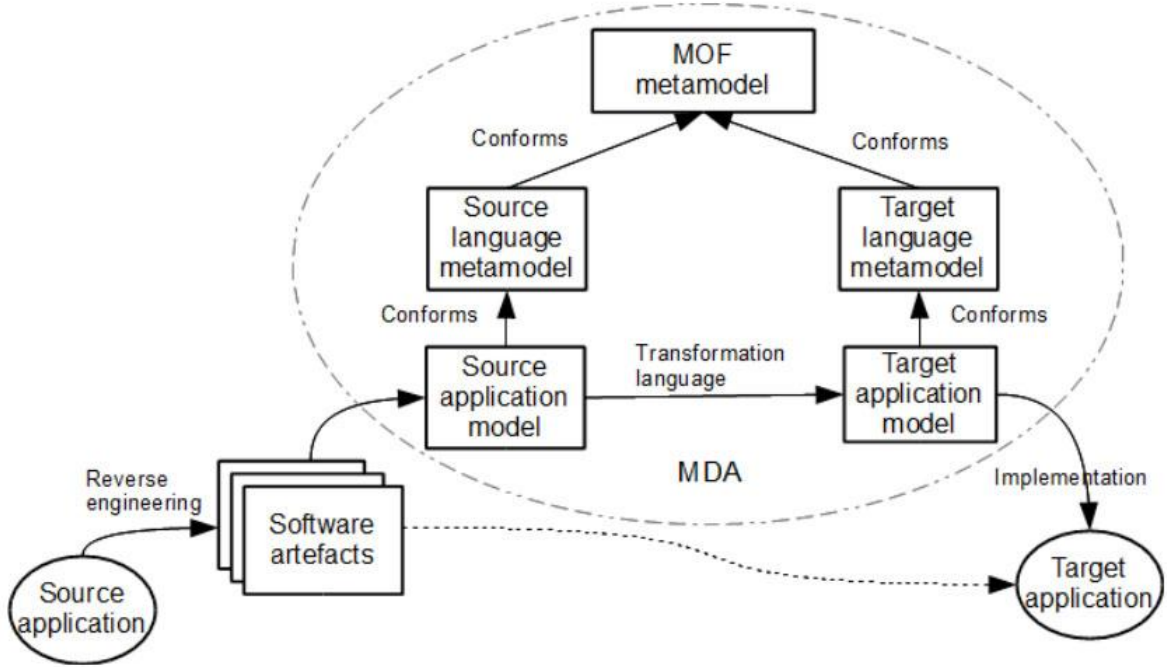
Günümüzde artık mobil cihazlar kullanıcılarıyla birlikte gelmekte ve her zaman her yerde bulunabilmekteler. Mobil cihazların bu yeni özellikleri arasında, küresel konumlandırma sensörleri (GPS-Global Positioning Sensors), kablosuz bağlantılar (wireless connectivity), dahili web tarayıcıları (built-in web browsers) ve fotoğraf/video/ses yetenekleri bulunmaktadır. Taşıdıkları işlemci ve ram kapasiteleri sebebiyle artık masaüstü uygulamalarının çalıştırdığı uygulamaları çalıştırabilmiş hale gelmişlerdir. Bu sebeple mobil tarafındaki iş geliştirme süreçleri son yıllarda oldukça artmıştır. Tabi bu gelişim de

beraberinde mobil uygulamalarının sınıflandırılmasını (taksonomi) gündeme getirmiş ve mobil uygulamaları 5 ana başlıkta irdelemişlerdir [1].

Mobil yayın: Farklı türdeki içeriklerin geniş bir alanda yayınlanması. Mobil bilgi: Genellikle uygulama üzerinden kullanıcıya aktarılan bilgileri oluşturur. Mobil işlem: Elektronik ödemeler, ürün alım/satımı gibi finansal işlemleri oluşturur. Mobil operasyon: Müşteri ile direkt temas kurmadan işletmenin operasyonları desteklemek içindir. Mobil İşbirliği: İşletmenin, başka işletmelerle sağladığı işbirliği platformudur [1].

Mobil uygulamalardaki temel zorluğun birden fazla android işletim sisteminin olması, ve her cihazda farklı işletim sistemlerinin koşmasıdır. Dolayısıyla hazırlanacak bir arayüzün, tüm (android) cep telefonlarında çalışabilir olması demek, ancak bağlam duyarlı (context aware) bir kodlama standartının belirlenmesiyle mümkün olabilir. Tabi bu kadar ayrı markada ve ayrı işletim sistemlerine sahip android uygulamalarının var olması, mobil geliştiriciyi de zor duruma sokmaktadır. Tam da bu noktada, MDD (Model Driven Development) bu zorlukların üstesinden gelmek için ümit verici bir yaklaşım olarak kabul edilir [1].

MDD'nin arkasında yatan asıl güç, sistem işlevselliği ile uygulamayı birbirinden ayırmaktan geçer. Ve takip edilmesi gereken yol, soyut modellerden uygulamalara doğru bir sürecin işletilmesidir. Ancak bu şekilde çoklu platform desteği sağlamak mümkün olur [1].



Şekil 1.1 MDD uygulamasında izlenen yeniden yapılanma operasyonu [1]

Şekil 1.1’de gösterilen şema, MDD uygulamasında izlenen yeniden yapılanma operasyonudur. Bu işlemi (reengineering process) aslında 3 fazda özetleyebiliriz. 1. Desktop uygulamasındaki kaynak kodun incelenmesi 2. Bu inceleme sonrasında soyut mimarinin oluşturulması. 3. Bu mimariye uygun olarak yeni mobil uygulamanın tasarlanması [1].

Biz ise MDD yöntemini, bu makaledeki aksine, “tamamen yeni bir projede” kullandık. MDD kullanmamızın temel sebebi, sahada yüzlerce kullanıcının değişik marka ve bu markanın değişik modellerinde cep telefonlarına sahip olmasıydı. Bu tür geniş alanda kullanıma sahip uygulamalarda, çoklu platform desteği kaçınılmaz bir problemdir. Bu sebeple uygulama geliştirmeye başlamadan önce, sahanın istekleri toplanmış, ihtiyaca göre bir fonksiyonel tasarım hazırlanmıştır. Daha sonra bu ihtiyaçlar doğrultusunda kodlamaya başlanmıştır. Geçmişte windows uygulaması geliştirenlerin de iyi bildiği üzere, işletim sistemlerine göre farklı tavır sergileyen masaüstü uygulamaları karşımıza çıkmaktaydı. Mesela, Windows XP işletim sistemine sahip kullanıcıların kimisinde SP2 (Service Pack 2) kurulu iken, kimisinde işletim sistemi sonrası herhangi bir Windows Update uygulanmıyordu. Android tarafında karşılaştığımız bu sorunu 2 şekilde çözüme kavuşturduk. Yeni eklediğimiz

ileri seviye özellikler (örneğin; bir çağrı geldiğinde arayan kişinin sipariş bilgilerinin ekrana getirilmesi) müşteriden belli izinlerin alınmasını gerektiriyordu. Manifest dosyasına bu izinleri ekledik. Ve bir popup pencere ile bu izinleri vermesi gerektiğini, aksi halde uygulamanın bu özelliğini kullanamayacağını anlattık. Müşteri, izin vermesi ya da vermemesi durumunda neyle karşılaşacağını bildiği için tercihi kendisine bırakmış olduk. İzin verirse, CallerID özelliğini kullanacak, vermezse bu özellik olmadan uygulamayı kullanmaya devam edecektir. Bir diğer konu da android kütüphanelerini zaman içinde yükseltme ihtiyacını hissetmemiz. Çünkü uygulama her geçen gün yeni özelliklerle çıkmaya devam etmekteydi. Bu sebeple müşterilerin android işletim sistemi versiyonlarının en az 4.4.2 olmasını istedik. Bu işletim sisteminin altında cep telefonuna sahip olanlara ise Play Store üzerinden kurulum hizmeti sunmadık. Çünkü 4.4.2'nin altında olanlarda zaten arama sonuçlarında görünmüyorduk. Ve son olarak, android 9 işletim sistemi ile birlikte Google Play Store üzerinde bazı katı kurallar getirildi. Uygulamada kullandığımız bazı zorunlu izinler play store'un yeni yayın politikasına takıldığına müşteriye .apk (android uygulamasının çalıştırılabilir dosyası) üzerinden kurulumları sağladık. Böylece işletim sistemi versiyonu sebebiyle oluşan problemleri de çözüme kavuşturmuş olduk.

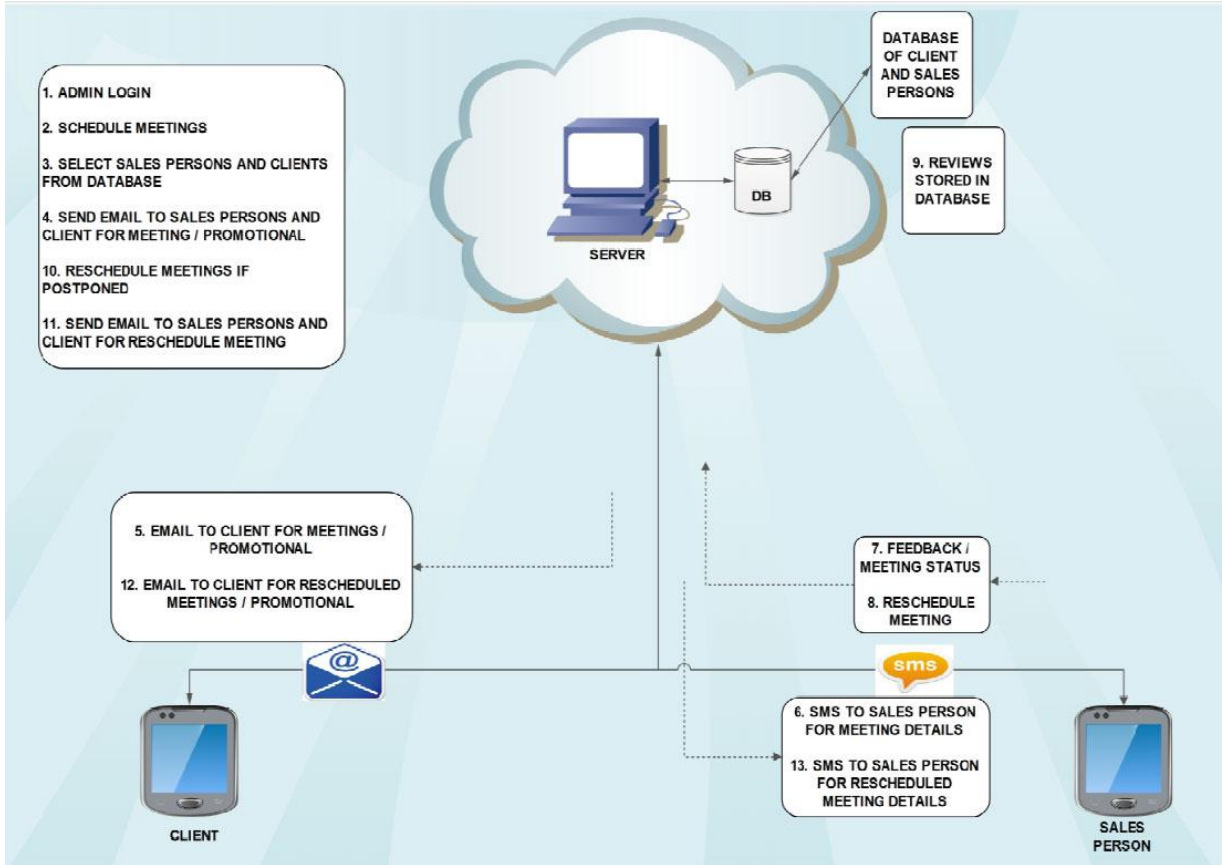
Diğer bir makalede, web tabanlı bir CRM uygulamasının toplantı ve iş atamalarını yapabildiği, saha personellerinin de android üzerinden atılan bu işlere katılabileceği anlatılmıştır [2].

Bu makalede, web uygulaması desteğinde bir mobil uygulama tasarlanmıştır. Amaç, satış yöneticileri ile satış görevlilerinin birlikte çalışabilecekleri merkezi bir otomasyon sistemi geliştirmektir. Uygulama sayesinde yöneticiler, toplantılarını planlamakta ve konuyla ilgili kişilere iş atamalarını yapabilmektedirler. Web üzerinde geliştirilen bu iş takip sisteminin mimarisi ise SAAS üzerine konumlandırılmıştır. Firma yöneticisi, iş ile alaklı herhangi bir atama yaptığında, bu bilgi sahadaki görevliye mail yoluyla iletilmektedir. Saha görevlisi maili inceledikten sonra gerekli aksiyonu kendi cep telefonundan android uygulaması üzerinden gerçekleştirir. Android uygulaması ise web uygulaması ile SOAP (simple object access protocol) üzerinden konuşur [2].

SAAS, Software as a Service (Hizmet olarak servis) mimarisidir. SAAS'taki yapı, uygulamaların internet üzerinden kullanımını mümkün kılmaktadır. Örneğin; e-posta, takvim veya ofis araçları birer örnektir. SAAS modelinin en güzel taraflarından biri de "kullandıkça



öde” yapısında olmasıdır. Hangi uygulamaya ihtiyacınız varsa, o uygulamayı kiralar ve bunu web tarayıcısı üzerinden kullanırsınız. Ana altyapı, yazılım ve veritabanının tamamı hizmet sağlayıcısının veri merkezinde yer alır. Hizmet sağlayıcısı, sizin için hem donanım hem de yazılım desteği sağlar. Size sadece kullandığınız tutar kadar ödeme sorumluluğu kalır.



Şekil 1.2 SAAS üzerinde konumlandırılan CRM projesi akış şeması [2]

Şekil 1.2’ de SAAS üzerinde konumlandırılan CRM projesi akış şeması gösterilmiştir. Web uygulaması, yöneticiler için hazırlanmıştır. Yönetici, kendi hesabıyla sisteme login olur ve gerekli gördüğü iş atamalarını ilgili kişilere yapar. Web uygulaması, temel olarak 4 fonksiyonu üstlenir. 1. Toplantı ayarlama 2. İşleri yönetme 3. Toplu mail gönderimi (tüm saha görevlilerine) 4. Sahadan geri dönüşler alma. Yönetici, sahadaki görevliye iş atadığında, görevli bu işi isterse öteleyebilir ya da tamamlayabilir. Her iki durumda da yöneticiye bir geri dönüş yapar. Yönetici, onay makamında olduğu için, işi ya kapatır, ya da başka bir takvim

belirler. Yapılan tüm işler kayıt altına alındığı için (logging), yönetici istediği zaman hangi çalışanın daha önceden hangi işleri tamamladığını, ne kadar süre çalıştığını görebilir, raporlayabilir. Yönetici, isterse işin muhteviyatına göre mail ile birlikte saha kullanıcılarına SMS bildirim de gönderebilir. Böylece kişi, dışarıdayken bile acil işleri SMS yoluyla takip edebilir [2].

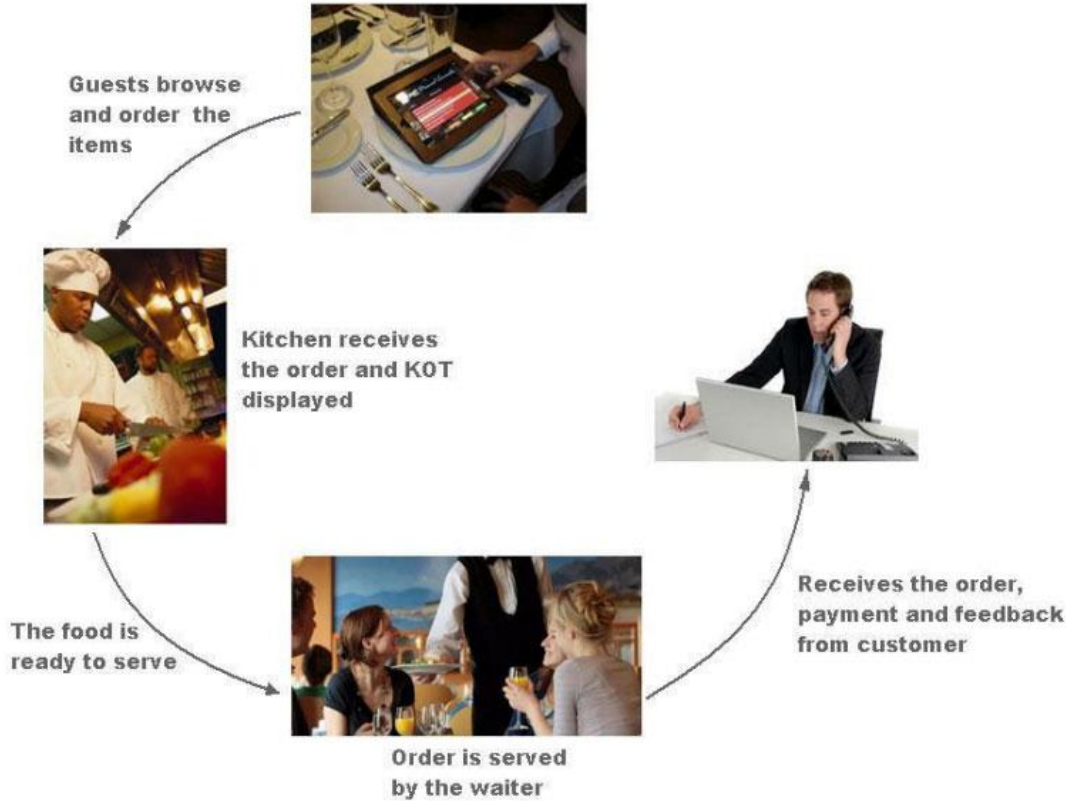
CRM (Customer Relationship Management) Müşteri ilişkileri yönetimi uygulamaları firmalara hem zaman tasarrufu sağladığı gibi hem de tüm toplantılarını veritabanında tutarak uzun vadede onlara bir profil çizmelerini sağlar. Web service (SOAP) sayesinde android ile web uygulamaları birbirleriyle kolayca iletişim kurarlar. Bu sayede merkez ile saha arasında kusursuz bir entegrasyon sağlanmış olur [2].

Bu uygulamada kullanılan SAAS altyapısı, cloud (bulut) çözümlerinden biridir. Mobil platformlar ile entegrasyonu da SOAP altyapısı kullanan web servisleriye sağlanmıştır. Biz ise teze konu olan projemizde PAAS altyapısında, mobil entegrasyonu için SOAP yerine RESTful'u tercih ettik. Bunun temel sebebi SOAP'ın zorunlu olarak XML altyapısını kullanması sebebiyle REST servislerine göre daha yavaş çalışmasıdır. REST servisleri, sadece XML değil, aynı zamanda JSON verileri üzerinden çalışabilmektedir. Cloud üzerinde mimari olarak tercih edeceğimiz 3 tür yapı bulunmaktadır. IAAS (Infrastructure as a service), PAAS (platform as a service) ve SAAS (software as a service). Bu türler arasındaki farkı en iyi şekilde anlatan grafiğimiz tezin sonunda gösterilmiştir.

Başka bir makalede, web servisleri kullanarak birçok platform ile entegrasyonun sağlandığı, oteller için çözüm sağlayan, mutfak ve müşteri ilişkilerinin yönetilebildiği android cihazlar üzerinden katılım sağlanabilen bir CRM projesi geliştirilmiştir [3].

Günümüzde web uygulamaları geniş bir şekilde kullanılmakta ve heterojen sistemlerin entegrasyonunda önemli bir rol üstlenmektedir. Bu makalede ise, bir otel rezervasyon takip sistemini web uygulamalarıyla nasıl hayata geçirildiği anlatılmaktadır. Dijital otel yönetimi sisteminde birden farklı platformlar ile entegrasyon sağlanmıştır. Mutfak sipariş sistemi, faturalandırma ve müşteri ilişkileri yönetimi bunlardan bir kaçıdır. Bu çözüm, her boyuttaki otel zincirlerinde kullanılabilir. Bu sayede müşteri memnuniyetinde kalite ve hızlı servis sağlanmış olur. Otel hizmetindeki kalite, müşterilerin istedikleri zaman kişiselleştirilmiş hizmet alması ile sağlanır [3].

Restoranlar en gözde mekanlardan biridir. Amaç, müşteriye en kısa sürede, verdiği siparişi eksiksiz teslim edebilmektir. Bu proje DOSRUA (Digital Ordering System for Restaurant Using Android) olarak adlandırılmıştır. Projenin uygulanışı, 3 noktada tabletin olmasını zorunlu kılar. 1. Kullanıcı tabletleri: Bu tabletler, restorana yemek için gelen müşteriler içindir ve içinde restoran menüsü yer alır. Wifi üzerinden internet bağlantısını kurar. Kullanıcı, menü üzerinde (fiyat, menü vs.) düzenleme yapamaz. Sadece ürünleri listelemek ve sipariş vermek üzere tasarlanmıştır. Uygulamanın arayüzü, her eğitimdeki kişilerin anlayabileceği şekilde düzenlenmiştir. 2. Yönetici tableti: Sadece restoran yöneticilerinde bulunur. Yönetici, bu uygulama ile tüm restoranı kontrol eder. İsteddiği tablete erişir, istediği fiyat ve menü değişikliğini yapabilir. 3. Mutfak tableti: Bu tablet de mutfak kısmında bulunur. Diğer tabletlere oranla daha büyük olması, görünmesi açısından önem taşır. Mutfak ekibi, bu tablet üzerinde, anlık siparişleri görüntüler [3].



Şekil 1.3 Restoran sipariş sistemi mimarisi [3]

Şekil 1.3’de görüldüğü gibi sistem mimarisini şu şekilde özetleyebiliriz : Müşteri, restorandan içeriye girer. Masasına oturur. Kendisine bir tablet uzatılır. Tablet üzerinde menülerde sörf yapar ve istediği ürünleri seçer. Ürün seçimlerini yaptıktan sonra son aşamada onaylamasını yapar ve siparişini mutfağa geçmiş olur. Bu sipariş anında mutfağa iletilir. Sipariş, mutfak tarafından hazırlandıktan sonra garsonlar tarafından servis edilir. Müşteri, yemeğini yedikten sonra toplam tutar tekrar tablete yansıtılır. Teknolojik altyapı da şu şekildedir; Android cihazlardaki işletim sisteminin minimum versiyonu 2.2.3 olmalıdır. Programlama dili Java’dır. Kodlamada kullanılan IDE ise Eclipse Indigo olmuştur. Veritabanı bağlantıları için JSP, Android üzerindeki veri operasyonları için de SQLite kullanılmıştır [3].

Bu makaledeki projede veritabanı olarak MySQL ve SQLite tercih edilmiştir. CRM tarafı JSP ile kodlandığından veritabanı olarak MySQL, Android üzerindeki veri operasyonları için de SQLite tercih edilmiştir. Bizim projemiz ise tamamen Microsoft tarafında yazıldığı için, CRM tarafında, Asp.net ile en uyumlu veritabanı olan MSSQL tercih edilmiştir. Android üzerindeki uygulamamızda da burada olduğu gibi tüm dünyanın gömülü sistemlerdeki tercihi SQLite olmuştur. Bizdeki Android versiyonunun minimum versiyonu 4.4.2’dir. Versiyonlardaki bu minimum gereklilik, android uygulaması içinde kullandığımız objeler doğrultusunda belirlenmiştir. Makaledeki projeye göre çok daha fazla (callerid, gps, bluetooth bağlantılar vs) kullandığımız için versiyonumuzun minimum değeri bu olmuştur. Android tarafındaki geliştirme Android Studio IDE’si üzerinden, web tarafındaki geliştirme ise Visual Studio üzerinden gerçekleştirilmiştir.

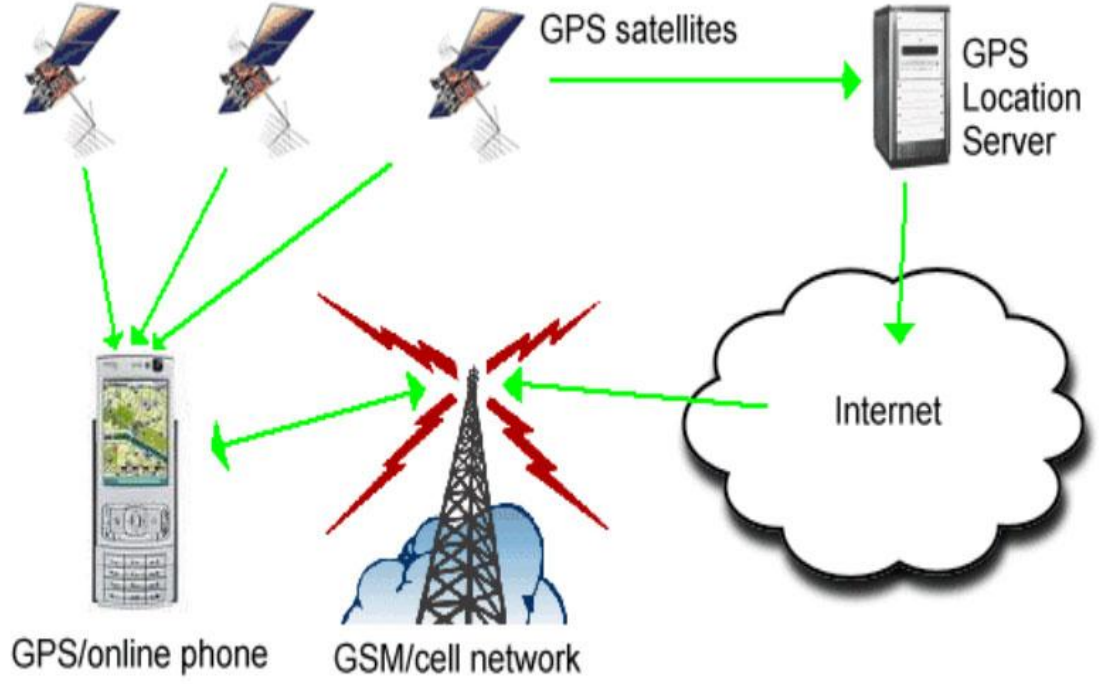
Farklı bir çalışmada, lokasyon bazlı bildirim servisleri üzerine kurulu bir mobil uygulamadan bahsedilmiştir. Bulunduğu konuma göre alacağı bildirimleri değişen bu yapıda, GPS sistemi kullanılmıştır. Örneğin, kullanıcıya en yakın otel ve restaurant tespiti yapılabilmektedir [4].

Lokasyon bazlı servisler, mobil kullanıcılara bir çok avantaj sunar. Hem buldukları konum hakkında bilgi alabilir, hem de kendisine en yakın otel, benzin istasyonu gibi yerlerin tespitini sağlar. Cep telefonlarından GPS ya da GPRS servislerini kullanarak lokasyon tabanlı hizmet alan kullanıcıların şu hizmetlere erişmesi mümkün : Bulunduğu konum ya da gideceği yerle ilgili trafik yoğunluğunu gösteren harita, bir noktadan başka bir noktaya gidilebilecek en kısa mesafenin tespiti (trafik yoğunluğunu da göz önünde bulundurarak) ya da kendisine en

yakın otellerin adresleri bunlara örnek olarak gösterilebilir. Bu makalede, lokasyon tabanlı hizmet sunan Google Web Servisleri ve Walk Score Transit API'si kullanılmıştır [4].

Cep telefonları, temel iletişim dışında 1990'ların başında internetin de dünyamıza girmesiyle birlikte çok daha değerli bir hizmete dönüşmüştür. Konum tabanlı servisler (LBS- Location Based Services) joğrafik konum bilgisinin alınmasıyla artık mümkün hale gelmiştir. Tabi bu da uygulama geliştiricilerin geniş bir alanda çalışmalarına imkan sağlamıştır. Örneğin, en yakın banka ya da otel hizmeti, trafik yoğunluğunu gösterir uygulamalar, belli bir alana girildiğinde otomatik yapılan bildirimler, çalınan telefonun konum bilgisi, bölge bazlı arkadaş bulma (ürün satışı, danışmanlık vs) [4].

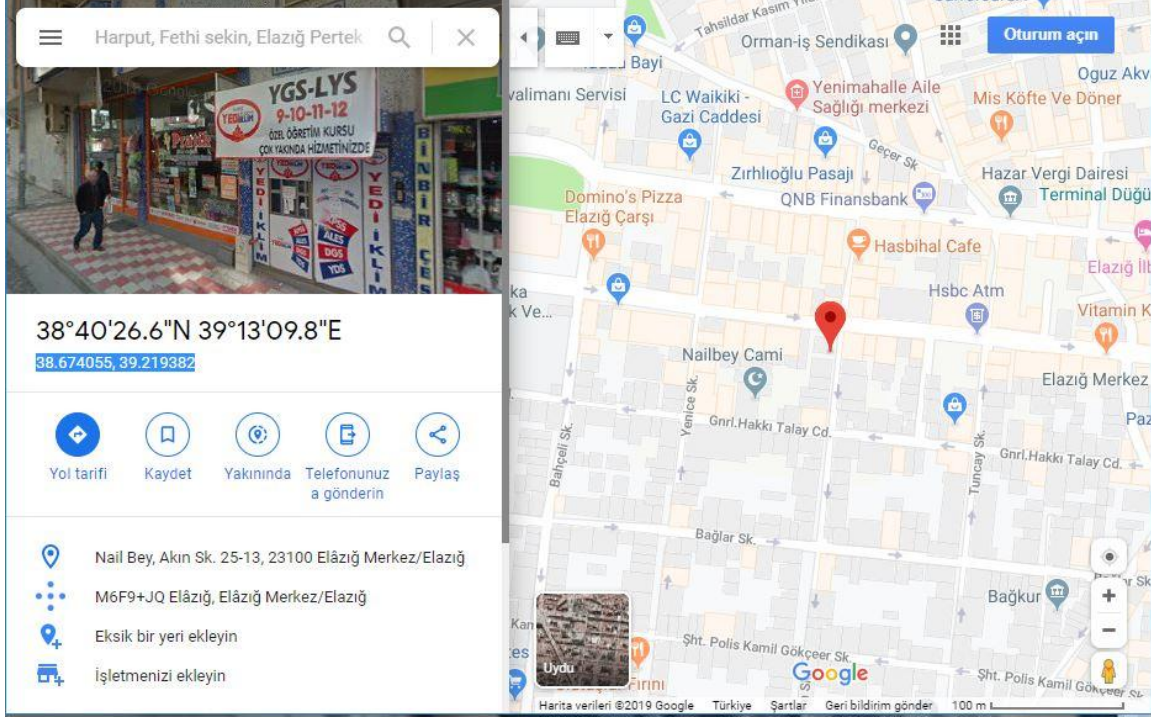
Lokasyon bilgisi 2 şekilde alınabilir. 1. Cep telefonunun servis sağlayıcı şebekesi: Buradaki yapı baz istasyonları (BTS-Base Transceiver Station) üzerinden yürür. Lokasyon belirlemenin en basit ve en ucuz yöntemidir. Avantajı, ekstra bir maliyetinizin olmamasıdır. Fakat konum bilgisi 2-20 km aralıkta bir yerde olabilir. Bu tamamen baz istasyonlarının sıklığı ile alakalı bir durumdur. 2. Uydular : GPS'in (Global Positioning System) çalışma yapısı, dünyayı dolaşan uydular üzerine kuruludur. Kullanıcının konumu, farklı uydu cihazlarının mesafe kombinasyonu ile hesaplanarak bulunur. Assisted-GPS (A-GPS) adlı yeni teknoloji, cep telefonunun servis sağlayıcı şebekesi ile birlikte uyduları kullanarak 5-10 metre hata ile konum belirlemeyi başarmıştır [4].



Şekil 1.4 A-GPS sisteminin mimarisi [4]

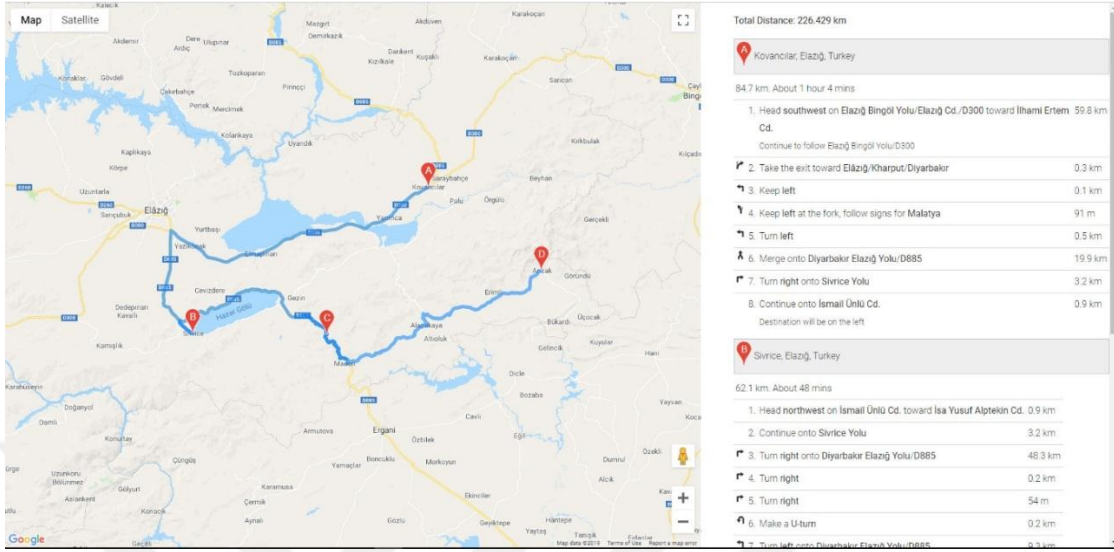
Şekil 1.4’te görüldüğü gibi Assisted-GPS ile hem servis sağlayıcı hem de uydular birlikte kullanılmıştır. Bizim projemizin en can alıcı noktalarından birisi de bu lokasyon bazlı hizmetleri, müşterinin kişiselleştirilmiş kullanımına açmak olmuştur. Halı yıkama sektöründe halı toplamak, halı dağıtmak dünyanın en zor işlerinden biridir. Asansörü olmayan bir apartmanın 5. Katına sırtında 5 tane halıyı çıkarmak ve bunu günde yaklaşık 30 evde icra etmek, herkesin yapabileceği bir iş yükü değildir. Bu sebeple, bu sektörde işçi mobilitesi çok yüksektir. Bu işi yapan kişilerin çok değil birkaç ay sonra işi bıraktığı, yerine başkalarının alındığı çok sık rastlanan bir durumdur. Yeni gelen şöför, şehrin haritasını, konumunu tam olarak bilmediğinden, halı almaya ya da teslim etmeye giderken saatlerini kaybedebilir. Geliştirdiğimiz sipariş takip otomasyonu uygulamasına Google Geocoding API’sini eklenmiştir. Web uygulamasında il, ilçe, mahalle, sokak ve kapı numarası girdiğinizde, bu API otomatik olarak o adresin enlem ve boylam bilgisini ekrana getirir. (Örn : 38.674055, 39.219382) Böylece, halısını yıkatacak kişinin adresi, sistemde artık koordinat bilgisiyle

tutulmaya başlanır. Halı almaya ya da teslimata giderken, android uygulamasındaki koordinat ikonuna bastığında, cep telefonu üzerinde (built-in support) zaten kurulu olan Google Haritalar uygulaması devreye girer ve belirtilen koordinata (enlem, boylam) bir rota oluşturulur. Buraya en hızlı yol (trafik yoğunluğu da göz önünde bulundurularak) google haritalar uygulaması üzerinden sağlanır.



Şekil 1.5 Google Geocoding API ile alınan enlem/boylam bilgisinin haritada görünüşü

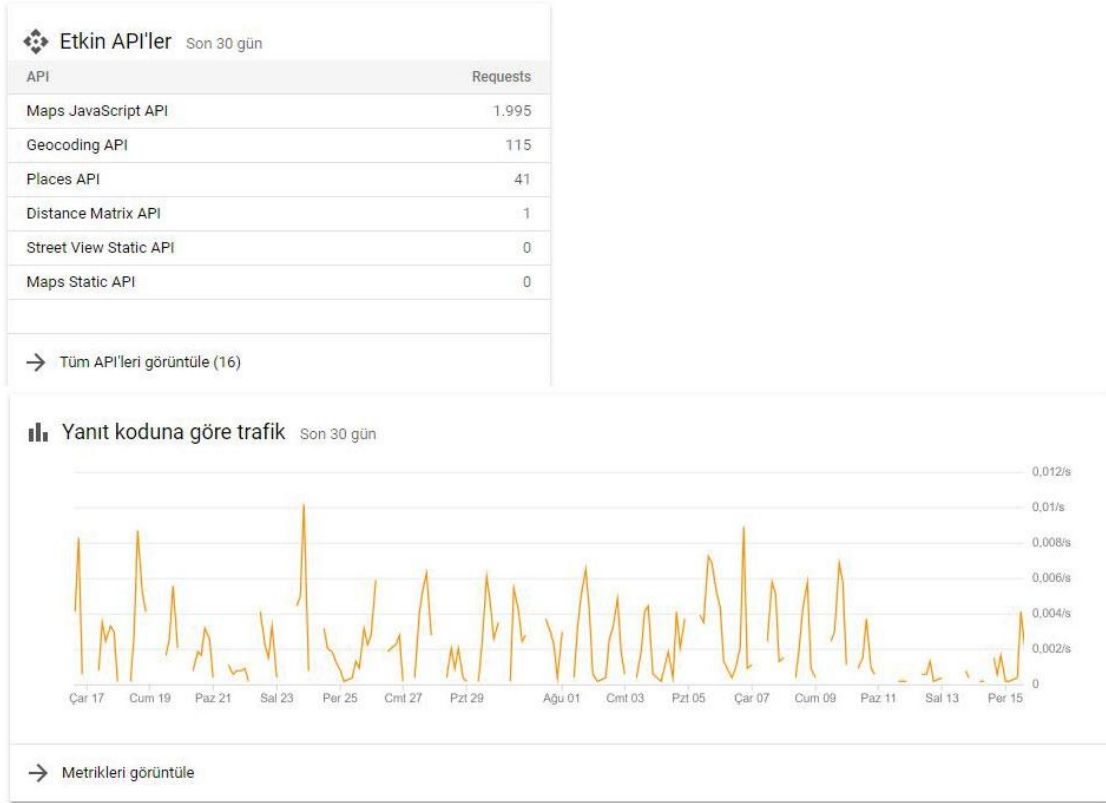
Şekil 1.5'te konum bilgisini google haritalar üzerinde gösterilmiştir. Geocoding API'ye ek olarak Direction API de kullanılmıştır. Ve asıl katma değer bu noktada oluşmuştur. Direction API ise, kendisine verilen onlarca koordinatı, merkez fabrikaya göre mesafesini hesaplayarak, aracın fabrikadan çıktuktan sonra hangi noktalara sırasıyla gitmesi gerektiğini belirleyen bir rota oluşturur.



**Şekil 1.6** Google Direction API iki nokta arasındaki en optimal rotanın belirlenmesi.

Şekil 1.6’da iki nokta arasında izlenebilecek en iyi rota gösterilmiştir. Kullandığımız bu API’lerin tüm istatistikleri, Google’ın sağladığı yönetim panelinden takip edilebilir ve hangi servisin ne kadar kullanıldığını görülebilir. Bu servisler günlük belirli bir sayıda talebi ücretsiz karşılar. Örneğin; günlük 1000 talep. Bu talebin üzerine çıktığında artık ücretli hale dönüşür ve “Kullandıkça öde” prensibince ay sonunda harcanan trafik kadar ödeme yapılır.





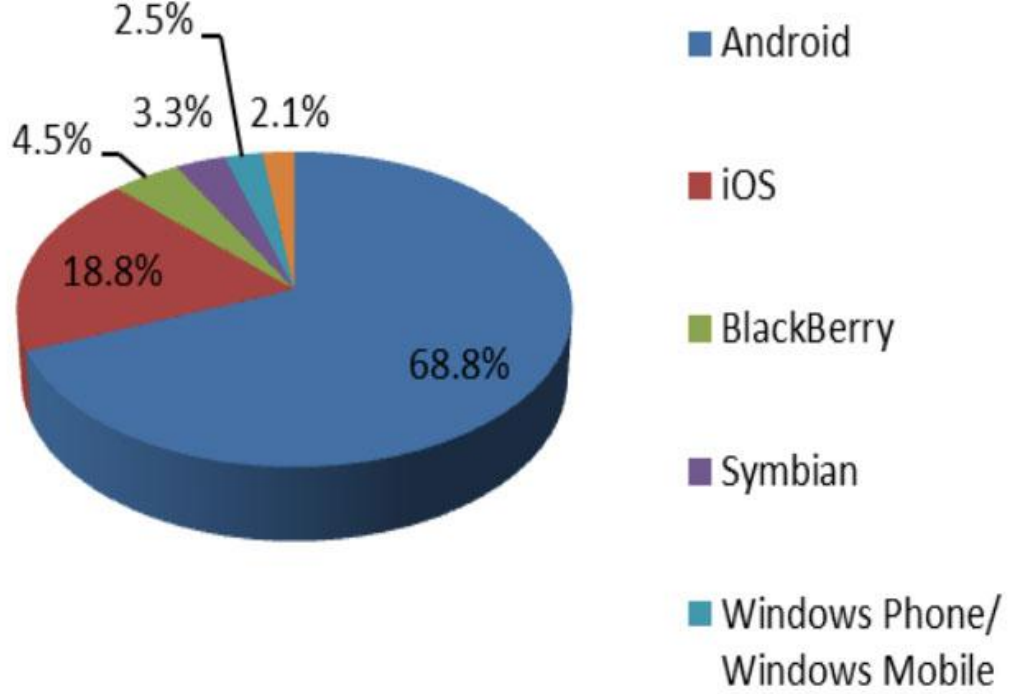
Şekil 1.7 Google API'lerinin istatistiklerini izlediğimiz bölüm

Şekil 1.7'de kullandığımız API'lerin kaç kez kullanıldığı gösterilmektedir.

Diğer bir çalışmada, web servislerini kullanan, REST servisleriyle çalışan, mobil provizyon sisteminden bahsedilmiştir [5].

Son yıllarda mobil cihazlarının işlemci gücü, hafıza kapasitesi oldukça artmıştır. Bununla birlikte kablosuz bağlantılar üzerinden geçen veri hızı da aynı doğrultuda artış göstermiştir. Ve bu kapasite gün geçtikçe daha da artacaktır. Bu sebeple artık, web servislerini akıllı telefonlar üzerinde host etmek (konumlandırmak) daha da kolaylaşmıştır. Dünyadaki akıllı telefon sayısının yaklaşık 1.8 milyar olduğu tahmin edilmektedir. Bu kadar çok cihaz demek, birbirinden farklı (heterojen) bir yapı anlamına gelmektedir. Bu, farklı işletim sistemleri ve farklı donanım kapasiteleri anlamına gelir. Şekil 1.8'de görüldüğü üzere (IDC Worldwide Mobile Phone Tracker, April 12, 2013), sektörde en az 5 farklı işletim sistemi bulunmaktadır [5].

## Mobile OS Market Share



Şekil 1.8 Dünyada kullanılan mobil işletim sistemleri [5]

Dünya nüfusunun %90'ında cep telefonu bulunmakta ve her geçen gün veri akış hızı da artış göstermektedir. 3G, 4G ve LTE (Long Term Evolution) bağlantılarına ek olarak Wi-Fi erişimleri sayesinde bu cihazlar her yerden ve her noktadan internete erişebilmektedirler. Tüm bu gelişmeler ve açık kaynak kodlu çalışmalar, mobil cihazları artık bir servis sağlayıcı konumuna getirmiştir. Mobil cihazların web servis sağlayıcı olarak kullanılmasının başlıca sebepleri şunlardır : 1. Web uygulamalarını, herhangi bir internet erişimine ihtiyaç olmadan lokalde test edebilme imkanının olması. 2. Üçüncü parti uygulamalar sayesinde telefona uzaktan bağlanma imkanının olması. Bu tür bağlantılar güçlü güvenlik katmanlarına ihtiyaç duyar. 3. GUI (Graphical User Interface-Grafiksel Kullanıcı Arayüzü)'ne ihtiyaç duymadan lokal android uygulamalarını kullanarak daha güçlü bir altyapıya sahip olabilmesi [5].

Bu çalışmanın asıl amacı, cep telefonlarını bir servis sağlayıcı olarak kullanmayı analiz etmektir. Cep telefonu üzerinde konumlandırılan (deployment) bu uygulama, bilindik web

arayüzleriyle tam uyumludur. Dolayısıyla son kullanıcı (client) bu arayüzün bir cep telefonu üzerinde host edildiğini anlayamaz. Mobil üzerindeki uygulamanın boyutu oldukça düşüktür. Ve çok daha az işlemci, hafıza ve pil kullanımına sahiptir. Ve bu makalede tamamen REST servisleri üzerine odaklanılmıştır. REST servisleri, SOAP'a göre çok daha ölçeklenebilir ve performansı yüksek bir altyapıya sahiptir.

<b>Load Level</b>	<b>Total No. of Reqt</b>	<b>Total No. of Resp</b>	<b>No. of Success Resp</b>	<b>% of Error Resp</b>	<b>Avg. Execu Time (MS)</b>
1	57	57	57	0.0%	28
5	107	107	107	0.0%	32
10	122	122	122	0.0%	24
15	136	136	136	0.0%	34
25	143	143	143	0.0%	67

Şekil 1.9 REST servislerinin performans analizi [5]

<b>Load Level</b>	<b>Total No. of Reqt</b>	<b>Total No. of Resp</b>	<b>No. of Success Resp</b>	<b>% of Error Resp</b>	<b>Avg. Execu Time (MS)</b>
1	57	57	57	0.0%	30
5	107	107	107	0.0%	35
10	122	122	122	0.0%	28
15	136	136	136	0.0%	39
25	143	143	143	0.0%	71

Şekil 1.10 SOAP servislerinin performans analizi [5]

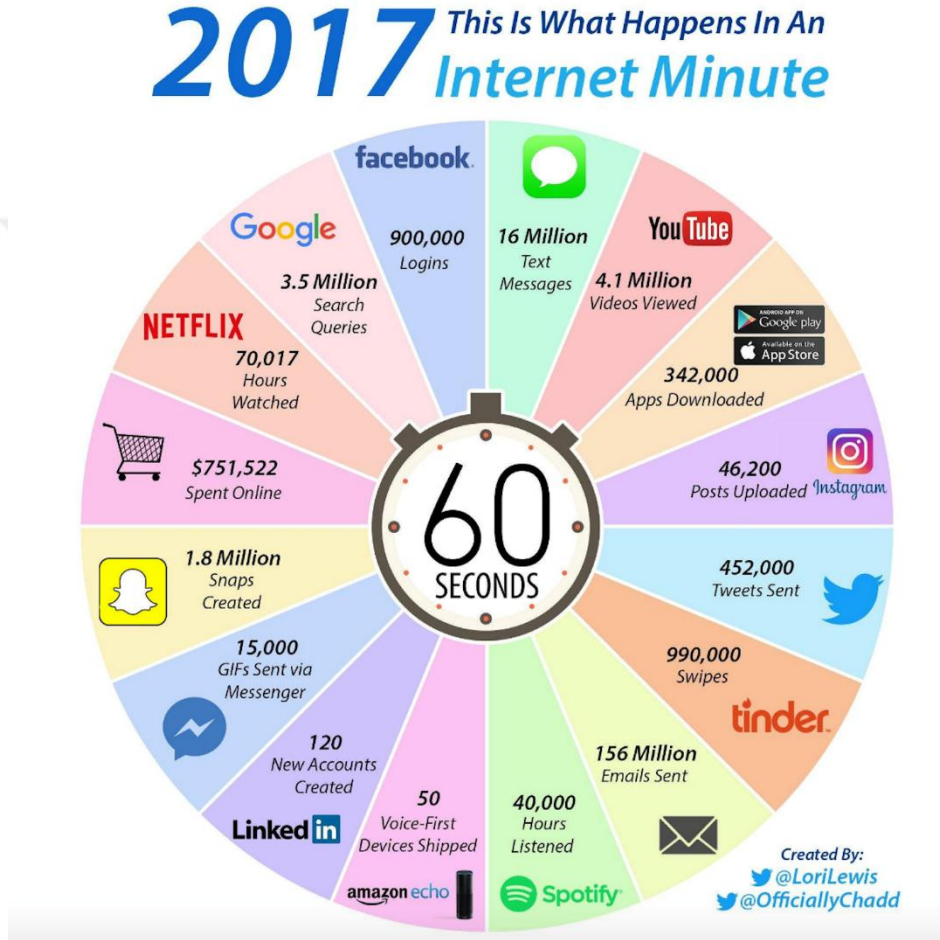
Şekil 1.9 ve 1.10'da da görüleceği gibi toplam talep sayısı (total no. of request) arttıkça REST'in ortalama dönüş süresi (Avg), SOAP'a göre çok daha hızlıdır.

Literatür taramalarını yaparken, bahsettiğimiz makalede [2] tercih edilen SOAP altyapısının yerine biz kendi projemizde neden REST kullandığımızı uzun uzadıya anlatmıştık. Son literatür incelemesinde [5] ise tercihlerinin REST'ten yana olduğunu görmemiz, kendi projemizde verdiğimiz kararın ne kadar doğru olduğunu göstermektedir. Biz de Microsoft Azure üzerinde PaaS altyapısında host ettiğimiz web uygulamamızı, sahadaki Android cihazlarla entegre ederken REST servislerini kullandık. Bu, eş zamanlı erişimin (concurrency) binlerce olduğu bir yapıda bize büyük kolaylık sağlamıştır. JSON verileri üzerinden konuşan iki platform sorunsuz ve yüksek performansta yıllardır sorunsuz çalışmaktadır.

## 1.2. Genel Bilgiler

Dünyanın gelişen konjonktürü ve rekabet ortamı, firmaların da iş yapış şekillerini değiştirmektedir. Teknolojinin ilerlemesi, artan işlem hacmi, şirketlerin mevcutta kullandığı

bilgi teknolojileri altyapılarının da hızla eskimesine veya yetersiz kalmasına sebep olmaktadır. Örneğin, son 2 yılda oluşan veri trafiği, kendisinden önceki 15 senenin veri trafiğinden çok daha fazladır.



Şekil 1.11. Dünyada 1 dakikada neler oluyor [6]

Şekil 1.11’de görüleceği gibi 1 dk içinde milyarlarca işlem oluşmaktadır. Uygulamalar artık, offline Windows uygulamalarından çıkıp web uygulamalara evirilmektedir [7]. Böylece her noktadan, her cihazdan erişim sağlanarak merkezi veriler üzerinde görüntüleme ve güncelleme yapılabilmektedir. Firmaların kendi sektörlerinde rekabette kalmalarını sağlayacak en önemli faktör, bilgiye istenildiği anda ulaşılabilmesi ve dağıtık bir modelle her noktadan veri operasyonları yapabilesidir.

Bilgi teknolojilerine yapılan yatırımlardaki en önemli konu, tabii ki sürdürülebilir maliyetlerdir. Yani, kullanılan cihazların bakımı, optimizasyonu (hızlandırılması), yetkin uzmanların görev alma zorunluluğu, internet, elektrik gibi hayati kavramların kesintisiz işleyebilmesini sağlayacak yedekleme sistemlerinin kurulmasıdır. Belli bir ölçeğin üzerinde yer alan kurumsal firmalarda bu konuların yürütülmesi sorun olmasa da küçük ve orta boyulu işletmelerde bu maliyetlere katlanmak, sürekliliğini sağlamak oldukça zordur. Çünkü fayda/maliyet analizi yaptığımızda, yaptığımız yatırımın size ne katacağı da önemlidir. Firmaların tek amacı vardır, kar etmek. Ve karlılığın da temel kuralı maliyetlerin düşürülmesi üzerine kuruludur. Dolayısıyla, iş üretiminde, dikkat edilmesi gereken konulardan birisi de kısmak yerine verimli hale dönüştürmeyi tercih etmekten geçer.

### 1.3. Bulut Bilişimin Faydaları

- Erişilebilirlik
- Güvenlik
- Düşük maliyet
- Yazılım kolaylığı
- Zamandan tasarruf



Şekil 1.12. Bulut bilişimin avantajları [9]

Şekil 1.12’de görüleceği gibi bulut bilişim size büyük bir konfor sağlar. İşte tam da bu noktada, web uygulamaları, servis tabanlı mimariler (SOA) bu maliyetleri kıyas kabul etmeyecek derecede düşüren çözümler olarak karşımıza çıkar. Merkezi uygulamaların, kendisine erişim sağlayan tüm cihaz ve platformlara destek vereceği bir yapıda olması sebebiyle, sadece bir noktada yatırım ve bakım maliyetlerini oluşturacaktır. Donanımsal yatırımların dışında, aynı server üzerinde konumlandırılmış birden farklı yazılım ve modül sayesinde bir firmanın ayrı ayrı birçok departmanına da çözümler sunabilme imkânı doğar. Dolayısıyla, hem fiziksel sunucular hem de sunucu üzerinde yayın yapan yazılımlar sayesinde maliyetler katlanarak azalacaktır.

Bilişim dünyasında Cloud Computing (bulut bilişim) olarak adlandırılan bu yapılar zaten temel olarak outsourcing (dış kaynak kullanımı) üzerine kurulmuştur. Firmalar, artan işlem hacimine göre istediği gibi kaynak kullanımını belirlemekte ve buna göre ödeme yapmaktadır. Böylece, fiziksel sunucuların kurulumu, bakımı, yetkili, alanında uzman kişilere bırakılmakta, son kullanıcı ise uygulamanın keyfini sürmektedir.

## 2. MATERYAL ve METOD

Teknolojik altyapıyı bir merkezde toplayarak hem yönetimi kolaylaştırmak hem de maliyetleri düşürmeyi amaçladık. Herhangi bir kurulum gerektirmeyen bu tür uygulamalar sayesinde müşteriler ve çalışanları istedikleri anda veriye ulaşabilmektedirler. Kapalı devre (offline) uygulamalarda böyle bir avantaj söz konusu değildir. İnternetin yaygınlaşmasıyla birlikte artık Windows uygulamaları (.exe) geri planda kalmış, yerini web uygulamaları almıştır. Böylece, hem geliştirme ortamı hem de son kullanıcı işletim sisteminden bağımsız çalışabilir hale gelmiştir. İşletim sistemlerinin farklılığı (Windows, Linux, Mac vs.) artık bir problem teşkil etmemektedir. Uygulamanın bir tarayıcı (browser) üzerinde çalışması, bize tüm noktalarda (cross platform) çalışma şansı sağlamıştır.

Günümüzde bulut bilişim (cloud computing) mimarisi ile çalışan onlarca örnek göstermek mümkündür. Örneğin, e-posta hizmetleri sunan g-mail, hotmail gibi servisleri bu kapsamda düşünebilir. Dropbox, dünyanın en büyük veri depolama sistemlerden birisidir. Trello, yine en fazla kullanılan iş takip (issue management) sistemlerine örnek gösterilebilir. Bir kaç web sunucusu ile tüm dünyanın kullanabildiği uygulamalar geliştirmek mümkündür.

### 2.1. ASP.NET

Klasik scripting (php, asp, cfm) dillerinden sonra bir devrim olarak yazılım dünyasına giren asp.net, bilindik interpreting (satır satır yorumlama) işleminin aksine compile (derleme) yöntemiyle çalışmakta ve başta hız olmak üzere geliştiriciye hatasız kodlama (strongly typed) imkânı da sağlamaktadır. Çünkü derlenen projelerde oluşan hatalar anında geliştiriciye yansıtılmakta ve gerekli düzeltmeleri yapmasına imkân vermektedir.

Scripting dillerine nazaran Asp.net tarafındaki önemli geliştirmelerden birisi de önyüz (frontend) ile arka taraf (backend) birbirinden ayrılmıştır. Böylece bir projede 2 farklı developer birbirinden bağımsız bir şekilde çalışabilmektedir. Önyüz tarafında css, javascript kodlaması yapılırken, arka tarafta C#, Vb.net gibi istenilen dil seçilebilir. Derleme sonrası hangi dilde yazılırsa yazılsın bir .dll dosyasına dönüştürüleceği için diller arasında performans farkı da oluşmayacaktır.



Scripting dillerini herhangi bir text editöründe (notepad, dreamviewer vs.) kodlayabilirken Asp.net uygulamalarını Visual Studio gibi efsane bir IDE üzerinde geliştirmek mümkündür. Bu size sürükle-bırak yöntemiyle hızlı arayüzler tasarlamayı sağladığı gibi, compile ve test süreçlerinde işinizi inanılmaz kolaylaştırır.

## **2.2. Microsoft SQL Server**

Microsoft'un geliştirdiği veritabanları içerisinde amiral gemisidir. RDMS (Relational Database Management System) yani "İlişkisel veritabanı yönetim sistemi" olarak çalışır. İlişkisel veritabanlarından kasıt, her satırın, bir başka tablodaki verilerle kurduğu birebir, birçok türündeki ilişkidir. Bu ilişkiler sayesinde yapılan sorgulamalarda bir veri ile alakalı diğer tablolardan veriler çekilebilir, raporlanabilir, grafikler oluşturulabilir.

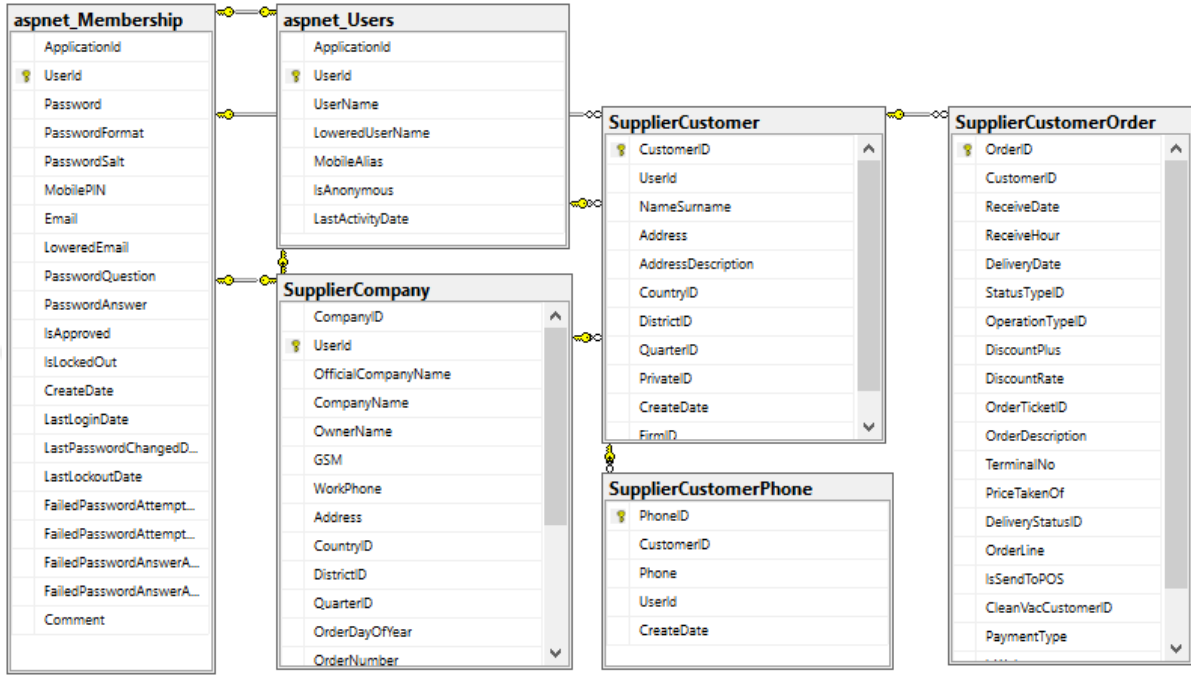
SQL server üzerindeki stored procedure (saklı yordam), trigger (tetikleyici), user defined function'lar (kullanıcı tanımlı fonksiyonlar) gibi özellikler sayesinde veri çekme ve verinin bütünlüğünü sağlama gibi işlemler kolaylaşır.

Veritabanında bulunan index kavramı, ve index istatistikleri, yapılan sorgulamaların hızlı bir şekilde cevap dönmesini sağlayan ana faktördür. Bu sebeple düzenli olarak veritabanı bakımı yapılmalı, index'ler ve index istatistikleri periyodik olarak "reorganize" ya da "rebuild" edilmelidir.

SQL Server'ın Cloud ortamındaki versiyonu ise Azure SQL'dir. Ve PAAS (Platform as a Service) olarak hizmet verir. Ki on-premises (firmanın kendi makinasına kurulan) sql versiyonlarına göre size inanılmaz bir kolaylık sağlar. Veritabanının tüm yönetimi Microsoft tarafındadır. Backup (full veya differential), Transaction Log Backup işlemleri otomatik yapılır. Makinanın tüm hardware kontrolleri (cpu, ram) Microsoft tarafından yönetilir. Bizim de bu projede tercihimiz Azure SQL'den yana olmuştur.

Tablolar arasındaki ilişkilerin kurulması, veri sorgu performansları için hazırlanan indexing işlemleri, her gün düzenli olarak çalışan otomatik görevler (job) ile tüm veri operasyonları gerçekleştirilmektedir.

## 2.3. Veritabanı Tasarımı



Şekil 2.1. Veritabanı tasarımı

Şekil 2.1’de de görülebileceği gibi üyelik sistemi, Asp.net ile birlikte gelen “membership framework” üzerine kurulmuştur. Kullanıcıların oluşturulması, şifrelerinin belirlenmesi, güvenlik soruları, hesap kilitleme, son giriş tarihleri gibi tüm yapılar aspnet\_users ve aspnet\_membership tabloları tarafından yürütülmektedir.

**aspnet\_membership**: Üyelerin (halı yıkama firmalarının) ana bilgilerinin tutulduğu tablodur. Her üyenin unique (tekil) UserId’si bulunur. Sistemdeki diğer tüm tablolarla ilişkilendirilmesi bu id sayesinde olur. Password alanı, şifrelerin tutulduğu yerdir. Email bölümü de unique bir değere sahiptir. Bu sebeple, aynı mail adresinden birkaç firma bulunamaz. Şifremi unuttum gibi özel bilgilerin paylaşımında oldukça önemli bir yer teşkil eder. IsApproved alanı, firmanın onaylı olup olmadığını gösterir. Şöyle ki; halı yıkama firması sistemimize üye olduğunda, belirlediği kullanıcı/şifre ile hemen giriş yapamaz. Sistem yöneticisi, gelen üyelik formunu analiz eder, sahte (fake) olmadığına kanaat getirirse üyeliğini onaylar. IsLockedOut alanı ise hesabının kilitli olup olmadığını belirler. Halı yıkama firmasının kurallara aykırı bir işlem/eylem içinde olduğu görüldüğünde hesabı kilitlenir.

Böylece kullanıcı adı/şifresi artık devre dışı olur. Createdate, üyeliğinin oluşturulma zamanını, LastLoginDate, en son sisteme ne zaman giriş yaptığını, LastLockOutDate ise en son ne zaman hesabının kilitlendiğini gösterir.

**aspnet\_Users:** UserName, firmanın kullanıcı adını tutarken, LastActivityDate ise en son ne zaman işlem yaptığını takip eder. Bu tablo, aspnet\_membership tablosunun diğer yarısıdır. Ve üyelik sistemi altyapısında en çok kullanılan 2 tablodur.

**SupplierCompany:** Halı yıkama firmasının üyelik formunu doldurduğunda topladığımız verileri sakladığımız tablodur. Adres, yetkili adı, cep telefonu gibi alanları içerir.

**SupplierCustomer:** Halı yıkama firmasının, müşterilerini kaydettiği tablodur. Bu tabloda halı yıkatan müşterilerin bilgileri yer alır.

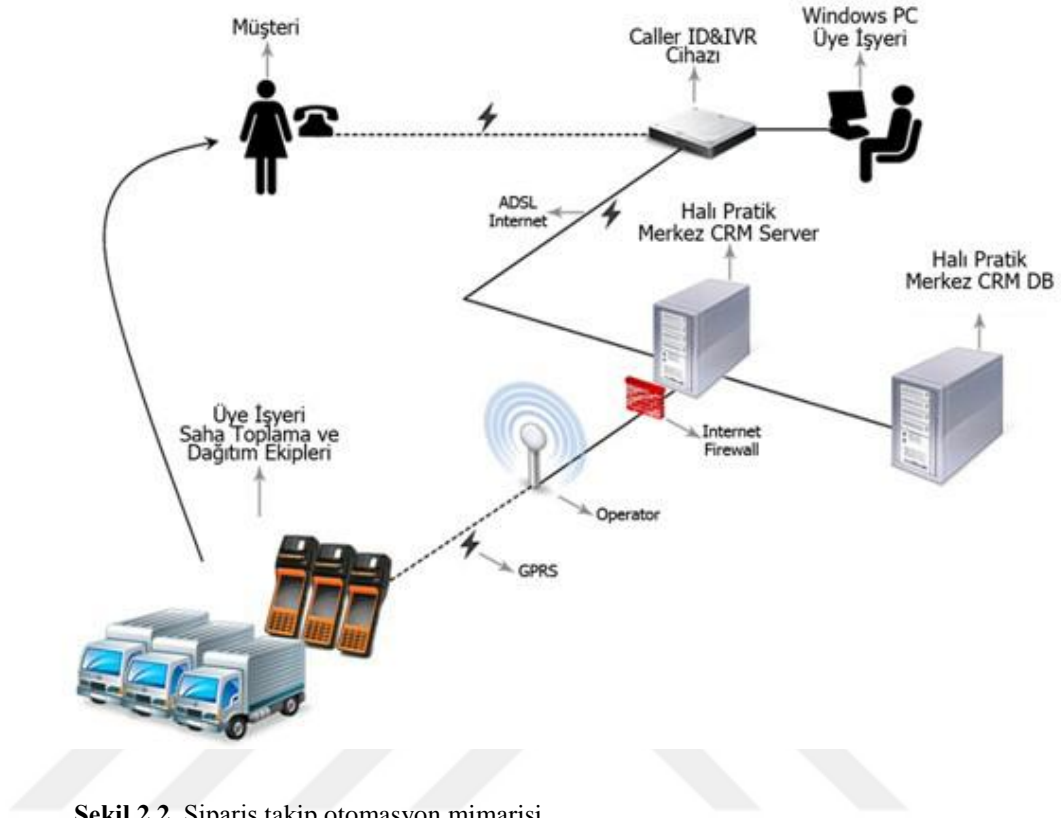
**SupplierCustomerPhone:** Müşterilerin telefonları yer alır. SupplierCustomer tablosu ile bire çok ilişki halindedir. Dolayısıyla bir müşteriye ait sınırsız telefon eklenebilir.

**SupplierCustomerOrder:** Müşteri siparişlerinin tutulduğu tablodur. Halısını yıkatan kişinin, geçmişte verdiği tüm siparişleri burada bulabilirsiniz. SupplierCustomer tablosu ile bire-çok ilişki halindedir. Bir müşterinin sınırsız siparişi olabilir.

## 2.4. İş akışı

Her şey müşterinin ilk telefonu ile başlar. Müşteri arama yaptığı anda firmanın bilgisayarlarına takılı CallerID cihazı sayesinde arayan kişinin bilgileri ekrana gelir.

CallerID cihazı, (sabit) telefon hattındaki sinyali yakalayarak USB kablosu üzerinden PC'ye iletir. PC üzerinde dinleme pozisyonunda olan bir uygulama (.exe) gelen numara bilgisini alıp işletim sisteminin varsayılan browserını tetikler. Açılan tarayıcıda bir URL üzerinde telefon bilgisi parametre olarak geçilir. CRM'in bulunduğu domain ise gelen requesti parametresine göre değerlendirip ilgili müşterinin hesabında sorgulama yapar.



Şekil 2.2. Sipariş takip otomasyon mimarisi

Şekil 2.2’de projemizin mimarisi gösterilmiştir. Gelen çağrının kim olduğunu müşteri temsilcisi ekranda gördükten sonra siparişi alınır. Temsilci, gelen siparişi sahadaki istediği cihaza terminal numarası seçerek yönlendirir. Sipariş, CRM’e girildiği anda, sahadaki android ya da pos uygulamasına otomatik bir bildirim (push servisleri) gider. Saha görevlisi gelen uyarıya istinaden cihaza baktığında, gitmesi gerektiği adresi görür. Temsilci, müşterinin kapısına gider ve ürünleri teslim alır. Ürünleri, 2 farklı cihazdan teslim alabilir. POS veya Android Uygulaması. POS uygulaması, CRM ile .net web servisleri üzerinden (SOAP protokolü) ile konuşurken, android uygulması RESTful servislerini kullanır. REST servisleri XML yapısı kullanmadığından .net web servislerine göre çok daha hızlıdır.

### 3. ONLINE SİPARİŞ TAKİP OTOMASYONU

Online Sipariş Takip Otomasyonu, temel olarak 3 ayak üstüne kuruludur. 1. Web Uygulaması, 2. Android Uygulaması 3. POS Uygulaması

Web uygulaması, halı yıkama firmasının ana üssüdür. Şöförlerin kullandığı Android ya da POS uygulamasına göre çok daha fazla yetki ve özellik taşır. Firma sahibi, web uygulaması üzerinden firmanın tüm giriş/çıkışlarını kontrol eder, Android kullanıcılarının (şöför) yetkilerini tanımlar, firmada yine web uygulamasını kullanacak olan sekreterlerin hangi menülere girip giremeyeceğini, hangi işlemleri yapıp yapamayacağını tanımlar. Yetki hiyerarşisinin en üzerinde firma sahibi bulunur. Web uygulaması'nda sol tarafta YÖNETİM başlığı altında yer alanlar sadece firma sahibinin girebileceği bölümdür. Ve ekstra bir yönetim şifresiyle korunur.

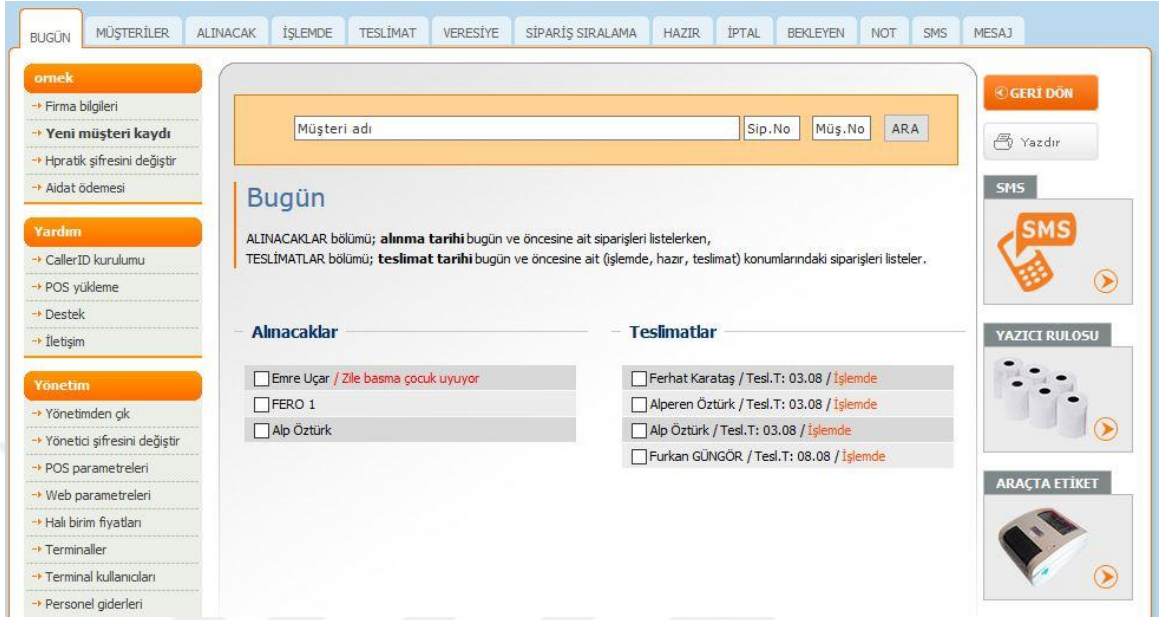
Yapılan uygulamayı birkaç başlıkta inceleyelim;

#### 3.1.Web uygulaması

Web uygulamasında, en sık kullanılan ve siparişlerin takibine ilişkin menüler üst sırada, yönetim ve konfigürasyona ilişkin menüler ise sol tarafta yer almıştır.

##### 3.1.1. Bugün

Firmanın günlük operasyonlarına bir bakıştır. Bugün sayfası, o gün içinde hangi müşteriden ürün alınacağını, hangisine ürün teslim edileceğini listeler. Müşteri isimlerinin yanında (varsa) özel notlar bulunur. “Zile basma çocuk uyuyor”, “Gitmeden yarım saat önce ara” gibi açıklamalardır. Bu da şöförün, ürün alımı/teslimi yaparken dikkat etmesi gereken noktaları tanımlar. Alınacaklar bölümü'nde Alınma Tarihi, Teslimatlar bölümünde ise Teslimat Tarihi işlem zamanı olarak kabul edilmiştir. Şekil 3.1'de Bugün ekranı gösterilmiştir.



Şekil:3.1 – Bugün ekranı

### 3.1.2. Müşteriler

Müşteriler üzerinde aramanın yapılabildiği, firmanın (varsa) alt firmalarına ait müşteri sayılarının görüntülediği alandır. Sayfada AJAX kütüphaneleri kullanılarak ilk 3 karakterden sonra arka planda otomatik tamamlama işlemi (autocomplete) gerçekleştirilmiştir. Böylece ismi yazmaya başladığınızda o isme benzer kayıtlar otomatik olarak altta listelenmektedir. Ek olarak bir dropdownlist yardımıyla; Adı ile, Telefon ile, Müşteri No ile, Sipariş No ile, Adres ile, Mahalle ile, İlçe ile arama opsiyonları eklenmiştir. Şekil 3.2’de Müşteriler ekranı gösterilmiştir.

**Müşteriler**

Toplam müşteri sayınız **13** kişidir. İlçelere göre müşteri sayıları için **tıklayınız**.  
Alt firma müşteri sayıları: EK ÖRNEK: **1**

Adı Soyadı	Müş.No
FERO 1	CUMHURİYET, vedat dalokay MERKEZ Müş.No:181
Furkan GÜNGÖR	ABDULLAHPAŞA, Çiçek sokak no 55 MERKEZ Müş.No:180
Alp Öztürk	ÜNİVERSİTE, Deneme Mh. 1234. Sk. No:18 Kat:1 MERKEZ Müş.No:179
Mali	ALTIN ÇEVRE KÖYÜ, Mali MERKEZ Müş.No:178
Furkan Güngör2	CİRİTTEPE, N.5 MERKEZ Müş.No:177
aaa	ATAŞEHİR, aaa MERKEZ Müş.No:176
Alperen Öztürk	ABDULLAHPAŞA, Uzun adres bilgisi mh. 1234. Sk. Pamukkale/DENİZLİ MERKEZ Müş.No:175
Zeynep	RÜSTEMPAŞA, 3 MERKEZ Müş.No:174
Saygı temizlik	ABDULLAHPAŞA, aaa gğgğg MERKEZ Müş.No:173
Ferhat Karataş	ÇANAKÇA KÖYÜ, GUL SK ALACAKAYA Müş.No:172
Şura Kara	ATAŞEHİR, Anakart sk. n.5 MERKEZ Müş.No:169
Emre Uçar	SÜRSÜRÜ, Teknokent sk. N.5 MERKEZ Müş.No:168
Furkan Güngör	ÇAYDAÇIRA, Cumhuriyet mh. Vedat Dalokay cd. N.6 MERKEZ Müş.No:149

Şekil:3.2 – Müşteriler ekranı

### 3.1.3. Alınacak

Hangi tarihte, hangi müşteriden halı alınacağını listeler. İstenildiğinde güncelleme yapılarak tarihler ötelenebilir. Gidilecek müşterilere istenildiğinde otomatik SMS bildirim yapılarak durum hakkında bilgi verilebilir. Örneğin; fabrikada oluşan bir elektrik ya da su kesintisi sebebiyle 20.08.2019 tarihinde alınacak bir sipariş, toplu SMS atılarak bir sonraki tarihe ertelenebilir. Ya da müşterinin evde olmasını garanti etmek için “Merhaba, gün içinde halılarınızı almak için uğrayacağız, lütfen evde bulunun” gibi uyarılar yapılabilir. SORGULA ve GÜNCELLE panelleri açılıp kapanabilir özelliktedir. Siparişler üzerinde istediğiniz filtreleri uygulayarak sorgulama yapabilir, gelen sonuçlarda düzenlemeler yaparak onları güncelleyebilirsiniz. Firma sahibi, istediği taktirde şöföre vermek üzere tüm sipariş listesini Excel’e aktarabilir, yazdırabilir. Şekil 3.3’de Alınacak ekranı gösterilmiştir.

Şekil:3.3 – Alınacak ekranı

### 3.1.4. İşlemdede

“Alınacak” işlemi eğer başarılı bir şekilde tamamlanmışsa o siparişin durumu otomatik olarak “İşlemdede” konumuna dönüşür. İşlemdede sayfasından, gün içinde (halısı) alınmış tüm siparişleri görüntüleyebilirsiniz. Yine, Sorgula ve Güncelle panelleri yardımıyla bu sipariş üzerinde de filtreler uygulayıp gelen sonuçları düzenleme imkanınız bulunmaktadır. İsminin en sağında köşeli parantez ile belirtilen örn: [174] rakamı, müşteri numarasını gösterir. İsminin başında yer alan 12517/3 formatında ise ilk bölüm sipariş numarasını, ikinci bölüm ise halı adedini gösterir. Şekil 3.4’de İşlemdede ekranı gösterilmiştir.



The screenshot shows the 'İŞLEMDE' (In Progress) screen of a POS system. The top navigation bar includes buttons for 'BUGÜN', 'MÜŞTERİLER', 'ALINACAK', 'İŞLEMDE', 'TESLİMAT', 'VERESİYE', 'SİPARİŞ SIRALAMA', 'HAZIR', 'İPTAL', 'BEKLEYEN', 'NOT', 'SMS', and 'MESAJ'. The left sidebar has three main sections: 'örnek' (Example) with links to 'Firma bilgileri', 'Yeni müşteri kaydı', 'Hpratik şifresini değiştir', and 'Aidat ödemesi'; 'Yardım' (Help) with links to 'CallerID kurulumu', 'POS yükeme', 'Destek', and 'İletişim'; and 'Yönetim' (Management) with links to 'Yönetimden çık', 'Yönetici şifresini değiştir', 'POS parametreleri', 'Web parametreleri', 'Halı birim fiyatları', 'Terminaller', 'Terminal kullanıcıları', and 'Personel giderleri'. The main content area features a search bar for 'Müşteri adı' with fields for 'Sip.No', 'Müş.No', and 'ARA'. Below this is a section titled 'İşlemdekiler' (In Progress Orders) with a 'Listeyi Excel'e aktar' button. It displays a summary: 'Belirtilen tarihte alınan siparişleri gösterir. Kayıt sayısı: 2, Halı adedi: 6, m2: 20,00, Toplam tutar: 185,00'. There are buttons for 'SORGULA' and 'GÜNCELLE'. A filter section allows selecting 'Önceki tarihleri de kapsa' (include previous dates) with a date range of '21.08.2019' and a 'POS listesi' dropdown. The main table has columns for 'Müşteri Adı', 'Alma T.', 'Teslim T.', 'POS', and 'İşlem Z.'. It lists two orders: 1) '3 Zeynep Karataş [174]' with details 'Not : - MERKEZ, RÜSTEMPAŞA / Tutar: 45,00 Battaniye: 3 ad.' and a timestamp of '21.08.19 21.08.19 500000 21.08.2019 13:38:45'. 2) '12517/3 Defne Karataş [176]' with details 'Not : - MERKEZ, ATAŞEHİR / Tutar: 140,00 Makina: 3 ad.' and a timestamp of '21.08.19 29.08.19 500000 21.08.2019 09:22:02'. The right sidebar contains buttons for 'GERİ DÖN', 'Yazdır', 'SMS', 'YAZICI RULOSU', and 'ARAÇTA ETİKET'.

Şekil:3.4 – İşlemde ekranı

### 3.1.5. Teslimat

Ürünü hazırlanmış, artık teslimat tarihi gelmiş siparişlerin listelendiği alandır. Müşteriye, hangi tarihe teslimat sözü verilmişse, sayfa bu duruma ilişkin bir görüntüleme yapar. Gecikmiş siparişlerde renklendirme yaparak firmayı uyarır. Teslimat konumuna gelmiş bir siparişin artık fiyatı belli olmuştur. Tüm siparişlerin toplamını oluşturan istatistiki bilgiler de (m2, adet, tutar vs) sayfanın başına konmuştur. Firma sahibi, hangi mahalleye kaç halı gideceğini SORGULA panelini kullanarak filtreleyebilir. Böylece bir arabaya sığacak siparişleri burada ayarlar. Artan siparişleri başka araçlara yönlendirmek için GÜNCELLE panelini kullanarak POS numarasını günceller. Şekil 3.5’de Teslimat ekranı gösterilmiştir.

The screenshot shows the 'Teslimatlar' (Deliveries) screen. The navigation menu on the left includes 'ornek', 'Yardım', and 'Yönetim' sections. The main content area features a search bar for 'Müşteri adı', 'Sip.No', and 'Müş.No'. Below the search bar, there's a summary of order statistics: 'Sipariş sayısı: 2, Hali adedi: 9, Toplam m2: 0,00, Toplam tutar: 45,00'. A table lists delivery records with columns for 'Müşteri Adı', 'İlçe', 'Mahalle', 'Alınma', 'Teslim', and 'POS'. Two records are visible: one for 'Zeynep Karataş' and another for 'Alperen Öztürk'. The right sidebar contains buttons for 'GERİ DÖN', 'Yazdır', 'SMS', 'YAZICI RULOSU', and 'ARAÇTA ETİKET'.

	Müşteri Adı	İlçe	Mahalle	Alınma	Teslim	POS
<input type="checkbox"/>	/ 3: Zeynep Karataş / Tutar : 45,00 Bahar sk. N.4 [174]	MERKEZ	RÜSTEMPAŞA	21.08.2019	21.08.2019 [İşlemede]	500000
<input type="checkbox"/>	12508 / 6: Alperen Öztürk / Tutar : 0,00 Uzun adres bilgisi m.h. 1234. Sk. Pamukkale/DENİZLİ [175] 08501111111	MERKEZ	ABDULLAHPAŞA	26.07.2019	21.08.2019	500001

Şekil:3.5 – Teslimat ekranı

### 3.1.6. Veresiyeler

Teslimat sonrasında müşteride kalan bakiyelerin listelendiği alandır. Ürün teslimi sonrasında eğer müşteri eksik ödeme yapmışsa bu bölümden izlenir. Böylece günlük olarak hangi müşteride ne kadar bakiyenin kaldığı takip edilebilir. Veresiye borcu olanlara hatırlatma yapmak için tek tuşta SMS gönderilebilir. SMS metni, herkesin borcuna göre otomatik düzenlenir. Eğer firma sahibi, kalan bakiyeyi küçük görürse, bu satırı işaretleyip İNDİRİM olarak kabul edebilir. Bu durumda bakiye sıfırlanmış olur. Filtrele bölümünden ise, istediği ilçe veya mahalleyi seçerek o bölgedeki alacaklarını görebilir. Şekil 3.6'da Veresiye ekranı gösterilmiştir.

BUGÜN MÜŞTERİLER ALINACAK İŞLEMDE TESLİMAT VERESİYE SİPARİŞ SIRALAMA HAZIR İPTAL BEKLEYEN NOT SMS MESAJ

örnek  
→ Firma bilgileri  
→ Yeni müşteri kaydı  
→ Hıpratik şifresini değiştir  
→ Aidat ödemesi

Yardım  
→ CallerID kurulumu  
→ POS yükleme  
→ Destek  
→ İletişim

Yönetim  
→ Yönetimden çık  
→ Yönetici şifresini değiştir  
→ POS parametreleri  
→ Web parametreleri  
→ Hali birim fiyatları  
→ Terminaller  
→ Terminal kullanıcıları  
→ Personel giderleri

Müşteri adı Sip.No Müş.No ARA

Veresiyeler ✕ Listeyi Excel'e aktar >

Toplam kayıt: 2, Toplam Tutar: 21,00

Seçili olanlara SMS gönder Seçili olanları POS'a gönder Seçili olanların Kalan'ını İndirim yap

Filtre : İlçe Mahalle 21.07.2019 21.08.2019 Getir

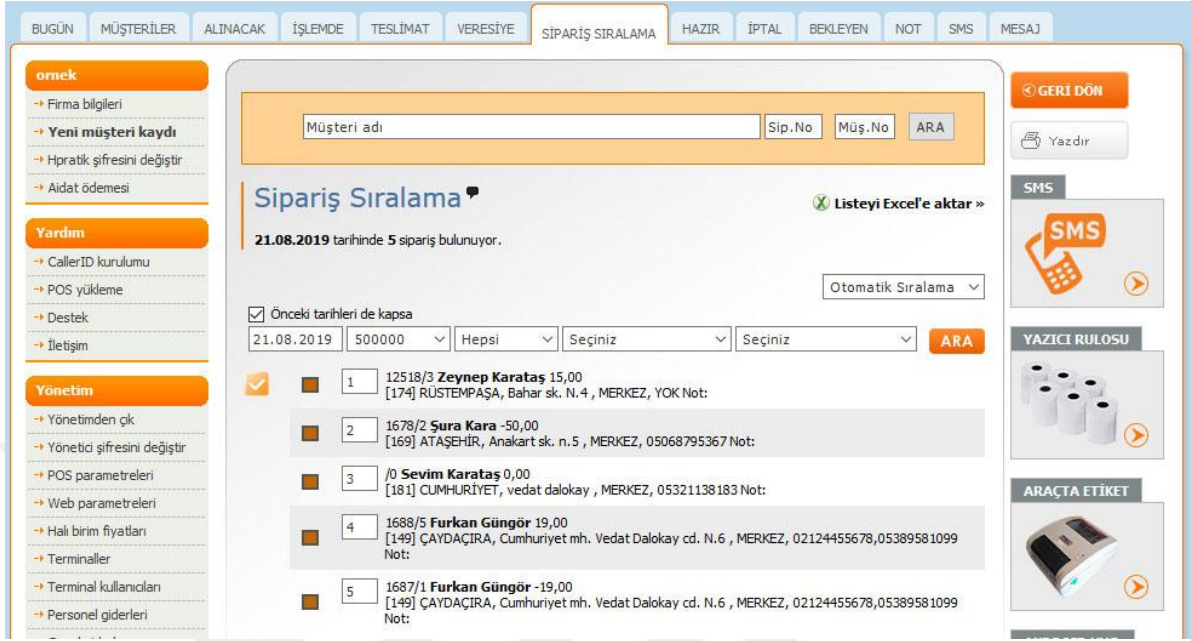
<input type="checkbox"/>	Adı Soyadı	Teslimat	POS	Tutar	Alınan	Kalan
<input type="checkbox"/>	Alperen Öztürk	21.08.2019	500001	96,00	90,00	6,00
<input type="checkbox"/>	Zeynep Karataş	21.08.2019	500000	45,00	30,00	15,00

GERİ DÖN  
Yazdır  
SMS  
YAZICI RULOSU  
ARAÇTA ETİKET

Şekil:3.6 – Veresiye ekranı

### 3.1.7. Sipariş Sıralama

Gün içinde hem müşteriden ürün alma (alınacak), hem de ürün teslim etme (teslimat) sözkonusu olabilir. Tüm bu operasyonları bir sayfada görebileceğiniz ekrandır. Sayfa üzerinde sürükle&bırak metodu ile, ya da satır numaraları vererek sıralama yapmak mümkün. Ek olarak açılan menüden Google Direction API'sini de seçebilirsiniz. Böylece siparişler, fabrikaya göre mesafe sıralamasına sokulur. Ve şoför, fabrikadan çıktıktan sonra bu listeye göre sırayla dolaşarak işini en kısa zamanda tamamlar. Şekil 3.7'de Sipariş Sıralama ekranı gösterilmiştir.



Şekil:3.7 - Sipariş sıralama ekranı

### 3.1.8. Hazırlar

“İşlemede” konumunda yer alan bir sipariş firmada (fabrikada) işlem görüyor anlamındadır. Eğer ürün yıkanıp teslim edilecek hale getirildiyse, müşteri aranır ve teslimat için onay alınır. Eğer müşteri o tarihte evde bulunmuyorsa, (tatile gitmişse vs) siparişin konumu “Hazır” a getirilerek bir köşede bekletilir. Hazır konumundaki siparişlerin özelliği, hiçbir şekilde saha operasyonlarını yapanların kullandığı cihazlara bir emir olarak düşmemesidir. Böylece teslimat tarihi gelmiş olsa bile ne Android ne de POS cihazlarında listelenmeyecek, şöförlerin gereksiz yere müşteriye gitmesine engel olacaktır. Ne zaman müşteri, artık halımı getirebilirsiniz diye firmayı ararsa, HAZIR konumundaki bu sipariş TESLİMAT konumuna getirilerek cihazlara düşmesi sağlanır. Şekil 3.8’de Hazır ekranı gösterilmiştir.

BUGÜN MÜŞTERİLER ALINACAK İŞLEMDE TESLİMAT VERESİYE SİPARİŞ SIRALAMA HAZIR İPTAL BEKLEYEN NOT SMS MESAJ

örnek  
→ Firma bilgileri  
→ Yeni müşteri kaydı  
→ Hpratik şifresini değiştir  
→ Aidat ödemesi

Yardımlar  
→ CallerID kurulumu  
→ POS yükleme  
→ Destek  
→ İletişim

Yönetim  
→ Yönetimden çık  
→ Yönetici şifresini değiştir  
→ POS parametreleri  
→ Web parametreleri  
→ Hali birim fiyatları  
→ Terminaller  
→ Terminal kullanıcıları  
→ Personel aidatları

Müşteri adı Sip.No Müş.No ARA

Hazırlar Listeyi Excel'e aktar »

Kayıt sayısı: 3, Hali adedi: 4, Toplam tutar: 156,00

İlçe Mahalle POS listesi

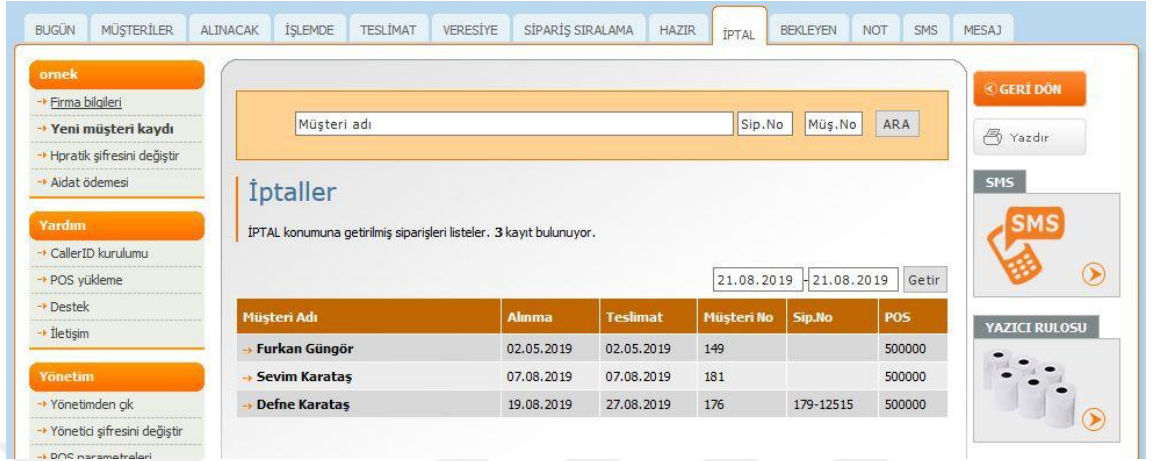
	Müşteri Adı	Mahalle	Alınma	Teslim	POS
<input type="checkbox"/>	12517/3 Defne Karataş [176] Tutar: 140,00 MERKEZ, ATAŞEHİR Gül sk. n.5 , 05555644778 NOT:	ATAŞEHİR	21.08.19	29.08.19	500000
<input type="checkbox"/>	12497/1 Alp Öztürk [179] Tutar: 16,00 MERKEZ, ÜNİVERSİTE Deneme Mh. 1234. Sk. No:18 Kat:1 , 08503047427 NOT:	ÜNİVERSİTE	26.07.19	03.08.19	500001
<input type="checkbox"/>	/0 Emre Uçar [168] Tutar: 0,00 MERKEZ, SÜRSÜRÜ Teknokent sk. N.5 , 05062865339,02125556644 NOT: Zile basma çocuk uyuyor	SÜRSÜRÜ	18.06.19	26.06.19	500001

GERİ DÖN  
Yazdır  
SMS  
YAZICI RULOSU  
ARAÇTA ETİKET

Şekil:3.8 – Hazır ekranı

### 3.1.9. İptaller

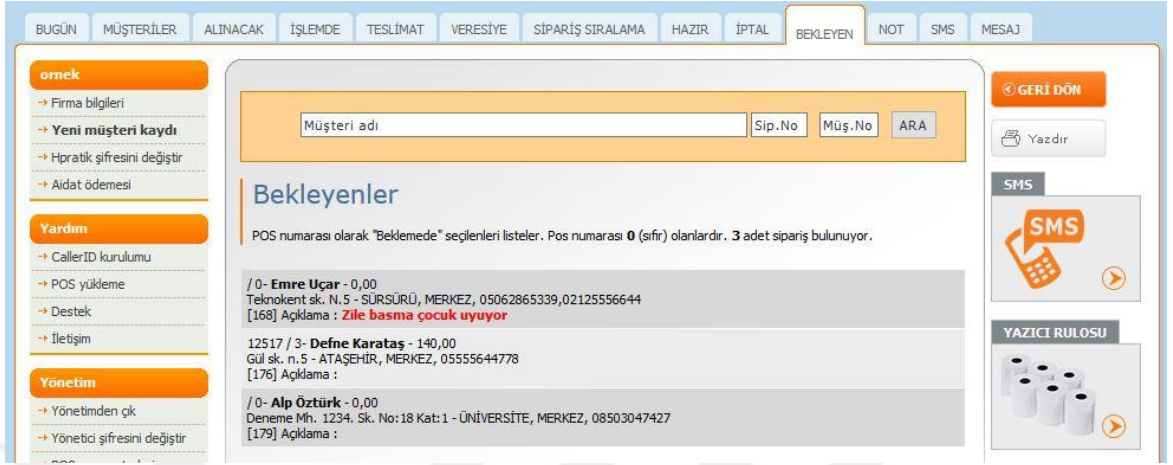
Müşteri firmayı arar ve siparişini verir. Firma, müşteriye ulaşmak üzere servis kaydı (alınacak) girer. Fakat müşteri, firmayı tekrar arayıp ürününü vermekten vazgeçtiğini söylerse, firma, bu siparişin konumunu İptal'e çeker. Böylece sipariş, hiçbir şekilde sahadaki cihazlara bir emir olarak düşmez. Müşterinin geçmişinde iptal konumlu sipariş olarak yer alır. Sonraki siparişlerinde firma, müşterinin performansını, ne kadar sipariş verdiğini, ne kadarını iptal ettiğini takip edebilir. Şekil 3.9'da İptal ekranı gösterilmiştir.



Şekil:3.9 – İptal ekranı

### 3.1.10. Bekleyen

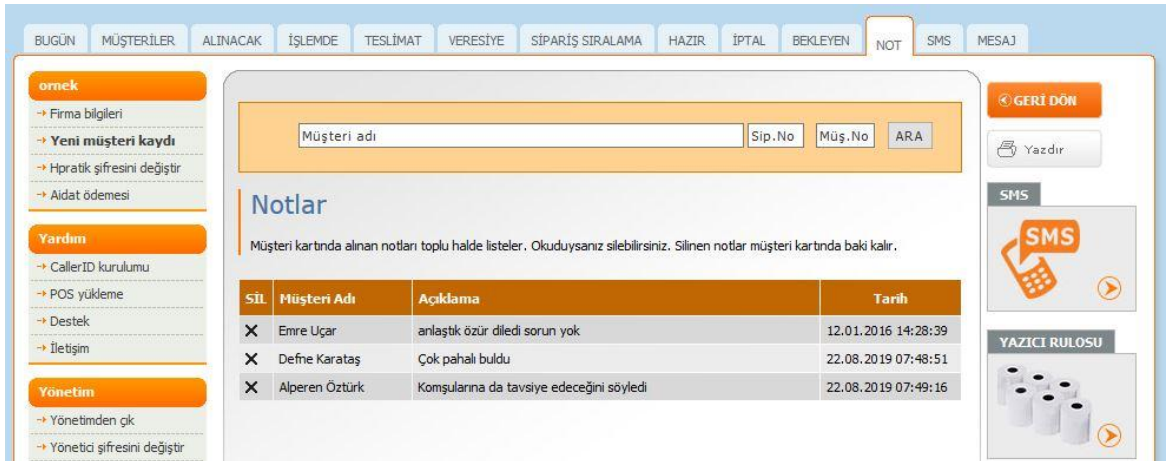
Saha operasyonlarını yapan (ürün alan, ürün teslim eden) firma çalışanlarının birer terminal numarası bulunur. Merkez, saha operasyonlarına yönlendirme yaparken bu terminal numaralarını kullanır. Örneğin, Kadıköy bölgesine 111222 nolu terminal bakarken, Üsküdar bölgesine 111000 nolu terminal bakar. Ve her saha sorumlusu elindeki cihazdan (POS, android vs) bu terminal numarasını kullanarak sisteme giriş sağlar. “Bekleyenler”, aslında 0 numaralı bir terminal numarasıdır. Firma, eğer siparişin hangi saha sorumlusu tarafından yürütüleceğini bilmiyorsa konumunu “bekleyenler” olarak belirleyebilir. Böylece gereksiz yere hiçbir şoförün önüne iş emri olarak düşmez. Hangi araç boştaysa “Bekleme” olan terminal numarası, ilgili terminal numarasına (örn:111222) atanır. Aslında bir nevi HAZIR pozisyonuna benzer. HAZIR pozisyonunda durum bilgisi değiştirilirken BEKLEYEN konumunda terminal numarası değiştirilir. Firma, kendi iş süreçlerine göre hangisini kullanacağına karar verir. Şekil 3.10’da Bekleyen ekranı gösterilmiştir.



Şekil:3.10 – Bekleyen ekranı

### 3.1.11. Not

Firma, her müşterisi için özel notlar alabilir. “Bu müşteriyle bir daha çalışma”, “Firma sahibidir dikkat edin”, “Akrabadır indirim yapın” gibi. Böylece firmaya gelen aramalarda, arayan kişinin bilgileri ekranda görüntülenir. Firma, müşterinin alınan bu notları doğrultusunda aksiyon alır. Tüm müşteriler hakkında alınan notları bir bütün olarak bu sayfada izleyebilir. Şekil 3.11’de Not ekranı gösterilmiştir.



Şekil:3.11 – Not ekranı

### 3.1.12. SMS

Sistem üzerinde 2 tür otomatik SMS bildirim yapılmaktadır. Birincisi, sipariş ilk kez sisteme girdiği durumda oluşur. Ve müşteriye “Siparişiniz sistemlerimize girmiştir, servisimiz gün içinde size ulaşacaktır”. İkincisi, ürün müşteriden alındığında oluşur “Siparişiniz alınmıştır, şu gün teslim edilecektir”. Bu bildirimlerin dışında, firma istediği zaman toplu SMS gönderimleri yaparak kampanyasını genişletebilir.

Toplu SMS gönderimlerde kullanabileceği özellikler şunlardır : Kayıt Tarihine göre gönderim; müşterinin sisteme kayıt edildiği tarihi baz alır. Mesela 2018 yılında kayıt yapan müşteriler gibi. İlçeye göre; sadece belirli bir ilçedeki müşterilerine yapılan gönderimdir. Mahalleye göre; sadece belli bir mahallede yaşayanlara gönderilen SMS’lerdir. Halı Yıkatanlar; son X aydır halısını yıkatmamış müşteriler kampanya yapmak için kullanılır. Örneğin, son 12 aydır halısını yıkatmamış müşteriler.

SMS altyapısı API (Application Programming Interface) kullanılarak tasarlanmıştır. Bu API’ler, BTK’dan (Bilgi Teknolojileri Kurumu) izni almış servis sağlayıcılar tarafından sağlanır. Müşteri, gelen SMS’ten rahatsız oluyorsa, istediği anda 4609’a bir mesaj göndererek sistemden çıkabilir. KVKK (Kişisel Verilerin Korunması Kanunu) kapsamında müşteriye istediği zaman çıkış hakkı tanınmak zorundadır. Şekil 3.12’de SMS ekranı gösterilmiştir.

Şekil:3.12 – SMS ekranı



### 3.1.13. Mesajlar

Uygulamayı tasarlayan ve yöneten yazılım firması ile, firma arasında günlük destek operasyonlarının yürütüldüğü alandır. Firmalar, geliştirme taleplerini, sorularını bu destek hattı üzerinden iletir. Yazılım firması da gün içinde gerekli desteği bu mesajlar üzerinden sağlar. Şekil 3.13’de Mesaj ekranı gösterilmiştir.

Mesaj	Tarih
HALI PRATİK : İyi bayramlar	11.08.2019 01:30:04
HALI PRATİK : BAYRAM TATİLİ	10.08.2019 09:32:21
HALI PRATİK : Sms Yükleme	7.08.2019 10:45:26
HALI PRATİK : Ramazan Bayramı	3.06.2019 12:06:44
HALI PRATİK : Web sitenizden mesaj var !	1.06.2019 13:15:44
HALI PRATİK : Sms	30.05.2019 15:13:09
HALI PRATİK : Webten sipariş	12.04.2019 12:00:31
HALI PRATİK : Bilgilendirme : POS kullanıcıları	3.02.2019 04:15:43
HALI PRATİK : Bilgilendirme : Bakım çalışması	2.02.2019 15:35:31
HALI PRATİK : paas	26.01.2019 23:26:01

Şekil:3.13 – Mesaj ekranı

### 3.2. Android Uygulaması

Android uygulaması, Android Studio IDE’si üzerinde Java ile geliştirildi. Birçok sayfadan oluşan bu uygulama sayesinde, saha görevlileri kendilerine atanan işleri tamamlayabilmekte ve yaptıkları işlemleri anında merkeze iletebilmektedirler.

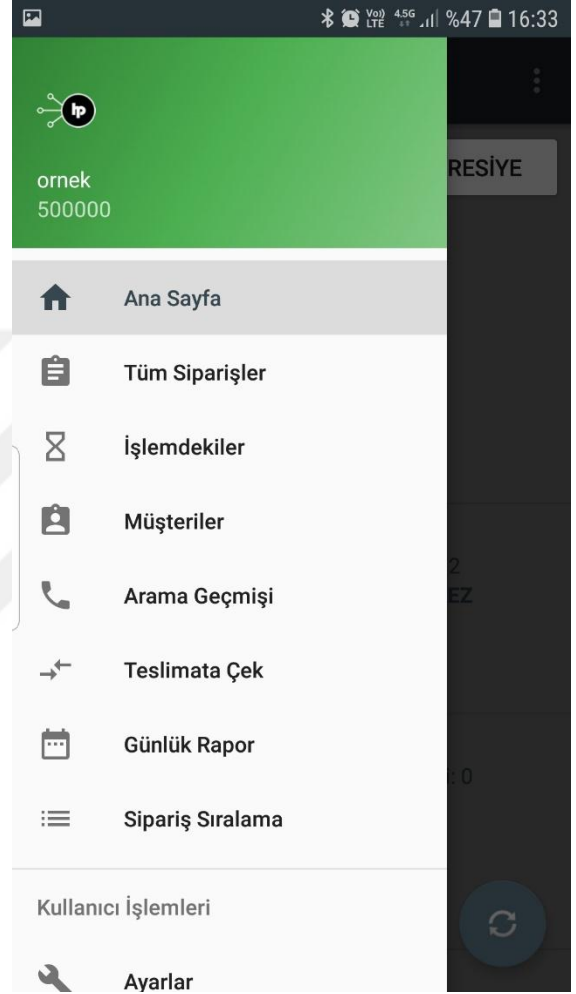


**Şekil 3.14.** Android Uygulaması

Şekil 3.14’te Android kullanıcı arayüzü ve kablosuz yazıcı gösterilmiştir. Play Store üzerinden kurulabilen uygulama, “materyal design” prensipleriyle hazırlandı. Kullanıcı tecrübelerinden (UX) yola çıkarak oldukça basit ve hızlı bir arayüz tasarlandı.



Şekil 3.15. Sipariş Detayları



Şekil 3.16. Uygulama menüsü

Şekil 3.15 ve Şekil 3.16’da Android uygulamasına ait bazı ekranlar gösterilmiştir. Burada, merkezi web uygulamasıyla bir URL üzerinden JSON formatında iletişim kurarak anlık sipariş bilgilerini görüntüleyen sayfa ve uygulamanın diğer bölümlerine geçişi sağlayan ana menümüz yer almaktadır.

### 3.3. POS uygulaması

POS cihazları da sahada kullanılan diđer bir cihazımızdır. Android uygulamasından farkı, tek cihaz üzerinde hem uygulama hem de yazdırma işleminin yapılabilmesidir. Pos cihazları, kendisine atanan siparişleri tamamlayarak sipariş fişlerini basabilmektedirler.



Şekil 3.17. POS Uygulaması

Şekil 3.17’de kullandığımız POS cihazı Nurit 8020 gösterilmiştir. Üzerinde bulunan SIM kart yuvası sayesinde, M2M hatlar kullanılarak GPRS üzerinden internete çıkan bu cihazlar, CRM uygulamasına .NET web servisleri üzerinden bağlanarak işlemlerini tamamlamaktadır.

## 4. ENTEGRASYONLAR

Online sipariş takip otomasyon sistemine entegre ettiğimiz modüller bulunmaktadır. Bu modüller sayesinde yazılım platformundan donanım tarafına erişim sağlanmakta ve bu cihazları da kullanma imkanı oluşmaktadır.

### 4.1 Bluetooth ile çalışan mobil mini yazıcı

Taşıma kolaylığı açısından POS cihazlarına göre daha konforludur. Saha operasyonlarını yapan kişiler kendi cep telefonlarını kullanarak, uygulama üzerinden bu yazıcılardan çıktı alıp bunu müşteriye verebilirler. Şekil 4.1’de Android uygulamasının kullandığı kablosuz yazıcı gösterilmiştir.



Şekil 4.1. Bluetooth yazıcı

### 4.2. Bluetooth ile çalışan mobil barkod yazıcı

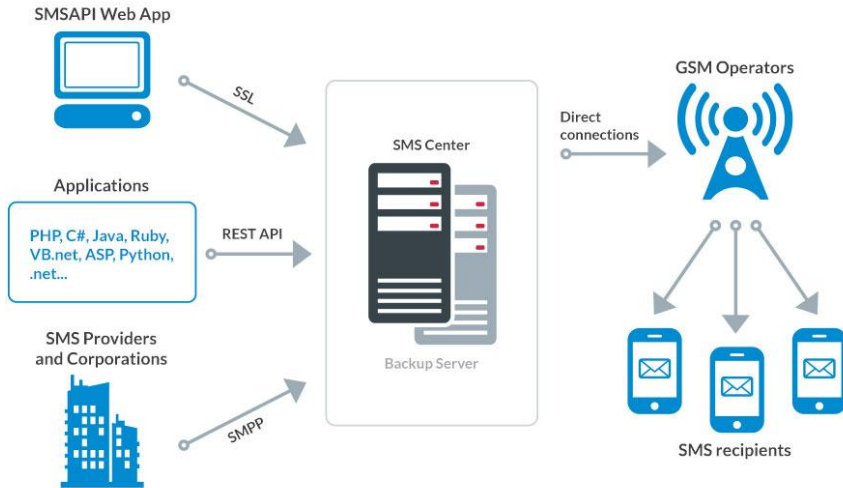
Araçlara konumlandırılan cihazlardır. Aracın çakmaklık adaptöründen beslenir. Android uygulaması üzerinden aldığı komutlarla ürün adedi kadar fiş basar. Basılan bu fişler kumaş etiketlerden oluşur. Etiketler yazdırılırken ribbon denilen bir tür kartuş kullanılır. Kumaş etiketlerde, yikanma durumunda silinme, solma olmaz. Kullanılan ribbon, D110A modelidir. Tişörtlerde gördüğümüz etiketlerde de kullanılmaktadır. Şekil 4.2’de araçlarda kullanılan etiket yazıcısı gösterilmiştir.



Şekil 4.2 Barkod yazıcı

### 4.3. SMS API entegrasyonu

Saha operasyonları esnasında, son kullanıcıyı bilgilendirmek amacıyla, hem POS cihazlarının altyapısında hem de android uygulamasında SMS API'leri kullanılmıştır. Böylece son kullanıcı, siparişinin gelişim sürecini gelen mesajlarla anlık takip edebilmektedir.



Şekil 4.3 SMS API entegrasyonu [8]

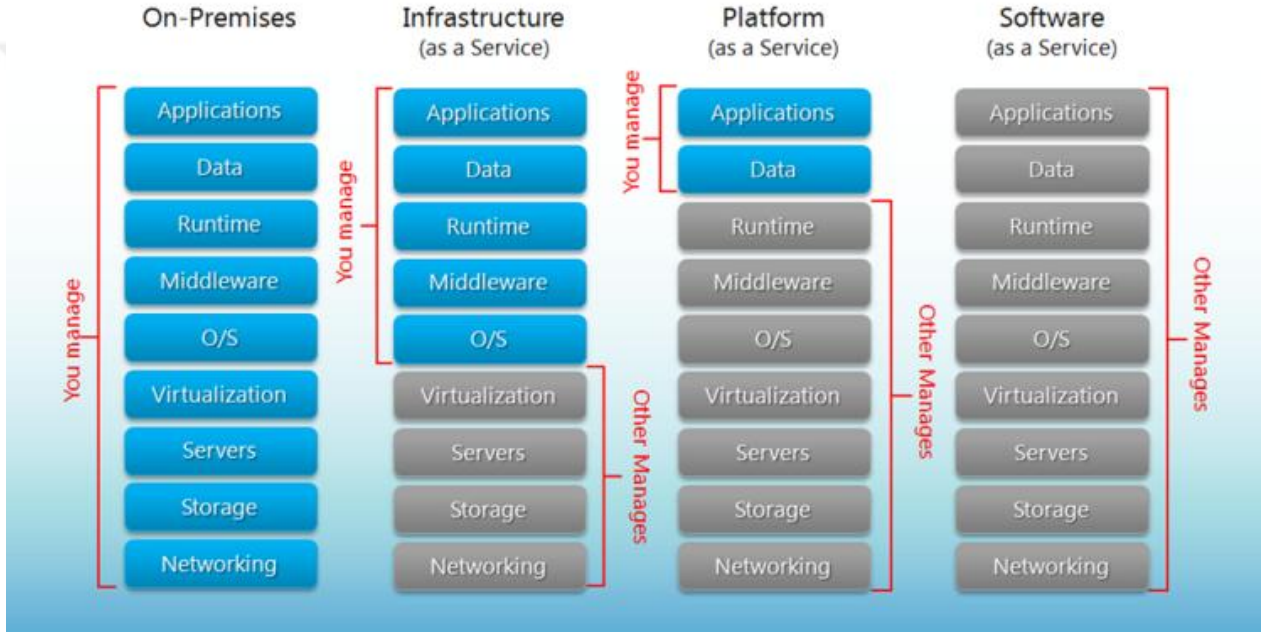
Şekil 4.3'te SMS'lerin nasıl gönderildiği gösterilmiştir. Web uygulaması, kendisine gelen talepleri (request) .net web servisleri yada REST servisleri üzerinden alır. Web uygulaması REST API'lerini kullanarak SMS Center'a mesaj gönderimini gerçekleştirir. SMS Center ise kuyrukta bekleyen bu mesajları GSM operatörüne iletir. Operatör ise kuyrukta bekleyen tüm mesajları sahiplerine dağıtır.



## 5. BULUT HİZMET TÜRLERİ

Bu projede, ölçeklenebilirlik ve yönetim kolaylığı açısından PAAS modelini tercih ettik. IaaS, PaaS ve SaaS farklarını inceleyelim.

### Separation of Responsibilities



Şekil 5.1 Bulut hizmet türleri [10]

Şekil 5.1’de bulut platformların birbirinden farkları gösterilmiştir. Dünyada Cloud platformu üzerinde söz sahibi büyük firmaların sayısı bir elin sayısını geçmez. Bunlar içinden en çok bilinen ve tercih edilenlerini Microsoft (Azure), IBM (SoftLayer, Bluemix), Google (Cloud Platform) ve Amazon (AWS) olarak saymak mümkündür. Hepsi de benzer işleri yaparlar. Amaç, geliştiriciden bağımsız katman yapısında bir platforma sunarak yazılımcının işini kolaylaştırmaktır.

**On-Premises:** Yazılım ve donanımların sorumluluğunun tamamen sizde olduğu yapılardır. Server üzerindeki power, network, yazılım gibi aklınıza ne geliyorsa sizin tarafınızdan



yönetilmesi gerekmektedir. İşletim sisteminin update'leri, veritabanlarının backup ve restore işlemleri vb. tüm sorumluluk sizdedir.

IAAS - Infrastructure As A Service: VM'ler (Virtual Machine) olarak da bilinir. Size bulut sistemler üzerinden harddisk, cpu, ram tanımlaması yapılır. Siz de bu makinalar üzerine işletim sistemini kurar ve projenizi deploy edersiniz. İstedığınız zaman bu kaynakları sanal olarak artırıp azaltabilirsiniz.

PAAS - Platform As A Service: Bizim bu projede kullandığımız modeldir. SQL server, IIS gibi ana yazılımların Microsoft tarafından yönetildiği, tüm update ve kritik güncellemelerin MS tarafından yapıldığı, sizin sadece veritabanınızı restore edip kullanabileceğiniz sistemlerdir. Örneğin, Azure SQL üzerindeki tüm backup işlemleri sizin için otomatik olarak yapılır.

SAAS - Software As A Service: Sadece yazılım tarafının kullanımını size sunar. Size tanımlanan hesap ile giriş yapar ve yazılımı kullanırsınız. Yazılımın server üzerinde konumlandırılması, veritabanı yönetimi gibi işlemleri servis sağlayıcı yapar. PAAS'ın bir ileri versiyonudur. Gmail servisleri, GoogleDocs buna örnek olarak gösterilebilir.

## 6. SONUÇLAR ve TARTIŞMA

Teknolojinin girdiği hemen her yerde zamandan ve maliyetten tasarruf söz konusudur. Saha sipariş takip otomasyonları da bunun örneklerinden sadece birisidir. Firma olarak, sahadaki örneğin 10 ayrı mobil temsilcinizin nerede olduğunu, hangi siparişlerle ilgilenmesi gerektiğini, koordinasyonunu, yük dağıtımını yapabilmemiz oldukça zor bir süreçtir. Bu tür dağıtık yapılarda kullanılacak en iyi çözümlerden birisi de merkezleştirilmiş yazılım projeleri kullanmaktır. Uygulamayı kullanan binlerce istemci olmasına rağmen, merkezde bir kaç server tüm yükü üstlenecek ve firma sahibine gerekli rapor, denetim, yönetim imkanı sunacaktır. Tabi en önemli avantaj da maliyet konusudur. Örneğin, yazılım üzerinde yapılacak bir güzergah belirleme işlemi, sahadaki çalışanların gereksiz dolaşmasının önüne geçecektir. İşte bulut bilişimin gücü de bu tür uygulamalarda kendini gösterecektir.

### 6.1. Bu Çalışmadaki Yenilik

Şimdiye kadar yapılan çalışmalarda sadece web ya da android platformu kullanılırken, biz bu çalışmada, web, android, hardware (bluetooth yazıcılar, POS cihazları) gibi birbirinden farklı platformları birbirleriyle konuşturup aynı projede işlem yapmasını sağladık. Bu mimaride ;

- Asp.net platformunda C# ile kodlanmış bir CRM uygulamasını yayına aldık. Uygulama üzerinde JQuery, AJAX kütüphaneleri aktif bir şekilde kullanıldı.

- .NET web servisleri (SOAP) ile public fonksiyonlar oluşturduk. Böylece, sahada işlem yapan POS cihazları merkezi veritabanı ile entegre edilmiş oldu. POS cihazları üzerindeki uygulama ise C ile geliştirildi. GPRS üzerinden internet erişimin sağlandığı bu cihazlarda, aylık 10-20 megabyte'lık veri transferleriyle tüm siparişleri yönetmek mümkün oldu. Cihazların internet erişimi M2M (Machine to Machine) SIM kartları ile sağlandı.

- Hazırladığımız REST servisleri ile, android platformu için JSON çıktıları verdik. Böylece android uygulamalarının CRM ile entegrasyonu tamamlanmış oldu. Android uygulaması, Android Studio üzerinde Java ile geliştirildi. Materyal Design prensiplerini kullandığımız UI ile son kullanıcıya keyifli bir kullanım sunduk.

- Mobil bluetooth yazıcılar üzerinden, android ile yapılan işlemlerden sonra fiş basılabilmektedir. Böylece müşteriye, yapılan işlemlerden sonra bir makbuz verilebilir.

- Bluetooth barkod yazıcılar üzerinden, işlem sonrası, kumaş etiketler basıldı. Bu etiketler, halıların üzerine zımbalandıktan sonra ürün teslim edilinceye kadarki süreçte takibi kolaylaştırmıştır.



## KAYNAKLAR

- [1] Améndola, F. and Favre, L. "Adapting CRM Systems for Mobile Platforms: An MDA Perspective," *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Honolulu, HI, 2013, pp. 323-328.doi: 10.1109/SNPD.2013.25
- [2] Bhalerao, S. Bhosale, B. Bhangale, G. Nigade, K. Nitturkar, P. "Cloud Based Sales and CRM Application for Builders A Web Enabled and Android CRM Tool" Shubhada Bhalerao et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 490-491
- [3] Bhargave, A. Jadhav, N. Joshi, A. Oke, P. Prof. Mr. Lahane, S. R. "Digital Ordering System for Restaurant Using Android" International Journal of Scientific and Research Publications, Volume 3, Issue 4, April 2013 1 ISSN 2250-3153
- [4] Singhal, M. Shukla, A. "Implementation of Location based Services in Android using GPS and Web Services" IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012 ISSN (Online): 1694-0814
- [5] Wagh, K. S. and Thool, R. C. Article: "Web Service Provisioning on Android Mobile Host". International Journal of Computer Applications 81(14):5-11, November 2013
- [6] <http://www.visualcapitalist.com/happens-internet-minute-2017/>
- [7] Jazayeri, M. (2007). Some Trends in Web Application Development. 199 - 213. 10.1109/FOSE.2007.26.
- [8] <https://www.bulksmsgateway.in/blog/bulks-sms-api-integration-in-php-8>
- [9] <http://www.sys.com.tr/tr/2016/02/05/bulut-bilisim-uygulamalari/>
- [10] <https://medium.com/cloud-and-servers/bulut-hizmetlerinde-iaas-paas-saas-nedir-3bb5620fd357>

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Adı Soyadı : Ferhat Karataş  
Doğum Yeri ve Tarihi : İstanbul / 1976

### Eğitim Durumu

Lisans Öğrenimi : İstanbul Üniversitesi / Siyasal Bilgiler Fakültesi - Maliye  
Yüksek Lisans Öğrenimi : Fırat Üniversitesi / Biyoteknoloji  
Bildiği Yabancı Diller : İngilizce  
Bilimsel Faaliyetleri : Karataş, F., & Korkmaz, S. A. (2018). Big Data: controlling fraud by using machine learning libraries on Spark. Int J Appl Math Electron Comput, 6(1), 1-5.

### İş Deneyimi

Stajlar : Bilgipark Yazılım Ltd. Şti, Fırat Teknokent / ELAZIĞ  
Projeler : Online sipariş takip otomasyonu  
Çalıştığı Kurumlar : Bilgeadam, Multinet, Aras Kargo

### İletişim

E-Posta Adresi : ferhat@bilgipark.com  
Tarih : 27.08.2019