



**ADAPTİF BAĞLAM BİLİNÇLİ MOBİL UYGULAMA GELİŞTİRME**

**Muhammed İkbâl BİLGİLİ**

**YÜKSEK LİSANS TEZİ  
BİLGİSAYAR BİLİMLERİ ANABİLİM DALI**

**GAZİ ÜNİVERSİTESİ  
BİLİŞİM ENSTİTÜSÜ**

**TEMMUZ 2014**

Muhammed İkbal BİLGİLİ tarafından hazırlanan “Adaptif Bağlam Bilinçli Mobil Uygulama Geliştirme” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ / OY ÇOKLUĞU ile Gazi Üniversitesi Bilgisayar Bilimleri Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

**Danışman:** Prof. Dr. M. Ali AKCAYOL

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum .....

**Başkan:** Doç. Dr. Bünyamin CİYLAN

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum .....

**Üye:** Öğr. Gör. Dr. Murat HACİÖMEROĞLU

Bilgisayar Mühendisliği, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/onaylamıyorum .....

Tez Savunma Tarihi: 21 / 07 / 2014

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....  
Doç. Dr. Nurettin TOPALOĞLU  
Bilişim Enstitüsü Müdürü

## ETİK BEYAN

Gazi Üniversitesi Bilişim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Muhammed İkbâl BİLGİLİ

21/07/2014



ADAPTİF BAĞLAM BİLİNÇLİ MOBİL UYGULAMA GELİŞTİRME  
(Yüksek Lisans Tezi)

Muhammed İkbal BİLGİLİ

GAZİ ÜNİVERSİTESİ  
BİLİŞİM ENSTİTÜSÜ

Temmuz 2014

ÖZET

Günümüzde bilgisayar uygulamaları, geleneksel masaüstü uygulamalarından, önce web uygulamalarına sonrasında ise mobil uygulamalara kaymıştır. Bunun en önemli sebebi mobil cihazların adı üstünde taşınabilir yani her an her yerde kullanıcının yanında ve erişilebilir olmasıdır. Ayrıca mobil cihazlar işlemci hızı ve kapasite bakımından masaüstü bilgisayarlara oldukça yaklaşmışlardır. Bu nedenle her geçen gün mobil cihazlara sahip kullanıcı sayısı artmakta sonucunda da mobil cihazlar için geliştirilen uygulamalar da bir o kadar önem kazanmaktadır. Bu uygulamaların daha etkin olabilmesi ve kullanıcının işini kolaylaştırabilmesi için mobil cihazın, kendisine sahip olan kullanıcının ve fiziksel ortamın durumundan yani bağlamdan haberdar olması gerekmektedir. Bu çalışmada Bağlam diye ele alınan kavram ve mobil uygulama geliştirmede ne şekilde kullanılacağı örneklerle ve ayrıntılarıyla ele alınmıştır. Ayrıca yapay sinir ağları kullanarak öğrenen ve bulunduğu ortama uyarlanabilir, kendini adapte edebilir, adaptif bağlam bilinçli bir mobil uygulama geliştirilmiştir.

Bilim Kodu : 902.1.014

Anahtar Kelimeler : Bağlam-bilinçli, mobil uygulama, adaptif, uyarlanabilir bağlam bilinçlilik, yapay sinir ağları

Sayfa Adedi : 99

Danışman : Prof. Dr. M. Ali AKCAYOL

# DEVELOPING AN ADAPTIVE CONTEXT AWARE MOBILE APPLICATION

(M. Sc. Thesis)

Muhammed İkbâl BİLGİLİ

GAZİ UNIVERSITY  
INFORMATICS INSTITUTE

July 2014

## ABSTRACT

The computer applications have previously changed from the traditional desktop applications into the web applications and then into the mobile ones nowadays. The most important reason of this is that the mobile devices, as the name implies, are removable, that is, they are by the side of the user at any moment or anywhere and accessible. Besides, mobile devices have caught up with desktop computers in terms of processor speed and capacity. Therefore, the number of users having mobile devices has been rising day by day, and consequently, the applications developed for these devices have gained importance as much again. The mobile device needs to be aware of the situation of the user owning itself and the physical environment, in other words, the context, so that, the application can be more effective and ease the user's work. In this study, the term which is dealt as the Context and how it will be used in developing mobile applications have been discussed in details and with examples. In addition, a mobile application which learns by using artificial neural network, is applicable to its location, can adapt itself and is adaptive context-aware has been developed.

Science Code : 902.1.014

Key Words : Context-aware, mobile application, adaptive context awareness, artificial neural network

Page Number : 99

Supervisor : Prof. Dr. M. Ali AKCAYOL

## TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren, önemli tecrübelerinden faydalandığım danışmanım Prof. Dr. M. Ali AKCAYOL'a, yüksek lisans yapmam konusunda beni sürekli teşvik eden babama ve manevi destekleriyle beni hiçbir zaman yalnız bırakmayan eşime ve aileme teşekkürü bir borç bilirim.



## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER .....	vii
ÇİZELGELERİN LİSTESİ .....	ix
ŞEKİLLERİN LİSTESİ .....	x
RESİMLERİN LİSTESİ .....	xi
SİMGELER VE KISALTMALAR.....	xiii
1. GİRİŞ .....	1
2. MOBİL CİHAZLAR VE MOBİL UYGULAMALAR.....	7
2.1. Mobil Cihazlar.....	7
2.1.1. Akıllı telefonlar ve tabletler.....	7
2.1.2. Mobil cihaz pazar payları .....	8
2.2. Mobil Uygulamalar .....	10
3. ADAPTİF SİSTEMLER.....	11
3.1. Adaptif Bağlam Bilinçli Uygulamalar .....	12
4. BAĞLAM.....	13
4.1. Bağlam Bilinçli Bilişim.....	17
4.2. Bağlam Bilinçli Sistemlerde Yer Bağlamının Tespiti.....	20
4.2.1. Dış alanda yer tespiti .....	20
4.2.2. İç alanda yer tespiti.....	21
4.2.3. Hibrit yer tespit sistemleri .....	22
5. BAĞLAM MODELLEME.....	23
5.1. Anahtar Değer Modelleri.....	24
5.2. İşaretleme Şema Modelleri.....	24
5.3. Grafik Modeller.....	24
5.4. Mantıksal Modeller .....	25
5.5. Nesne Yönelimli Modeller .....	25
5.6. Ontoloji Yönelimli Modeller.....	25

6. ADAPTİF BAĞLAM BİLİNÇLİ MOBİL UYGULAMA (MOBİL ADAPTİF ASİSTAN-MAA) .....	27
6.1. MAA Sistem Modülleri.....	27
6.1.1. Mobil istemci modülü.....	27
6.1.2. Öğrenme için yapay sinir ağları modülü .....	29
6.1.3. Web sunucusu uygulama modülü.....	29
6.1.4. Güvenlik modülü .....	29
6.2. MAA Sistem Mimarisi .....	31
6.3. MAA Mobil Uygulama Verileri.....	33
6.4. MAA Mobil Uygulama İşleyiş Özellikleri.....	35
6.5. MAA Mobil İstemci ve Web Sunucu Uygulamaları.....	36
6.5.1. MAA mobil istemci uygulaması.....	36
6.5.2. Web sunucu uygulaması.....	69
6.6. Yapay Sinir Ağları ile Öğrenme.....	77
6.6.1. Yapay sinir ağları mimarisi .....	78
6.6.2. Yapay sinir ağları eğitimi ve öğrenmesi.....	80
6.6.3. Yapay sinir ağları yapısı .....	80
6.6.4. Yapay sinir ağları MAA öneri sistemi.....	82
7. DENEYSEL SONUÇLAR.....	83
7.1. Sistemin Yer Bağlamını Kullanarak Adaptif Olması.....	83
7.2. Sistemin Yapay Sinir Ağları Kullanarak ve Öğrenerek Adaptif Olması .....	85
8. SONUÇ VE ÖNERİLER.....	91
KAYNAKLAR .....	93
ÖZGEÇMİŞ .....	99

## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 2.1. 2013 yılı akıllı telefon işletim sistemleri pazar payları.....	8
Çizelge 2.2. 2013 yılı tüm cihazlar pazar payları ve 2014, 2015 tahminleri .....	9
Çizelge 4.1. Farklı bağlam tipleri için kullanılan sensör örnekleri .....	17
Çizelge 5.1. Bağlam modelleme kategorileri.....	23
Çizelge 6.1. ANN öğrenmede kullanılan öznelikler tablosu .....	33
Çizelge 6.2. MAA kullanılan tablolar ve özellikleri .....	82
Çizelge 6.3. Öğrenme parametrelerinin değişimine göre öğrenme sonucu .....	88
Çizelge 6.4. Öğrenme parametrelerinin azaltılmasıyla öğrenme sonucu.....	88
Çizelge 6.5. Öğrenme parametrelerinin artırılmasıyla öğrenme sonucu .....	89

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Bilgisayar, akıllı telefon ve tablet küresel kullanım oranları.....	9
Şekil 3.1. Adaptif sistemler.....	11
Şekil 3.2. Bağlam-bilinçli, adaptif bağlam bilinçli ve geleneksel adaptif sistemlerin alt küme ilişkisi.....	12
Şekil 4.1. Bağlam sınıflandırılması (mekansal, zamansal, fiziksel, sosyal, çevresel, kimlik) .....	14
Şekil 4.2. Geleneksel bilgisayar sistemleri .....	18
Şekil 4.3. Bağlam bilinçli sistemler .....	19
Şekil 6.1. MAA modülleri ve arasındaki ilişkiler .....	30
Şekil 6.2. Üç katmanlı mimari .....	32
Şekil 6.3. MAA sistem mimarisi.....	33
Şekil 6.4. Uygulama verileri veri tabanı tablo ilişki diagramı.....	35
Şekil 6.5. Yapay nöron sinir hücresi yapısı .....	79
Şekil 6.6. Yapay sinir ağları yapısı .....	81

## RESİMLERİN LİSTESİ

<b>Resim</b>	<b>Sayfa</b>
Resim 4.1. AndroSensor uygulaması ile Samsung P3100 tablet üzerindeki sensörlerin anlık görüntüsü ve ulaşılan bağlamlara ilişkin veriler .....	16
Resim 4.2. AndroSensor uygulaması ile Samsung S4 akıllı telefon üzerindeki sensörlerin anlık görüntüsü ve ulaşılan bağlamlara ilişkin veriler .....	16
Resim 6.1. Samsung Galaxy Tab2 P3100 3G tablet.....	28
Resim 6.2. Eclipse IDE.....	37
Resim 6.3. MAA uygulamasının çalıştırılması .....	38
Resim 6.4. MAA uygulaması karşılama ekranı .....	39
Resim 6.5. MAA uygulamasının sisteme giriş ekranı .....	39
Resim 6.6. MAA uygulaması yeni kayıt ekranı.....	40
Resim 6.7. MAA uygulaması program seçme ekranı .....	43
Resim 6.8. MAA uygulaması öneri yer bilgisi seçme ekranı .....	44
Resim 6.9. MAA uygulaması ana menü ekranı .....	45
Resim 6.10. MAA uygulaması profil bilgilerini düzelt ekranı.....	46
Resim 6.11. MAA uygulaması program tanımla ekranı.....	48
Resim 6.12. MAA uygulaması program göster-tercihleri göster ekranı .....	49
Resim 6.13. MAA uygulaması seçilen yer etkinlikler ekranı.....	50
Resim 6.14. MAA uygulaması seçilen tercihleri göster ekranı.....	51
Resim 6.15. MAA uygulaması ilgi yeri sorgula (gps) ekranı.....	52
Resim 6.16. MAA uygulaması ilgi yeri sorgula (gps) – haritada göster ekranı .....	53
Resim 6.17. MAA uygulaması ilgi yeri sorgula (gps) – yer için yeni etkinlik ekle ekranı .....	54
Resim 6.18. MAA uygulaması ilgi yeri sorgulama (il ilçe) ekranı .....	55
Resim 6.19. MAA uygulaması ANN önerileri getir ekranı.....	56

<b>Resim</b>	<b>Sayfa</b>
Resim 6.20. MAA uygulaması konum uyumsuzluk hali.....	57
Resim 6.21. MAA uygulaması ANN önerileri getir ekranı.....	58
Resim 6.22. MAA uygulaması programları takvimde göster ekranı .....	59
Resim 6.23. MAA uygulaması takvim etkinlik ekleme.....	60
Resim 6.24. MAA uygulaması profil sabitleri göster ekranı.....	61
Resim 6.25. MAA uygulaması öğrenmede bu parametreleri kullan seçimi .....	62
Resim 6.26. MAA uygulaması öneri alınacak program görüntüle/değiřtir ekranı .....	63
Resim 6.27. MAA uygulaması öneri hatırlatma servisi ANN kapalı ve açık .....	68
Resim 6.28. MAA uygulaması öneri hatırlatma servisi puanlama.....	69
Resim 6.29. Netbeans IDE.....	70
Resim 6.30. Biyolojik nöron sinir hücresi yapısı.....	79
Resim 7.1. Yer tespiti işlemleri .....	83
Resim 7.2. GPS network yer tespiti ve öneri işlemi .....	84
Resim 7.3. 30000 düğüm ve 0,0703 hata oranıyla öğrenme sonucu öneriler .....	86
Resim 7.4. 13745 düğüm ve 0,0199 hata oranıyla öğrenme sonucu öneriler .....	87
Resim 7.5. 20849 düğüm ve 0,005 hata oranıyla öğrenme sonucu öneriler .....	90

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

### Simgeler

### Açıklama

**KPa**

Kilo Pascal

**Ghz**

Giga Hertz

**m/s<sup>2</sup>**

Metre / saniye kare

### Kısaltmalar

### Açıklama

**3D-ID**

3D-visualization and branding IDentity

**ANN**

Artificial Neural Network

**API**

Application Programming Interface

**APK**

Android Application Package File

**AVD**

Android Virtual Device

**CSMA/CA**

Carrier Sense Multiple Access with Collision Avoidance

**DVC**

Digital Video Camera

**ER**

Entity - Relationship

**GNP**

Global Network Positioning

**GPS**

Global Positioning System

**GSM**

Global System for Mobile

**HTTP**

HyperText Transfer Protocol

**HTTPS**

HyperText Transfer Protocol Secure

**IEEE**

Institute of Electrical and Electronics Engineers

**IR**

Infrared

**iOS**

iPhone Operating System

**JAX-RS**

Java API for XML REST Services

**JSON**

JavaScript Object Notation

**LAN**

Local Area Network

**Kısaltmalar****Açıklama****MAA**

Mobil Adaptive Assistant

**NFC**

Near Field Communication

**ORM**

Object Relational Mapping

**RDF**

Resource Description Framework

**RDFS**

Resource Description Framework Schema

**REST**

Representational State Transfer

**RESTful WS**

Representational State Transfer Web Service

**SOA**

Service-Oriented Architecture

**UML**

Unified Modeling Language

**URI**

Uniform Resource Identifier

**W3C**

World Wide Web Consortium

**WiFi**

Wireless Fidelity

**XML**

Extensible Markup Language



## 1. GİRİŞ

İnsanoğlunun varoluşundan günümüze kadar her daim, bilgiye ulaşma ve bilgiyi paylaşma ihtiyacı olmuştur. Bu nedenle bilgiye ulaşmak ve bilgiyi paylaşmak için her zaman en hızlı, en ucuz ve en etkili yolu araştırmış, çalışmalarını bu doğrultuda yoğunlaştırmıştır. Tabii ki ulaşılması ve paylaşılması istenen bilginin öncelikle ham olan ve işlenmemiş veriden bilgiye dönüştürülmesi, yani karar verici için anlamlı ve kullanışlı hale getirilmesi gerekmektedir [1]. İhtiyacı olan kişiye, ham ve işlenmemiş olan verinin hiçbir yararı bulunmamaktadır, bundan dolayı verinin bilgiye dönüştürülebilmesi ve bu işlemin daha etkin, hızlı, doğru ve kullanılabilir olması için bilgi sistemlerine ihtiyaç duyulmuştur. Günümüzde bilginin bu denli çeşitli ve çok olmasından dolayı amaca ve ihtiyaca yönelik bilgiye hızlı bir şekilde ve her yerden ulaşmak önem arz etmektedir. Bunun sonucunda bu şekilde bir bilgiye ulaşma ihtiyacı bilgisayar destekli bilgi sistemlerinin gelişmesini kaçınılmaz kılmıştır.

Bilgisayar destekli bilgi sistemleri, ihtiyacı hız ve doğruluk yönünden karşılmasına karşın, ilk olarak 1980 yılında kullanıcıların, kendi bilgisayarlarını üzerlerinde taşıyacakları ve başka aygıtlarla da bağlantı sağlayabilecekleri ve bu şekilde bilgiye her an ve her yerden en doğru şekilde ulaşabilecekleri fikriyle mobil bilişim kavramı ortaya çıkmıştır. Mobil Bilişim, bilgisayar uygulamalarındaki kullanıcıların, donanımın, verinin ve yazılımların taşınabilirliği ile ilgilidir. Weiser (1999), *ubiquitous computing* kavramıyla incelediği mobil bilişimi “bilgisayarlara fiziksel çevrelerini kullandırırken bu işlemleri kullanıcıdan etkili bir şekilde gizleyerek bilgisayar kullanımını daha etkin sağlayabilme metodu” şeklinde tanımlamıştır [2].

Mobil bilişimde yer alan ve her geçen gün her yaşta birey tarafından daha da yaygın olarak kullanılan mobil cihazlar, daha etkin kullanılabilmeleri için aynı zamanda mobil uygulamalara olan ihtiyacı da beraberinde getirmiştir. Nasıl bir uygulamanın başarısı eldeki kaynakları en etkin bir şekilde kullanarak kullanıcının işini kolaylaştırabilmesi ve daha fazla kitleye ulaşabilmesi ile ölçülüyorsa, günümüzde mobil cihazlar için de bu durum benzerdir. Hatta mobil cihazlar, kullanıcının yanında her yerde ve her zaman bulunduğu için cihazın kullanıcı ile paylaşacağı durum, yani etkileşim daha fazla olacaktır. Bu da mobil uygulamalara olan ilgiyi ve mobil uygulamaların başarısını artırmaktadır.

Mobil uygulama, eğer kullanıcının bulunduğu ortamın farkında olarak kullanıcıya hissettirmeden kullanıcının işini kolaylaştırabiliyorsa, mobil cihazın daha verimli kullanılmasını sağlayabilmektedir. Bu tip sistemler, bağlam bilinçli sistemler olarak adlandırılmaktadır. Yani mobil cihaz kullanıcısının bulunduğu bağlamın farkında olan ve bu bağlama göre kullanıcıya hissettirmeden kullanıcının işini kolaylaştıran uygulamalar, bağlam bilinçli uygulamalardır.

Bağlam bilinçli uygulamalar, kullanıcının geçmiş tercihlerinden, yine kullanıcının işini kolaylaştırmak için geleceğe dair sonuçlar çıkarabildiğinde adaptif olma özelliği kazanmakta, yani kendini bulunduğu bağlama uyarlayabilir hale getirebildiğinde adaptif bağlam bilinçli uygulama adını almaktadır.

### Motivasyon

Günümüzde mobil cihazların oldukça yaygınlaşmasıyla mobil uygulamalara olan ilgi ve ihtiyaç artmıştır. Mobil cihazlar, donanım yönüyle farklı sensörlere sahip olması dolayısıyla bağlam bilinçli ve kullanıcının yanında her daim buldukları için daha fazla etkileşim içinde olarak bulunulan bağlamın farkında olabilmektedirler. Mobil cihazlar için geliştirilen mobil uygulamalar ise, bağlam bilgilerini kullanıcıya hissettirmeden kullandıklarında daha etkin ve başarılı bir sonuca ulaşmaktadırlar. Bu nedenle bu çalışmanın amacı, daha çok kişiye ulaşabilmek ve daha çok kişinin kullanacağı ve yararına olabilecek bağlam bilinçli mobil bir uygulama geliştirmektir. Ayrıca bağlam bilinçli mobil bir uygulama geliştirirken, aynı zamanda uygulamanın adaptif özelliğe sahip olmasının sağlanmasıyla, kullanıcıya, bulunulan bağlama göre değişen dinamik sonuçlar göstermenin yanında, geçmiş kullanıcı tercihlerine ve bulunulan bağlama göre sistemin, öğrenerek kullanıcının seçebileceği önerileri ön görebilmesini sağlamaktır.

### Bağlam Bilinçli Uygulamalar

Son on yılda bilişim sektörü özellikle mobil kullanıcılara odaklanmış durumdadır. Giderek gelişen farklı kapasitelere sahip mobil cihazlar, artık birçok farklı kablosuz bağlantı formlarını da desteklemektedir. (Örneğin Wifi, IR, GSM, IEEE 802.11, Bluetooth). Bu da geleneksel masaüstü bilişimden, daha fazla bağlantı formlarını destekleyen ve daha fazla bağlama ait verileri elde edip kullanıcının işini kolaylaştıran mobil cihazların kullanıldığı

mobil bilişime yüksek oranda bir kayma olduğunu göstermektedir. Özellikle mobil bilişim cihazlarının sahip olduğu tüm çevre birimleri, bağlamın değişim durumuna karşı oldukça hassastır. Bu değişim durumu kullanıcıların hareketlerine göre bulunduğu çevrelerin değişmesiyle ilgilidir. Ayrıca mobil cihazların davranışlarının, değişen bu çevreye ve kaynaklara göre uyum sağlayabilmesi gerekmektedir. Bu da mobil bilişimin araştırma alanına girmektedir. Bu kapsamdan bakıldığında iki konu ortaya çıkmaktadır. Bu konular, adaptif mobil uygulamalar ve bağlam bilinçli sistemlerin/uygulamaların geliştirilmesi konularıdır.

Adaptasyon kelimesi 90'ların ilk yarısından itibaren önemli bir araştırma konusu haline gelmiştir [3, 4]. Bu dönemde özellikle mobil uygulama dünyasında adaptasyon için gerekli olan birçok bilgi elde edilmiştir ve adaptasyon çoklu ortam uygulamalarından ziyade ağ bağlantısı ve güç desteği için uygulanmıştır.

Bağlam bilinçli uygulamaların ilk doğuşu, Weiser (1993) tarafından gelecekteki mobil uygulamalar için ihtiyaç duyulan bağlam bilinçli özelliği *ubiquitous (her yerde olan) bilişim* kavramının açıklamasına dayanmaktadır [5]. Sonrasında ise mobil bilişimde kullanıcının çevresini algılayacak ve ona göre davranışı şekillendirecek uygulamaların ortaya çıkmasıyla yeni ilgiler doğmuş ve yeni teknolojilere ilgi artmıştır. (portatif bilgisayarlar, çevreye duyarlı teknolojiler vs.)

Mobil bilişimin gelişim sürecinde daha yüksek seviyede farklı mobilite durumuyla uğraşan birbirinden bağımsız iki paralel yol ortaya çıkmıştır. Bunlardan ilki mobil teknolojilerin doğal kısıtlarının (ağ kalitesi, sınırlı pil ömrü vs.) üstesinden gelebilmek için adaptif mobil uygulamaların geliştirilmesiyle ilgilenmiştir. İkincisi ise yön etkileşimi olmaksızın kullanıcının bulunduğu bağlama adapte olabilecek ve çevresel görüntüleme teknolojilerini kullanarak mobil kullanıcının işlerini kolaylaştıracak uygulamalar geliştirilmesiyle ilgilenmiştir. İkinci çalışmanın esas amacı dış değişimlere göre davranışını değiştirebilen ve kullanıcının bağlamına göre uyum sağlayabilen mobil uygulamalar geliştirmektir. Günümüzdeki araştırmalarda ele alınan esas konu, gelecekteki mobil sistemlerin bu iki özelliği birleştireceği, yani kısıtlı kaynakları iyileştirirken, dış kaynaklara uyum sağlayabilen mobil uygulamaların önem kazanacağı yönündedir. Bu yüzden günümüzde geliştirilen her mobil uygulamanın, değişen sistem ya da bağlamsal değişkenlerle

tetiklenen uyarlamaları gerçekleştirmek ve bunları gerçekleştirirken de kısıtlı sistem kaynaklarına göre daha az kaynak tüketmek için geliştirilmesi gerekmektedir.

Mobil uygulamalarla sistemin bulunduğu bağlam hakkında bilgiye ulaşmak kolay olmasına rağmen uygulama programcıları için en büyük sorun o bilginin etkin bir şekilde nasıl kullanılacağıdır. Schilit, Adams ve Want (1994) bağlam bilinçli uygulamaları şu şekilde kategorize etmişlerdir [6]:

- Yakınlık Seçimi
- Yakında bulunan nesnelerin belirlenebildiği kullanıcı ara yüzü tekniği ya da yakındaki nesnelerin daha kolay seçilebilmesi
- Otomatik bağlamsal düzeltimi
- Yeni bileşenlerin eklenebildiği var olanların kaldırılabilirdiği ya da değişen bağlama göre bağlantıların değiştirilebildiği işlemlere imkân tanınması
- Bağlamsal bilgi ve komutlar
- Bulunulan bağlama göre farklı sonuçların üretilebilmesi
- Bağlam tetiklemeli eylemler
- Bağlam bilinçli sistemlerin nasıl adapte olacağına basit IF-THEN kurallarıyla belirlenebilmesi

Yukarıdaki sınıflandırma bağlam bilinçli uygulamaların sınıflarını belirlerken Pascoe, Ryan ve Morse (1997) bağlam bilinçli uygulama özelliklerini aşamalı olarak şu şekilde tanımlamışlardır [7]:

- Bağlam algılama
- Bağlam uyarlama
- Bağlam kaynaklarının keşfi
- Bağlamı zenginleştirme: kullanıcı bağlamı ile eldeki dijital verileri birleştirme

Dey (2001) ise bağlam bilinçli sistem özelliklerini özetle şu şekilde tanımlamıştır [8]:

- Bulunulan bağlama göre kullanıcıya bilgi ya da servislerin sunulması  
Örnek: Zaman bilgisi ve kullanıcının yanındaki restoranlar
- Belirli bir bağlam için herhangi bir servisin otomatik çalıştırılması

Örnek: Telefonun kullanıcı için alarmı otomatik olarak ayarlaması

- Bağlam ile ilgili bilgileri sonraki bir kullanım için etiketleme

Örnek: Telefon tarafından arama kaydının tutulması

Kotz ve Guanling (2000) ise bağlam bilinçli sistemleri iki kategoride incelemişlerdir [9]:

- Aktif Bağlam Bilinçlilik: Uygulamanın otomatik olarak davranışını değiştirerek kendi belirlediği ve bulunduğu bağlama uyum sağlamasıdır.
- Pasif Bağlam Bilinçlilik: Uygulamanın kullanıcıya yeni ya da güncellenmiş bağlamı sunması ya da var olan bağlamı daha sonra kullanabilmek için saklamasıdır.

Aktif bağlam bilinçlilik, pasif bağlam bilinçliliğe göre mobil cihazlar üzerinde yeni uygulamalara ve daha fazla altyapı desteğine ihtiyaç duymakta ve eldeki kısıtlı kaynakları daha fazla tüketmektedirler.

Literatürde tanımlanan bağlam bilinçli sistem özelliklerinden hareketle, etkin ve kullanışlı bağlam bilinçli bir sistem modelinin özellikleri ve geliştirilen uygulama aşamaları özetle aşağıdaki gibi sıralanabilir.

#### Bağlam Bilinçli Sistem için Hazırlanan Modelin Özellikleri

- a) Bağlamı kolay bir şekilde anlayabilmek ve kullanabilmek için bağlam bilgilerini toplayan ara yüzleri kullanabilme
  - b) Bağlama ait bilgiye ulaşma hassasiyetini artırma ve en doğru bilgiye ulaşabilme
  - c) Bağlam verisinin toplaması, işlenmesi ve veriyi kullanarak sonuç üretilmesiyle uygulamada kullanılacak veri kümesinin belirlenmesi
  - d) Bulunulan bağlamın modellenmesi, sunulması ve saklanması
  - e) Diğer bağlam bilinçli uygulamalara kolay haberleşebilir şekilde uyum sağlayabilme
  - f) Günümüz popüler servisleriyle kolay haberleşebilme
- Örneğin: servis mimarisiyle geliştirilen (SOA) servislere uygulamayı entegre edebilme
- g) Veri Güvenliği ve doğruluğunu sağlayabilme
  - h) Veriye ulaşmak için 2G,3G ya da WiFi bağlantı tiplerinden hangisine en iyi, en etkin ve en ucuz şekilde ulaşılabilirse bunu belirleyip kullanabilme
  - i) Kullanıcı ara yüzlerini daha kullanıcı dostu ve anlaşılabilir bir şekilde sunabilme

- j) Farklı platformlara kolay taşınabilme ve uyarlanabilme
- k) Tüm bu işlemleri yaparken kısıtlı sistem kaynaklarını çok etkin ve verimli bir şekilde kullanabilme

#### Bağlam Bilinçli Uygulama Geliştirme Aşamaları

- a) Cihazın sahip olduğu donanım araçlarıyla cihazın bulunduğu bağlama ait bilgilerin toplanması (sensörler, Kamera, GPS vs.)
- b) Cihazda koşan sistemin çalışma zamanında bağlama ait verilerin toplanmasında yardımcı donanımsal araçlara ulaşımın kolaylaştırılması
- c) Bağlamı zenginleştirecek ve karar vermeyi kolaylaştıracak varsa dijital verilerin toplanması ve bu verilere daha az kaynak tüketerek erişim için gerekli servislerin yazılması
- d) Sistem içerisinde yer alan hangi bilginin ne şekilde kullanılacağına ve sunulacağına karar verilmesi
- e) Kullanıcıya bildiri yapılacak bilginin kullanımının ne şekilde ve hangi durumlarda tetikleneceğine karar verilmesi

## 2. MOBİL CİHAZLAR VE MOBİL UYGULAMALAR

### 2.1. Mobil Cihazlar

Mobil, kelime anlamı olarak hareketli, taşınabilir anlamına gelmektedir. Mobil cihazlar ise küçük, elle tutulabilir, dokunmatik bir ekranı ya da küçük bir klavyesi olan, kullanıcının kolaylıkla yanında taşıyabileceği cihazların genel adıdır. Günümüzde amaca göre farklılık gösteren çok çeşitli mobil cihazlar bulunmaktadır. Bu cihazların çoğunluğunun ortak özelliği belirli bir işletim sistemlerinin olması, farklı amaçlar için yazılmış mobil uygulamaların üzerlerinde çalıştırılabilmesi, Wifi, bluetooth, GPS, NFC gibi birçok bağlantı özelliklerini desteklemesi ve bu bağlantı özelliklerini kullanarak diğer cihazlarla haberleşebilmesidir. Güç kaynağı olarak genellikle lithium pil kullanan bu cihazlar dış dünyayla sensörler, kamera, mikrofon ve özel amaçlı üretilmiş donanım arayüzlerini kullanarak etkileşim kurabilirler. Mobil cihaz tipleri, en genel anlamda aşağıdaki şekilde beş temel başlıkta sıralanabilir.

- Mobil Bilgisayarlar (tabletler, laptoplar, netbooklar, hesap makinası, akıllı saatler, taşınabilir, kişisel dijital asistan pda, taşınabilir çoklu ortam yürütücüleri, ultramobiller vs. bu kapsamda değerlendirilebilir)
- Dijital Kameralar (DVC)
- Mobil Telefon
- Çağrı Cihazı
- Navigasyon Cihazları

Bu çalışmada incelenen sistemlerde kullanılan mobil cihazlar, tüm mobil cihazlar içinde kaynak yönünden en gelişmişleri olan mobil bilgisayarlar ve mobil telefonlardır. Günümüzde mobil cihazlar içinde en yaygın kullanılanı, mobil telefon kategorisinde sayabileceğimiz akıllı telefonlardır.

#### 2.1.1. Akıllı telefonlar ve tabletler

Akıllı telefonlar, sıradan cep telefonlarına göre çok daha ileri düzeyde donanım kaynaklarına, işletim sistemi özelliklerine sahip olan ve çok fazla uygulamayı çalıştırabilen, hem Wifi bağlantı hem de sahip olduğu GSM sağlayıcının sağladığı hücresel

iletişim özelliklerini kullanarak internete girebilen taşınabilir cihazlardır. Günümüzde mobil cihazlar arasında en yaygın kullanılanları iPhone'lar ve Android akıllı telefonlardır.

### 2.1.2. Mobil cihaz pazar payları

Mobil Cihazlar Pazar payları bakımından, cihaz tipi, kullandıkları işletim sistemleri ve üreticisi bakımından değerlendirildiğinde Amerikan Gartner Araştırma şirketinin Şubat 2014 araştırma raporuna göre, Çizelge 2.1.'de görüldüğü üzere mobil cihazlar içinde en yaygın kullanılan akıllı telefonların işletim sistemi piyasasında Android işletim sisteminin payı %78.4'le büyük bir paya sahiptir. iPhone'un sahip olduğu iOS' un ise piyasa payı % 15,6'dır [10] . Tüm cihazlarda değerlendirildiğinde Android işletim sisteminin pazar payı % 37,9 ile yine ilk sıradadır.

Çizelge 2.1. 2013 yılı akıllı telefon işletim sistemleri pazar payları

<b>İşletim Sistemi</b>	<b>2013 (Birim X 1000)</b>	<b>2013 Pazar Payı (%)</b>	<b>2012 (Birim X 1000)</b>	<b>2012 Pazar Payı (%)</b>
Android	758 719,9	78,4	451 621,0	66,4
iOS	150 785,9	15,6	130 133,2	19,1
Microsoft	30 842,9	3,2	16 940,7	2,5
BlackBerry	18 605,9	1,9	34 210,3	5,0
Diğer İşletim Sistemleri	8 821,2	0,9	47 203,0	6,9
<b>Toplam</b>	<b>967 775,8</b>	<b>100,0</b>	<b>680 108,2</b>	<b>100,0</b>

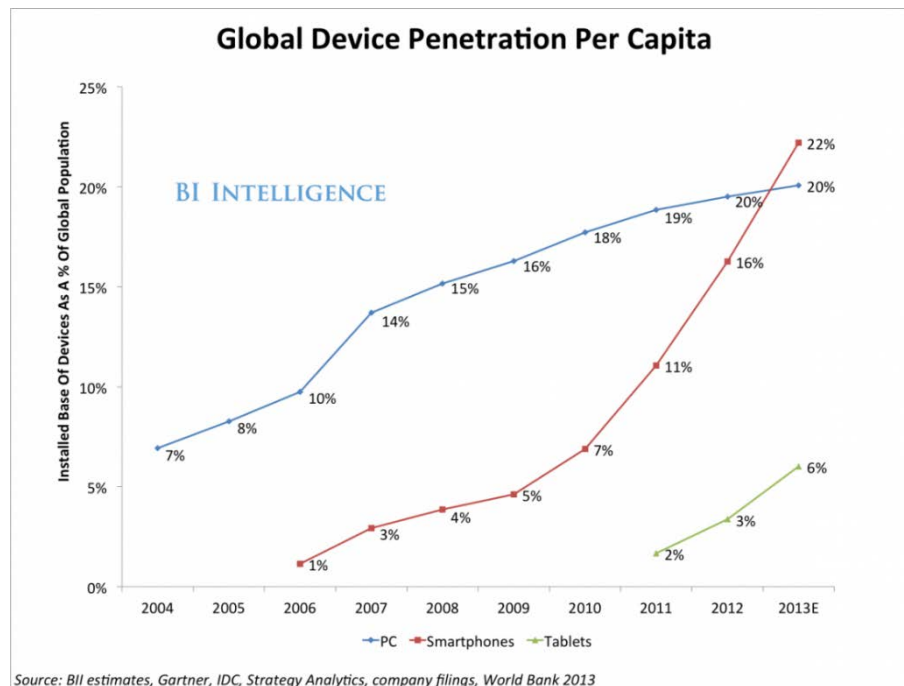
Ayrıca kişisel bilgisayarlar, tabletler, diğer ultramobiller ve mobil telefonlar kategorilerinde mobil cihazlar değerlendirildiğinde tüm cihazlarda mobil telefonlar 1,8 milyar kullanıcıyla %77,9'la en büyük pazar payına sahiptir [11].



Çizelge 2.2. 2013 yılı tüm cihazlar pazar payları ve 2014, 2015 tahminleri

Cihaz Tipi	2013	2014	2015
Geleneksel Bilgisayarlar	296,1	276,7	263,0
Tabletler	195,4	270,7	349,1
Mobil Telefonlar	1 807,0	1 895,1	1 952,9
Diğer Ultramobiller	21,1	37,2	62,0
<b>Toplam</b>	<b>2 319,6</b>	<b>2 479,8</b>	<b>2 627,0</b>

En genel sonuç olarak, günümüzde mobil cihazlar oldukça yaygın kullanılmakta ve Şekil 2.1.'den görüldüğü üzere dünya üzerinde her 5 kişiden birinin akıllı telefonu her 17 kişiden birinin tableti bulunmaktadır [12]. Tüm mobil cihazlar arasında en yaygın olarak Android işletim sistemli cihazlar bulunmakla birlikte cihaz üreticisi olarak akıllı telefonlarda ilk sırayı %31,5 oranla Samsung almaktadır [10].



Şekil 2.1. Bilgisayar, akıllı telefon ve tablet küresel kullanım oranları [12]

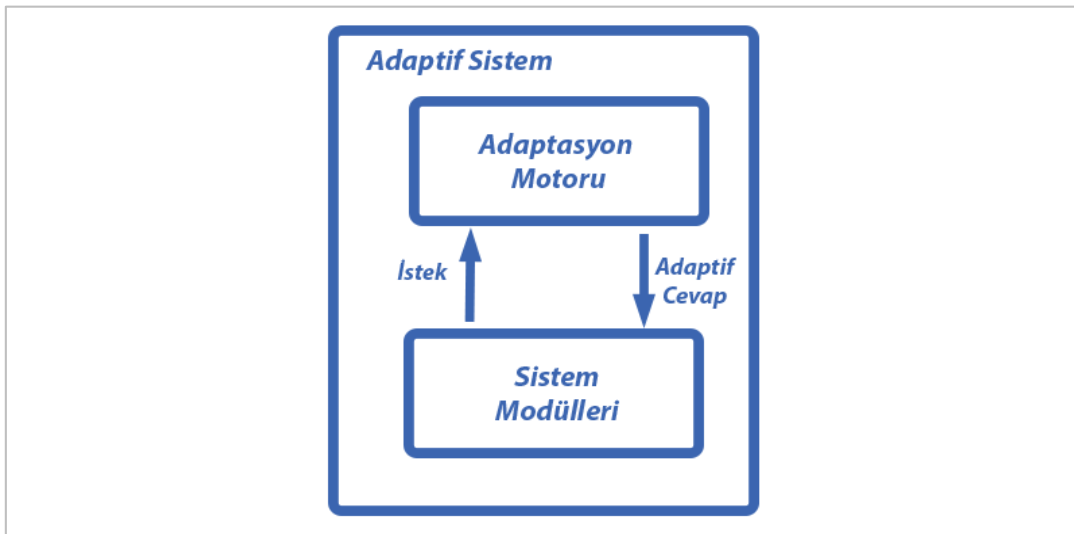
## 2.2. Mobil Uygulamalar

İnternet teknolojisi neredeyse hayatın her anında kullanılan ve vazgeçilemeyen bir ihtiyaç haline gelmiştir. Günümüzde aklımıza gelen her şey internet üzerinden aranabilmekte ve bu yolla bilgiye zahmetsizce ulaşılabilmektedir. Alışveriş yapmak istendiğinde, eğlence, eğitim ve aklımıza gelen daha nice şeyler için internet vazgeçilmez bir bilgi kaynağıdır. Bunun sonucunda internetin hayatımızın bir vazgeçilmezi olmasıyla, bu teknolojiyi yanımızda taşıma ihtiyacı doğmuştur. Bu ihtiyacı günümüzde sağlayan araçlar mobil cihazlar olmakla birlikte mobil cihazların daha etkin ve verimli amacına uygun kullanılabilmesi için tasarlanmış yazılımlara “*Mobil Uygulamalar*” denir. Mobil uygulamalar, kullanılan cihazın işletim sistemine göre farklı kaynaklardan elde edilebilmektedir. Örneğin; Apple iOS işletim sistemi kullanan iPhone, iPad gibi cihazlar için Apple App Store’dan, Android işletim sistemi kullanan cihazlar için Google Play Store’dan, diğer mobil işletim sistemi kullanan cihazlar için ise Windows Phone Store ve BlackBerry App World gibi mağazalardan mobil uygulamalar ücretli ya da ücretsiz edinilebilmektedir. Ücretsiz uygulamalar direkt mobil cihaza indirilip kurulabildiği gibi, ücretli uygulamalar için ilgili ücret kredi kartıyla ödendikten sonra uygulama cihaza indirilip kurulabilmektedir. Mobil uygulamalar buldukları marketlerde farklı kategorilere bölünmüşlerdir. Bu da uygulamaların amaçlarına uygun olarak sınıflandırılmasını sağlamakta ve kullanıcı için bulmayı kolaylaştırmaktadır.

Günümüzde normal web sitelerin de mobil versiyonları veya mobil uygulamaları vardır. Mobil uygulamalar normal web sitelerinden daha az veri kullanarak aynı veriye ulaşabilmemizi sağlamakta bu da günümüzde sınırlı internet kullanan mobil cihazlar için daha cazip hale gelmektedir. Mobil uygulamalar sayesinde sosyal paylaşımlar, bulunulan yer, bir tıkla paylaşabilmekte, istenirse mobil cihazın donanımında başka amaç için tasarlanmış bir özelliğin (örneğin kamera flaşı) farklı başka bir amaç için (örneğin yol aydınlatması) kullanılması mümkün olabilmektedir. Mobil cihazları cazip kılan ve mobil cihazlara ilgiyi artıran en önemli faktör, üzerlerinde çalışan mobil uygulamaların çeşitliliği, sayısı ve bunun sonucunda bilgisayarda çalışan ve kullanıcının hoşuna giden uygulamanın, mobil uygulamasının da olmasıyla kullanıcının her daim yanında olması ve hayatını daha da kolaylaştırabilmesidir.

### 3. ADAPTİF SİSTEMLER

Doğada canlılar arasında milyonlarca yıldır süre gelen ve acil ve normal durumlarda bulunduğu doğal olarak ortama adapte olabilme özelliği, yazılım tasarımcılarının dikkatini çekmiş ve bu özelliğin bilgisayar sistemlerine de uyarlanabileceği fikri doğmuştur [13]. Adaptif sistem konusu 60'lı yılların ortalarından beri detaylıca araştırılan ele alınan bir konudur. Geçen son 10 yılda ise bu konu daha fazla önem kazanmış ve adapte olabilme ilgili olan sistemler ve teknolojiler daha popüler hale gelmiştir. Genel olarak adaptif sistemler, çevresine duyarlı, çevresindeki durumlara göre davranışını, vereceği cevabı değiştirebilen sistemlerdir. Tüm adaptif sistemler sistemin davranışını bulunduğu ortama göre değiştirebilecek bir iç, yardımcı mekanizmaya sahiptirler. Bu yardımcı mekanizma, sistemin bulunduğu dış ortamdan gelen uyarıları, çevreden gelen her türlü bilgiyi alarak vereceği cevabı buna göre şekillendirebilmekte ve bulunduğu ortama adapte olarak çıktı üretebilmektedir. Bu tip sistemler kullanıcıdan karmaşıklığı tamamen gizleyerek çevresel uyarılara göre kendini ayarlayabilmeli, düzenleyebilmeli, koruyabilmeli ve yenileyebilmelidir. Bu özellikleri kullanan bir sistem kullanıcının işini daha da kolaylaştıracak ve daha etkin bir kullanıcı etkileşimi sağlayacaktır. Adaptif sistemler, kullanıcı sistem etkileşimini indirgeyerek gereksinimleri ve eğilimleri belirleyen kapalı döngü sisteminden (örneğin kendini sistemde gelen devamlı değişikliklere göre çalışma zamanında değiştirebilmesi) oluşur. Adaptif sistemin kavraması kullanıcının gereksinimlerine, sistem ve çevresel özelliklere bağlıdır [14].

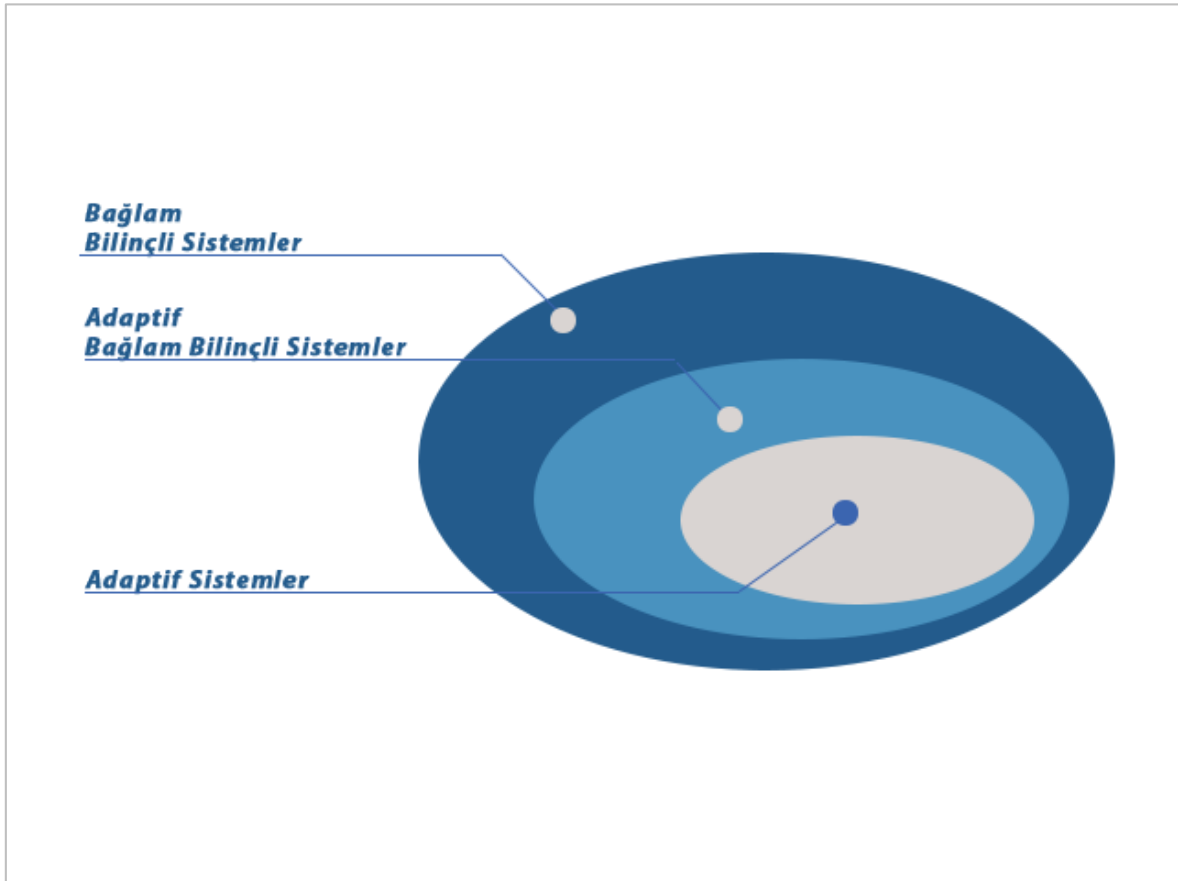


Şekil 3.1. Adaptif sistemler

Şekil 3.1.'de görüldüğü gibi adaptif sistemde üretilen adaptif cevap, adaptasyon motoru diye gösterilen ve yukarıda da bahsedilen sistemin davranışını ortama göre belirleyen ve düzenleyen yardımcı mekanizmaya göre şekillendikten sonra belirlenir.

### 3.1. Adaptif Bağlam Bilinçli Uygulamalar

Adaptif bağlam bilinçli uygulamalar uygulama bağlamının değişmesine göre davranışlarını değiştirebilen (adapte olabilen) uygulamalardır. Bu tanıma göre adaptif olarak tanımlanan bir uygulama gerçekte bağlam bilinçli bir uygulamadır. Ve adaptif bağlam bilinçli uygulamalar kümesi bağlam bilinçli uygulamaların bir alt kümesi ve geleneksel bağlam bilinçli uygulamaların bir üst kümesidir. Ancak tüm bağlam bilinçli uygulamalar adaptif sistemler diye tanımlanan kavramın içine girmez. Bağlam-bilinçli, adaptif bağlam bilinçli ve geleneksel adaptif sistemlerin alt küme ilişkisi Şekil 3.2.'de gösterilmiştir.



Şekil 3.2. Bağlam-bilinçli, adaptif bağlam bilinçli ve geleneksel adaptif sistemlerin alt küme ilişkisi

## 4. BAĞLAM

Bağlam, ilk kez Schilit ve Theimer (1994) tarafından “*kullanıcının yerine, çevresindeki insanlar ve nesnelere ve bu nesnelereki tüm değişimlere göre adapta olabilen uygulama*” tanımıyla gündeme gelmiş ve ancak bu tanımda bağlam, yer ve yakınlık olarak sınırlı bir şekilde tanımlanmıştır [15]. Brown (1996) bağlamı, “*yer ve kullanıcının çevresindeki kişiler, zaman, mevsim, sıcaklık vs.*” şeklinde tanımlarken [16], Pascoe ve diğerleri (1997), “*belirli bir varlığa olan ilginin fiziksel ve kavramsal durumlarının bir alt kümesi*” şeklinde tanımlamıştır [7]. Dey (1998) yaptığı çalışmasıyla bağlam tanımına kullanıcının duygusal durumu, dikkati, yeri, yönü, zaman ve çevresindeki insanlar ve nesnelere de katmış [17], Kotz ve Guanling (2001) ise bağlamı, “*Hem uygulamanın davranışlarını hem de uygulamanın hangi olayları oluşturduğunu belirleyen çevresel durum ve ayarlar kümesi*” şeklinde tanımlamışlardır [9]. Schmidt ve Laerhoven (2001) bağlamı “*çevresini, durumunu ve az ölçüde de yerini kapsayan kullanıcı ve bilgi sisteminin durumu hakkındaki bilgi*” şeklinde tanımlamışlardır [18].

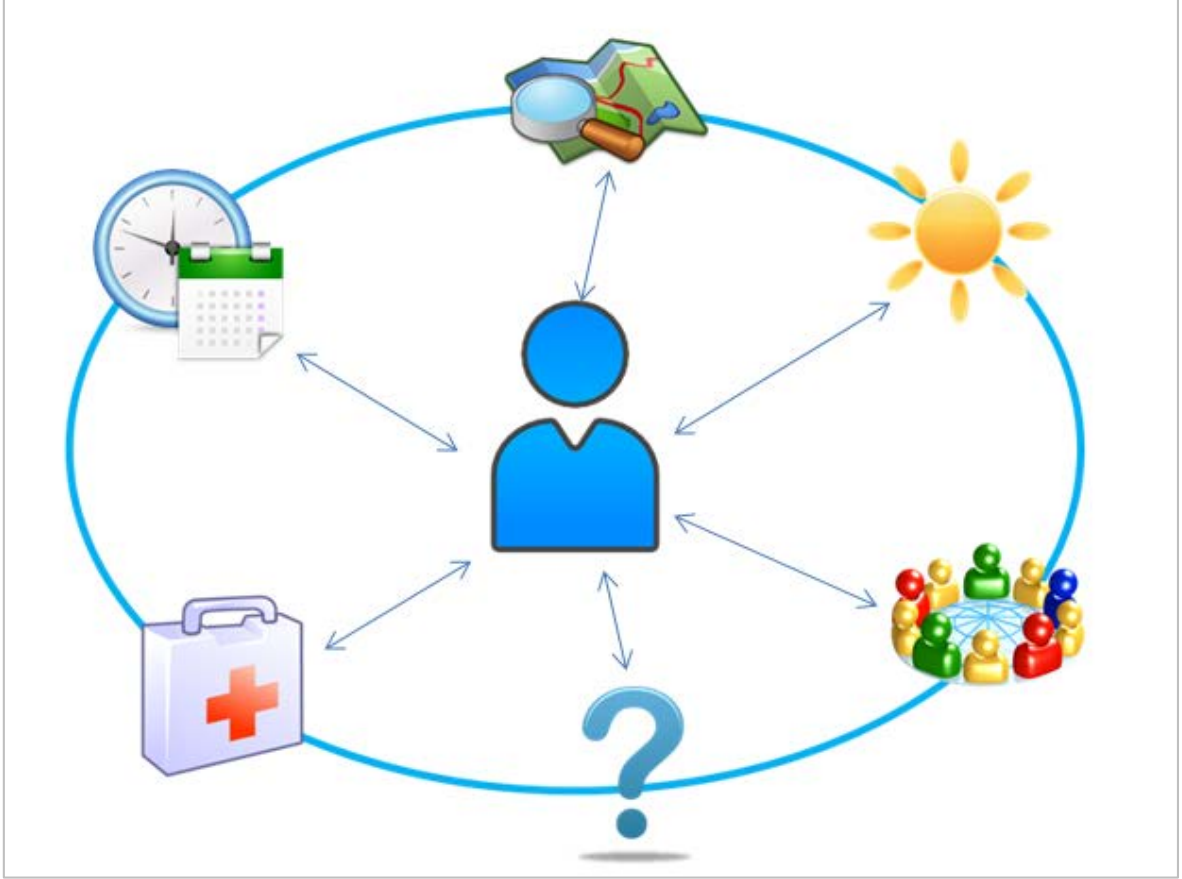
Ancak bağlamı günümüzde de birçok çalışmada da kabul gören şekliyle Dey (2001) “*Varlığın durumunu karakterize eden herhangi bir bilgidir. Varlık uygulama ve kullanıcı arasındaki etkileşime göre bir kişi, yer ya da nesne olabilir.*” şeklinde tanımlanmıştır [8]. Geçmiş tanımlar da göz önünde tutularak en genel kapsamda bağlam, bilgisayar destekli bilgi sistemleriyle bu sistemlerin kullanıcıları arasındaki her türlü durum etkileşimlerini betimleyen her türlü bilgi olarak tanımlanabilir ve bu her türlü bilgi içerisinde kullanıcının kimliği, mekânsal, zamansal, çevresel, sosyal ve fiziksel durumu da sayılabilir.

Bağlam için verilebilecek bazı örnekler şu şekilde sıralanabilir;

1. Kullanıcının kimliği
2. Kullanıcının ve yakınında bulunan insanların ve nesnelere buldukları yerler ve yönelimleri vs.
3. Zaman, tarih, mevsim, vs.
4. Işık, gürültü, basınç, ses, sıcaklık vs.
5. Kullanıcının veya bilgi sisteminin bulunduğu sosyal çevre, kullanıcının yakınındaki insanlar, aktiviteler, takvim, randevular vs.

6. Kullanıcının duygusal durumu, dikkati ve yaptığı işler vs.

7. Kullanıcının kan basıncı, kalp atışı, ses tonu, kan şekeri vs.



Şekil 4.1. Bağlam sınıflandırılması (mekansal, zamansal, fiziksel, sosyal, çevresel, kimlik)

Günümüzde kullanıcıyla bilgi sistemleri arasındaki etkileşimler, mobil cihazlarla daha rahat sağlanabildiğinden yani mobil cihazlarla, kullanıcısı arasındaki etkileşimlerin daha çok olması ve mobil cihazların daha çok etkileşim bilgilerini toplayacak sensörlere kendiliğinden sahip olması nedeniyle, mobil cihazlarla bağlam belirlenmesi daha ucuz ve daha çok tercih edilmektedir. Mobil cihazlarda bulunan sensörler bağlam belirlemede çok etkin bir rol oynamaktadırlar. Mobil cihazlar içinde de en yaygın olarak kullanılanlar akıllı telefonlardır. Mobil cihazlarda bağlam belirlemede en etkin araç olan sensörlerin tiplerini şu şekilde sıralayabiliriz.

- Işık: ortam aydınlatmasını lüks olarak açıklayan bir ortam ışık sensörüdür.
- Yakınlık: Cihaz ve hedef nesne arasındaki mesafeyi metre olarak gösteren bir yakınlık sensörüdür.

- Kamera: Görüntünün alınmasını sağlayan araçtır.
- Mikrofon: Sesin alınmasını sağlayan araçtır.
- Dokunma: Ekran dokunmalarını algılayan sensördür.
- Yer
  - GPS
  - Wifi
  - Network
  - NFC, Bluetooth
- Hız: Mevcut ivmeyi üç eksen doğrultusunda  $m/s^2$  olarak döndüren üç eksenli bir ivmeölçerdir.
- Manyetik: mevcut manyetik alanı üç eksen doğrultusunda mikrottesla olarak bulan bir manyetik alan sensörüdür.
- Jiroskopik Sensör: Mevcut cihaz yönelimini üç eksen derece olarak döndüren bir jiroskopik sensördür.
- Basınç: cihaz üzerinde etki eden mevcut basıncı KPa (KiloPascal) olarak bir tek değer ile döndüren basınç sensörüdür.
- Sıcaklık: sıcaklığı santigrat derece cinsinden döndüren bir termometredir.
- Nem: Ortamın nem miktarını algılayan sensördür.
- Yönelim: cihaz yönelimini üç eksen derece olarak döndüren bir yönelim sensörüdür.
- Yerçekimi: Ortamın yerçekimini belirleyen sensördür.

Örnek olarak Android işletim sistemli mobil bir cihazda bulunan sensörleri ve sensörlerin belirlediği bağlam değerlerini Resim 4.1.'deki gibi inceleyecek olursak cihazda bulunan 9 farklı bağlama ilişkin verilere ulaşılabilir. Örnekteki cihaz tablet olduğu için 9 farklı bağlama ilişkin veriler toplanabilmiştir ancak günümüz akıllı telefonlarda kullanılabilen sensör ve ulaşılabilen bağlam sayısı artmaktadır. Resim 4.2.'de ise akıllı telefon üzerindeki sensörlere ilişkin bilgilere bakıldığında 15 farklı bağlama ilişkin verilere ulaşılabilir. Bu bağlam bilgileri kullanılarak istenilen amaç doğrultusunda bağlam bilinçli mobil uygulamalar geliştirilebilir.



Resim 4.1. AndroSensor uygulaması ile Samsung P3100 tablet üzerindeki sensörlerin anlık görüntüsü ve ulaşılan bağlantılara ilişkin veriler



Resim 4.2. AndroSensor uygulaması ile Samsung S4 akıllı telefon üzerindeki sensörlerin anlık görüntüsü ve ulaşılan bağlantılara ilişkin veriler



Genel olarak akıllı mobil cihazlarda bulunan yukarıdaki sensör tiplerinin yanı sıra, örneğin kullanıcının fiziksel durumu ile ilgili sağlık bilgilerini veren bio sensörler gibi belirli amaçlar için hazırlanmış, belirli amaçlara hizmet eden özel sensör tipleri de bulunabilmektedir.

Farklı bağlam tipleri için kullanılabilen sensör örnekleri Çizelge 4.1.'de verilmiştir.

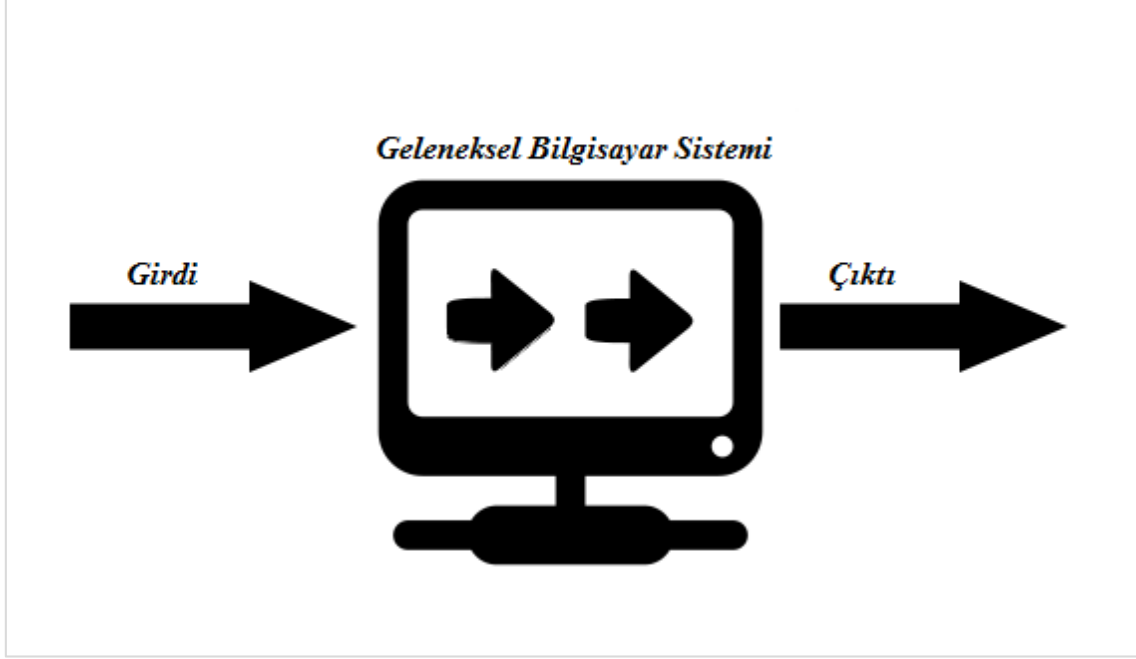
Çizelge 4.1. Farklı bağlam tipleri için kullanılan sensör örnekleri

	<b>Bağlam Tipi</b>	<b>Kullanılabilen Sensörler</b>
1.	Işık	Fotodiyotlar, renkli sensorler, IR ve UV sensörler vs.
2.	Görsel bağlam tipi	Farklı Kamera Tipleri
3.	Ses	Mikrofon
4.	Hareket, Hız	Civa düğmeleri, açısal sensörler, ivmelendiriciler, hareket belirleyiciler, manyetik sensörü, yerçekimi sensörü vs.
5.	Yer	Dış Alan: GPS, GSM İç Alan: Aktif işaret sistemleri vs.
6.	Dokunma	Mobil cihazlardaki dokunmatik ekranlar
7.	Sıcaklık	Termometreler
8.	Fiziksel nitelikler	Biosensörler

#### **4.1. Bağlam Bilinçli Bilişim**

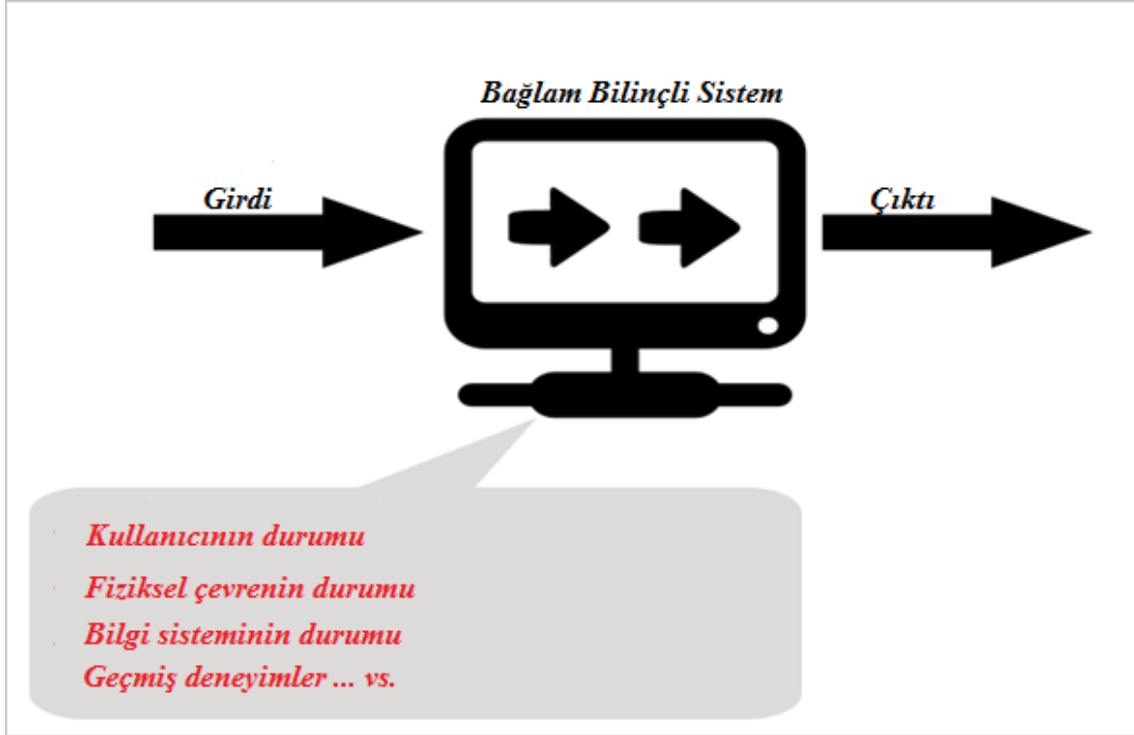
Bağlam Bilinçli Bilişim kavramının 20 yıllık bir geçmişi olmasına rağmen günümüzde bu teknoloji akıllı telefonların da yaygınlaşmasıyla en üst seviyesine ulaşmaktadır. Masaüstü veya mobil uygulama fark etmeden bulunduğu bağlama duyarlı uygulamalar, bağlam bilinçli uygulamalar olarak adlandırılırlar. Her ne kadar bağlam kullanıcının yeri olarak

bilinse ve düşünülse de kullanıcının yerinden çok daha fazlasını kapsamaktadır. Bu kapsamda düşünüldüğünde bağlam bilinçli sistemler sadece kullanıcının yer bilgisine göre değil aynı zamanda sensörler aracılığıyla farklı bağlamları belirleyip ona adapte olabilen ve bağlama göre hizmet verebilen servisleri kullanan uygulamalardır.



Şekil 4.2. Geleneksel bilgisayar sistemleri

Geleneksel bilgisayar sistemleri, herhangi bir girdiye Şekil 4.2.'de görüldüğü gibi belirli işlemler doğrultusunda çıktı üreten sistemlerdir. Kullanıcının bu tip bir sisteme etkisi sadece veri girdisinin belirlenmesinde olabilmektedir. Bunun dışında kullanıcı bu sistemin hiçbir aşamasında yer almamaktadır.



Şekil 4.3. Bağlam bilinçli sistemler

Ancak bağlam bilinçli sistemler belirli bir girdiye belirli bir çıktı üretirken Şekil 4.3.'de görüldüğü gibi kullanıcının, sistemin, fiziksel çevrenin vs. bulunduğu durumun ve geçmiş kullanıcı sistem etkileşimlerinin de farkında olarak çıktı üretmektedirler. Böylece kullanıcının isteğine, durumuna ve beklentisine uygun sonuçlar elde edilebilmektedir. Bilgisayar sistemi, bulunulan durumu otomatik olarak belirler, yani bağlam hakkındaki tüm alabildiği bilgileri toplar, geçmiş tecrübelerle bu bilgileri birleştirerek, bu şekilde bulunulan duruma adapte olabilen çıktılar üretebilir. Bu şekilde davranan mobil bir bilgi sistemine, kullanıcının takvim olaylarını tarayıp takvim olaylarına göre meşgul olduğu zaman aralığında telefonunu otomatik olarak sessize alması verilebilir. Bu örnekte kullanılan bağlamlar, kullanıcı kimliği, takvim, zaman şeklinde sıralanabilir. Bu şekilde hazırlanan uygulama ise bağlam bilinçli mobil uygulama şeklinde tanımlanabilir.

Bağlam bilinçli bilişim ilk kez Schilit ve Theimer (1994) tarafından incelenmiştir [15]. Sonrasında Dey (2001) bağlamın tanımını yaptığında [8], bağlamın bu tanımı pratik olarak herhangi bir etkileşimde katılımcının durumunu karakterize eden herhangi bir bilgiyi temsil etmektedir.

Dey'e (2001) göre bağlam bilinçli uygulamalar şu özellikleri desteklemelidir [8]:

- Kullanıcıya bilgi ve servislerin sunumu
- Kullanıcı için bir servisin çalıştırılması
- Sonraki kullanımlar için bir bilginin bağlam için etiketlenmesi

Yukarıdaki listede yer alan işlemler, bağlam bilinçli uygulamalar tarafından kullanıcıdan gizlenerek etkin bir şekilde gerçekleştirilmeli ve uygulamaların davranış değişikliğine göre adapte olabilmelidir.

## **4.2. Bağlam Bilinçli Sistemlerde Yer Bağlamının Tespiti**

Bağlam, günümüzde de en çok kabul gören şekliyle Dey (2001) tarafından tanımlanana kadar sadece yer, konum olarak algılanmakta ve bağlam bilinçli sistemlerde konum bağlamı üzerinden uygulamalar hazırlanmaktaydı [8]. Konum günümüzdeki bağlam bilinçli uygulamalar için tek bağlam olmamasına karşın yine en önemli ve en çok bilgisine ihtiyaç duyulan bağlamdır. Kullanıcının hareketine göre değişen yer, çok önemli bir bağlam olduğundan, yer izleyen sistemler ve bağlam bilinçli uygulamalar için de oldukça önemlidir. Eğer kullanıcı isterse kullanıcının yer bilgisine ulaşmak günümüzde oldukça kolaydır. Kullanıcının bir işaret taşıması ya da bir odaya girerken ya da çıkarken parmağıyla bir parmak izi okutucusuna basması bunlardan bir kaçıdır. Sistem kullanıcıyı sisteme giriş yaptığı andan itibaren izlemeye başlar. Bu metotlar daha az doğrulukla elde edilen yer bilgisini verebilir. (örneğin kullanıcı odadan ayrılırken parmağıyla okutucuya bildirimini unutmuş olabilir, bu da sonucun doğruluğunu azaltacaktır) Yer bilgisinin tespit metodu iç alanda ve dış alanda farklılık göstermektedir.

### **4.2.1. Dış alanda yer tespiti**

Dış alan yer tespiti için en iyi tercih Küresel Yer Belirleme Sistemi (GPS) dir. Çünkü GPS alıcısına sahip cihazın dünya üzerindeki kesin yeri tam doğrulukla tespit edilebilmektedir. GPS düzenli olarak kodlanmış bilgi yollayan bir uydu ağıdır ve uydularla arasındaki mesafeyi ölçerek kullanıcının dünya üzerindeki kesin yerini tespit etmeyi mümkün kılar. Bu sistem, Amerika Birleşik Devletleri Savunma Bakanlığı'na ait, yörüngede sürekli olarak dönen 24 uydudan oluşur ve bu uydular çok güçlü radyo sinyalleri yayarlar.

Yeryüzündeki GPS alıcısı, bu sinyalleri alır ve böylece konum belirlenmesi mümkün olur. Bu sistemin ilk olarak tamamen Amerikan hükümetinin askeri amaçlarını gerçekleştirmek için kurulmuşken 1980'lerde GPS sistemi sivil kullanıma da açılmıştır. Amerikan hükümeti eskiden sivil kullanıcılara 10-20 m doğrulukla bir yer tespiti imkânı vermekteyken günümüzde tam koordinatlarıyla yer tespitine imkân verilmektedir. Bu da 10 kat daha doğrulukla yer tespiti yapılabildiğini göstermektedir. Otomobil navigasyon sistemleri ve GPS alıcısına sahip mobil cihazlar bu yeni politikadan faydalanabilmektedir [19].

#### **4.2.2. İç alanda yer tespiti**

Kullanıcının dünya üzerindeki konumunu belirlemede en etkin yol olan GPS iç alanlarda kullanılamaz. Çünkü GPS sinyal gücü binalardan, binaların dış duvarlarından dolayı oldukça zayıflar. Gerçi yansımalarla bina içi okumak bazen mümkün olsa da çok yollu yansımalar güvenilir sonuçlar vermeyebilir. Bu nedenlerden dolayı iç alanlarda ideal bir yer sensörü geliştirmek günümüzde uğraşılan önemli bir problemdir. Bununla ilgili birçok araştırma projeleri kendi yer izleme sistemlerine sahiptirler. Örneğin The Olivetti Active Badge system [20], Xerox ParcTab [21], ve Cyberguide project [22] kızılötesine bağlı (IR) bir izleme sistemi kullanırlar. Kişisel alışveriş asistanı [23] radyo frekansları (RF) kullanarak yakın alan etiketleme sistemini kullanır. Ticari olarak ta kullanılabilen 3D-ID sistemi, radyo frekansı kullanarak 10 metre doğrulukla yer tespiti yapabilmektedir.

Ayrıca Olivetti ve Oracle Araştırma laboratuvarlarında ultrasonik ve radyo sinyallerine bağlı 15 cm doğrulukla yeni bir yer tespit sistemi geliştirilmiştir [24]. Her 200 ms de kontrolcü mobil cihazlardaki tüm alıcılara bir radyo mesajı gönderir ve hangi alıcıdan cevap aldığına bakar. Bu şekilde ultrasonik ve radyo sinyalleriyle cihazın yeri ve yönü tespit edilebilmektedir. Bunun bir prototipi AT&T laboratuvarlarında Cambridge Bat Sistem olarak geliştirilmiştir [24].

Cricket yer tespit sistemi MIT laboratuvarlarında yine ultrasonik ve radyo sinyallerini kullanarak geliştirilmiştir. Sistem kullanıcının yerini tespit ederken aynı zamanda her mobil cihaz da kendi yerini bilebilmektedir [25]. Mobil cihaz radyo ve ultrasonik sinyalleri dinler ve gelen iki sinyallere göre merkezi istasyonla arasındaki uzaklığı hesaplar. Sonuç doğruluğu bir kaç ayak uzunluğundadır.

Yukarıdaki anlatılan iç alan yer tespit sistemleri hücrenel yaklaşıma göre tasarlanmıştır. Tabii ki bunun dışında farklı yaklaşımlar da vardır. Microsoft Araştırma'dan RADAR sistemi, herhangi bir yer izleme sistemi kurmadan radyo frekanslarıyla gönderici ve alıcı arası uzaklığı tespit edebilmektedir [26]. Yer, bilinen gönderici pozisyonları için radyo frekans gücününün merkezi bir veritabanından sorgulanmasıyla tespit edilmektedir.

Aynı şekilde Castro ve Muntz gönderici ve farklı istasyonlar arası SNR (signal-to-noise ratio) değerlerini hesaplamışlardır [27]. Diğer hücrenel olmayan yaklaşımlar Active Floor ve Smart Floor projeleridir ki, kişileri ayak adımlarına göre belirlemeye çalışır [28]. Hareketli insanlarda doğruluğu %90'a ulaşmaktadır. Sadece insanlar için değil nesnelere ve mobil cihazlar için de çalışmaktadır. Bazı çalışmalarda ise kullanıcının yerini tespit için kameralar da kullanılmıştır [29, 30].

#### **4.2.3. Hibrit yer tespit sistemleri**

Özellikle iç ya da dış alan kullanımı için tasarlanmamış her iki alanda da kullanılacak hibrit yer tespit sistemleri de mevcuttur. GUIDE projesi, kablosuz ethernet sistemiyle (IEEE 802.11) bir hücrenel sistem kurmuştur [31]. Hücreler, WaveLAN 802.11'in baz istasyon erişimine göre kurulmuştur. Kablosuz LAN teknolojisi kullanılan büyük alanlara kurulabilmektedir. Mobil cihazlarda bulunan yer sensörü hücrenel sunuculardan gelen sinyalleri dinleyerek yeri tespit edebilmektedir. Bu bağlantı yaklaşımı aynı zamanda Bluetooth kullanılarak da yapılabilir. Ayrıca GSM Hücre Bilgisi ile Yer Belirleme Sistemi ya da küresel ağ yer belirleme sistemi olarak ta adlandırılan (GNP) hem iç hem de dış alanda kullanılabilir. GSM mobil telefonların yerini belirlemede kullanılan bu sistem, mobil telefonun taşıdığı GSM hattının hizmet aldığı baz istasyonunun konum bilgisinden yararlanarak tam doğrulukla olmasa da kullanıcının yer tespitini mümkün kılmaktadır.

## 5. BAĞLAM MODELLEME

Bağlam bilinçli bilişim alanında iyi tasarlanmış bir bağlam modelleme, sistemdeki bağlamı belirlemede önemli bir rol oynamakta ve iyi bir bağlam modelleme de bağlam bilinçli sistemin başarısını artırmaktadır. Bazı modeller bağlam içindeki kullanıcının durumunu (örneğin toplantıda, uykuda vs.) incelerken, bazıları da bulunduğu fiziksel çevresini (örneğin bulunduğu yer vs.) inceler. Ancak bağlam modellemeye başlamadan önce ilk olarak bağlam “Kim, Nerede, Ne zaman, Ne, Nasıl, Niçin” sorularıyla altı kategoride incelenmelidir [32]. Bağlama ait sensörlerden, kullanıcılardan elde edilen veriler ve bu verilerden elde edilen tüm bilgiler bu altı kategoride ele alınmalıdır. Çizelge 5.1.’de bu kategori tanımları sırasıyla gösterilmiştir.

Çizelge 5.1. Bağlam modelleme kategorileri

<b>Kategori</b>	<b>Tanım</b>	<b>Örnekler</b>
Kim	Kullanıcıya ait temel bilgiler	Id, Ad Soyad, Cinsiyet, Doğum Tarihi vs.
Nerede	Yer	Koordinat enlem boylam, şehir, bölge
Ne zaman	Zaman	Yıl, Mevsim, Zaman, Tarih
Ne	İlgili nesnelere	Uygulamaya bağlı uygulamalar, servisler
Nasıl	Devam eden işlemler	Sensörlerden elde edilen veriler
Niçin	Kullanıcının niyetleri	Stres, duygusal durum, Kullanıcının gelecek planlı aktiviteleri

Kullanıcının girdilerinden ve cihaz sensörlerinden elde edilen veriler direkt bilgiler, bu verilerin, servisler ve uygulama modülleri tarafından işlendikten sonraki elde edilen bilgileri ise dolaylı bilgiler olarak adlandırılırlar.

Bağlam modelleme ile ilgili olarak bulunulan bağlamda tanımlanacak, saklanacak ve işlenecek bağlam verilerinin, modellenme veri yapılarına göre farklı gösterim yaklaşımları mevcuttur [33].

### 5.1. Anahtar Değer Modelleri

Bağlama ait bilgilerin modellenmesinde kullanılan en basit veri yapısıdır. Anahtarın bağlam parçasını, değerini de bağlamın gerçek değerini temsil ettiği anahtar-değer çiftiyle temsil edilmiştir. Örneğin bağlam BAGLAM = (temp,20). Burada BAGLAM anahtar ve ısı değeri değişkeni (temp) ve temp değeri 20 değer çifti, BAGLAM bağlamındaki değer çiftlerini gösterir. Bu modeli tasarlamak ve yönetmek basit olmasına rağmen bağlam sonrasını elde etmek ya da bağlam değişikliği yapmak zordur. Yani karmaşık yapılarda etkin bağlam elde etme algoritmalarının uygulanmasında yetersizdir.

### 5.2. İşaretleme Şema Modelleri

Bağlam modellemede kullanılan bir diğer yaklaşım da işaretleme şema modelleridir. Bu yaklaşımda XML'in türevi olan işaretleme dilleri kullanılır. Tüm işaretleme dillerinde ortak olan içerik ve niteliklerle oluşturulan işaret etiketleriyle etiketler oluşturularak bir veri yapısı oluşturulur. Bu yaklaşımda kullanılan kaynak tanımlama çatısı (RDF) kaynakları ve ilişkilerini modelleyebilen çok bilinen bir yapıdır. RDF kaynakları tanımlarken XML yazımını, yapısı içinde RDF şema ya da RDFS şema dilini kullanır. RDFS, kaynaklar için sınıf, domain ve özellik tanımları yapabilir. Web ontoloji dili (OWL) RDF ve RDFS üzerine geliştirilmiş kaynakları tanımlamak için oldukça esnek bir yapıdır. OWL den farklı RDFS kelimeleriyle geliştirilen diğer tipik örnekler CC/PP [34] ve kullanıcı temsilci profilidir [35].

RDF ile bir temp ısı sensörünün nasıl tanımlanılacağı örneği şu şekildedir:

```
<?xml version="1.0"?>
<rdf:RDFxmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#" xmlns:sensor="http://www.ibilgilim.com/temp#">
<rdf:Description rdf:about="temp">
<sensor:tempOkuma>20</sensor: tempOkuma > </rdf:Description>
</rdf:RDF>
```

### 5.3. Grafik Modeller

Grafik modeller kullanmanın avantajı, okunabilir ve yapısının kolay olmasıdır. Böyle modellenmiş bir bağlamı bilgisayarda değerlendirmek zor olurken bazı yazarlar, genellikle modelleme amaçlı kullanılan ve güçlü grafik bileşenlerine sahip olan Unified Modeling



Language (UML) denilen birleştirilmiş modelleme dilini kullanırlar. Kapsamlı yapısıyla UML bağlam modelleme için oldukça uygundur. Bağlam modelleme için kullanılan diğer bir örnek ise Henricksen tarafından tanıtılan Object-Role Modeling Nesne rol model (ORM) yaklaşımıdır [36].

#### 5.4. Mantıksal Modeller

İlk mantıksal tabanlı bağlam modelleme McCarthy (1993) tarafından yapılmıştır [37]. McCarthy bağlamı yapay zekâda kullanılan özellikleriyle soyut matematiksel varlıklar olarak tanımlamıştır ve matematiksel yaklaşımla bağlamı formülize etmeye çalışmıştır. Bu yaklaşımda temel ilişki  $ist(c,p)$  ilişkisidir.  $p$  durumu *true* olduğunda bağlamın  $c$  olduğu sonucuna varılır. Genel olarak bu yaklaşımda bağlam, olaylar, ifadeler ve kurallar şeklinde tanımlanır. Mantıksal modeller, var olan kurallara bağlı yeni gerçekleri üretmek için bağlam nedenlerini kullanabilirler ancak yüksek seviyeli biçimselliğe sahip olmalarına rağmen geçerlilik kontrolü kolay değildir. Ancak küçük ölçekli bilgisayarlarda tam bir mantıksal sonuçlamanın olmaması çoğunlukla bir sorundur [38].

#### 5.5. Nesne Yönelimli Modeller

Bu tarz modellerin kullanılmasının en önemli sebebi nesne yönelimli tasarımın tüm avantajlarından faydalanmaktır (sarmalama, kalıtım, tekrar kullanılabilirlik) . Bağlama ait herhangi bir işlem nesne seviyesinde sarmalanır ve diğer bileşenlerden gizli tutulur. Bağlama ait bilgilere ulaşmak için arayüzler kullanılır. Diğer bileşenler işlemin nasıl yapıldığından ziyade ne yapıldığı ile ilgilenirler. Dağıtık bağlam bilinçli sistemler geliştirmek için ideal bir yaklaşımdır [39].

#### 5.6. Ontoloji Yönelimli Modeller

Ontolojiler kavramların ve ilişkilerinin tanımlarını temsil ederler [40]. Özellikle bağlama ait bilgiyi tanımlamak ve bilgi sistemleri tarafından kullanılabilir veri yapısıyla elde edilen günlük hayat bilgilerini kullanmak için uygundur. Ontolojiyle ilgili bağlam modellemedeki yaklaşımlardan birisinde psikolojik çalışmalar yapılmış ve bağlam bilgileriyle hatırlama ve tanıma arasındaki fark bu yaklaşımla incelenmiştir [41]. Ontoloji tabanlı bağlam modelleme yaklaşımının kullanıldığı öne çıkan önemli bir çalışma CoBrA sistemidir [42]. CoBrA sistemi bağlamda yer alan kişileri, yerleri ve diğer nesnelere karakterize etmek için

ontolojik kavram kümelerini oluşturur ve çalışma zamanında bağlam bilinçli sistemlere destek için merkezi bir ajan mimarisi kullanır. Bu yaklaşım bağlam bilinçli sistemlerde bağlam modelleme için oldukça uygundur.

## **6. ADAPTİF BAĞLAM BİLİNÇLİ MOBİL UYGULAMA (MOBİL ADAPTİF ASİSTAN-MAA)**

İstemci-sunucu mimarisi kullanılarak geliştirilen mobil uygulama, kullanıcının bulunduğu yer, zaman ve gezi programı bağlamlarına ve kullanıcı bilgilerine göre öneri yapan bir sistemdir. Uygulama “Mobile Adaptive Assistant” kısaltmasından MAA olarak kısaltılmıştır. MAA tarafından sistemi kullanmak isteyen kullanıcının bilgileri üye olarak sisteme kaydedilir ve üye kullanıcısı sisteme giriş yaptıktan sonra kullanıcı profiline uygun adaptif olarak yapılan önerileri kabul etmesi ya da etmemesi durumuna göre MAA sistemi öğrenir ve sonraki önerilerini yeni duruma göre yapabilmektedir. Uygulama, kullanıcının bulunduğu yer, zaman ve gezi programına duyarlı olması yönüyle bağlam bilinçli, bulunduğu yere ve kullanıcının geçmiş tercihlerine göre öğrenerek öneri yapabilmesi yönüyle adaptiftir. Ayrıca daha fazla kişiye ulaşmak için tasarlanmış ve geliştirilmiş tüm bu sistem mobil platformlarda kullanılabilir.

### **6.1. MAA Sistem Modülleri**

Geliştirilen adaptif bağlam bilinçli mobil uygulama genel olarak dört ana modülden oluşmaktadır.

#### **6.1.1. Mobil istemci modülü**

Android işletim sistemi kullanan mobil cihazlarda çalışacak şekilde Java ile kodlanmış Android uygulamasıdır. Mobil istemci tarafından sağlanan bağlama ait bilgiler, web servisler aracılığı ile web sunucusuna iletilir ve web servisi veri katmanındaki verileri kullanarak elde ettiği bilgileri mobil istemci ekranında gösterir. Kullanıcıdan alınan girdiler tekrar yorumlanmak üzere yine servisler aracılığı ile web sunucusuna iletilir. Bilgi sistemi kullanıcı etkileşimi en çok bu modülde sağlanmaktadır. MAA uygulamasının geliştirmesi yapılırken Samsung Tab2 P3100 Modeli kullanılmıştır. Ancak geliştirilen uygulama Android işletim sistemi kullanan tüm cihazlarda çalışabilmektedir.



Resim 6.1. Samsung Galaxy Tab2 P3100 3G tablet

Cihaz, ekranı daha büyük olduğu ve 3G ve WiFi üzerinden internet özelliklerini desteklediği için tercih edilmiştir. MAA, internet bağlantısına sahip olmayan cihazlarda hizmet verememektedir. Çünkü tüm veriyi çevrimiçi olarak merkezi bir sunucudan getirmekte ve sunmaktadır. Günümüzde internet bağlantısı iki farklı yolla sağlanabilmektedir.

- **3G Bağlantısı:** 3G bağlantısı, mobil cihazlardan veya bilgisayarlardan hızlı internet bağlantısı sağlayan ve birçok servis hizmeti sunan üçüncü nesil mobil iletişim altyapısına verilen genel addır. 3G ile GSM'den farklı olarak yeni bir frekans bandından ve daha fazla bant genişliğinden faydalanarak multimedya uygulamalarını mobil alana taşımak mümkün hale gelmektedir. 3G bağlantısı GSM operatörü tarafından sağlanan kullanma maliyetiyle kullanılabilen hizmettir.
- **WiFi Bağlantısı:** "Wireless Fidelity" kelimelerinin kısaltması olup kablosuz bağlılık veya kablosuz bağlantı anlamlarına gelmektedir. WiFi ürünlerin kablosuz bağlantı sağlayabildiğini gösteren bir uyumluluk göstergesidir ve IEEE 802.11a, IEEE 802.11b, IEEE 802.11g ve IEEE 802.11n standartlarına göre belirlenmektedir. WiFi, yakınlarındaki tüm cihazların kablosuz erişim noktaları aracılığıyla yerel alan ağına bağlanmasını ve varsa internet bağlantısını ek bir maliyete ihtiyaç duymaksızın kullanmasını sağlamaktadır. Bağlantı, kablosuz erişim noktaları ve cihazın ortak desteklediği, IEEE 802.11 protokolüne bağlı olarak 2.4 GHz veya 5 GHz radyo frekansında gerçekleştirilmekte ve veri iletimi, CSMA/CA protokolüne uygun

gönderilip alındığı için paketlerin iletimi sırasında hata oluşması sorunu da ortadan kaldırılmış olur.

Cihazın üzerinden entegre bulunan GPS alıcısı tarafından yer bilgisi elde edilmekte ve bu yer bilgisi internet bağlantısı aracılığıyla Web sunucusu uygulama modülüne iletilerek sonuçların mobil istemcide görüntülenmesi sağlanmaktadır. Mobil istemci ve Web sunucusu uygulama modülü arasındaki veri alışverişi güvenli HTTPS protokolü üzerinden sağlanmaktadır.

### **6.1.2. Öğrenme için yapay sinir ağları modülü**

Sistemin kullanıcısı tarafından belirlenen yapay sinir ağları parametrelerine göre web uygulama sunucusunda koşan yapay sinir ağları modülü, kullanıcının geçmiş tercihlerine göre öğrenip, kullanıcının yer, tarih, zaman ve gezi programına uygun verilerden seçebileceğini düşündüklerini belirleyerek web servisler aracılığıyla kullanıcıya sunar. Kullanıcının yeni tercihlerine göre de öğrenme belirli aralıklarla yenilenerek daha etkin ve adaptif bir öneri sistemi amaçlanmıştır.

### **6.1.3. Web sunucusu uygulama modülü**

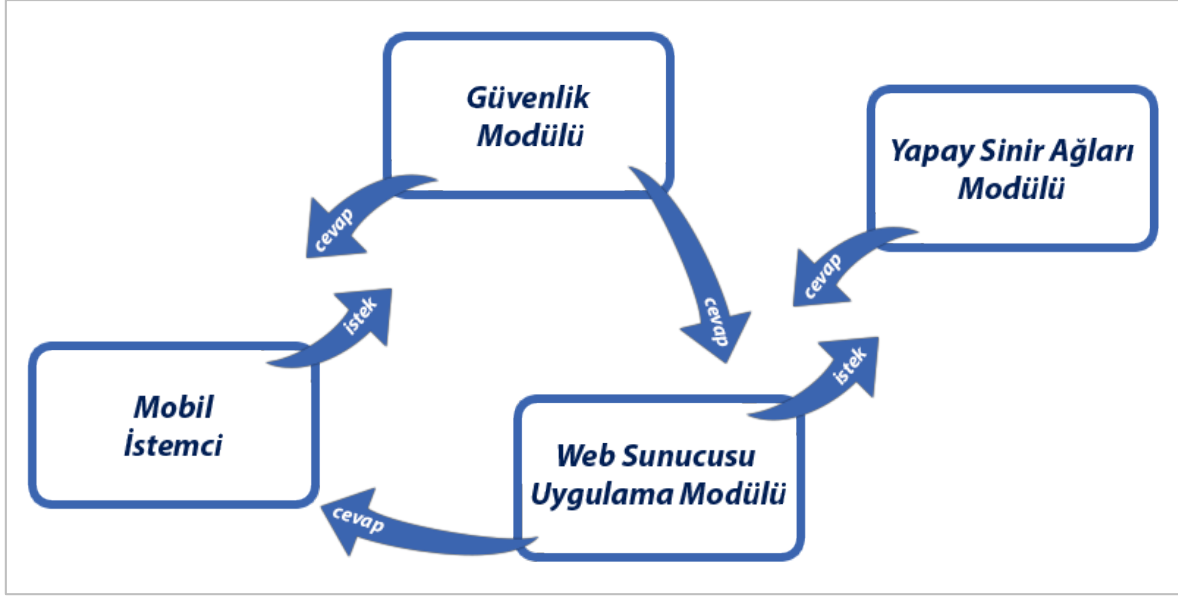
Aslında bütün iş mantığının döndüğü ve kullanıcı etkileşiminin ekran anlamında hiç olmadığı modüldür. Bu modül aracılığıyla, mobil istemcinin sağladığı bağlama ait bilgiler internet aracılığıyla bu modüle iletilir ve bu modül tarafından kullanılarak veri katmanından bağlama uygun veriler belirlenir, yapay sinir ağları modülü kullanılarak daha önceki kullanıcı tecrübeleri dikkate alınıp veriler süzülür ve sonra yapılacak tercihler için öğrenme mekanizması yeni tercihlerle birlikte eğitilerek güncellenir. Son olarak ta mobil istemci ekranına elde edilmiş son bilgi yine web servisler aracılığıyla sunulur.

### **6.1.4. Güvenlik modülü**

Web sunucusu üzerinde koşan tüm servisler, yazılan bir güvenlik modülüyle koruma altına alınmıştır. Bu modülde gelen tüm isteklerin yetkili olup olmadığı kontrol edilmekte, yetkili değilse yetkisizlik hatası verilmektedir. Daha önce sisteme kayıt olmuş olan kullanıcı, kullanıcı adı ve şifre bilgilerini girdiğinde, sistem tarafından kendisine bir eşsiz anahtar numara (oturum id) verilerek servisleri kullanma oturumu açılır ve sonraki tüm isteklerini

belirli zaman aralığında yani anahtar numaranın yaşam süresi boyunca bu anahtar numara ile yapmak zorundadır. Aksi halde servisleri kullanamayacaktır.

Geliştirilen adaptif bağlam bilinçli mobil uygulamada tüm modüller, birbirleriyle dinamik olarak veri alışverişi sağlayabilmekte ve haberleşebilmektedir.



Şekil 6.1. MAA modülleri ve arasındaki ilişkiler

Modüller üzerinde olan süreç ilişkileri Şekil 6.1.'de görüldüğü üzere aşağıdaki gibidir.

1. Mobil istemci aracılığıyla kullanıcı web sunucudan bir istek yapmaya kalktığında ilk olarak güvenlik modülü tarafından karşılanır.
2. Güvenlik modülü isteğin tipine göre yani isteğin güvenlik gerektirip gerektirmediğine göre isteği gruplayıp güvenlik gerektirmiyorsa direkt olarak isteği web sunucuya iletir, aksi takdirde güvenlik gerektiren bir istekse güvenlik için gerekli kullanıcı adı ve şifreyi veya yetkili anahtarı arar. Test sonunda güvenlik gereksinimlerini sağlayan kullanıcı isteği web sunucu uygulama modülüne iletilir ve hizmet verilir. Aksi halde mobil istemci modülüne yetkisiz erişim hatası gönderilir.
3. Güvenlik gereksinimlerini sağlayan kullanıcı isteği, web sunucusu uygulama modülü tarafından gerçekleştirilir ve cevap olarak mobil istemciye sonuç gönderilir.
4. Mobil istemci tarafından sağlanan bağlam bilgileri web sunucusu modülü tarafından oturum boyunca saklanır ve kullanıcının belirlediği aralıklarla kullanıcı isteği

olmaksızın mobil istemciye yapay sinir ağıları modülü tarafından yapılan öğrenme sonucundaki tahmin önerisi sunulur.

5. Öneri sonucunda kullanıcının tercihi ya da tercih etmemesi web sunucu modülü tarafından sisteme kaydedilir ve sonraki öğrenmede kullanılmak üzere yapay sinir ağıları modülüne iletilir.
6. Tüm yukarıdaki süreçler devam ederken belirli aralıklarla web sunucusu modülü, her istekte uzun zaman alabilecek yapay sinir ağılarını eğitmeyi gerçekleştirmemek için yapay sinir ağıları modülünü kullanarak genel kullanımlar için eğitim işlemini gerçekleştirir.
7. Ayrıca web sunucusu modülü ömrü dolan kullanıcı eşsiz oturum anahtarlarını, kullanıcıya ait öğrenmiş yapay sinir ağını ve gereksiz kullanılmayan bilgileri yok ederek hafızanın etkin kullanılmasını sağlar.

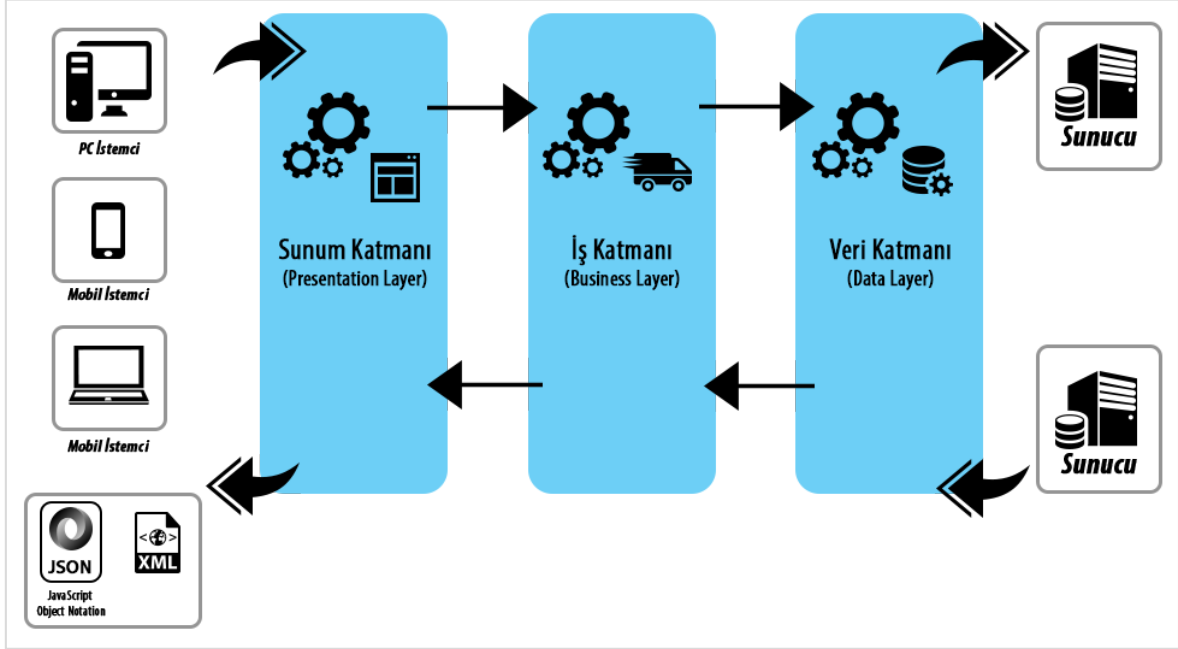
## 6.2. MAA Sistem Mimarisi

Geliştirilen Mobil Uygulama en genel şekliyle incelendiğinde uygulama sunucusu üzerinde üç katmanlı sistem mimarisi görülecektir. Sunum Katmanı (Presentation Layer), İş Katmanı (Business Layer) ve Veri Katmanı (Data Layer) dır. Veri katmanında Oracle veritabanı yönetim sistemi kullanılmıştır ve veriler buradan alınmakta ve buraya kaydedilmektedir. İş katmanında ise güvenlik modülü, yapay sinir ağıları modülü ve uygulama sunucusunun sunduğu tüm fonksiyonel işlemler yani servisler yer almaktadır. Sunum katmanında ise mobil istemcilerin kullanabileceği şekliyle veriler servislerle JSON veya XML olarak sunulmaktadır.

Gelen XML veya JSON verisi mobil istemci uygulaması tarafından yorumlamakta ve sunulmaktadır. JSON, insanlar tarafından kolayca okunabilen hafif bir veri değişim formatıdır. Makinaların okuması, yaratabilmesi ve sunması oldukça kolaydır. JavaScript Programlama Dili, Standard ECMA-262 3.Yayın - Aralık 1999 [43], versiyonunun alt kümesi üzerine kurulmuştur. JSON, tamamen programlama dillerinden bağımsız, ancak C türevi dillere (C, C++, C#, Java, JavaScript, Perl, Python ve daha pek çoğu), yazılış bakımından çok benzeyen bir veri tanımlama formatıdır. Bu özellikler, JSON'u veri değiş tokuşu için ideal hale getirmektedir.

XML'de insanlar ve bilgi sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan, W3C tarafından tanımlanmış bir standarttır. Bu özelliği ile veri

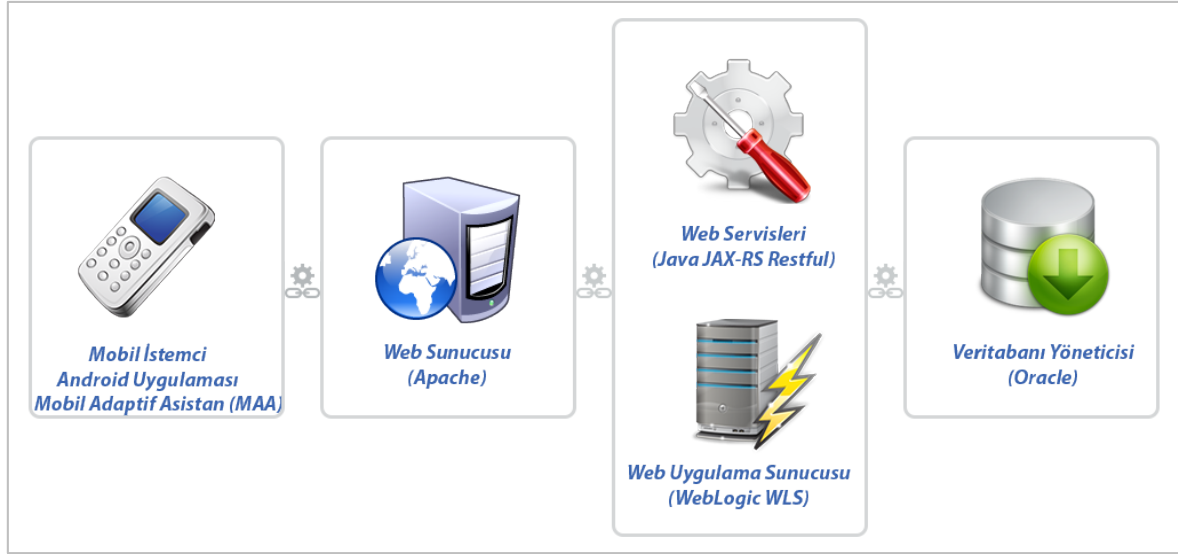
saklamanın yanında farklı sistemler arasında veri alışverişi yapmaya yarayan bir ara format görevi de görür. Bu nedenlerden dolayı hizmet veren web servislerinin veriş alış verişinde JSON ve XML formatları kullanılmıştır.



Şekil 6.2. Üç katmanlı mimari

Mobil Uygulama sistem mimarisi daha ayrıntılı incelendiğinde merkezi bir uygulama sunucusunun istemcilere hizmet verdiği bir sistemden bahsedilebilir. Bu hizmeti verirken verileri veritabanından java REST web servisi aracılığıyla getirmekte ve JSON veya XML olarak sonuçları istemciye sunmaktadır. REST web servis teknolojisi, ulaşılabacak her bilginin kendine has bir adresi yani URI 'si olan ve 2000 yılında Roy Thomas Fielding'in hazırladığı doktora tezinde tanımlanan bir yazılım mimarisidir [44]. REST mimari de üretilen servislere RESTful servisler denir. RESTful tasarlanan web servisler web üzerindeki linklere dönüşmüşlerdir ve istemci ile sunucu arasındaki iletişim genellikle HTTP üzerinden sağlanır. HTTP protokolünün sunduğu metotlarla GET, POST, PUT, DELETE ile hizmet vermektedirler. RESTful Web servislerin Java dünyasındaki standartları JAX-RS spesifikasyonu ile çizilmiştir. MAA mobil istemcinin tükettiği servisler güvenli HTTPS protokolü üzerinden sağlanmıştır.





Şekil 6.3. MAA sistem mimarisi

### 6.3. MAA Mobil Uygulama Verileri

Türkiye’de bulunan 81 il ve 957 ilçe için koordinat bilgileri, 64 farklı ilgi alanı, farklı navigasyon yazılımlarından elde edilen 31035 farklı ilgi merkezi, ve her ilgi merkezine rastgele atanan beş farklı etkinliklerle birlikte 155175 farklı etkinlik ile veri kümesi hazırlanmış ve veri tabanına kaydedilmiştir.

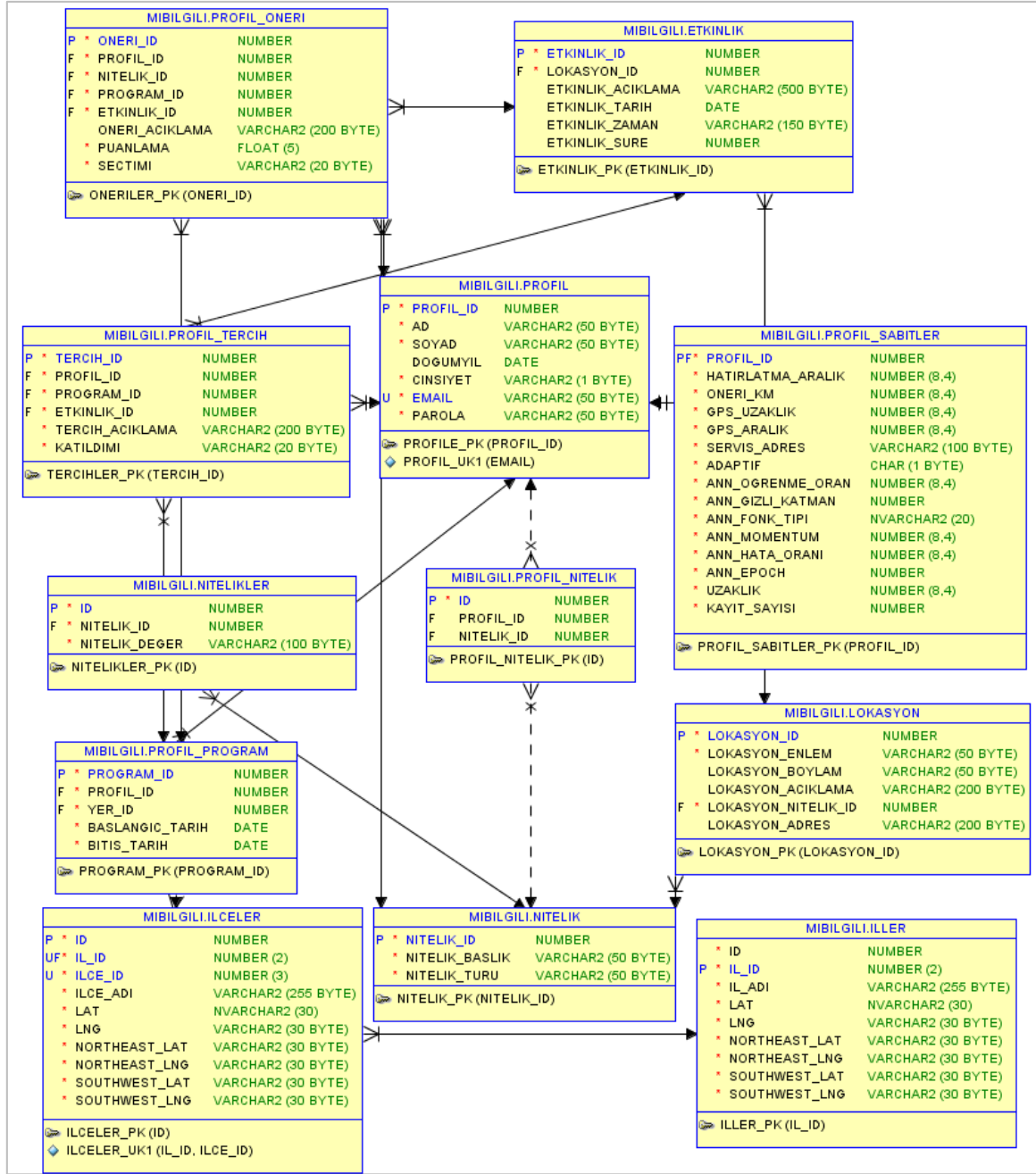
Çizelge 6.1. MAA kullanılan tablolar ve özellikleri

Tablo Adı	Açıklama
PROFIL	MAA kullanıcı bilgilerinin tutulduğu tablodur.
PROFIL_PROGRAM	MAA kullanıcıya ait planlı program bilgilerinin tutulduğu tablodur.
PROFIL_NITELIK	MAA kullanıcısının öneri almak istediği yer nitelik bilgilerinin tutulduğu tablodur.
PROFIL_TERCIH	MAA sistemi tarafından kullanıcıya önerilen ve kullanıcının tercih ettiği etkinlik bilgilerinin tutulduğu tablodur.
PROFIL_SABITLER	MAA kullanıcısının gerek sistemin genel işleyişi, gerekse yapay sinir ağlarına ait belirlediği parametreler için belirlediği bilgileri içeren tablodur.

Çizelge 6.1.(devam) MAA kullanılan tablolar ve özellikler

PROFIL_ONERI	Kullanıcıya otomatik olarak hatırlatıcı aracılığıyla önerilen tüm önerilerin tutulduğu tablodur.
NITELIK	İlgi merkezlerine ait 64 farklı nitelik bilgisinin tutulduğu tablodur.
NITELIKLER	İlgi merkezlerine ait daha ayrıntılı nitelik bilgilerinin tutulduğu tablodur.
ETKINLIK	Her konum bilgisi için rastgele atanmış beş adet etkinlik olmak üzere toplamda 155175 farklı etkinliğin bulunduğu tablodur.
LOKASYON	Farklı navigasyon cihazlarından elde edilmiş ve düzenlenmiş 31035 farklı ilgi merkezine ait bilgilerin tutulduğu tablodur.
ILLER	Türkiye’de bulunan 81 adet ilin bilgilerinin tutulduğu tablodur.
ILCELER	Türkiye’de bulunan 957 adet ilçe bilgilerinin tutulduğu tablodur.

MAA Sistemi veri tabanında yer alan tablo adları ve özellikleri Çizelge 6.1.’de, tablolar arası ilişkileri gösteren ER diyagramı Şekil 6.4.’de gösterilmiştir. MAA Mobil istemci kullanıcısı Profil tablosunda kayıtlı olup tüm tablolar doğrudan yada dolaylı bir şekilde profil\_id alanı üzerinden kullanıcıya bağlanmaktadır. Bu tabloda yer almayan kullanıcılar sistemi kullanmaya yetkili değildir.



Şekil 6.4. Uygulama verileri veri tabanı tablo ilişki diagramı

#### 6.4. MAA Mobil Uygulama İşleyiş Özellikleri

MAA mobil uygulamanın işleyişi en genel şekliyle aşağıdaki gibi sıralanabilir.

1. Kullanıcı profiline göre kullanıcının bulunduğu yer, dışarıda dahili GPS alıcının içeride GSM ağının sağladığı koordinat bilgisiyle (GNP) konumunu tespit edip, kullanıcının çevresinde ilgisine ve programına uyan etkinlikleri belirli aralıklarla kullanıcıya tavsiye eder.

2. Kullanıcının seçtiği aktiviteye göre tercihlerini belirleyip kullanıcı profiline uyan ancak daha önce ilgilenmediği aktivitelerden öğrenip önerisini buna göre düzenler.
3. Tüm bilgiler veritabanında saklanıp mobil uygulama yalnızca istemci olarak uygulama sunucusundaki servislerle bu bilgilere ulaşabilir.
4. Kullanıcının tanımladığı program dahilinde belirli aralıklarla kullanıcıya öneri bildirimini yapar ve tercihlerini kaydeder.
5. Bazı aralıklarla da kişinin profilinde bulunmayan fakat diğer kullanıcılar tarafından çoğunlukla tercih edilmiş olan aktivitelerin de kullanıcıya önerilmesi yapılarak, tercih edip etmemesi durumuna göre sistemin öğrenmesi sağlanır.
6. Uygulamada kullanılacak tüm parametreler için uygulamada profil bazında parametre giriş ekranı bulunmaktadır. Bu da uygulamada kod değişikliği yapmaksızın sistem parametrelerine göre sistemin davranış değişikliğini mümkün kılmaktadır.
7. Uygulama sunucusunda tüm servisler yetkisiz kullanıma karşı korunmakta ve tüm trafik güvenli iletim olan HTTPS protokolü üzerinden sağlanmaktadır. Uygulama kullanımı kullanıcı adı ve şifreyle profil bazında güven altına alınmıştır.
8. Kullanıcı dostu ekranlar hazırlanmış, kullanıcı bilgisayar etkileşimi en üst düzeyde tutulmaya çalışılmıştır.

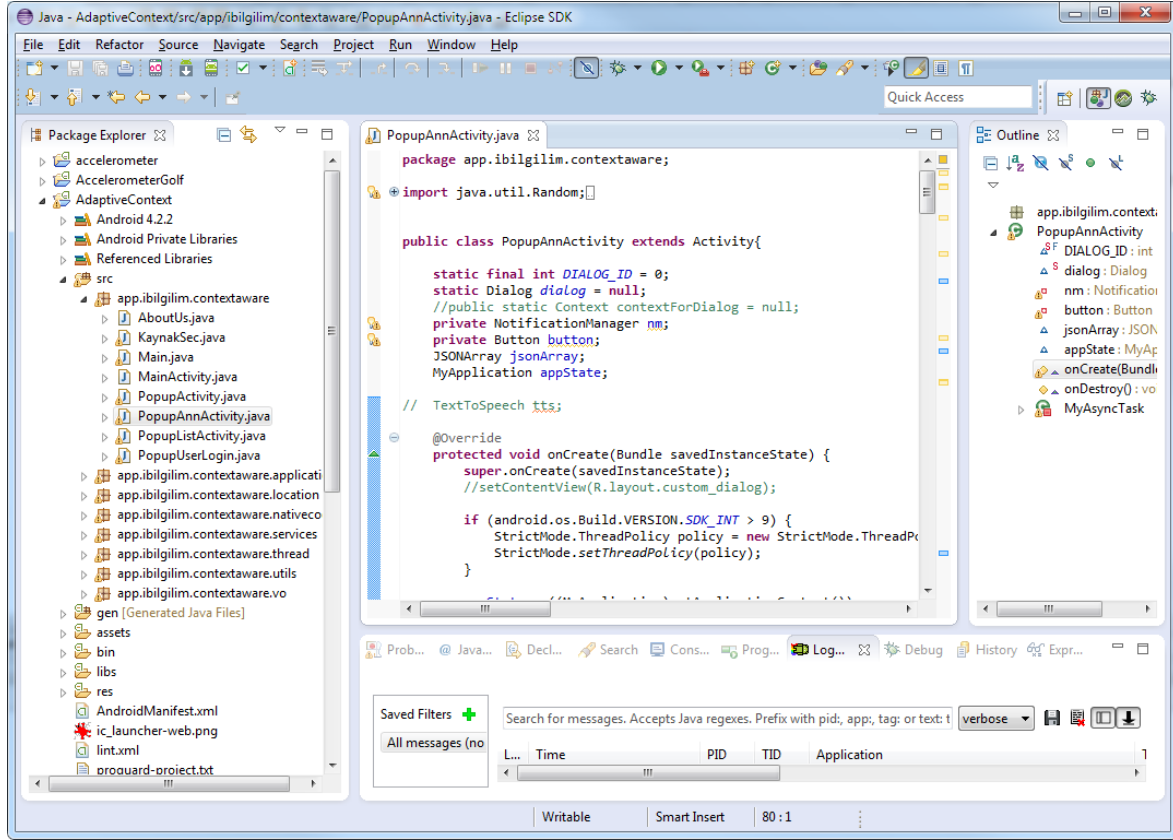
## **6.5. MAA Mobil İstemci ve Web Sunucu Uygulamaları**

### **6.5.1. MAA mobil istemci uygulaması**

Mobil istemci uygulaması, Java programlama diliyle Android SDK kullanılarak geliştirilmiştir. Geliştirme ortamı olarak Eclipse IDE kullanılmıştır. Android platformlara uygulama geliştirmek için öncelikle geliştirme yapılacak ortamda aşağıdaki işlemler sırasıyla yapılmalıdır.

- Java Development Kit (JDK) kurulumu yapılmalıdır [45].
- Eclipse IDE kurulumu yapılmalıdır [46].
- Android Software Development Kit (AndroidSDK) kurulumu yapılmalıdır [47].
- Kurulumu yapılmış olan Eclipse IDE'sine Android kod geliştirme için gerekli plugin yüklenmelidir [48].
- Android SDK ve Android SDK ile gelen AVD Manager Ayarları yapılmalıdır.

- Uygulama geliştirme fiziksel Android cihazında yapılmayacaksa, AVD Manager kullanılarak Sanal Android Cihazı oluşturulmalıdır.



Resim 6.2. Eclipse IDE

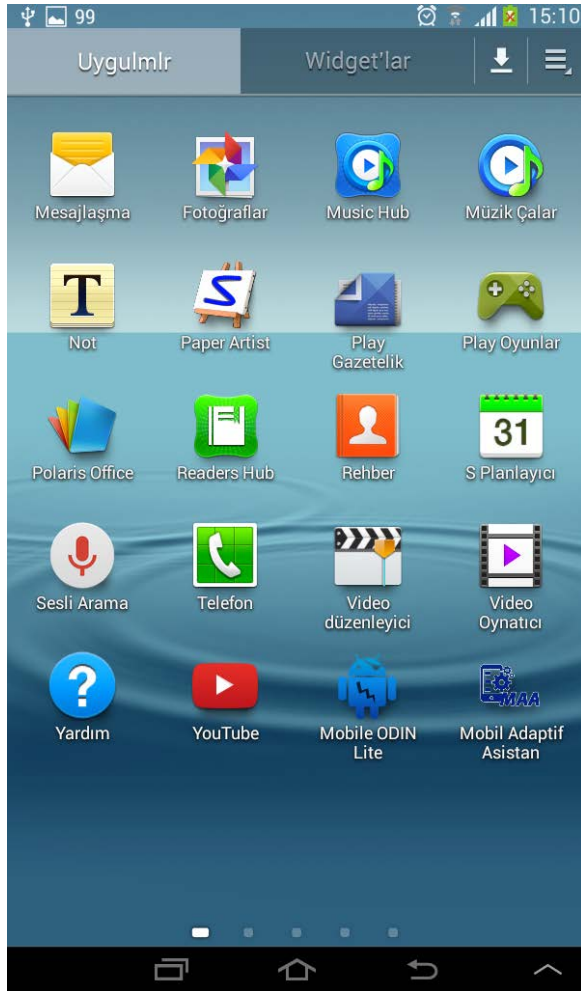
MAA, mobil istemci uygulaması için gerekli kurulum dosyası APK istenirse Eclipse IDE'si aracılığıyla üretilerek istenilen android işletim sistemli cihaza kurulabilmektedir. Veya cihaz bağlantı kablosu aracılığıyla geliştirme ortamına bağlanarak IDE üzerinden Android uygulaması olarak çalıştırıldığında otomatik olarak cihaz üzerine kurulacaktır. Uygulama cihazda Mobil Adaptif Asistan adıyla yer almaktadır.

Mobil uygulamanın iki katmanı bulunmaktadır. Birinci katman genellikle birçok ekranın hazırlandığı HTML5 standartlarında JQueryMobile Kütüphanesi kullanılarak hazırlanmış ekran kodları ve scriptleri, ikinci katman ise Android kaynaklarının kullanılabilmesi için geliştirilen native java kodları içermektedir [49]. Standart HTML5 sayfaları ile Android'in haberleşmesi için PhoneGap javascript kütüphanesi kullanılmıştır [50]. PhoneGap kullanılarak javascriptler aracılığıyla pluginler hazırlanabilmekte ve bu pluginler aracılığıyla html sayfaları ile android kaynakları arasında alışveriş sağlanabilmektedir. Bu

da uygulama ekranlarının standart teknolojilere uygun olarak daha hızlı ve etkin hazırlanmasını sağlamakta, farklı platformlara uygulamalar birkaç ufak çalışmayla kolayca entegre edilebilmektedir. Bu iki katmanda mobil platform üzerinde entegreli ve uyumlu bir şekilde konuşabilmekte ve veri alışverişi sağlayabilmektedir.

### MAA Kullanıcı Arayüzleri

Mobil platforma kurulan uygulamanın çalıştırılabilmesi için MAA uygulaması android uygulamalar teysisinden seçilmelidir.



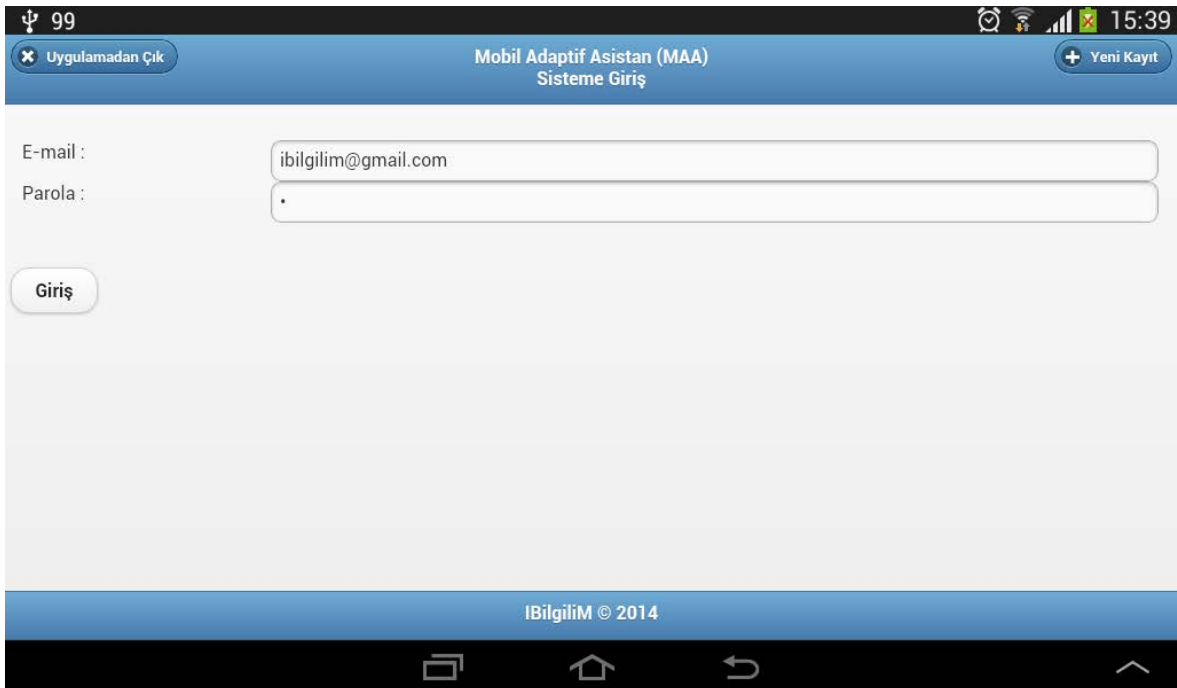
Resim 6.3. MAA uygulamasının çalıştırılması

Uygulama çalıştırıldıktan sonra ilk olarak karşılama ekranı gelecektir. Bu ekran ekrana dokunulmadığı takdirde beş saniye sonunda kaybolacaktır. Kullanıcı ekrana dokunduğunda ise karşılama ekranı kapatılacak ve kullanıcıyı bir sonraki ekran karşılayacaktır.



Resim 6.4. MAA Uygulaması Karşılama Ekranı

Kullanıcı uygulama karşılama ekranından geçtiğinde sistemi kullanmaya yetkisi olup olmadığı belirlenmek üzere sisteme giriş ekranına yönlendirilecektir. Bu ekran aracılığıyla kullanıcı isterse var olan kullanıcı adı ve şifresiyle sisteme girip sistem hizmetlerinden faydalanabilmekte, eğer sisteme daha önceden üyeliği yoksa Yeni Kayıt butonu aracılığıyla sistemi kullanmak için kendi profilini sisteme tanıtabilmektedir.



Resim 6.5. MAA uygulamasının sisteme giriş ekranı

Eğer kullanıcı kendini sisteme tanıtmak isterse ekranda yer alan yeni kayıt butonuna basması yeterli olacaktır. Yeni kayıt butonuyla birlikte açılan ekranda kullanıcıya ait kullanıcı adı, şifre, ad soyad, doğum tarihi, cinsiyet ve öneri almak istediği ilgi noktaları sorulmakta ve sisteme kaydedilmektedir. Sistem kaydı gerçekleştirildiğinde kullanıcı giriş sayfasına yönlendirilmekte ve isterse yeni kullanıcı adı ve şifresiyle sisteme giriş yapabilmektedir. Tüm bu işlemler sırasında gerçekleşen veri alışverişi sırasında hiçbir şekilde mobil istemci tarafında veri tutulmamakta tüm veriler, HTTPS protokolü üzerinden web sunucusu uygulamasına web servisler aracılığıyla gönderilip alınmaktadır.

Resim 6.6. MAA uygulaması yeni kayıt ekranı



The screenshot shows the 'Mobil Adaptif Asistan (MAA) Profil Tanımla' registration screen. The form contains the following fields and values:

E-mail :	deneme@deneme.com
Parola :	.....
Ad :	Test
Soyad :	Test
Doğum Tarihi :	1982-06-03
Cinsiyet :	

A success message popup is displayed over the form, stating: "Profil başarıyla oluşturuldu! Giriş yapabilirsiniz!". Below the form is a blue button labeled "Kaydet". At the bottom of the screen, there is a footer with the text "IBilgiliM © 2014".

Resim 6.6.(devam) MAA uygulaması yeni kayıt ekranı

Mobil istemci tarafından veriler, web sunucu uygulamasına iletilirken JSON nesnelere dönüştürülüp gönderilmektedir. Android tarafından verileri web sunucusuna göndermek için yukarıda bahsedilen iki mobil istemci katmanı için iki farklı dilde iki farklı metot kullanılmıştır.

- Birinci Katmanda kullanılan JSON gönderme örnek metodu (HTML Sayfalarından Javascript Koduyla)

```
function kaydet(jsonObject) {
    var urlKaydet = restServiceUrl+"profil/kaydet";
    $.ajax({
        type: "POST",
        url: urlKaydet,
        dataType: "json",
        contentType: 'application/json',
        data: jsonObject,
        success: function (data) {
            $("#lblMesaj").text("Profil başarıyla oluşturuldu! Giriş yapabilirsiniz!");
            $("#popupCloseLeft").popup("open");
            $("#popupCloseLeft").on({
                popupafterclose: function(event, ui) {
                    window.location.href = "login.html";
                }
            });
        }
    });
}
```

```

    }, error: function (xhr) {
        $("#lblMesaj").text("Hata oluřtu");
        $( "#popupCloseLeft" ).popup("open");
    }, complete:function(){
        //window.location.href = "login.html";
    }
});
}

```

- İkinci Katmanda kullanılan JSON gönderme örnek metodu  
(Android Java Koduyla)

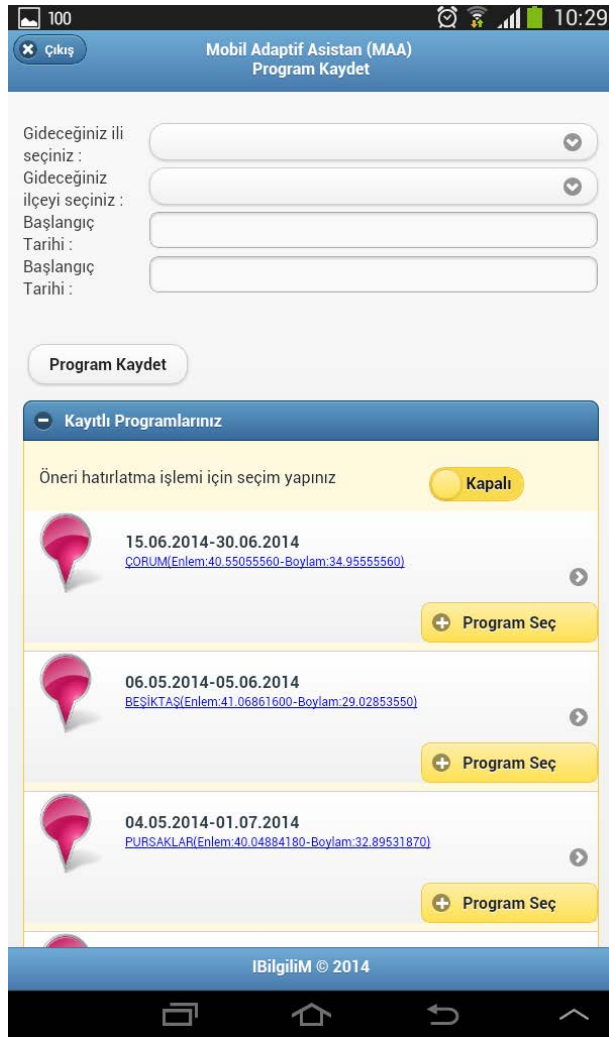
```

public static boolean sendJSONToRest(String url, JSONObject
object) {
    boolean result = false;
    try {
        HttpPost post = new HttpPost (new URI(url));
        HttpClient client = new DefaultHttpClient();
        client=sslClient(client);
        // HTTP Post İsteđini alıřtır
        String message = object.toString();
        post.setEntity(new StringEntity(message, "UTF8"));
        post.setHeader("Accept", "application/json");
        post.setHeader("Content-type", "application/json");
        HttpResponse resp = client.execute(post);
        if (resp != null) {
            if (resp.getStatusLine().getStatusCode() == 204)
                result = true;
        }
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (ParseException e) {
        e.printStackTrace();
    } catch (URISyntaxException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return result;
}

```

Kullanıcı, MAA sistemine kayıt olduktan sonra sisteme giriş yapabilmektedir. Kullanıcı, sisteme giriş yaptıđında tüm aktif oturum süresi boyunca kendisine sistem tarafından yapılacak önerilere ilişkin geçerli olacak programını seçmek zorundadır. Gelen ekranda

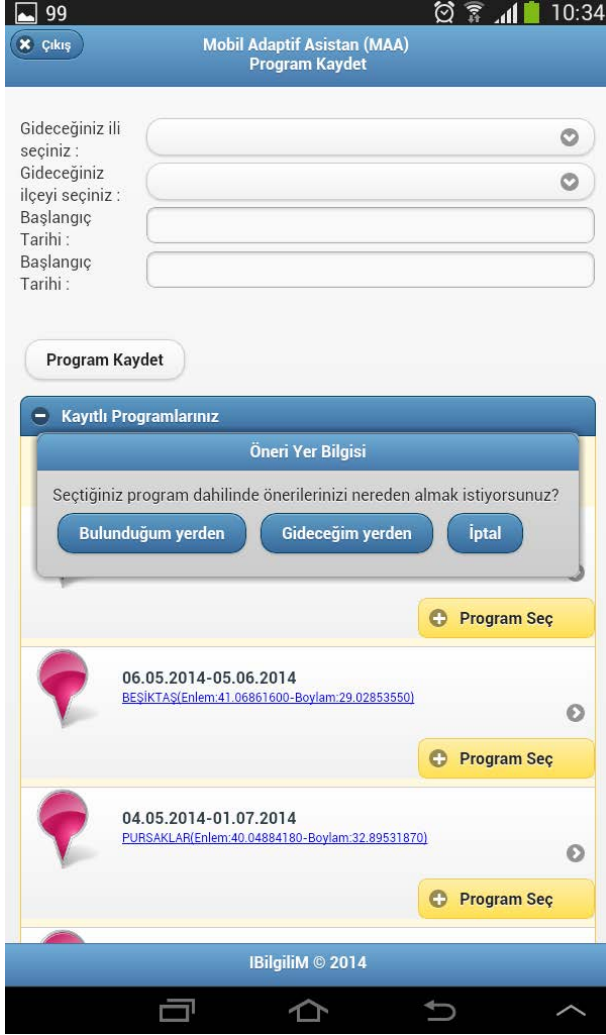
varsa daha önceki sonlanmamış yani bulunulan tarihi içine alan programlar listelenir. İsterse yeni bir program tanımlayabilir ya da var olan programlardan birini seçebilir.



Resim 6.7. MAA uygulaması program seçme ekranı

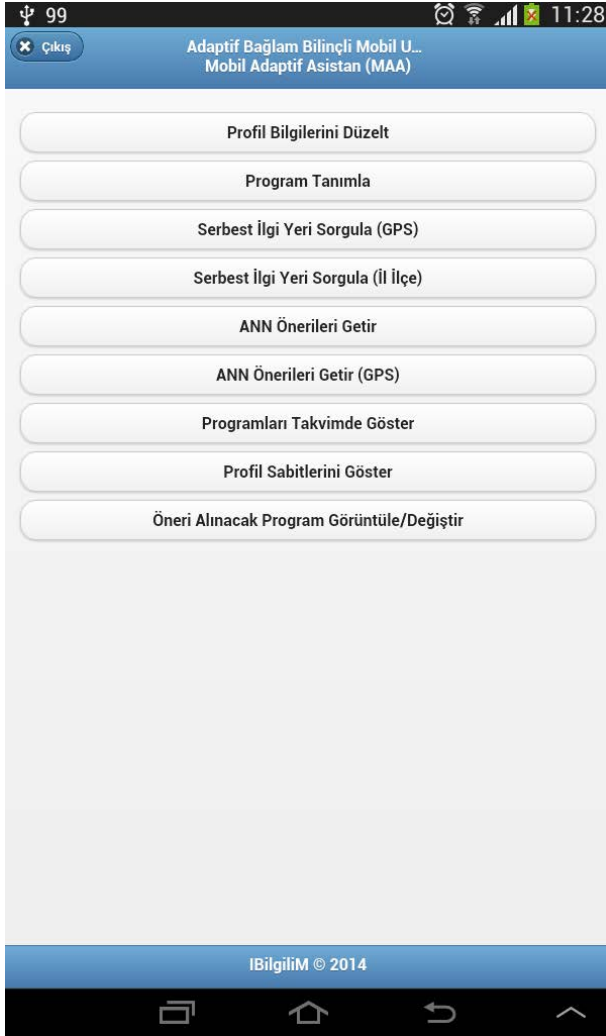
Kullanıcı, var olan programlardan birini seçtiğinde “Seçtiğiniz program dahilinde önerilerinizi nereden almak istiyorsunuz?” sorusu gelmekte, kullanıcının cevabı oturum boyunca saklanmaktadır. Çünkü öneriler için kurulacak öneri metodunu bu parametre şekillendirecektir. Kullanıcının “Gideceğim yerden” seçeneğini seçmesi, yapılacak önerilerin programın yapılacağı konum bilgisine göre, “Bulduğum yerden” seçeneğini seçmesi ise cihazın dahili GPS alıcısıyla elde edilecek, cihazın o anda bulunduğu konum bilgisine göre yapılacağı anlamına gelmektedir. Eğer kullanıcı bulunduğu yerden öneri almak isterse seçtiği programın yer bilgisi ile bulunduğu yer yakınlık göstermelidir aksi takdirde program konum bilgilerine uyan sonuçlar önerilemeyecek genel öneri listesinden etkinlikler önerilecektir. İsterse kullanıcı yine bu ekran aracılığıyla sistemin öneri

hatırlatma sistemini aktif hale getirebilmektedir. Kullanıcı, öneri hatırlatma işlemini açık olarak seçtiğinde kendi belirleyebileceği zaman aralıklarında sistem tarafından belirlenen öneriler popup pencereleriyle karşısına gelecektir. Aksi takdirde kullanıcı sistem önerilerini ilgili ekranlara girdiğinde görüntüleyebilecektir.



Resim 6.8. MAA uygulaması öneri yer bilgisi seçme ekranı

Kullanıcı tarafından öneri almak istenilen program seçildiğinde mobil istemci ve web sunucu uygulaması üzerinde kullanıcı için bir oturum açılacak ve kullanıcı sistemden çıkış yapana kadar yada 120 saniye işlem yapmadan durana kadar bu bilgiler her iki tarafta da saklanacaktır. Kullanıcı doğru kullanıcı adı ve şifreyi sisteme girdiğinde sistemde yetkili olması dolayısıyla sistemde yetkili kullanıcıların kullanabildiği tüm işlemlere gidilebilen ana menü ekranı gelmektedir.



Resim 6.9. MAA uygulaması ana menü ekranı

### *Profil Bilgilerini Düzelt*

Kullanıcı, isterse bu ekran aracılığıyla sistemde kayıtlı olan kullanıcı bilgilerini değiştirebilmektedir. Ayrıca ilgi alan başlıklarını da bu ekran aracılığıyla güncelleyebilmektedir. Var olan ilgi başlığını kaldırabilir ya da yeni bir ilgi başlığı tanımlayabilmektedir. Böylece MAA tarafından kullanıcıya yapılacak öneriler, kullanıcının yeni ilgi alanlarına göre olacaktır.

100 Mobil Adaptif Asistan (MAA) Profil Tanımla 10:27

Geri

E-mail : ibilgilim@gmail.com

Parola : \*

Ad : Muhammed İkbal

Soyad : Bilgili

Doğum Tarihi : 1982-06-04

Cinsiyet : Erkek

İlgiler

Lütfen öneri almak istediğiniz ilgi alanlarınızı seçiniz.

Alışveriş

Gezi

Araba

Banka

Ulaşım

Yemek

Petrol

Elektronik

Eğlence

Otel

ArabaKiralama

Trafik

İletişim

Diğer

Resim 6.10. MAA uygulaması profil bilgilerini düzelt ekranı

Tüm uygulamada olduğu gibi bu ekranda da veriler merkezi bir web uygulama sunucusundan JSON nesnelere olarak getirilmekte ve gönderilmektedir. JSON nesnesi olarak veriyi web servisler aracılığıyla mobil istemciye çekmek için aşağıdaki kodlar kullanılmaktadır.

- Birinci Katmanda kullanılan JSON dizisi getirme örnek metodu (HTML Sayfalarından Javascript Koduyla)

```
function nitelikleriGetir(profilId) {
    getJSON(restServiceUrl + 'profilnitelik/profilId/' +
profilId, function(data) {
        ilgiler = data;
        $('#ilgiler').html("");
        $.each(ilgiler, function(index, nitelik) {
```

```

        var selectNitelikler = '<option value="" +
nitelik.nitelikId.nitelikId + '>' +
nitelik.nitelikId.nitelikBaslik + '</option>';
        $('#ilgiler').append(selectNitelikler);
    });
    $('#ilgiler').selectmenu('refresh', true);
});
}

```

- İkinci Katmanda kullanılan JSON dizisi getirme örnek metodu  
(Android Java Koduyla)

```

public static JSONArray getJSONArrayFromUrl(String url) {

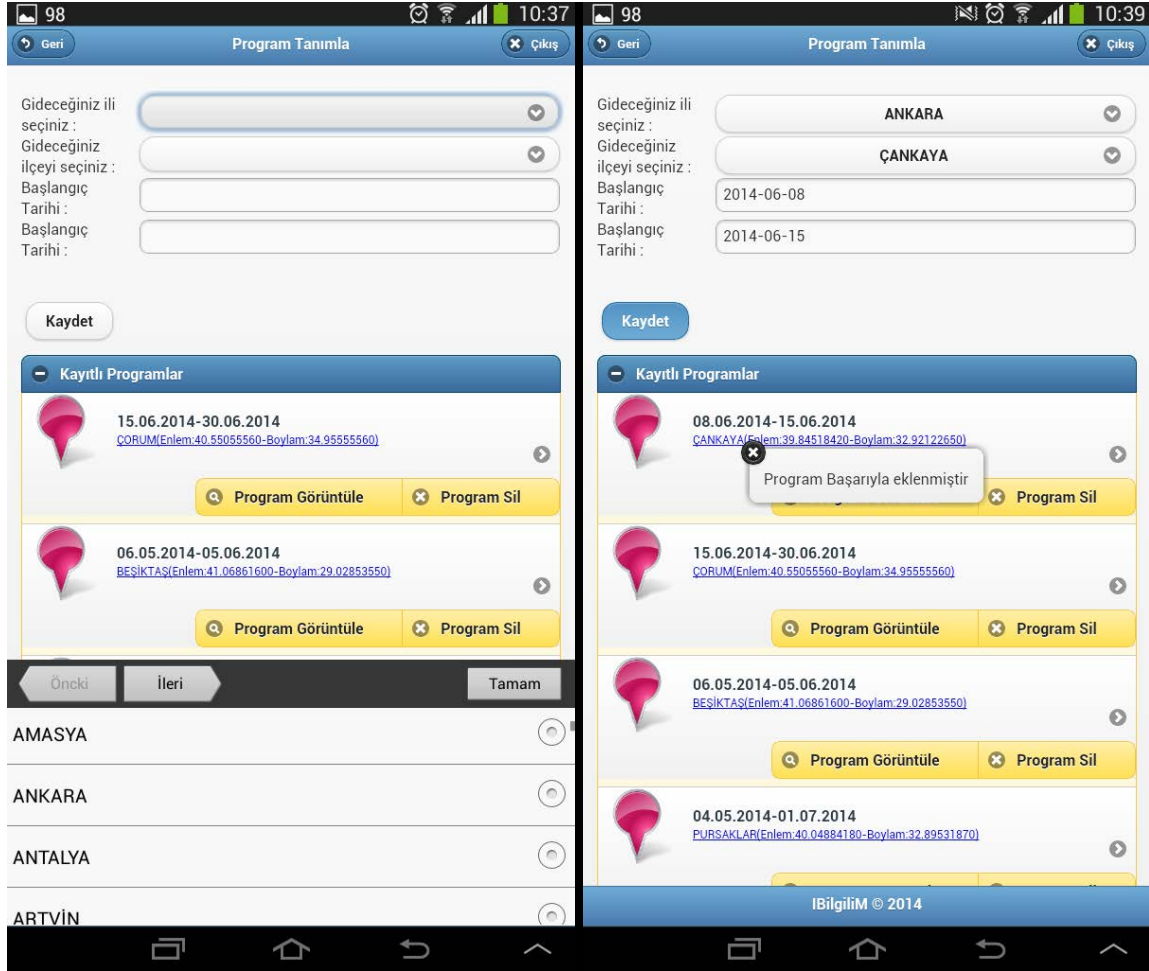
    JSONArray ja=null;
    try {
        HttpGet post = new HttpGet(new URI(url));
        HttpClient client = new DefaultHttpClient();
        client=sslClient(client);
        HttpResponse result = client.execute(post);

        String text =
EntityUtils.toString(result.getEntity(), HTTP.UTF_8);
        ja = new JSONArray(text);
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (ParseException e) {
        e.printStackTrace();
    } catch (URISyntaxException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return ja;
}

```

### *Program Tanımla*

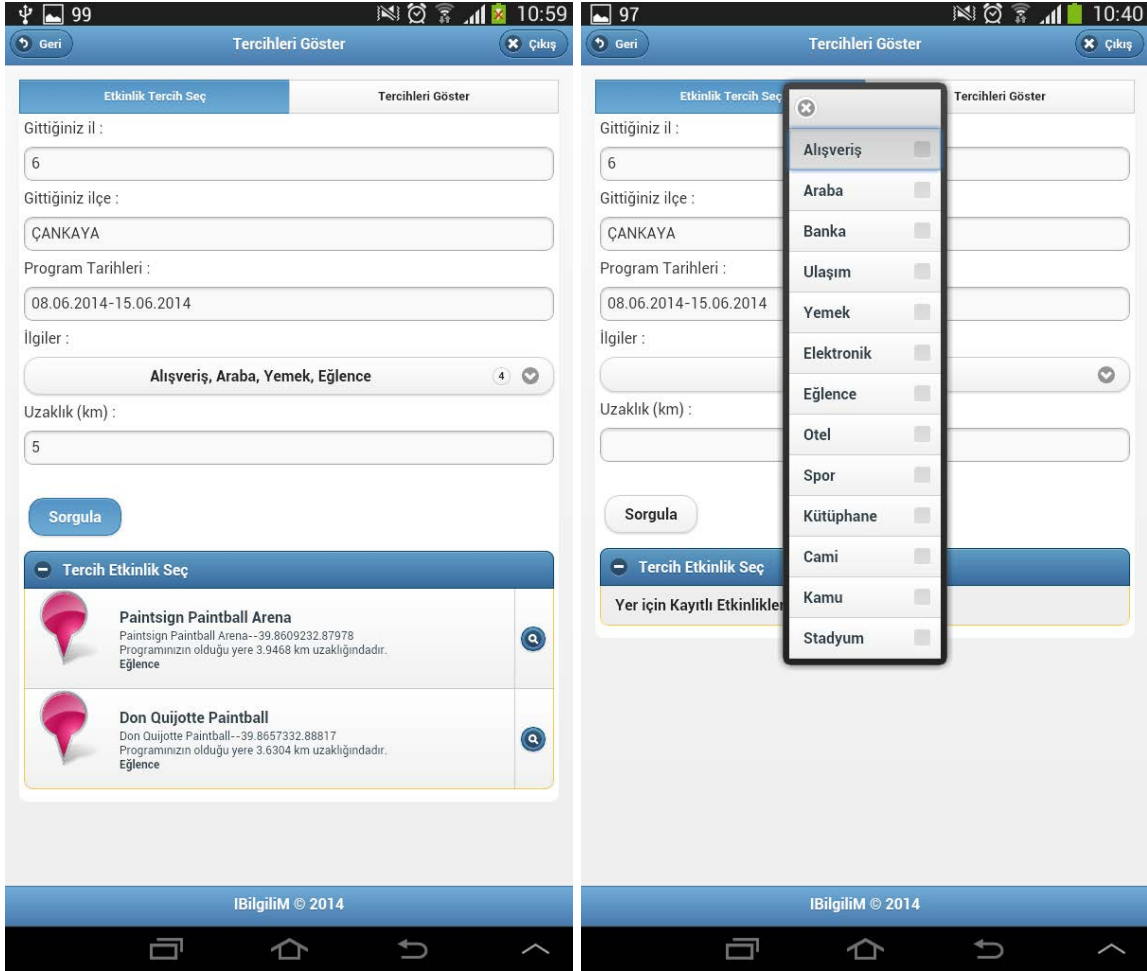
Bu ekran, bulunulan tarihi içine alan programlar ve yeni tanımlanan planlı programlar üzerinde her türlü işlemin yapılabilirdiği ve yeni planlı programların tanımlanabildiği bölümdür. Program, Türkiye içinde gidilecek il ve ilçe seçilip, programın hangi tarihler arasında gerçekleşeceği girildikten sonra kaydedilebilmektedir.



Resim 6.11. MAA uygulaması program tanımla ekranı

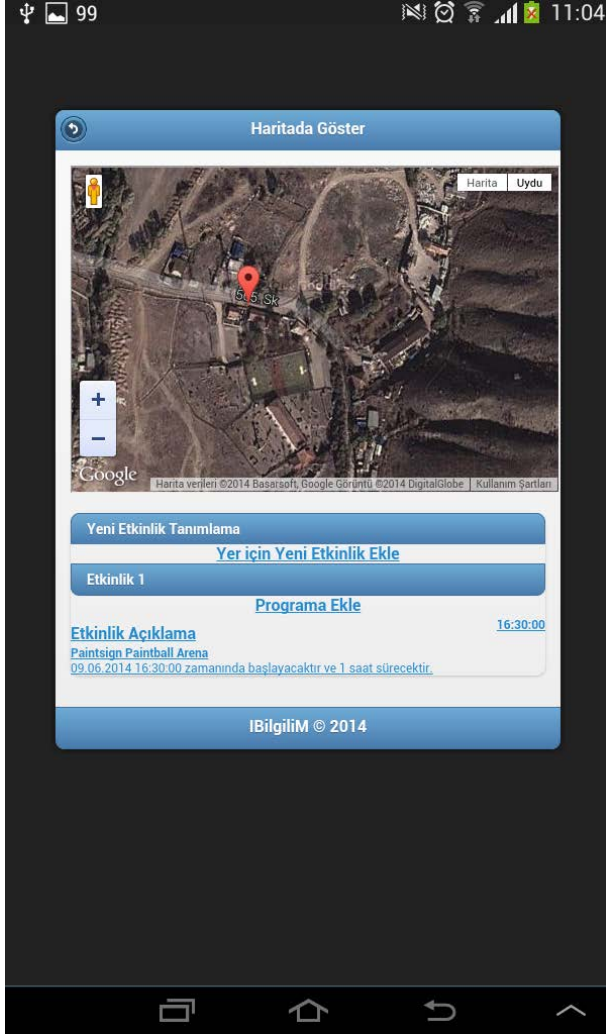
Program kaydedildikten sonra üzerine tıklanarak harita üzerinde görüntülenebilmekte ve program sil butonuyla tamamen sistemden silinebilmektedir. Program görüntüle butonu aracılığıyla programa sistemde kayıtlı olan yerlerdeki etkinlikler eklenebildiği gibi, yer etkinliklerinden bağımsız program dahilinde kişisel toplantılar, ziyaretler vs. de kaydedilebilmektedir.





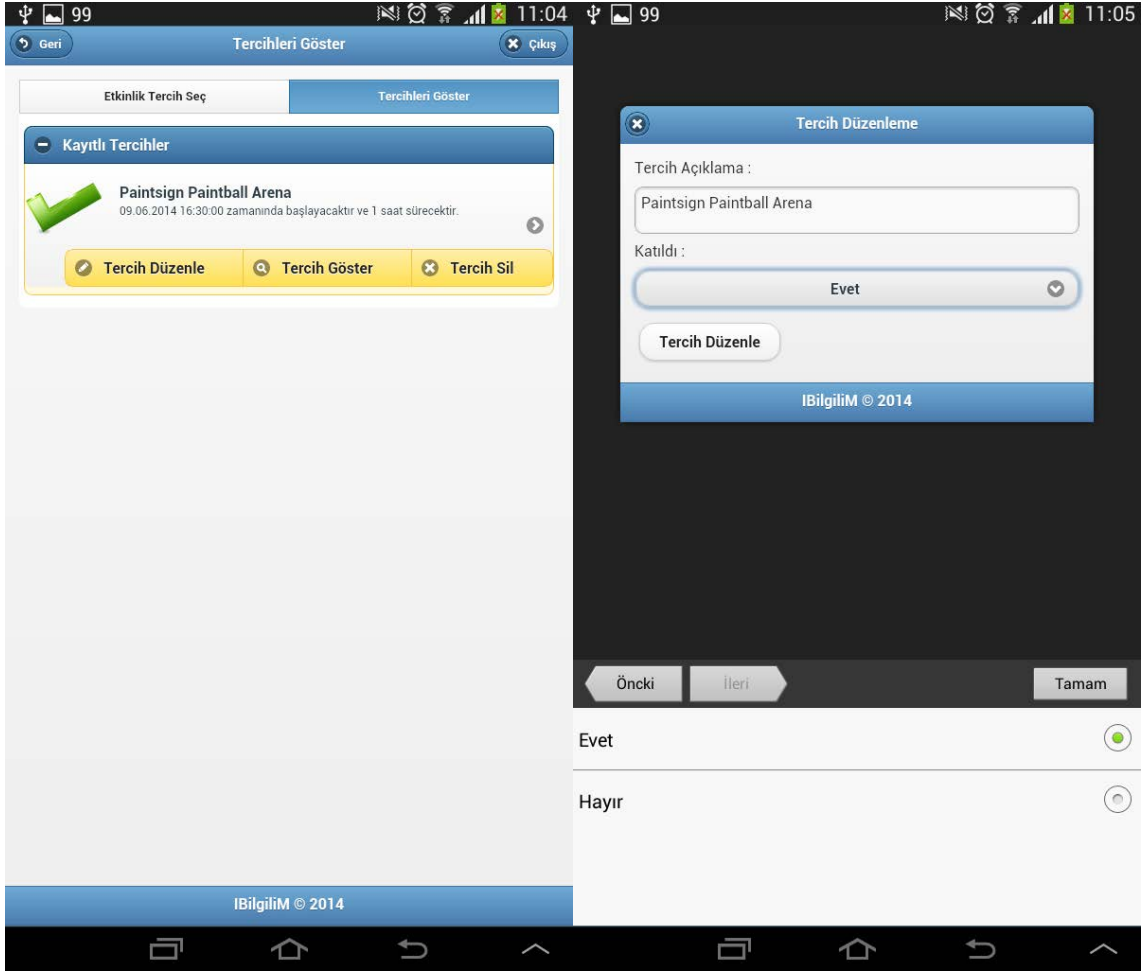
Resim 6.12. MAA uygulaması program göster-tercihleri göster ekranı

Program tanımlama ekranından program göster seçilip Etkinlik Tercih Seç seçildiğinde kullanıcı programına uygun bilgilerle birlikte serbest olarak kullanıcının kendi ilgi alanlarından istediklerini seçerek ve istediği bir uzaklık çevresinde sorgulama yapabileceği ekran gelmektedir. Bu ekranlarda sorgulanıp belirlenen yerlere tıklanarak yer içerisinde kullanıcı program tarihlerine uyan etkinlikler görüntülenebilmekte, seçilebilmekte, istenirse yer için yeni etkinlik tanımlayıp seçilebilmektedir.



Resim 6.13. MAA uygulaması seçilen yer etkinlikler ekranı

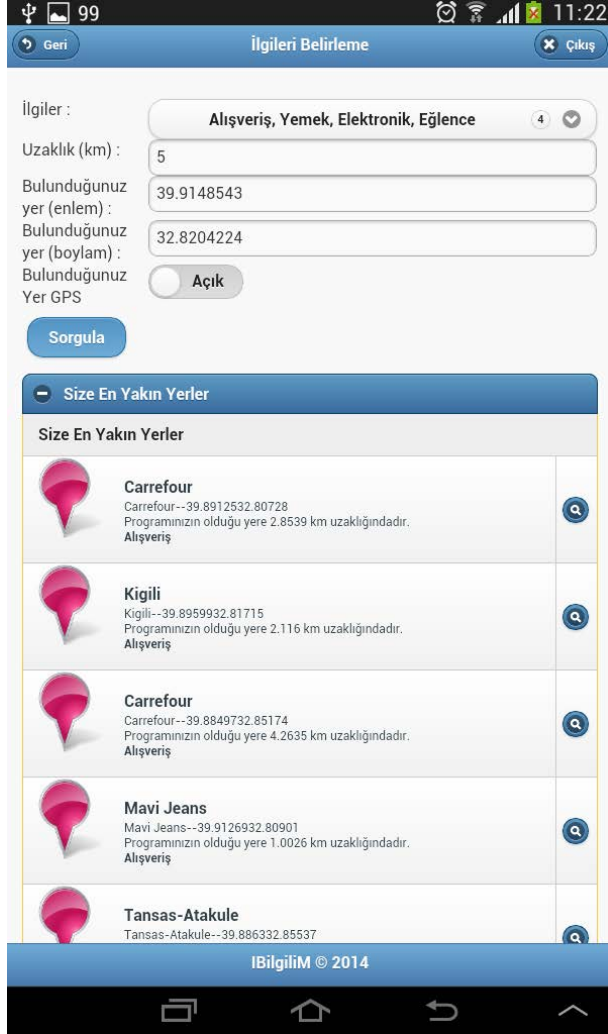
Etkinlik programa eklendiğinde ekran otomatik kapanacak ve bir önceki ekrana dönecektir. İstenirse önceki ekranda bulunan Tercihleri Göster bölümüyle programa eklenmiş olan etkinlikler görüntülenebilmekte, güncellenebilmekte ve silinebilmektedir.



Resim 6.14. MAA uygulaması seçilen tercihleri göster ekranı

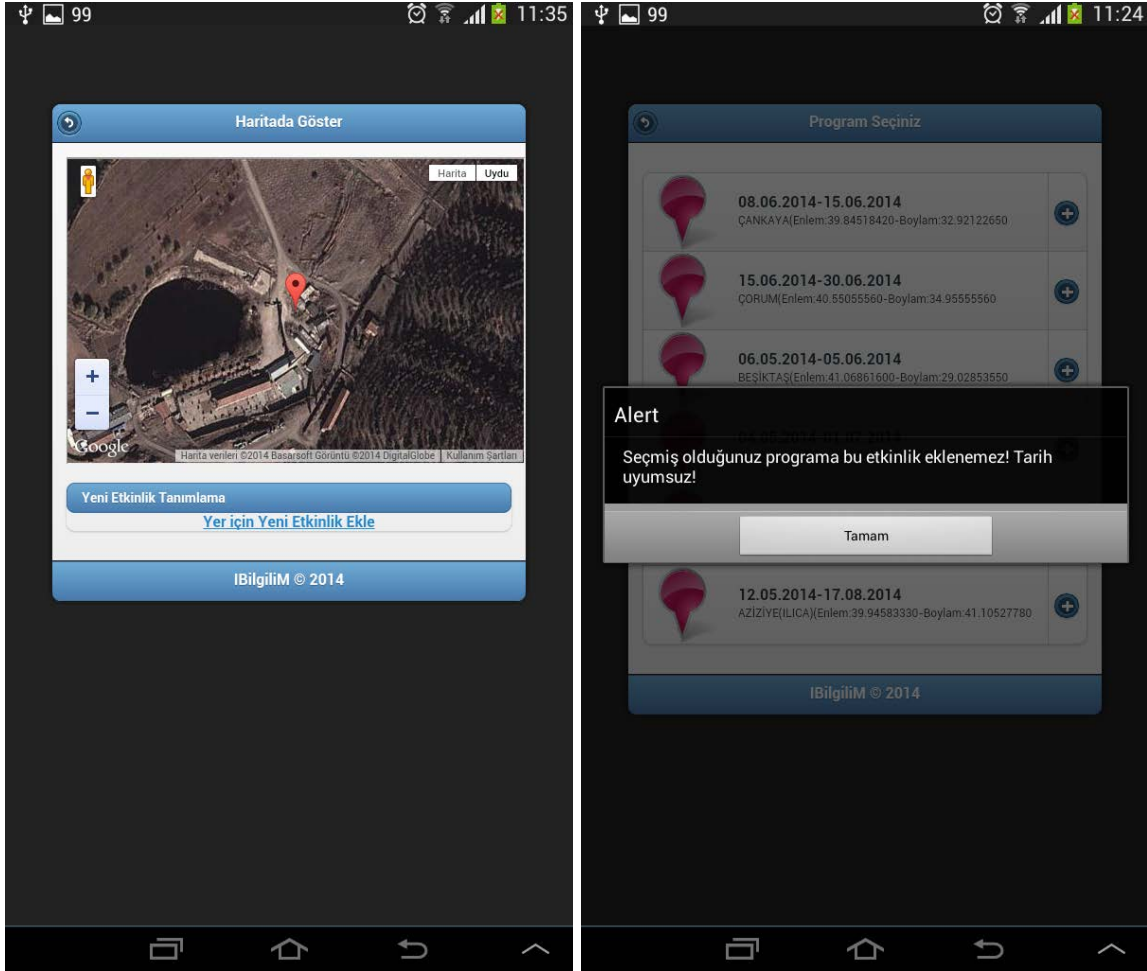
### *Serbest İlgi Yeri Sorgula (GPS)*

Serbest ilgi yeri sorgulama ekranında bir önceki program tanımlama ekranının tüm işlevselliği bulunmakta ancak bu ekran daha çok ilgi yeri ve konuma yoğunlaşmaktadır. Konum bilgisi dahili GPS aracılığıyla alınarak ekrana otomatik olarak getirilmekte ve sorgulamalar bu konum bilgisine göre belirlenen uzaklık çevresinde yapılabilmektedir. GPS konum bilgisi kapalı tutularak, manuel olarak elle enlem boylam bilgisi de girilebilmekte ve sorgulama yapılabilmektedir. Sorgulama sonucunda belirlenen konum çevresinde yer alan ilgi merkezleri bilgileriyle birlikte liste halinde görüntülenmektedir.



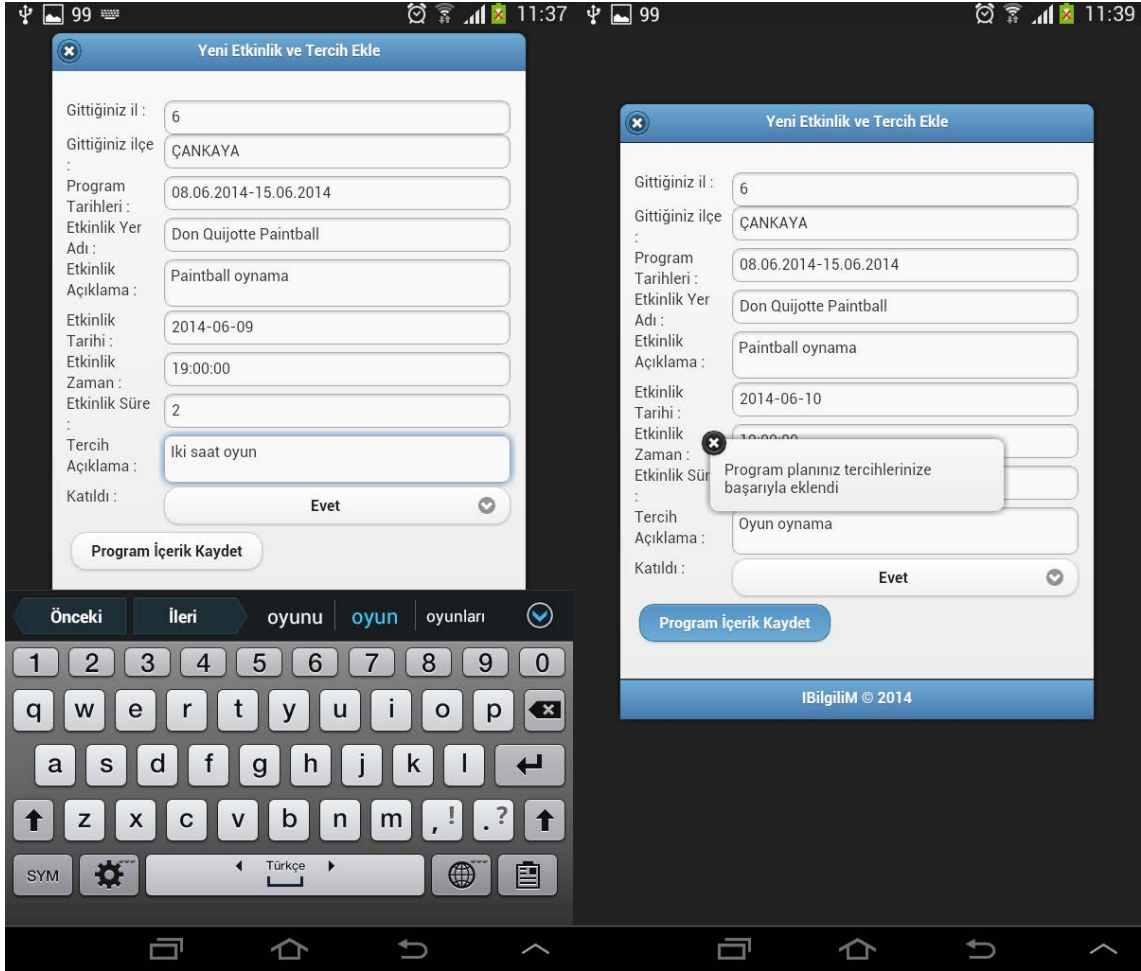
Resim 6.15. MAA uygulaması ilgi yeri sorgula (GPS) ekranı

Ekranda görüntülenen ilgi merkezlerine tıklanarak harita üzerinde görüntülenebilmekte programa uygun yer etkinliklerinden seçilebilmekte ve yer için yeni etkinlik tanımlayıp seçim yapılabilmektedir. Yer için yeni etkinlik tanımlanmak istendiğinde, serbest giriş ekranı olduğundan dolayı var olan ve ileriki programlar için ekleme yapılmaya izin verildiğinden kullanıcının aktif tüm programları görüntülenmekte ve istenilen programa ekleme yapılabilmektedir. Ancak eğer kullanıcı var olan etkinliklerden birini programa ekleme çalışır ve sonrasında gelen programlarda bu etkinlik tarihleriyle uyumsuz bir program seçerse sistem buna engel olmaktadır.



Resim 6.16. MAA uygulaması ilgi yeri sorgula (gps) – haritada göster ekranı

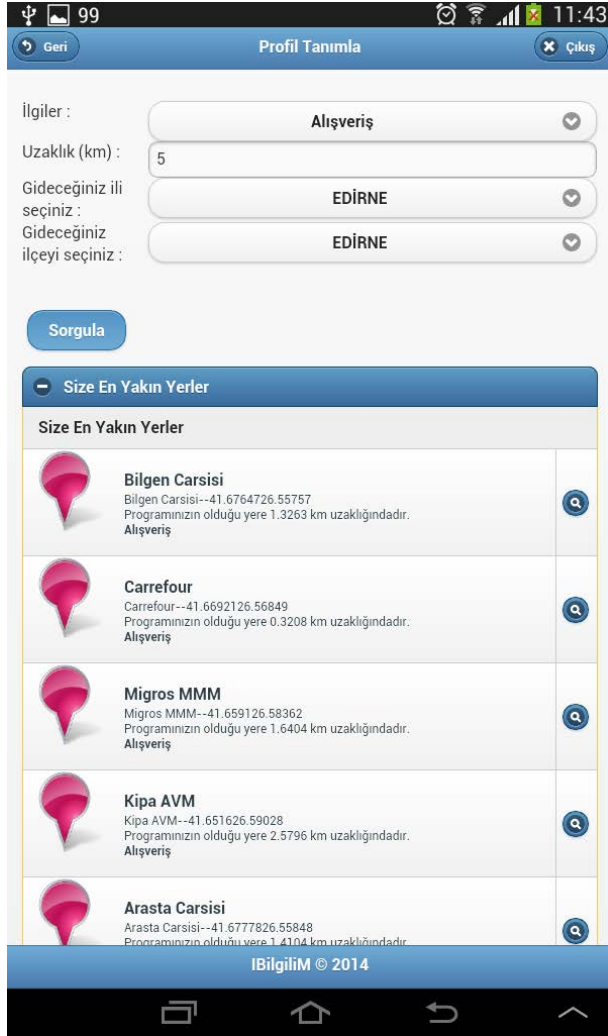
Yer için yeni etkinlik eklemek için program seçildikten sonra yeni etkinlik ve tercih ekle ekranı ile programa ilgili yerde kullanıcının belirlediği bir etkinlik tanımlanabilir ve kullanıcı tercihlerine bu etkinlik eklenebilir.



Resim 6.17. MAA uygulaması ilgi yeri sorgula (gps) – yer için yeni etkinlik ekle ekranı

### *Serbest İlgi Yeri Sorgula (İl İlçe)*

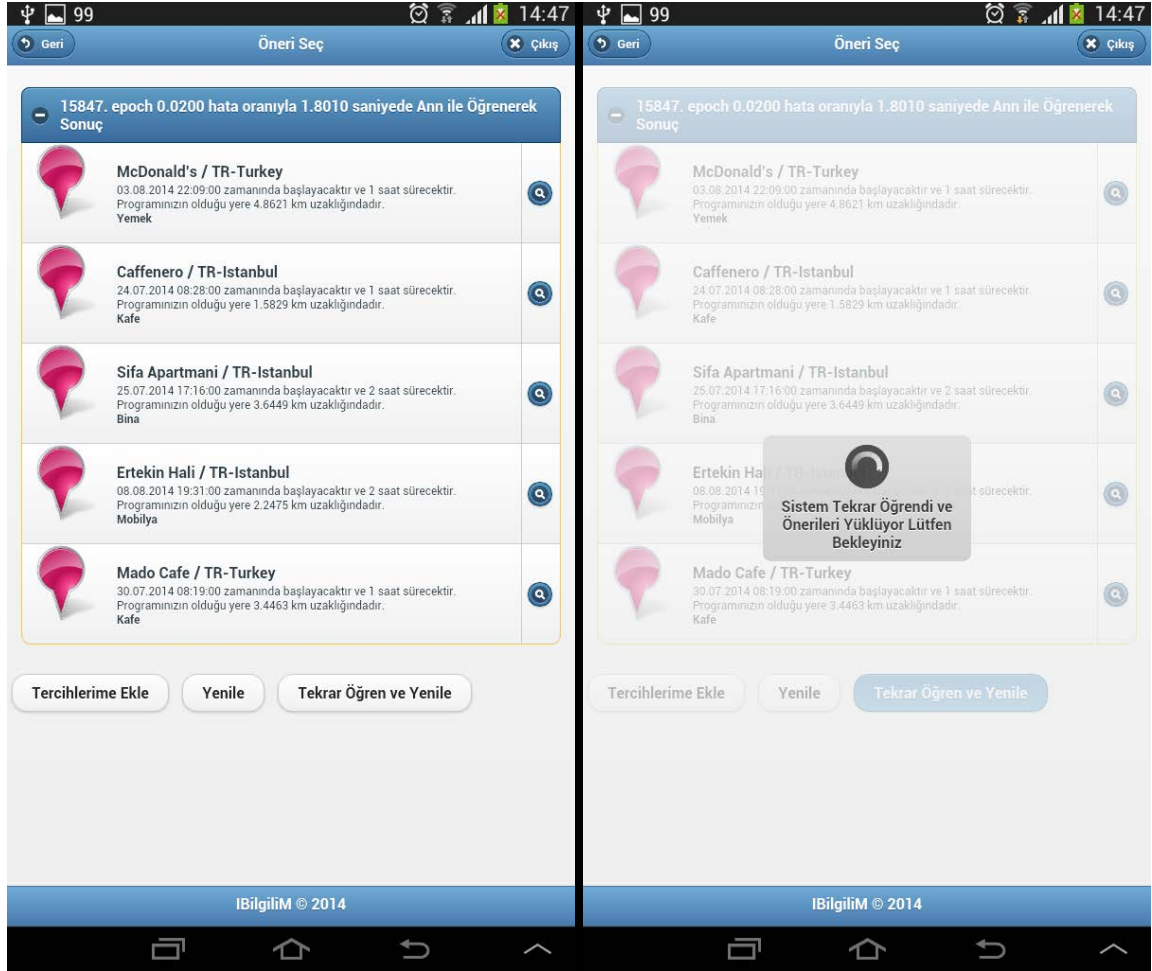
Bu ekran Serbest İlgi Yeri Sorgulama(GPS) ekranının tüm işlevselliğine sahiptir. Ancak burada sorgulama GPS konum bilgileri üzerinden değil kullanıcının belirleyeceği il ve ilçe üzerinden yapılmaktadır.



Resim 6.18. MAA uygulaması ilgi yeri sorgulama (il ilçe) ekranı

### *ANN Önerileri Getir*

Kullanıcı sisteme giriş yaptıktan sonra kullanıcının geçmiş tercihlerine göre öğrenmiş olan yapay sinir ağı, bu ekranda kullanıcının sistem girişinde belirlemiş olduğu programa göre yaptığı adaptif önerileri bu ekranda göstermektedir. Yapılan önerilerden katılmak istenilenler kullanıcı tarafından seçilip kullanıcı tercihlerine toplu olarak eklenebilmektedir. Yenile butonuyla yeni öneriler istenilebilir. Tekrar öğren ve yenile butonuyla ise kullanıcının tercih etmiş olduğu son tercihlerine göre yapay sinir ağı öğrenmesi yenilenir ve öneriler sıralanmaktadır. Liste başlığında ANN öğrenmesinin ne kadar hatayla ve kaçınıcı düğüm için gerçekleştiği bilgisi yer almaktadır.



Resim 6.19. MAA uygulaması ANN önerileri getir ekranı

Eğer kullanıcı program seçme ekranında önerileri bulunduğu yerden almak isteyipte kullanıcının bulunduğu yer ile seçtiği program konumları uyumsuzsa öğrenme gerçekleşir ancak konum uyumsuzluğundan kullanıcının seçebileceği öneri olmadığı için tüm uzaklıktaki öneriler aşağıdaki gibi görüntülenmektedir. İsterse kullanıcı program tarihlerine uyan ancak konum itibariyle program konumuna uzak etkinlikleri de inceleyebilmekte ve tercihlerine ekleyebilmektedir.



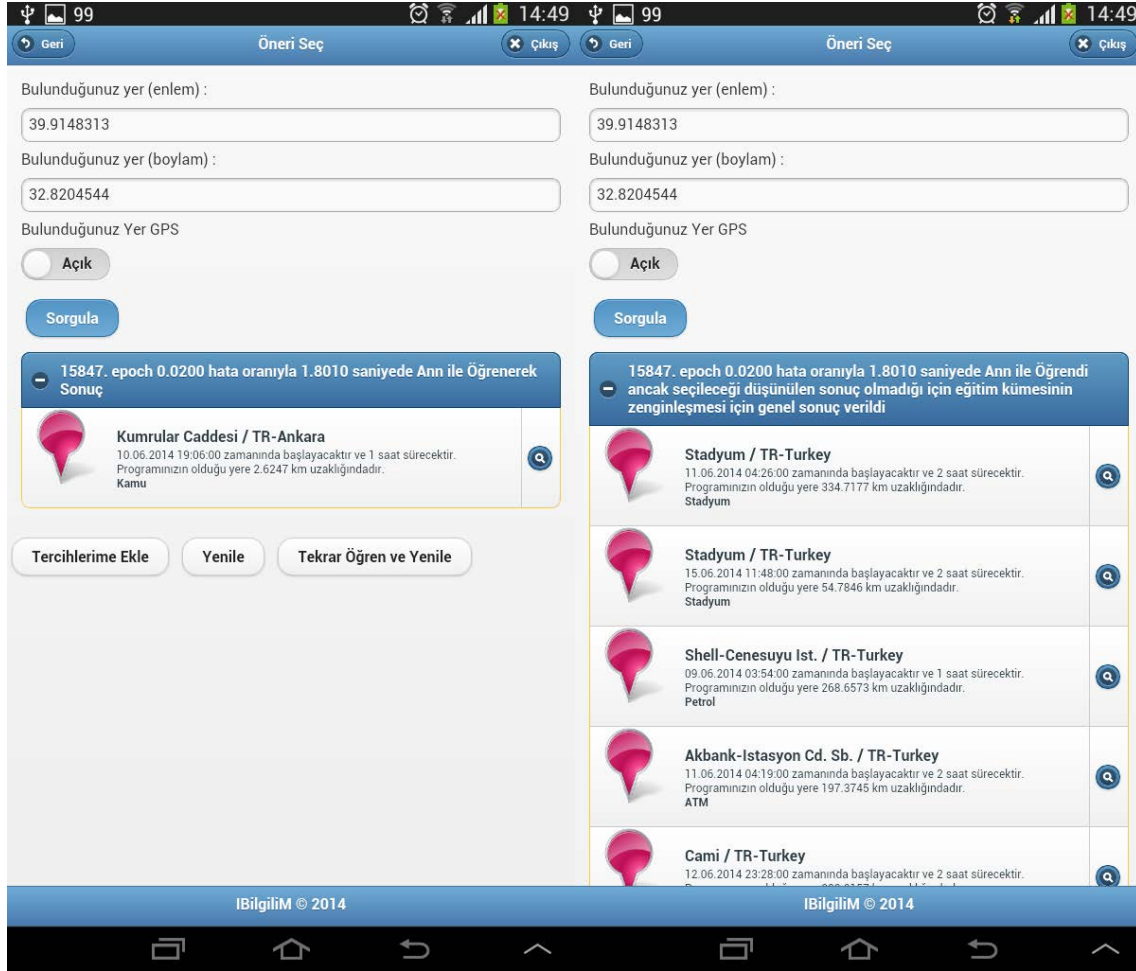


Resim 6.20. MAA uygulaması konum uyumsuzluk hali

Ekranda kullanıcıya programı için önerilen öneri sayısı ve programa olan uzaklığı kullanıcının Profil Sabitleri ekranında belirlediği Öneri maksimum kayıt sayısı ve uzaklık bilgilerine göre değişmektedir. Bu ekranda istenirse önerilerin haritadaki konumları da görüntülenebilmektedir.

### *ANN Önerileri Getir (GPS)*

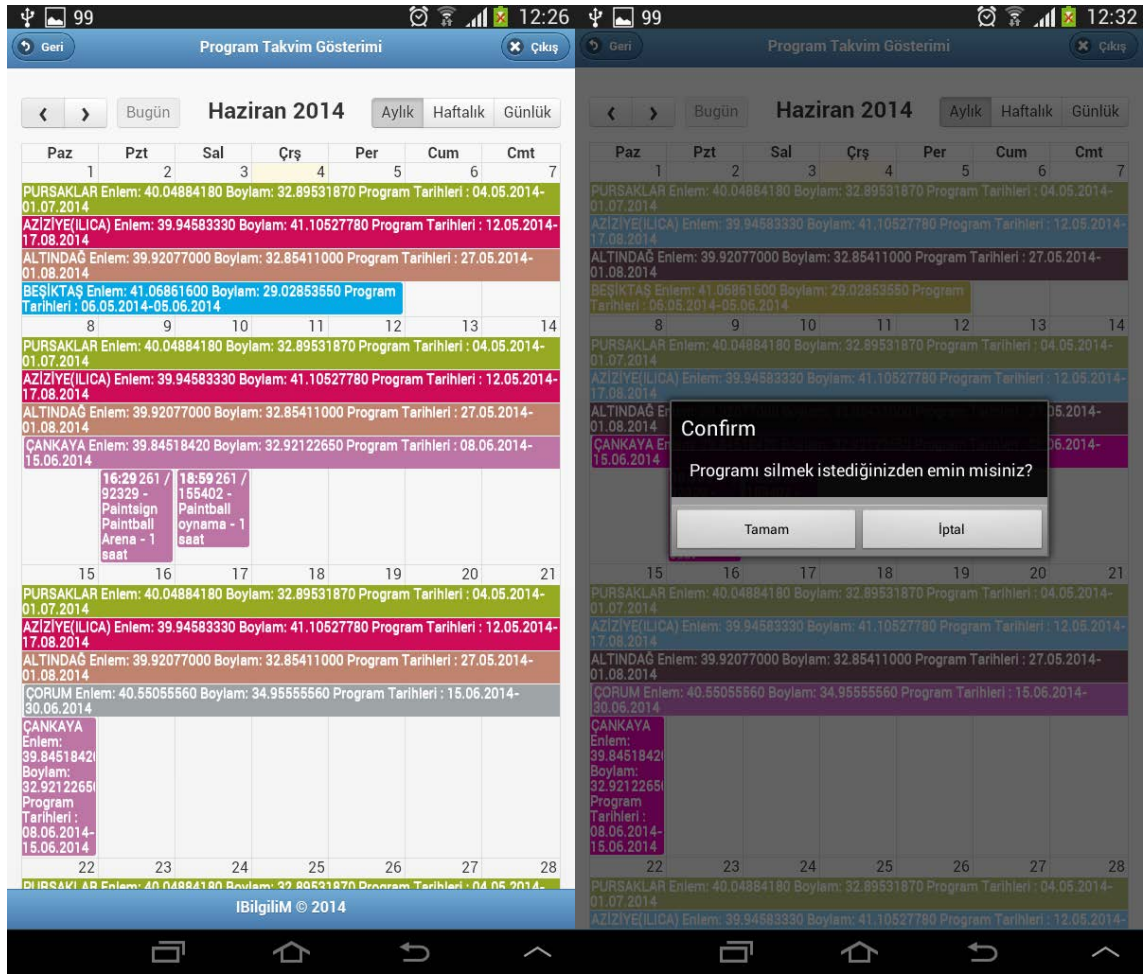
ANN Önerileri Getir ekranıyla işlevsellik bakımından aynı olan bu ekran sadece dahili GPS alıcısından alınan veya manuel olarak girilen konum bilgilerine göre önerileri yine adaptif yapay sinir ağları öğrenmesiyle yapan bölümdür. Bu ekranda da önerilerden istenilenler toplu olarak seçilip program tercihlerine eklenebilir.



Resim 6.21. MAA uygulaması ANN önerileri getir ekranı

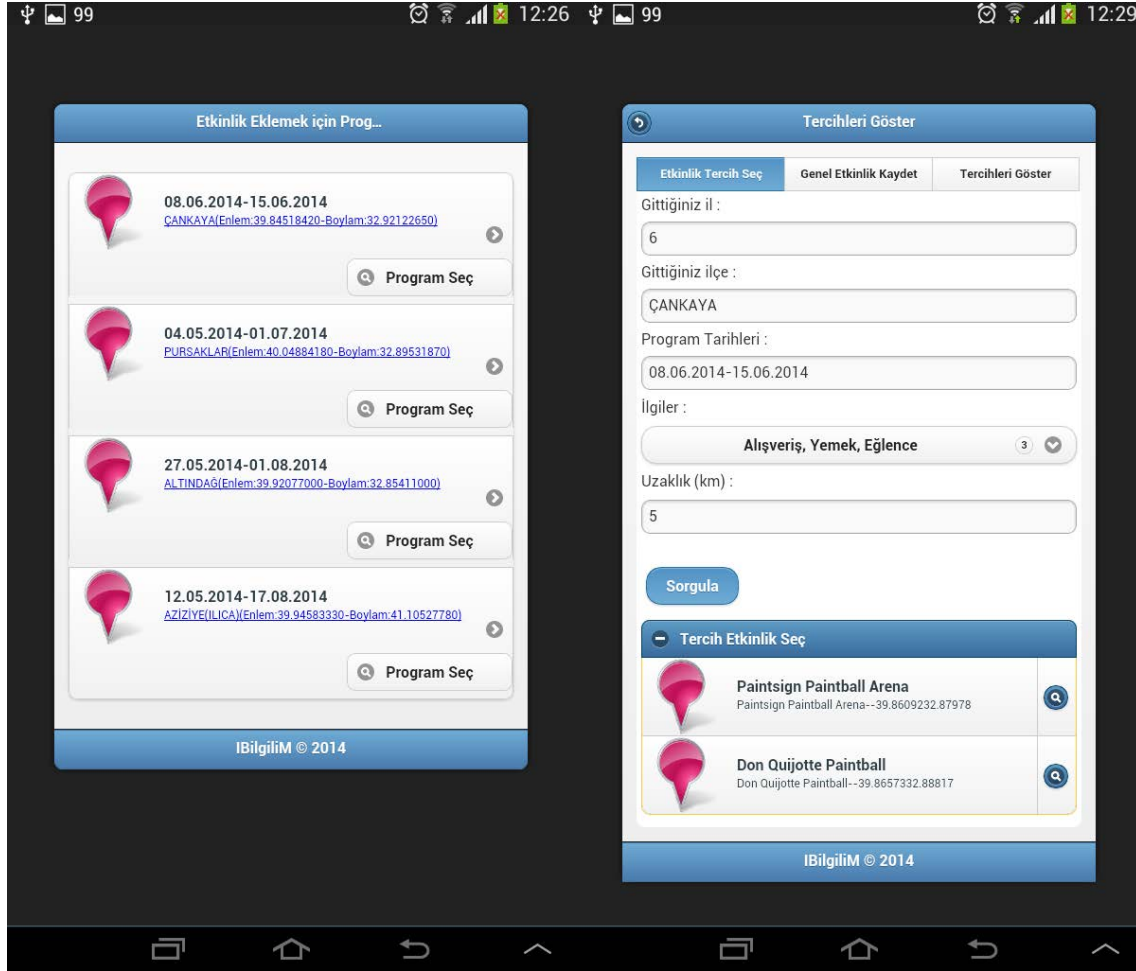
### *Programları Takvimde Göster*

Kullanıcının sistemde tanımlamış olduğu tüm programlarını özet bir şekilde takvim üzerinde görebildiği ekrandır. Programlar gerçekleştirilecekleri zaman aralıklarında rastgele üretilen farklı renklerle temsil edilmektedir. Silinmesi istenen programın üzerine tıklanarak tüm program ve bağlı etkinlikler silinebilmektedir.



Resim 6.22. MAA uygulaması programları takvimde göster ekranı

Ayrıca ekran üzerinde gün ve zaman aralıklarına tıklanarak etkinlik kaydı da yapılabilmektedir. Etkinlik tercihi ekleme mekanizması diğer ekranlarda olduğu gibidir. Seçilen tarih diliminde kayıtlı olan program listesi gelmekte, seçilen program listesine göre ilgilerin belirlendiği ve sorgulama yapılabildiği ekranlar üzerinden tercihler belirlenebilmektedir. İstenirse ilgilere bağlı olmadan programa kullanıcının kendi etkinliği tanımlanabilmekte ve seçilebilmektedir.



Resim 6.23. MAA uygulaması takvim etkinlik ekleme

### *Profil Sabitleri Göster*

Bu ekran aracılığıyla tüm uygulamada kullanılan parametre sabitleri görüntülenmekte ve güncellenmektedir. Bu parametreler arasında yapay sinir ağlarına ait sabitlerinin yanında öneri hatırlatma servisinin, hatırlatma aralığı, konum uzaklık bilgileri, GPS alıcının yer değişimini algılaması için gerekli zaman ve uzaklık bilgileri bulunmaktadır. Ayrıca hatırlatma servisinin yapacağı öneri hatırlatmasının adaptif mi yoksa sıradan mı yapılacağı seçimi de bu ekrandan belirlenmektedir. Bu ekran ile kullanıcıya uygulamanın işleyişini değiştiren ana parametreleri değiştirme imkânı tanınarak, uygulama kodlarında değişiklik yapılmaksızın farklı durumlarda uygulamanın nasıl davrandığı kontrolü kolayca yapılabilmektedir.

99 12:33

Geri Profil Sabitleri Güncelleme

Öneri: 0.8

Hatırlatma Aralık (dk):

Öneri Uzaklık (km): 3

GPS Aralık (dk): 1

GPS Uzaklık (km): 0.1

Adaptif Öneri: Evet

Servis Adresi: https://m.csgb.gov.tr/

Öneri Max Kayıt Sayısı: 10

Öneri Max Uzaklık: 5

**ANN Parametreleri**

ANN Öğrenme Oranı: 0.3

ANN Momentum: 0.7

ANN Gizli Katman Sayısı: 9

ANN Epoch Sayısı: 30000

ANN Hata Oranı: 0.02

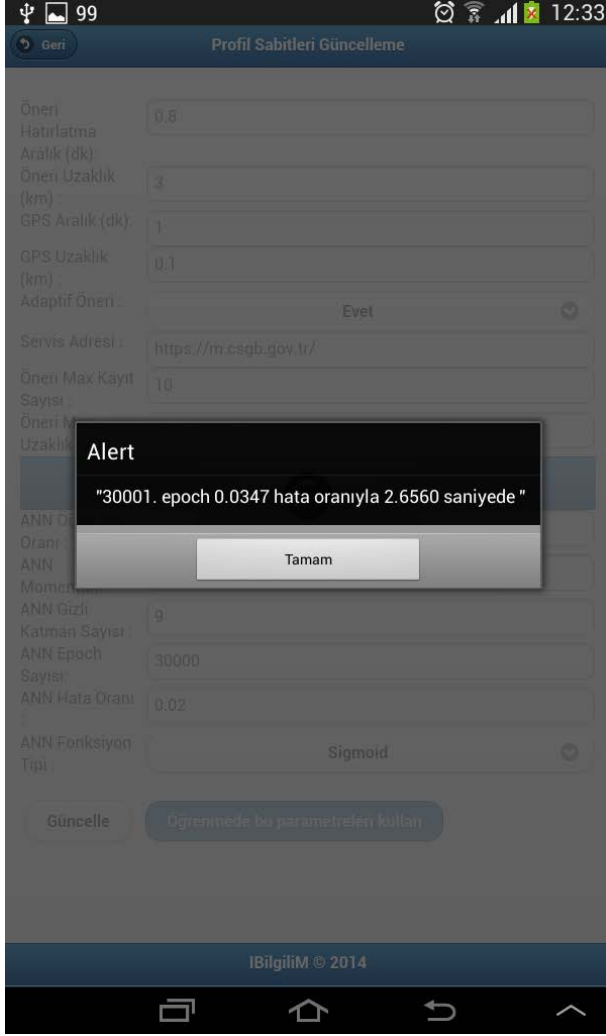
ANN Fonksiyon Tipi: Sigmoid

Güncelle Öğrenmede bu parametreleri kullan

IBilgiliM © 2014

Resim 6.24. MAA uygulaması profil sabitleri göster ekranı

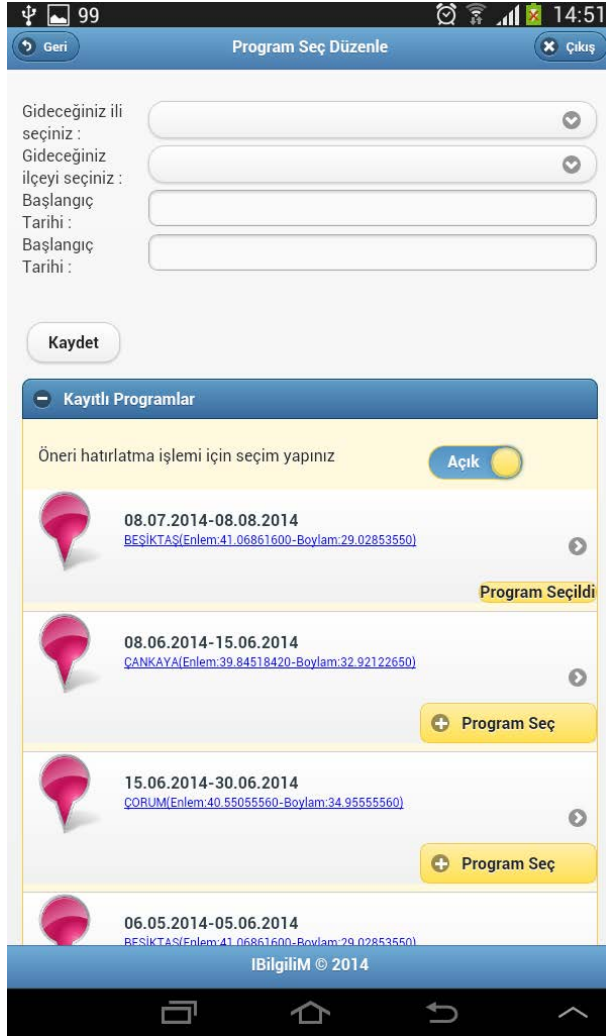
Yapay sinir ağları için parametreler değiştirildiğinde eğer öğrenmede bu parametrelerin kullanılması isteniyorsa “Öğrenmede bu parametreleri kullan” seçiminin yapılması gerekmektedir. Bu seçim yapıldığında öğrenmenin hangi özelliklerle gerçekleştirildiğine dair bu mesaj gelmektedir. Bu mesaj öğrenmenin bundan sonra yeni parametrelerle yapılacağı anlamına gelmektedir. Yapay sinir ağları parametreleri de değiştirilerek uygulama kodlarında herhangi bir değişiklik yapılmadan öğrenmenin parametrelere göre değişimi gözlenebilmekte ve en etkin öğrenme parametreleri kolayca belirlenebilmektedir.



Resim 6.25. MAA uygulaması öğrenmede bu parametreleri kullan seçimi

### *Öneri Alınacak Program Görüntüle/Değiştir*

Sisteme giriş yaptıktan sonraki program seçme ekranından seçilmiş olan program değiştirilmek istendiğinde bu ekran kullanılır. Program değişimi yapıldığında hatırlatma servisinin yapacağı öneriler artık yeni program için olacaktır.



Resim 6.26. MAA uygulaması öneri alınacak program görüntüle/değiştir ekranı

Ayrıca MAA uygulamasında bir hatırlatma servisi bulunmaktadır. Program seçme ve değiştirme ekranlarından hatırlatma servisi aktif ya da pasif hale getirilebilmektedir. Kullanıcı programı seçtiğinde önerileri almak istediği yere göre ve profil sabitlerinde bulunan adaptif önerinin açık yada kapalı olmaması durumuna göre hatırlatma servisi farklı metotları çalıştırmaktadır. Eğer önerilerin alınmak istendiği yer “gideceğim yer” olarak seçilmişse, hatırlatma servisi hatırlatma aralığı ve yapacağı önerilerde kullanacağı uzaklık bilgilerini profil sabitlerinden alarak aktif hale getirilir. Bunun için Android üzerinde çalışan java kodu şu şekildedir;

```
public class ReminderThread extends Thread implements
Runnable{
private Context context;
public boolean isRunning = false;
public static long interval;
```

```

public ReminderThread(Context context, long interval){
    isRunning = true;
    this.context = context;
    this.interval = interval;
}
@Override
public void run() {
    while (isRunning){
        try {
            Application application =
(Application)MyApplication.getContext();
            MyApplication app =
(MyApplication)application;
            Intent newIntent = null;

            if(app.getKullanici().getProgramSabitleri().getAdaptif().equals(Sabitler.HAYIR)){
                newIntent = new
Intent(context.getApplicationContext(),
PopupActivity.class);
            }else{
                newIntent = new
Intent(context.getApplicationContext(),
PopupAnnActivity.class);
            }

            newIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

            newIntent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

            context.getApplicationContext().startActivity(newIntent)
;

                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
}

```

Eğer önerilerin alınmak istendiği yer “bulduğum yer” olarak seçilmişse, mobil cihazın GPS alıcısı aktif hale getirilir ve profil sabitlerinde bulunan hatırlatma aralık dakika bilgisine ve GPS konum değişikliğinde meydana gelecek değişikliğe göre öneri tetiklenmektedir. GPS alıcısının aranacağı alıcı Android platformu üzerinde şu şekilde kodlanır.



```

public void registerReceiver(){
    messageReceiver = new LocationReceiver();
    registerReceiver(messageReceiver, new
IntentFilter("LOCATION_UPDATED"));
    AlertDialogManager alert = new AlertDialogManager();

    long minDistanceForUpdates =
Math.round(kullanici.getProgramSabiti().getGpsUzaklik()*1000
);//metre
    long minTimeForUpdates =
Math.round(kullanici.getProgramSabiti().getGpsAralik()*60*10
00);//ms
    gps = new GPSTracker(this,
minDistanceForUpdates,minTimeForUpdates);
    if (gps.canGetLocation()) {
        Log.d("Yeriniz", "latitude:" + gps.getLatitude() +
", longitude: " + gps.getLongitude());
    } else {
        alert.showAlertDialog(getApplicationContext(), "GPS
Durumu","Yer Bilgisi alınamadı. Lütfen GPS aktif yapın",
false);
        return;
    }
}
}

```

GPS alıcısından bilgileri almak için ise geliştirilen java metotları şu şekildedir;

```

public Location getLocation(long minDistanceForUpdates, long
minTimeForUpdates) {
    boolean gpsCanGetLocation = false;
    try {
        locationManager = (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);
        isGPSEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVID
ER);
        isNetworkEnabled =
locationManager.isProviderEnabled(LocationManager.NETWORK_PR
OVIDER);
        if (!isGPSEnabled && !isNetworkEnabled) {
        } else {
            this.canGetLocation = true;
            if (isGPSEnabled) {
                if (location == null) {

                    locationManager.requestLocationUpdates(LocationManager.G
PS_PROVIDER, minTimeForUpdates, minDistanceForUpdates,
this);

```

```

        Log.d("GPS", "GPS Enabled");
        if (locationManager != null) {
            if(mobileLocation==null){
                location =
locationManager.getLastKnownLocation(LocationManager.GPS_PRO
VIDER);

                if (location != null) {
                    latitude =
location.getLatitude();
                    longitude =
location.getLongitude();
                }

                gpsCanGetLocation = false;
            }else{
                latitude =
mobileLocation.getLatitude();
                longitude =
mobileLocation.getLongitude();

                if(mobileLocation.getProvider().equals("gps"))
gpsCanGetLocation = true;
                else gpsCanGetLocation =
false;
            }
        }
    }
}
if (isNetworkEnabled) {

    locationManager.requestLocationUpdates(LocationManager.N
ETWORK_PROVIDER, minTimeForUpdates,
minDistanceForUpdates, this);
    Log.d("Network", "Network Enabled");
    if (locationManager != null) {
        if(mobileLocation==null){
            location =
locationManager.getLastKnownLocation(LocationManager.NETWORK
_PROVIDER);

            if (location != null) {
                latitude = location.getLatitude();
                longitude = location.getLongitude();
            }
        }else {
            if(gpsCanGetLocation == false){
                latitude =
mobileLocation.getLatitude();
                longitude =
mobileLocation.getLongitude();
            }
        }
    }
}

```

```

        }
    }
}

} catch (Exception e) {
    e.printStackTrace();
}
return location;
}

@Override
public void onLocationChanged(Location location) {
    if(location!=null) {
        this.mobileLocation = location;
        Intent i = new Intent( "LOCATION_UPDATED");

        LocationVO loc = new LocationVO();
        loc.setLatitude(location.getLatitude());
        loc.setLongitude(location.getLongitude());
        loc.setAccuracy(location.getAccuracy());
        loc.setAltitude(location.getAltitude());
        Date date = new Date( location.getTime());
        android.text.format.DateFormat df = new
android.text.format.DateFormat();
        String time = "Tarih Zaman: " +
df.format("dd.MM.yyyy hh:mm:ss", date).toString();
        loc.setTime(time);
        loc.setBaglantiTipi(location.getProvider());

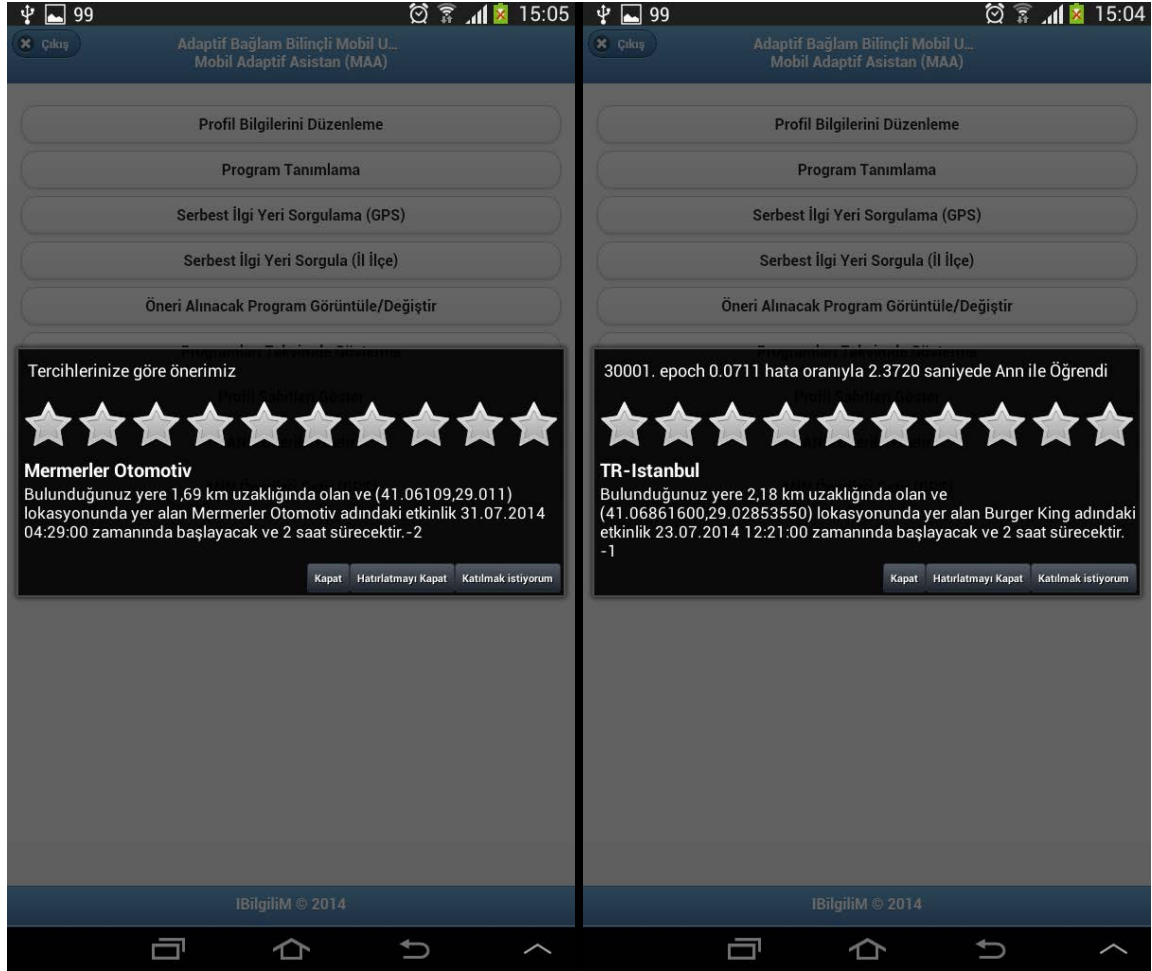
        i.putExtra("mobileLocation",loc);
        mContext.sendBroadcast(i);
    }else mobileLocation = null;
}
}

```

Metotlarda yer alan `minTimeForUpdates`, `minDistanceForUpdates` parametreleri profil sabitlerinden okunmaktadır.

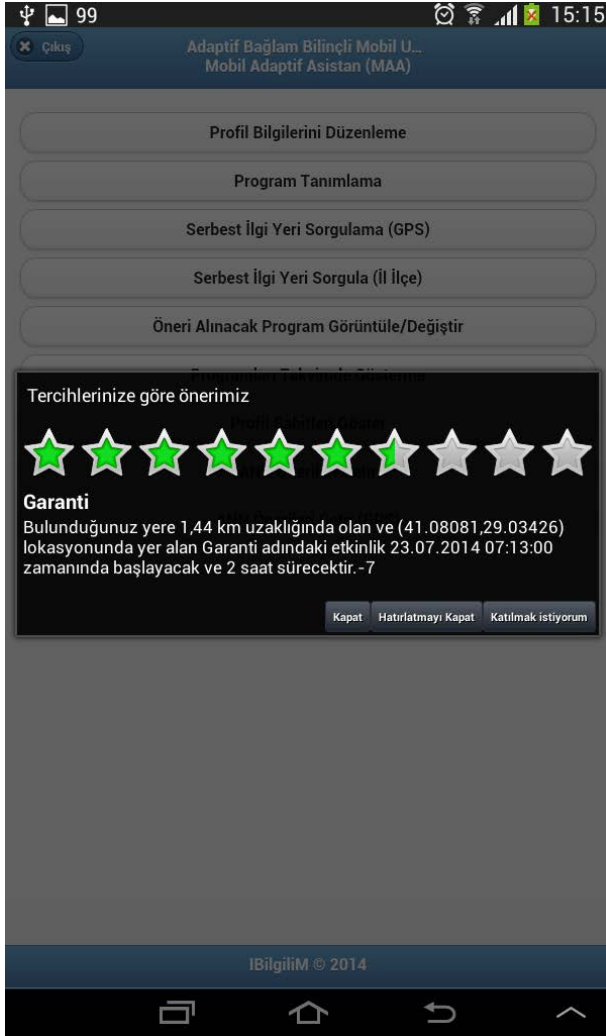
Öneri hatırlatma işlemi aktif hale getirildiğinde gerekli tüm parametreler profil sabitlerinden okunarak kullanıcının önerileri almak istediği yer tipine göre hatırlatma servisi aktif hale getirilecektir.

Öneri yer tipinin gideceğim yer olduğu durumda öneriler belirli aralıklarla kullanıcıya aşağıdaki şekilde görüntülenecektir. İki farklı hatırlatma ekranı profil sabitlerinden adaptif önerinin kapalı ve açık olması durumuna göre değişmektedir.



Resim 6.27. MAA uygulaması öneri hatırlatma servisi ANN kapalı ve açık

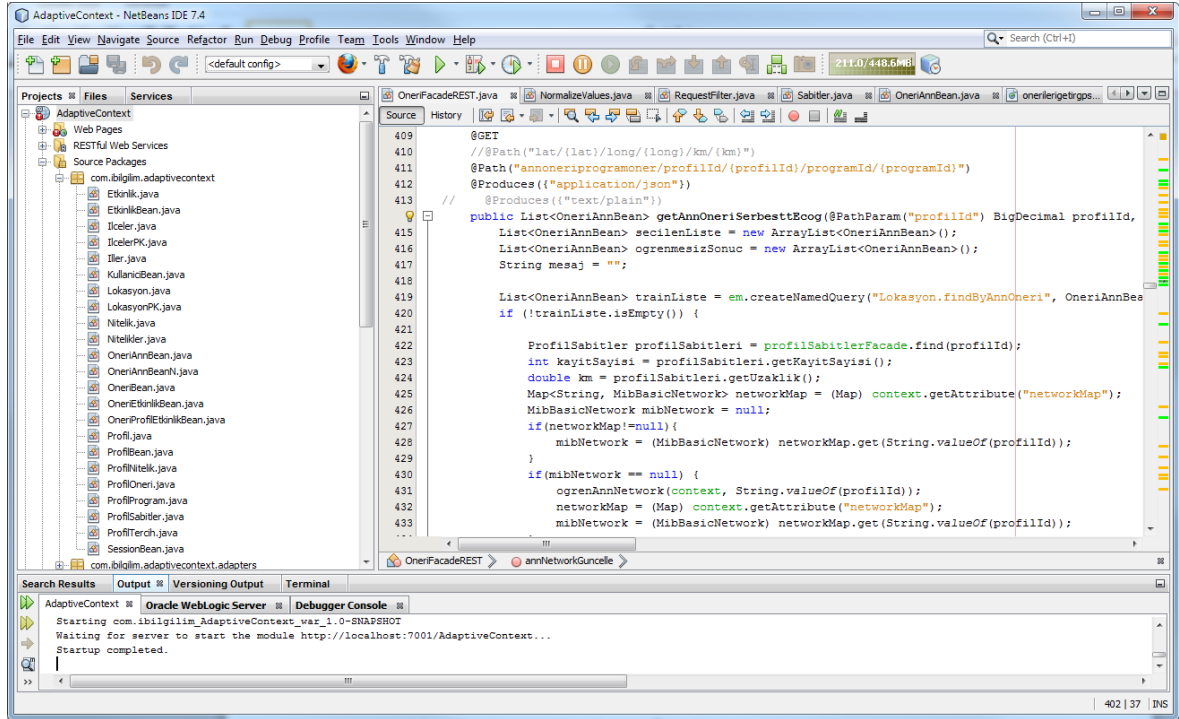
İstenirse öneri yapılan ekrandan etkinlik için puanlama yapılabilmekte ve sisteme girişteki seçilen program içine etkinlik eklenebilmektedir.



Resim 6.28. MAA uygulaması öneri hatırlatma servisi puanlama

### 6.5.2. Web sunucu uygulaması

Web Sunucu Modülü Uygulaması, Java programlama diliyle Netbeans IDE'si kullanılarak geliştirilmiştir. Uygulama WebLogic uygulama sunucusunda koşturmaktadır. Uygulama mobil istemcide bulunan tüm ekranlardaki her işlem için restful web servislerini içermektedir. Uygulama üzerinde kullanıcı arayüzü bulunmamakla birlikte istenirse mobil istemcide kullanılan kullanıcı ara yüzleri cihaz kaynaklarını kullanmayı gerektiren durumlar hariç web sunucunda da çalıştırabilir.



Resim 6.29. Netbeans IDE

Web sunucu uygulaması çalıştırıldığında saatte bir uygulama üzerindeki yaşam süresi dolmuş kullanıcıları belirleyip öldürecek servis ve uygulama yolunun sonunda “/adaptive” kelimesi bulunan URI’leri karşılayacak filtreleyici çalışmaya başlamaktadır. Uygulama servislerinin güvenliği için karşılayıcı bir filtreleyici geliştirilmiş, servisler gelen tüm istekler bu filtreleyici tarafından karşılanmakta, herkesin yetkili olduğu servisler hariç herhangi bir istek geldiğinde parametre olarak kullanıcı oturum id’si aranmaktadır. Aksi takdirde istek yapana hizmet verilmemektedir. Kullanıcı oturum id’si yetkili kullanıcı sisteme ilk giriş olduğunda benzersiz olarak üretilmekte ve yaşam süresi boyunca hafızada saklanmaktadır. Ayrıca giriş başarılı olduğu andan itibaren kullanıcının yaşam süresi boyunca hafızada saklanan yapay sinir ağları eğitimi yapıp saklanır. Sisteme ilk giriş, oturum id üretilmesi ve yapay sinir ağlarının eğitiminin sağlandığı web servis metodu aşağıdaki şekildedir.

```
@PUT
@Path("giris")
@Consumes({"application/json"})
@Produces({"application/json"})
public Profil getProfilByEmailAndParola(KullaniciBean
kullanici, @Context HttpServletRequest req) throws
Exception{
    try {
```

```

        Profil profil =
em.createNamedQuery("Profil.findByEmailParola",
Profil.class).setParameter("email",
kullanici.getEmail()).setParameter("parola",
kullanici.getParola()).getSingleResult();
        Map<String, Profil> kullaniciMap = (Map)
req.getServletContext().getAttribute("kullaniciMap");
        if(kullaniciMap == null){
            kullaniciMap = new HashMap<String,
Profil>();
                UUID idOne = UUID.randomUUID();
                SessionBean sessionBean = new
SessionBean(Sabitler.SESSIONBEAN_LIFE_SECONDS,
idOne.toString());
                profil.setSessionBean(sessionBean);
                kullaniciMap.put(idOne.toString(),
profil);

req.getServletContext().setAttribute("kullaniciMap",
kullaniciMap);
        }else{
            UUID idOne = UUID.randomUUID();
            SessionBean sessionBean = new
SessionBean(Sabitler.SESSIONBEAN_LIFE_SECONDS,
idOne.toString());
            profil.setSessionBean(sessionBean);
            kullaniciMap.put(idOne.toString(),
profil);

req.getServletContext().setAttribute("kullaniciMap",
kullaniciMap);
        }

oneriFacade.annNetworkGuncelle(profil.getProfilId());
        return profil;
    } catch (NoResultException e) {
        return null;
    }
}

```

Yapay sinir ağlarının kullanıcı bazında eğitilip bellekte saklandığı java metodu şu şekildedir;

```

private String ogrenAnnNetwork(ServletContext
servletContext, String profilIds) {
    Map<String, MibBasicNetwork> networkMap = (Map)
context.getAttribute("networkMap");
    String mesaj = "";
    BigDecimal profilId = new BigDecimal(profilIds);

```

```

        List<OneriAnnBean> trainListe = new
ArrayList<OneriAnnBean>();
        List<Profil> profiller = profil.findAll();
        for (Profil profil : profiller) {
            List<OneriAnnBean> list =
em.createNamedQuery("Lokasyon.findByAnnOneri",
OneriAnnBean.class).setParameter("profilId",
profil.getProfilId()).getResultList();
            trainListe.addAll(list);
        }

        if (!trainListe.isEmpty()) {
            trainListe =
NormalizeValues.normalizeOnerilerTrain(trainListe);
            ProfilSabitler profilSabitleri =
profilSabitlerFacade.find(profilId);
            int kayitSayisi =
profilSabitleri.getKayitSayisi();
            int inputNeurons = 10;
            int outputNeurons = 1;
            ActivationFunction fonkTipi =
Util.getAnnEcogFonkTipi(profilSabitleri.getAnnFonkTipi());
            MibBasicNetwork network = new
MibBasicNetwork();
            network.addLayer(new BasicLayer(fonkTipi,
true, inputNeurons));
            network.addLayer(new BasicLayer(fonkTipi,
true, profilSabitleri.getAnnGizliKatman()));
            network.addLayer(new BasicLayer(fonkTipi,
false, outputNeurons));
            network.getStructure().finalizeStructure();
            network.reset();
            MLDataSet trainingSet = new
BasicMLDataSet();

            for (OneriAnnBean oneriAnnBean : trainListe)
            {
                OneriAnnBeanN annOneri =
oneriAnnBean.getNormalizedValues();
                double[] girisVeriler =
annOneri.getGirisVeriler();
                double[] cikisVeriler =
annOneri.getCikisVeriler();
                MLData input = new
BasicMLData(girisVeriler);
                MLData output = new
BasicMLData(cikisVeriler);
                MLDataPair pair = new
BasicMLDataPair(input, output);
                trainingSet.add(pair);
            }
        }
    }
}

```



```

        long started = System.currentTimeMillis();
        //ResilientPropagation train = new
ResilientPropagation(network, trainingSet);
        Backpropagation train = new
Backpropagation(network, trainingSet,
profilSabitleri.getAnnOgrenmeOran(),
profilSabitleri.getAnnMomentum());
        int epoch = 1;
        double hataOrani = 1.0;
        do {
            train.iteration();
            hataOrani = train.getError();
            if (hataOrani > train.getError()) {
                hataOrani = train.getError();
            }
            if (epoch >
profilSabitleri.getAnnEpoch()) {
                break;
            }
            epoch++;
            //System.out.println(epoch+". epoch "+
hataOrani);
        } while (train.getError() >=
profilSabitleri.getAnnHataOrani());
        train.finishTraining();

        Encog.getInstance().shutdown();
        long finish = System.currentTimeMillis();
        long elapsed = finish - started;
        double e2 = elapsed / 1000.0;
        NumberFormat f =
NumberFormat.getNumberInstance();
        f.setMinimumFractionDigits(4);
        f.setMaximumFractionDigits(4);
        mesaj = epoch + ". epoch " +
f.format(hataOrani) + " hata oranıyla " + f.format(e2) + "
saniyede ";
        network.setMesaj(mesaj);
        //System.out.println(mesaj);
        //servletContext.setAttribute("annNetwork",
network);
        if(networkMap==null){
            networkMap = new HashMap<String,
MibBasicNetwork>();
            networkMap.put(profilIds, network);
        }else{
            if (networkMap.containsKey(profilIds)) {
                networkMap.remove(profilIds);
                networkMap.put(profilIds, network);
            } else {
                networkMap.put(profilIds, network);
            }
        }
    }
}

```

```

        }
    }
    servletContext.setAttribute("networkMap",
networkMap);
    }
    return mesaj;
}

```

Yapay Sinir Ağları eğitimi yapılırken kullanılan 11 adet öznitelik verileri normalize edilerek eğitimde kullanılmıştır. Normalize edilirken tüm örneklerde o verilerin aldığı en büyük değerle, en küçük değer belirlenmiş ve değer en küçük değerden çıkarılıp, en büyük ve en küçük değer farkına orantılanmıştır. Kullanıcının tercihine katılıp katılmadığı bilgisi için ise E değeri 1, H değeri 0 olarak kabul edilmiştir. Verilerin toplu olarak normalizesi için kullanılan java metodu şu şekildedir;

```

public static double normalizeValues(double value,
double[] values){
    double minValue = Util.minArray(values);
    double maxValue = Util.maxArray(values);
    double normalizedValue = 0.0;
    if(minValue != maxValue )
        normalizedValue = (value - minValue)/(maxValue-
minValue);
    return normalizedValue;
}

```

Yapay sinir ağlarının sonuç değeri 1 ise kullanıcının öneriyi kabul edeceği 0 ise kabul etmeyeceği tahmini yapılmıştır. Örnek olarak kullanıcının programına göre adaptif öneri yapan ve sonuçları listeleyen servis metodu şu şekildedir.

```

@GET
@Path("annoneriprogramoner/profilId/{profilId}/programId/{pr
ogramId}")
@Produces({"application/json"})
public List<OneriAnnBean>
getAnnOneriSerbesttEcog(@PathParam("profilId") BigDecimal
profilId, @PathParam("programId") BigDecimal programId) {
    List<OneriAnnBean> secilenListe = new
ArrayList<OneriAnnBean>();
    List<OneriAnnBean> ogrenmesizSonuc = new
ArrayList<OneriAnnBean>();
    String mesaj = "";
}

```

```

        List<OneriAnnBean> trainListe =
em.createNamedQuery("Lokasyon.findByAnnOneri",
OneriAnnBean.class).setParameter("profilId",
profilId).getResultList();
        if (!trainListe.isEmpty()) {
            ProfilSabitler profilSabitleri =
profilSabitlerFacade.find(profilId);
            int kayitSayisi =
profilSabitleri.getKayitSayisi();
            double km = profilSabitleri.getUzaklik();
            Map<String, MibBasicNetwork> networkMap = (Map)
context.getAttribute("networkMap");
            MibBasicNetwork mibNetwork = null;
            if(networkMap!=null){
                mibNetwork = (MibBasicNetwork)
networkMap.get(String.valueOf(profilId));
            }
            if(mibNetwork == null) {
                ogrenAnnNetwork(context,
String.valueOf(profilId));
                networkMap = (Map)
context.getAttribute("networkMap");
                mibNetwork = (MibBasicNetwork)
networkMap.get(String.valueOf(profilId));
            }

            mesaj = mibNetwork.getMesaj();

            List<OneriAnnBean> testListe = new
ArrayList<OneriAnnBean>();
            try {
                Query query =
em.createNamedQuery("Lokasyon.findByAnnProfilEtkinliklerSerb
est", OneriAnnBean.class);
                query.setParameter("profilId", profilId);
                query.setParameter("programId", programId);

                ogrenmesizSonuc = query.getResultList();
                List<OneriAnnBean> secilenSonuclar = new
ArrayList<OneriAnnBean>();
                for (OneriAnnBean oneriAnnBean :
ogrenmesizSonuc) {
                    double kmm =
GpsUtil.distance(Double.parseDouble(oneriAnnBean.getLokasyon
Enlem()),
Double.parseDouble(oneriAnnBean.getLokasyonBoylam()),
Double.parseDouble(oneriAnnBean.getEnlem()),
Double.parseDouble(oneriAnnBean.getBoylam()), 'K');
                    if (kmm <= km) {
                        oneriAnnBean.setUzaklik(kmm);
                    }
                }
            }
        }
    }
}

```

```

        secilenSonuclar.add(oneriAnnBean);
    }
}

if(!secilenSonuclar.isEmpty()){
    for(int i=1;i<=kayitSayisi;i++){

testListe.add(secilenSonuclar.get(new
Random().nextInt(secilenSonuclar.size())));
    }
}
if (!testListe.isEmpty()) {
    testListe =
NormalizeValues.normalizeOnerilerTest(testListe);
    MLDataSet testSet = new
BasicMLDataSet();
    for (OneriAnnBean oneriAnnBean :
testListe) {

        OneriAnnBeanN annOneri =
oneriAnnBean.getNormalizedValues();
        double[] girisVeriler =
annOneri.getGirisVeriler();
        double[] cikisVeriler =
annOneri.getCikisVeriler();
        MLData input = new
BasicMLData(girisVeriler);
        MLData output = new
BasicMLData(cikisVeriler);
        MLDataPair pair = new
BasicMLDataPair(input, output);
        testSet.add(pair);

        final MLData networkOutput =
mibNetwork.compute(pair.getInput());

        if
(NormalizeValues.quantizeResults(networkOutput.getData())[0]
== 1.0) {
            secilenListe.add(oneriAnnBean);
        }
    }
}

} catch (Exception e) {
    e.printStackTrace();
}

if (secilenListe.isEmpty()) {

```

```

        mesaj += "Ann ile Öğrendi ancak seçileceği
düşünülen sonuç olmadığı için eğitim kümesinin zenginleşmesi
için genel sonuç verildi";
        if (testListe.isEmpty()) {
            for(int i=1;i<=kayitSayisi;i++){
                OneriAnnBean oneriAnnBean =
ogrenmesizSonuc.get(new
Random().nextInt(ogrenmesizSonuc.size()));
                double kmm =
GpsUtil.distance(Double.parseDouble(oneriAnnBean.getLokasyon
Enlem()),
Double.parseDouble(oneriAnnBean.getLokasyonBoylam()),
Double.parseDouble(oneriAnnBean.getEnlem()),
Double.parseDouble(oneriAnnBean.getBoylam()), 'K');
                oneriAnnBean.setUzaklik(kmm);
                testListe.add(oneriAnnBean);
            }
        }
        testListe.get(0).setMesaj(mesaj);
        return testListe;
    } else {
        mesaj += "Ann ile Öğrenerek Sonuç";
        secilenListe.get(0).setMesaj(mesaj);
        return secilenListe;
    }
} else {
    trainListe =
em.createNamedQuery("Lokasyon.findByAnnOneriSerbestNotEtk",
OneriAnnBean.class).setParameter("programId",
programId).getResultList();
    mesaj += "Kişi programına uygun sonuç
önerilemeyeceği için genel öneri listesinden öneriler
verildi";
    trainListe.get(0).setMesaj(mesaj);
    return trainListe;
}
}
}

```

## 6.6. Yapay Sinir Ağları ile Öğrenme

Yapay Sinir Ağları, insan beyninin biyolojik sinir ağlarının, bilgi sistemleri için modellenmesiyle ortaya çıkan bilgisayar sistemleridir. İlk kez Donald Hebb (1949) tarafından yapılan beynin nasıl çalıştığı ile ilgili çalışmalarda, beynin iki sinir hücresinin nasıl birbiriyle korelasyon sergiledikleri incelenmiş ve sinir ağları teorisi bu temel üzerine inşa edilmiştir. Bu çalışma günümüzde hala devam eden beynin nasıl çalıştığı ile ilgili

çalışmaların temelini oluşturmaktadır. Bu temel kullanılarak günümüzde oldukça başarılı sonuçlar veren farklı teorilerle geliştirilmiş yapay sinir ağ modelleri mevcuttur.

### **6.6.1. Yapay sinir ağları mimarisi**

Yapay Sinir Ağlarının bilgi sistemleri için modellenmesi yazılım ya da donanım için olabilmektedir. Bilgi sistemlerinin modellenmesi için kullanılan yapay sinir ağlarını anlayabilmek için öncelikle biyolojik sinir hücresinin, nöronların incelenmesi gerekmektedir. Biyolojik sinir sistemlerinin yanında yapay sinir ağlarının da temelini oluşturan olan nöronların yapısı dört ana bölümden oluşmaktadır; dendrit, akson, çekirdek ve Sinapsis bağlantılar.

#### Dendrit

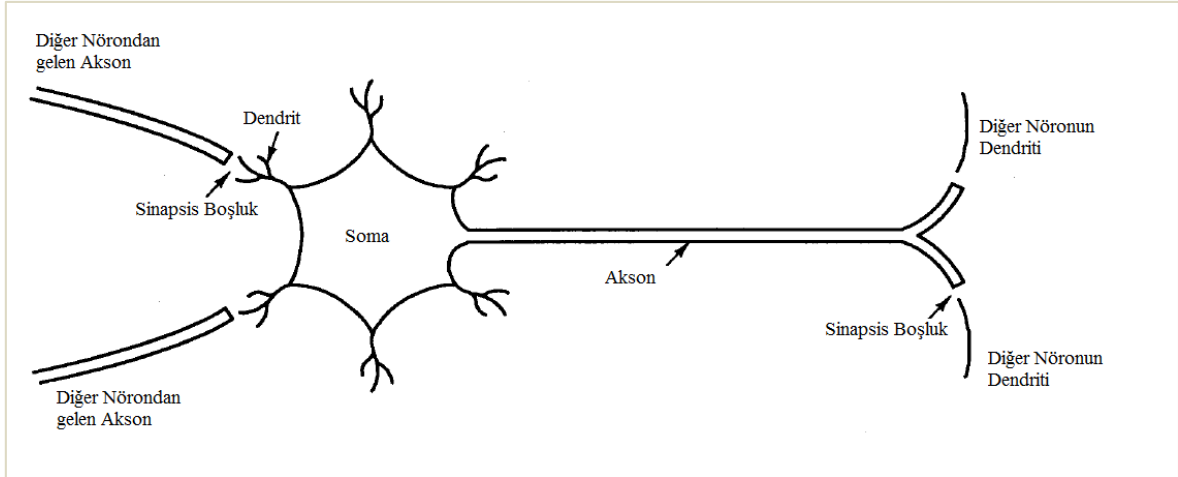
Dendritler, sinir hücresinin ucunda bulunan ve ağaç kökü görünümüne sahip bir yapıya sahiptir. Dendritlerin görevi bağlı olduğu diğer nöronlardan veya duyu organlarından gelen sinyalleri çekirdeğe iletmektir.

#### Çekirdek ve Akson

Çekirdek (Soma) dendrit tarafından gelen sinyalleri bir araya toplayarak aksone iletir. Toplanan bu sinyaller akson tarafından işlenerek nöronun diğer ucunda bulunan sinapsis denenen birimlere gönderilir.

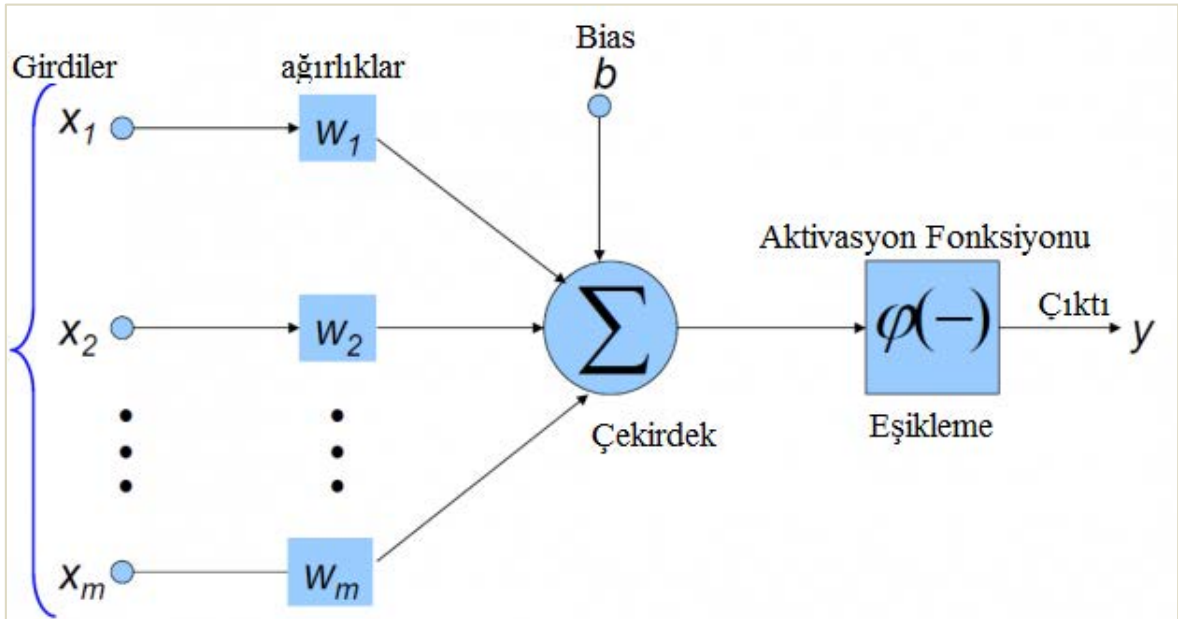
#### Sinapsis

Aksondan gelen toplam bilgiyi ön işlem den geçirdikten sonra diğer sinir hücrelerinin dendritlerine iletmekle görevlidir. Sinapsis tarafından gerçekleştirilen bu ön işlem oldukça önemlidir çünkü bu ön işlem gelen toplam sinyalin belirli bir eşik değerine göre düzenlenmesini içermektedir. Bu da her gelen toplam sinyal ile dendritlere iletilen sinyaller arasında bir korelasyon oluşturur. Bu korelasyon sonucunda yapay sinir ağları teorilerinde, *öğrenme*, sinapsisler ve dendritler arasında yer alan ağırlık katsayılarının güncellenmesi şeklinde algılanmıştır.



Resim 6.30. Biyolojik nöron sinir hücresi yapısı [51]

Biyolojik sinir hücresinden asıl konumuz olan yapay sinir hücresine geçmek istediğimizde, yapay sinir hücresi de biyolojik sinir hücresi üzerinden modellenmiş ve aynı çalışma prensiplerine dayandırılmaya çalışılmıştır [51].



Şekil 6.5. Yapay nöron sinir hücresi yapısı

Şekil 6.5.'de görünen yapay sinir hücresinin dendritleri  $x_m$  ve herbir dendritin ağırlık katsayısı (önemlilik derecesi)  $w_m$  ile gösterilmiştir. Böylece  $x_m$  girdi sinyallerini,  $w_m$  ise o sinyallerin ağırlık katsayılarının değerlerini içermektedir. Çekirdek ise tüm girdi sinyallerinin ağırlıklı toplamlarını elde etmekte ve tüm bu toplam sinyal sinapsise eşiklenme fonksiyonuna girdi olarak yönlendirilmiştir. Sinapsis üzerindeki eşikleme

fonksiyonundan çıkan sonuç sinyali  $y$  ile belirtilmiş ve diğer hücreye beslenmek üzere yönlendirilmiştir.

### **6.6.2. Yapay sinir ağları eğitimi ve öğrenmesi**

Yapay Sinir Ağları öğrenmesi aktivasyon olarak nitelendirilebilmektedir. Sinir hücresine giren sinyallerin toplamı, ilgili sinir hücresinin aktif olabileceği değerin altında ya da üzerinde olması durumuna göre öğrenme ya da sınıflandırma gerçekleştirilebilecektir. Örneğin eğer toplam değer, sinir hücresini aktifleştirebilecek değerden büyük ise hücre aktif yani sonuç 1 ( $y=1$ ), küçük ise hücre pasif sonuç 0 ( $y=0$ ) olarak kabul edilebilir. Bunun sonucunda basit anlamda belirli girdi örüntü gruplarına 1 ya da 0 şeklinde sınıflandırma yaparak sonuç üretebilen bir hücre, teorik anlamda da karar verebilen, bir hücredir. Karar verme ve sınıflandırma öğrenme sürecinin temel yapı taşlarını oluşturduğundan bu şekilde davranabilen bir yapay sinir hücresi öğrenebilir anlamına gelmektedir.

İnsan beyni tüm yaşamı boyunca dış uyaranlardan elde ettiği sonuçları yorumlamakta ve bu sonuçları başka davranışlarında da kullanmaktadır. Yapay sinir ağlarının da bu şekilde davranabilmesi için yukarıda bahsedilen şekliyle çıktıyı ürettikten sonra hata oranı tespit edilerek bu hata oranının minimize edilmeye çalışılması gerekmektedir. Bu hata oranı farklı algoritmalarla azaltılmaya çalışılarak gerçek çıkışa yaklaşılmaya çalışılır. Bu işlemler sırasında yenilenen yapay sinir ağlarının ağırlıklarıdır. Ağırlıklar her çevrimde yenilenerek istenilen amaca götüren, en az hata oranı elde edilmeye çalışılır. Eğer yapay sinir ağları istenilen sonuç değerlerine ulaşmışsa en son ulaşılan ağırlık değerleri saklanmaktadır. Ağırlıkların sürekli yenilenerek istenilen sonuca ulaşılan kadar geçen zamana öğrenme adı verilir. Yapay sinir ağı öğrenme işlemi gerçekleştikten sonra, daha önce eğitimde kullanılmayan girdi değerleri verilip, gerçek çıkış değeriyle karşılaştırılır. Gerçek çıkış değeriyle aynı değere ulaşılmışsa yapay sinir ağı öğrenmiş demektir.

### **6.6.3. Yapay sinir ağları yapısı**

Yapay Sinir Ağları, birden fazla yapay sinir hücresinin birbiriyle bağlanmasıyla oluşturulan yapılardır. Bu özelliğiyle yapay sinir ağları eş zamanlı birden fazla karmaşık

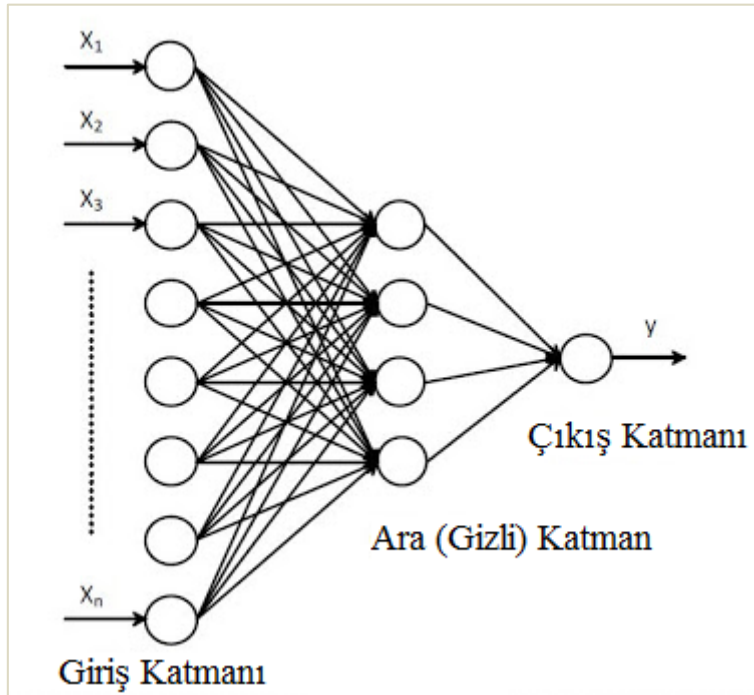


işlemi aynı anda gerçekleştirebilmektedir. Yapay sinir ağları üç ana kısımda incelenebilmektedir. Giriş Katmanı, Ara (Gizli) Katmanlar ve Çıkış Katmanı.

Giriş Katmanı: Yapay sinir ağının dış dünyadan beslendiği girdilerinin geldiği katmandır. Bu katmana gelen girdiler herhangi bir işleme uğramadan alt katmanlara iletilmektedir.

Ara (Gizli) Katman(lar): Giriş katmanından çıkan bilgiler bu katmana gelmekte ve ara katman sayısı ağdan ağa değişebilmektedir. Bazı yapay sinir ağlarında ara katman bulunmadığı gibi bazı yapay sinir ağlarında ise birden fazla ara katman bulunabilmektedir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde daha etkin kullanılabilmesini sağlamaktadır.

Çıkış Katmanı: Ara katmanlardan gelen bilgilerin işlenerek ağ çıktılarının üretildiği katmandır. Bu katmanda üretilen çıktılar dış dünyaya bu katmandan gönderilmektedir. Geri beslemeli ağlarda bu katmanda üretilen çıktı kullanılarak ağın yeni ağırlık değerleri hesaplanmaktadır.



Şekil 6.6. Yapay sinir ağları yapısı

#### 6.6.4. Yapay sinir ağıları MAA öneri sistemi

Sistem eğitime başlanmadan önce, öncelikle kullanıcı tercihlerinden yapay sinir ağlarının eğitimi için veri kümesinden Çizelge 6.2.' de gösterilen 11 adet öznitelik oluşturulmuştur. Bu özniteliklere göre sistem, kullanıcısı tarafından belirlenen yapay sinir ağıları parametrelerine göre eğitilmiş ve sonraki tercihlerinde Öznitelik 11 olan kullanıcının etkinliği tercih edip etmeyeceği tahminine göre tercih edeceği etkinlikler gösterilmektedir. Yapay Sinir Ağları öğrenmesinde geriye yönelik besleme kullanılmıştır.

Çizelge 6.2. ANN öğrenmede kullanılan öznitelikler

<b>Öznitelik 1</b>	Kullanıcı ID
<b>Öznitelik 2</b>	Seçilen Etkinliğin Nitelik Tipi
<b>Öznitelik 3</b>	Seçilen Etkinliğin Bulunduğu Enlem Bilgisi
<b>Öznitelik 4</b>	Seçilen Etkinliğin Bulunduğu Boylam Bilgisi
<b>Öznitelik 5</b>	Seçilen Etkinliğin Zamanı
<b>Öznitelik 6</b>	Seçilen Etkinliğin Süresi
<b>Öznitelik 7</b>	Kullanıcının Bulunduğu İl
<b>Öznitelik 8</b>	Kullanıcının Bulunduğu İlçe
<b>Öznitelik 9</b>	Kullanıcının Bulunduğu Enlem
<b>Öznitelik 10</b>	Kullanıcının Bulunduğu Boylam
<b>Öznitelik 11</b>	Kullanıcının Etkinliği Tercih Edip Etmediği

Belirlenen özniteliklere ait veriler öncelikle normalize edilmekte sonra öğrenmede kullanılmaktadır. Kullanıcı tarafından belirlenen parametrelerle MAA aracılığıyla sistem tekrar tekrar eğitilebilmekte ve güncel tutulabilmektedir. Öğrenme mekanizmasının hızlı çalışabilmesi için öğrenme işlemi kullanıcı sisteme girdiğinde yapılmakta ve tercih yaptıkça arka planda web sunucusu üzerinde bu öğrenme sistemde aktif olan kullanıcılar için belirli aralıklarla gerçekleştirilmektedir. MAA üzerinde, kullanıcının yapay sinir ağıları eğitiminde kullanılacak parametreleri değiştirebileceği ekran hazırlanmıştır. Bu ekran aracılığıyla kullanıcı isterse eğitim için gerekli öğrenme oranı, momentum, gizli katman sayısı, öğrenme fonksiyon tipi parametrelerini belirleyebilmekte, öğrenmenin devam edeceği düğüm sayısı ve hata oranı bilgilerini değiştirebilmektedir. Bu değişimlerde sistemin nasıl davrandığı deneysel sonuçlar bölümünde incelenmiştir.

## 7. DENEYSSEL SONUÇLAR

MAA mobil uygulamasından elde edilen deneysel sonuçlar, sistemin ne şekilde adaptif olabildiği ve bu adaptif olabilmenin ne şekilde etkin ve hızlı olabileceği yönüyle incelenmiştir. Ayrıca kullanıcıya ait sistem parametre sabitlerinin ve sistem işleyiş özelliklerinin değiştirilmesine bağlı olarak sistemin ne şekilde davrandığı ve adaptif özelliğinin ne şekilde değiştiği ele alınmıştır. Sistemin genel olarak adaptif olma özelliği, sistemin bulunduğu yere duyarlı olması ve kullanıcının önceki tercihlerinden öğrenmesi olarak iki başlıkta özetlenebilir.

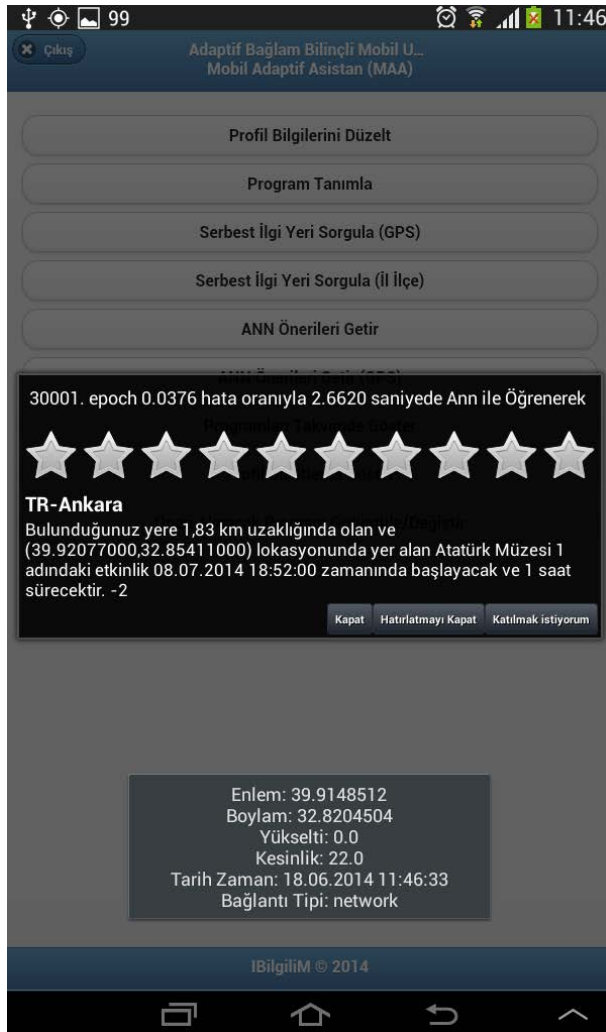
### 7.1. Sistemin Yer Bağlamını Kullanarak Adaptif Olması

MAA sisteminin bulunduğu yere duyarlı olması üç farklı metotla elde edilmiştir. Açık alanlarda GPS alıcı sistemiyle, kapalı alanlarda ve açık alanlarda mobil şebeke tarafından sağlanan yer bilgileriyle ve gidilecek program dahilindeki yerlerin veritabanında bulunan önceden girilmiş enlem boylam yer bilgileriyle. Bu üç metotla yer elde edilmesi örnekleri şu şekildedir. Örnek olarak Şekil 7.1.'de görüldüğü üzere, MAA kullanıcısı programını Ankara Altındağ ilçesi olarak seçmiş ve veritabanında yer bilgisi olarak Ankara Altındağ için (39,92077000;32,85411000) koordinat bilgileri yer almaktadır.



Resim 7.1. Yer tespiti işlemleri

MAA kullanıcısı bu program için öneri almak istediğinde kendisini karşılayan Öneri Yer Bilgisi Ekranından “Gideceğim yerden” seçeneğini seçtiğinde kendisine yapılacak olan öneriler, kullanıcı nerede olursa olsun programın bulunduğu koordinat bilgilerine göre olacaktır. MAA tarafından yapılacak öneri kullanıcının belirlediği parametrelere göre öğrenen bir sistem tarafından belirlenip yine kullanıcının belirlediği zaman aralıklarında önerilmektedir. “Bulduğum yerden” seçeneği tercih edildiğinde ise mobil cihazın bulunduğu yere göre GPS alıcısı ya da şebeke tarafından elde edilecek yer bilgilerine göre aşağıdaki gibi öneri alınacaktır. Bu öneri mobil cihaz kapalı alanda bulunduğu ve GPS alıcısından gerekli bilgi elde edilemediği için bağlantı tipi “network” yani mobil ağ sağlayıcı tarafından elde edilen (39,9148512;32,8204504) koordinat bilgileri kullanılarak yapılmıştır. MAA tarafından yapılacak öneri kullanıcının belirlediği parametrelere göre öğrenen bir sistem tarafından belirlenip yine kullanıcının belirlediği GPS koordinat bilgilerinin okunacağı zaman aralığı ve uzaklığa göre önerilmektedir.



Resim 7.2. GPS network yer tespiti ve öneri işlemi

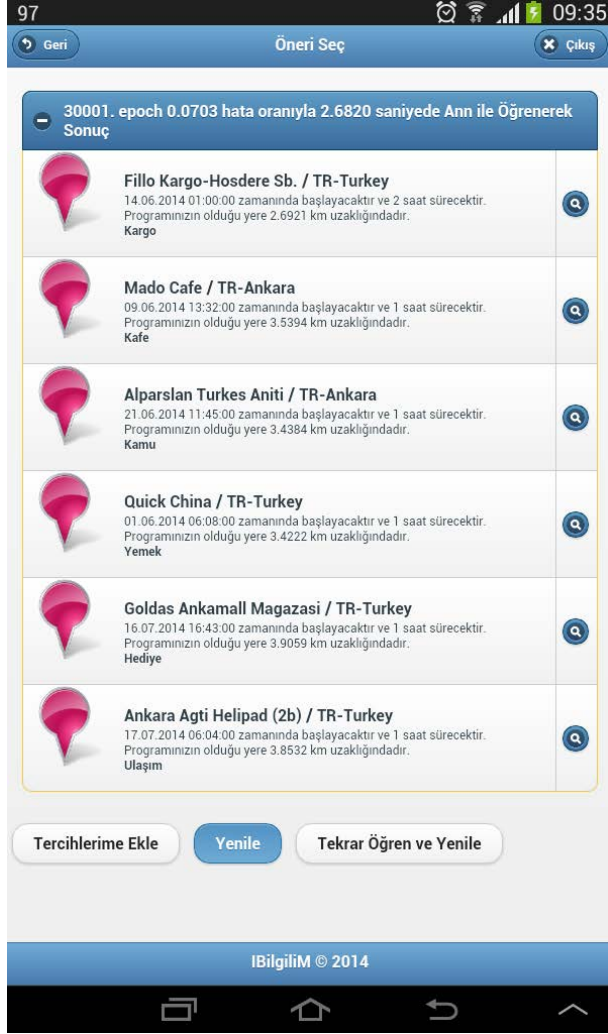
Yapılan önerinin yapay sinir ağı öğrenmesine göre yapıp yapılmayacağı yine kullanıcı sabitlerinde yer alan Adaptif Öneri seçeneğinin aktif edip edilmemesine göre değişmektedir.

Öneri hatırlatma sistemini kullanmadan kullanıcı eğer önerileri liste halinde almak isterse yine kendi belirlediği parametrelere göre öğrenmeli ya da öğrenmesiz şekilde sistemden öneri alabilmektedir. Öğrenme yapılmadan yapılan önerilerde yalnızca yer bilgisine sistem adapte olmaktadır.

## **7.2. Sistemin Yapay Sinir Ağları Kullanarak ve Öğrenerek Adaptif Olması**

MAA sistemi, kullanıcının belirlediği yapay sinir ağı parametrelerini ve kullanıcının önceki tercihlerini kullanarak öğrenmekte ve istenirse önerilerini bu şekilde kullanıcıya sunmaktadır.

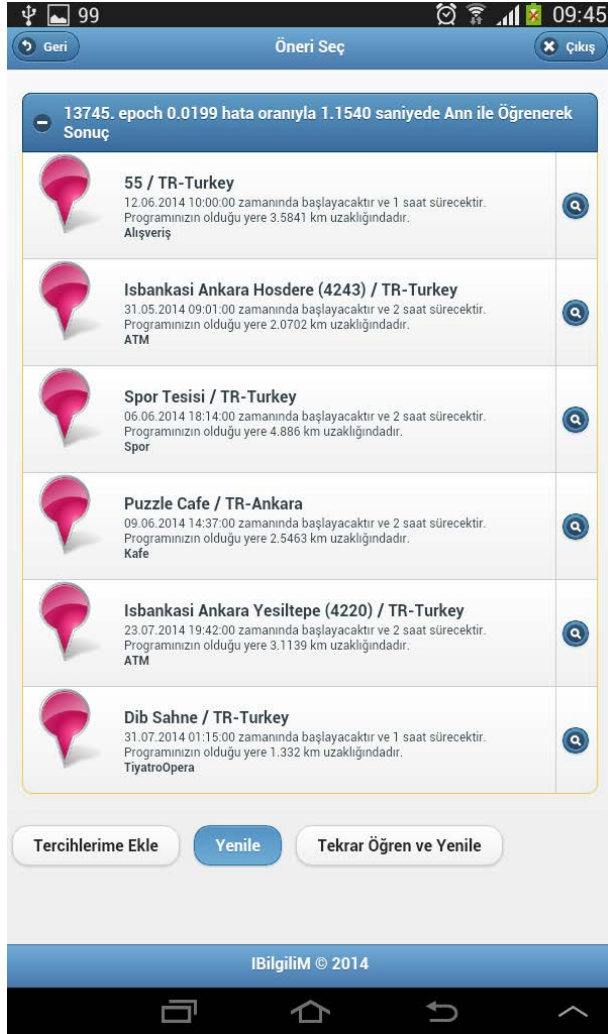
Örnek olarak 30000 düğüm ve 0,02 hata oranı parametreleriyle sistemin öğrenmesi istendiğinde 30000 düğüm tamamlandığında 0,07 hata oranıyla öğrenme gerçekleşmiş ve en fazla 10 tane öneri istediğimiz durumda sistem farklı kategorilerden Resim 7.3.'deki gibi etkinlik önermelerini yapmıştır.



Resim 7.3. 30000 düğüm ve 0,0703 hata oranıyla öğrenme sonucu öneriler

Resim 7.3.'te görüntülenen etkinlik önerileri incelendiğinde en yakını 2,6921 km en uzağı 3,9059 km olan farklı kategorilerde altı farklı etkinlik önerisi kullanıcıya sunulmuştur.

Sistem tekrar öğrenmeye zorlandığında ise 13745. düğümde 0,0199 hata oranını yakaladığı için öğrenmeyi durdurmuş ve farklı kategorilerden Resim 7.4.'deki gibi etkinlik önermelerini yapmıştır.



Resim 7.4. 13745 düğüm ve 0,0199 hata oranıyla öğrenme sonucu öneriler

Resim 7.4.'te görüntülenen etkinlik önerileri incelendiğinde en yakını 1,332 km en uzağı 4,886 km olan farklı kategorilerde yine altı farklı etkinlik önerisi kullanıcıya sunulmuştur. Bu öneriler içerisinde kullanıcının sadece kendi ilgilerine uygun öneriler değil olabilecekse başka ilgilerden tercih edebilir düşüncesiyle tercih edilebileceği düşünülen farklı ilgi alanlarındaki etkinlikler de sunulmuştur. Bu şekilde öğrenme yapılırken kullanıcının tercih örnek veri kümesini zenginleştirme amaçlanmıştır.

### Yapay Sinir Ağları parametre değişimine göre Öğrenmenin İncelenmesi

Yapay Sinir Ağları parametrelerinden öğrenme oranı, momentum düşürüldüğünde, gizli katman sayısı sabit bırakıldığında daha düşük bir hata oranıyla daha hızlı etkin bir öğrenme gerçekleştiği, ilgili parametreler artırıldığında ise hata oranının çok yükseldiği öğrenmenin çok etkin sağlanamadığı gözlemlenmiştir.

Çizelge 6.3. Öğrenme parametrelerinin değişimine göre öğrenme sonucu

<b>Öğrenme Parametreleri</b>		
<b>Öğrenme Oranı</b>	0,3	
<b>Momentum</b>	0,7	
<b>Gizli Katman Sayısı</b>	9	
<b>Düğüm Sayısı</b>	30000	
<b>Hata Oranı</b>	0,02	
<b>Fonksiyon Tipi</b>	Sigmoid	
<b>Öğrenme Sonucu</b>		
<b>Kaçıncı Düğüm</b>	<b>Hata Oranı</b>	<b>Öğrenme Zamanı (saniye)</b>
30001	0,05	3,25
30001	0,0407	2,85
24362	0,02	2,14
10758	0,0196	0,995
30001	0,1363	36,140

Çizelge 6.4. Öğrenme parametrelerinin azaltılmasıyla öğrenme sonucu

<b>Öğrenme Parametreleri</b>	
<b>Öğrenme Oranı</b>	0,1
<b>Momentum</b>	0,5
<b>Gizli Katman Sayısı</b>	9
<b>Düğüm Sayısı</b>	30000
<b>Hata Oranı</b>	0,02
<b>Fonksiyon Tipi</b>	Sigmoid



Çizelge 6.4. (devam) Öğrenme parametrelerinin azaltılmasıyla öğrenme sonucu

<b>Öğrenme Sonucu</b>		
<b>Kaçıncı Düğüm</b>	<b>Hata Oranı</b>	<b>Öğrenme Zamanı (saniye)</b>
21006	0,02	1,78
12986	0,0199	10,990
11810	0,02	1,18
16249	0,02	1,37
13419	0,02	11,290

Öğrenmede fazla gizli katman sayısı kullanıldığında öğrenmenin doğruluk oranı ve çalışma süresi açık bir şekilde artmaktadır. Katman sayısı artırıldığında, daha az bir hata oranıyla öğrenmenin gerçekleştirilebilmesi için öğrenme düğüm sayısı artırılması gerekir. Ancak bu da fazladan işlem maliyeti gerektirir.

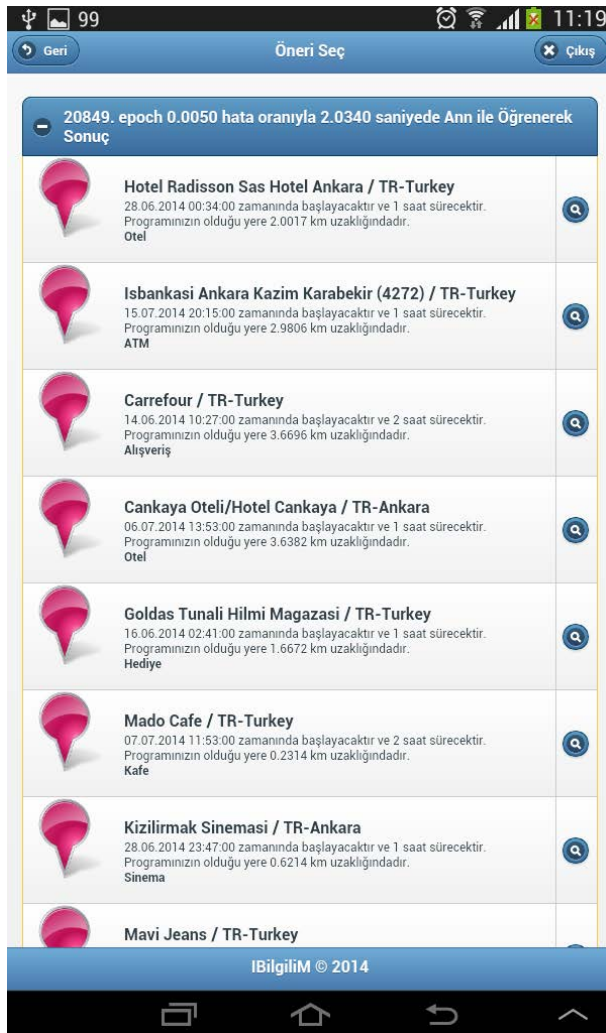
Çizelge 6.5. Öğrenme parametrelerinin artırılmasıyla öğrenme sonucu

<b>Öğrenme Parametreleri</b>		
<b>Öğrenme Oranı</b>	1	
<b>Momentum</b>	0,9	
<b>Gizli Katman Sayısı</b>	12	
<b>Düğüm Sayısı</b>	30000	
<b>Hata Oranı</b>	0,02	
<b>Fonksiyon Tipi</b>	Sigmoid	
<b>Öğrenme Sonucu</b>		
<b>Kaçıncı Düğüm</b>	<b>Hata Oranı</b>	<b>Öğrenme Zamanı (saniye)</b>
30001	0,2039	65,880
30001	0,2625	77,880
30001	0,6369	70,110

Çizelge 6.5.(devam) Öğrenme parametrelerinin artırılmasıyla öğrenme sonucu

30001	0,2623	7,28
60001	0,1921	266,450

Yukarıdaki tüm çıkarımlardan elde edilen en etkin oranlarla hata oranını % 0,05 tutarak öğrenmeyi gerçekleştirdiğimizde 20849. Düğümde % 0,05 hata oranıyla öğrenme gerçekleşmiş ve aşağıdaki gibi öneriler alınmıştır.



Resim 7.5. 20849 düğüm ve 0,005 hata oranıyla öğrenme sonucu öneriler

Resim 7.5.'te görüntülenen etkinlik önerileri incelendiğinde en yakını 0,2314 km en uzağı 3,9497 km olan farklı kategorilerde dokuz farklı etkinlik önerisi kullanıcıya sunulmuştur.

## 8. SONUÇ VE ÖNERİLER

Günümüzde mobil cihazlar çok yaygınlaştığı ve her yerde kullanıcının yanında olduğu için mobil cihazlar için uygulama geliştirme oldukça önem kazanmıştır. Mobil cihazların daha etkin bir şekilde kullanılabilmesi için bağlam bilinçli olmaları gerekmektedir. Geliştirilen uygulamanın da daha etkin bir şekilde kullanıcısının işlerini kolaylaştırabilmesi ve kişiselleştirilebilmesi için bu bağlam bilinçli servisleri en etkin bir şekilde kullanabilmesi ve ilgili bağlam ile ilgili sonrasında kullanmak üzere veriler saklayabilmesi gerekmektedir.

Bağlam bilinçli bilişim kavramı ilk olarak sadece 20 yıl önce gündeme gelmesine rağmen sırf bu ihtiyaçlardan ve gelişmelerden dolayı oldukça popüler hale gelmiştir. Gelecekte de bu alanla ilgili özellikle sağlık, hukuk, asistan sistemler alanlarında yapılacak uygulamaların artacağı düşünülmektedir. Bu nedenle bu tez çalışmasında adaptif bağlam bilinçli mobil uygulama geliştirme üzerinde durulmuş ve tez kapsamında hazırlanan uygulamada kullanıcının yer zaman bağlamlarına duyarlı olabilen ve kullanıcının planladığı seyahatlerinde kullanabileceği bir asistan öneri sistemi geliştirilmiştir. Bu öneri sisteminin, kullanıcı tercihlerine göre öğrenmesi sağlanmış ve kendini kullanıcının önceki isteklerine, yer zaman ve program bilgilerine göre adapte edebilmesi sağlamıştır. Kullanıcı için kullanıcı dostu kolay kullanılabilir ekranlar hazırlanmış, öneri hatırlatma sistemiyle de kullanıcının, uygulama ekranını kullanmadan, her an belirlediği aralıklarla öneriler alması sağlanarak, esnek bir kullanım olanağı sağlanmıştır. Tüm bu işlemler sağlanırken uygulama ve veri güvenliği en üst seviyede tutulmuş ve günümüz mobil ve web uygulamalarındaki en son popüler teknolojiler ve web standartları kullanılmaya çalışılmıştır. Ayrıca uygulamanın öğrenme, hatırlatma ve öneri sunma işlemlerindeki kullanılan değişkenler, uygulama değişimi gerektirmeksizin parametrik olarak kullanıcı tarafından belirlenebilmektedir.

Bu kapsamda yapılacak ileriki çalışmalarda, kullanıcının tercihlerine bağlı yapay sinir ağları öğrenme öznelikleri, sınıflandırma için daha belirleyici şekilde ayarlanırsa ve sistem, konum bilgilerine göre gerçek etkinliklerle anlık olarak beslenebilirse daha iyi ve başarılı sonuçlar elde edilebileceği düşünülmektedir. Bu öğrenme yapısının ayrıca kullanılan veri kümesi ve bağlamlar değiştirilerek ya da zenginleştirilerek daha etkin bir şekilde kullanılabilmesi öngörülmektedir.



## KAYNAKLAR

1. Gupta, U.G. (1996). *Management information systems: A managerial perspective*, United States, West Pub. Co. , 3-69.
2. Weiser, M. (1999, July). Some Computer Science Issues in Ubiquitous Computing, *Communications of the ACM*, 3(3): 12-22.
3. Diot, C., Huitema, C. and Turletti, T. (1995). *Multimedia Applications should be Adaptive*. In Proceedings of the 3rd IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS'9), 23–25. Mystic, Connecticut.
4. Campbell, A. and Coulson, G. (1994). A Quality of Service Architecture. *ACM Computer Communications Review*, 24(2): 6–27.
5. Weiser, M. (1993). Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 75–84.
6. Schilit, B., Adams, N. and Want, R. (1994). Context-Aware Computing Applications. In Proceedings of the Workshop on Mobile Computing Systems and Applications, 85–90. *IEEE Computer Society*, Santa Cruz, CA.
7. Ryan, N., Pascoe, J. and Morse, D. (1997). *Enhanced reality fieldwork: the context-aware archaeological assistant*, Proceeding of the 25th Anniversary Computer Applications in Archaeology, <http://www.caaconference.org/>.
8. Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1): 4–7.
9. Guanling, C. and Kotz, D. (2000, November). *A Survey of Context-Aware Mobile Computing Research.*, Dartmouth Computer Science Technical Report TR2000-381.
10. Internet : *Gartner1 . Gartner1*. (2014, Şubat).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.gartner.com%2Fnewsroom%2Fid%2F2665715&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
11. Internet : *Gartner2 . Gartner2*. (2014, Mart).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.gartner.com%2Fnewsroom%2Fid%2F2692318&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
12. Internet : *IDC . IDC*. (2013, Aralık).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.businessinsider.com%2Fsmartphone-and-tablet-penetration-2013-10%23%21J7D2z&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014

13. Nakagawa, H., Ohsuga, A., Honiden, S., (2008). Constructing Self-Adaptive Systems Using a KAOS Model. *In Proceedings of the 2008 second IEEE international conference on self-adaptive and self-organizing systems workshops (132–137): IEEE Computer Society.*
14. Weyns, D. , Iftikhar, M.U. , Malek, S., Andersson, J., Naqvi, M. et al., (2012). Claims and supporting evidence for self-adaptive systems – A literature review. *In: Proceedings of the 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2012. ACM, 47, New York*
15. Schilit, B. and Theimer, M. (1994). Disseminating Active Map Information to Mobile Hosts. *IEEE Network, 22–32.*
16. Brown, P.J. (1996). The stick-e document: a framework for creating context-aware applications, *Proceedings of the Electronic Publishing, 259–272, Palo Alto.*
17. Dey, A.K. (1998). Context-aware computing: the CyberDesk project, *Proceedings of the AAAI, Spring Symposium on Intelligent Environments, 51–54, Menlo Park, CA.*
18. Schmidt, A. and van Laerhoven, K. (2001). How to build smart appliances?, *IEEE Personal Communications, 8(4): 66–71.*
19. Internet : *WikiPedia. GPS.*  
<http://www.webcitation.org/query?url=http%3A%2F%2Ftr.wikipedia.org%2Fwiki%2FGPS&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
20. Want R., Hopper A., Falcão V., and Gibbons J. (1992, January). The Active Badge location system. *ACM Transactions on Information Systems, 91-102.*
21. Want, R., Schilit, B. N., Norman, I. A., Rich G., Petersen, K., Goldberg, D., Ellis, J.R., and Weiser M. (1995, December). An overview of the PARCTAB ubiquitous computing experiment. *IEEE Personal Communications, 28-43.*
22. Abowd, Gregory D., Atkeson, Christopher G., Hong J., Long S., Kooper R., and Pinkerton M. (1997, October). Cyberguide: A mobile context-aware tour guide. *Wireless Networks, 421-433.*
23. Abhaya A., Cravatts M., and Krzyzanowski P. (1994, December). An indoor wireless system for personalized shopping assistance. *In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, 69-74, Santa Cruz, California.*
24. Ward A., Alan J., and Hopper A. (1997, October). A new location technique for the active office. *IEEE Personal Communications, 4(5): 42-47.*

25. Priyantha Nissanka B., Chakraborty A., and Balakrishnan H. (2000, August). *The Cricket location-support system*. In Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking, Boston, MA.
26. Paramvir B. and Venkata N.P.R. (2000, March). *An in-building RF-based user location and tracking system*. In Proceedings of IEEE INFO COM 2000, Tel-Aviv, Israel.
27. Castro P. and Muntz R. (2000, October). Managing context data for smart spaces. *IEEE Personal Communications*, 44-46.
28. Addlesee Michael D., Jones A., Livesey F., and Samaria F. (1997, October). The ORL Active Floor. *IEEE Personal Communications*, 35-41.
29. Stillman S., Tanawongsuwan R., and Essa I. (1999, April). *A system for tracking and recognizing multiple people with multiple cameras*. In Proceedings of Second International Conference on Audio-Visionbased Person Authentication, Washington, DC.
30. Brumitt B., Meyers B., Krumm J., Kern A., and Shafer S. (2000, September). *EasyLiving: Technologies for intelligent environments*. In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, 12-29, Bristol, UK.
31. Davies N., Cheverst K., Mitchell K., and Friday A. (1999, February). *Caches in the air: Disseminating tourist information in the GUIDE system*. In Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, Louisiana.
32. Dongpyo H., Hedda R. S., Woontack W. (2007). *Linking Context Modelling and Contextual Reasoning*, GIST, U-VR Lab, South Korea.
33. Sorensen, C.F., Wu, M., Sivaharan, T., Blair, G. S., Okanda, P., Friday, A., and Duran-Limon, H. (2004). *A context-aware middleware for applications in mobile ad hoc environments*. In MPAC '04: Proc. of the 2nd workshop on Middleware for pervasive and ad-hoc computing, 107–110, New York, NY, USA.
34. Capra, L., Emmerich, W., and Mascolo, C. (2003, October). Carisma: context-aware reflective middleware system for mobile applications. *IEEE Transactions on Software Engineering*, 29(10):929 – 45.
35. Barton, J. and Kindberg, T. (2001, February). *The Cooltown user experience.*, Hewlett Packard.
36. Henriksen, K., AND Indulska, J. (2004). *Modelling and Using Imperfect Context Information*. In Workshop Proceedings of the 2nd IEEE Conference on Pervasive Computing and Communications, Washington, DC, USA.

37. McCarthy, J. (1993). *Notes on formalizing contexts*. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, R. Bajcsy, Ed., Morgan Kaufmann (Ed.), 555–560, San Mateo, California.
38. Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., and K., Nahrstedt. (2002, October). A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing*, 74–83.
39. A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. H. Campbell, and M. D. Mickunas. (2004). Middlewhere: A middleware for location awareness in ubiquitous computing applications. In H.-A. Jacobsen, (Ed.), *Lecture Notes in Computer Science*, 3231: 397–416. Springer.
40. Gruber, T. G. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2): 199–220.
41. Öztürk, P., and Aamodt, A. (1997, February). *Towards a model of context for case-based diagnostic problem solving*. In Context-97; Proceedings of the interdisciplinary conference on modeling and using context, 198–208, Rio de Janeiro.
42. Chen, H., Finin, T., and Joshi, A. (2003). *Using OWL in a Pervasive Computing Broker*. In Proceedings of Workshop on Ontologies in Open Agent Systems.
43. Internet : *JSON . JSON*. (2011, June).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.ecma-international.org%2Fpublications%2Ffiles%2Fecma-st%2FECMA-262.pdf&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
44. Internet : *Roy Thomas Fielding*. University of California, Irvine. (2000).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.ics.uci.edu%2F~fielding%2Fpubs%2Fdissertation%2Ftop.htm&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
45. Internet : *JDK, JDK*. (2000).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.oracle.com%2Ftechnetwork%2Fjava%2Fjavase%2Fdownloads%2Findex.html&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
46. Internet : *Eclipse. Eclipse*. (2014, June).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fwww.eclipse.org%2Fdownloads%2F&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
47. Internet : *AndroidSDK. AndroidSDK*. (2014, June).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fdeveloper.android.com%2Fsdk%2Findex.html&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014



48. Internet : *AndroidPlugin. AndroidPlugin.* (2014, June).  
<http://www.webcitation.org/query?url=https%3A%2F%2Fdl-ssl.google.com%2Fandroid%2F eclipse%2F&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
49. Internet : *jquerymobile. jquerymobile.* (2014, June).  
<http://www.webcitation.org/query?url=http%3A%2F%2Fjquerymobile.com%2F&date=2014-06-23>, Son Erişim Tarihi: 23 Haziran 2014
50. Internet : *PhoneGap. PhoneGap.* (2014, June).  
[http://www.webcitation.org/query?url=http%3A%2F%2Fdocs.phonegap.com%2Fen%2Fedge%2Fguide\\_platforms\\_index.md.html&date=2014-06-23](http://www.webcitation.org/query?url=http%3A%2F%2Fdocs.phonegap.com%2Fen%2Fedge%2Fguide_platforms_index.md.html&date=2014-06-23), Son Erişim Tarihi: 23 Haziran 2014
51. Fausset L. (1994) , *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*, Prentice-Hall, Inc. Upper Saddle River, NJ, USA



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : BİLGİLİ, Muhammed İkbal  
 Uyuğu : T.C.  
 Doğum tarihi ve yeri : 03.04.1982, Ulubey  
 Medeni hali : Evli  
 Telefon : 0 (555) 309 44 16  
 e-mail : ibilgilim@gmail.com

### Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi / Bilgisayar Bilimleri	Devam Ediyor
Lisans	Trakya Üniversitesi / Bilgisayar Mühendisliği	2004
Lise	Çorum Anadolu Öğretmen Lisesi	2000

### İş Deneyimi

Yıl	Yer	Görev
2005-2006	Sosyal Sigortalar Kurumu	Bilgisayar Mühendisi
2006-2010	Sosyal Güvenlik Kurumu	Bilgisayar Mühendisi
2010-2014	Çalışma ve Sosyal Güvenlik Bakanlığı	Bilgisayar Mühendisi
2014-Halen	Sosyal Güvenlik Kurumu	Bilgisayar Mühendisi

### Yabancı Dil

İngilizce - Almanca





*GAZİ GELECEKTİR..*