

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

**HARİTALAR ÜZERİNDE
COĞRAFİ BİLGİ SİSTEMİ
PORTALI**

YÜKSEK LİSANS TEZİ

Can Alper ÖZTÜRK

**Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM
MÜHENDİSLİĞİ**

Tez Danışmanı : Prof. Dr. Cemil ÖZ

Mayıs 2019

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HARİTALAR ÜZERİNDE
COĞRAFİ BİLGİ SİSTEMİ
PORTALI

YÜKSEK LİSANS TEZİ

Can Alper ÖZTÜRK

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM
MÜHENDİSLİĞİ

Bu tez 27.05.2019 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Prof. Dr.
Cemil ÖZ
Jüri Başkanı



Doç. Dr.
Sezgin KAÇAR
Üye



Dr. Öğr. Üyesi
Ali GÜLBAĞ
Üye



BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Can Alper ÖZTÜRK

30.05.2019

TEŐEKKÜR

Yüksek lisans eğitiminin boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Prof. Dr. Cemil ÖZ'e teşekkürlerimi sunarım.

Bilgi ve deneyimlerinden faydalandığım Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü öğretim elemanlarından Arş. Gör. Musa BALTA'ya teşekkür ederim.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	v
TABLolar LİSTESİ	viii
ÖZET	ix
SUMMARY	x
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
COĞRAFİ BİLGİ SİSTEMLERİ	2
2.1. Giriş	2
2.1.1. Coğrafi bilgi sistemlerinin Türkiye'deki örnekleri	4
2.1.2. Coğrafi bilgi sistemlerinin alt bileşenleri	8
2.1.2.1. Coğrafi veri tabanları	8
2.1.2.2. Harita web servisleri	10
2.1.2.3. Masaüstü uygulamalar	11
2.1.2.4. Web/Mobil uygulamalar	13
2.2. Yazılım Mimarisi	15
2.2.1. Katmanlı mimari	16
2.2.2. Tasarım desenleri	17
2.2.3. Yazılım güvenliği.....	18
2.3. Yazılım Çalışma Ortamı ve Konteyner Yapısı	19

BÖLÜM 3.

CBS YAZILIM MİMARİ GELİŞİRMESİ	22
3.1. Problemin Tanımı ve Önerilen Model.....	22
3.2. Kullanılan Materyaller.....	24
3.2.1. Ön tanımlı veri setleri.....	24
3.2.2. Çalışma içinde oluşturulan veri setleri.....	25
3.3. Yöntem	28
3.3.1. Veri tabanı katmanı.....	30
3.3.2. Harita katmanı.....	34
3.3.3. İş katmanı	35
3.3.4. Uygulama katmanı.....	36
3.3.5. Uygulama çalışma ortamı.....	61

BÖLÜM 4.

TARTIŞMA VE SONUÇ	62
KAYNAKLAR	64
ÖZGEÇMİŞ	67

SİMGELER VE KISALTMALAR LİSTESİ

AOP	: Aspect Oriented Programming
CBS	: Coğrafi Bilgi Sistemleri
GIS	: Geographical Information Systems
HTTP	: Hyper Text Transfer Protocol
MAKS	: Mekansal Adres Kayıt Sistemi
MVC	: Model View Controller
OGC	: Open Geospatial Consortium
ORM	: Object Relational Mapping
OWASP	: Open Web Application Security Project
QGIS	: Quantum gis
SAKUS	: Sakarya akıllı ulaşım sistemleri
SLD	: Styled Layer Descriptor
SQL	: Structured Query Language
SRS	: Spatial Reference System
SRID	: A spatial reference identifier
TKGM	: Tapu Kadastro Genel Müdürlüğü
TUCBS	: Türkiye Ulusal Coğrafi Bilgi Sistemi
UML	: Unified Modelling Language
WCS	: Web Coverage Service
WFS	: Web Feature Service
WKT	: Well Known Text
WMS	: Web Map Service

ŞEKİLLER LİSTESİ

Şekil 2.1. Dünyanın enlem ve boylamlara bölünmüş şekli	2
Şekil 2.2. Dünyanın iki boyutlu düzleme dönüştürülmesi	3
Şekil 2.3. Veriyi aynı kaynaktan alan farklı projeksiyon dönüşümlü haritalar	3
Şekil 2.4. Mahalli bileşenler uygulama şeması	4
Şekil 2.5. TUCBS coğrafi veri temaları	5
Şekil 2.6. TKGM uygulama ekranı	6
Şekil 2.7. Sakus uygulaması	6
Şekil 2.8. Proje Haritası	7
Şekil 2.9. İstanbul Büyükşehir Belediyesi trafik yoğunluk haritası	7
Şekil 2.10. Coğrafi bilgi sistemleri bileşenleri	8
Şekil 2.11. Coğrafi veri tipleri	9
Şekil 2.12. Geoserver web uygulama arayüzü	11
Şekil 2.13. Quamtum Gis CBS uygulaması	12
Şekil 2.14. ArcGis masaüstü harita uygulaması	12
Şekil 2.15. Örnek Leaflet ekranı	13
Şekil 2.16. Openlayers kütüphanesi UML diagramı	14
Şekil 2.17. Openlayers harita ekranı	14
Şekil 2.18. Asp.net Core platformu	15
Şekil 2.19. Çevik Yazılım Yaşam Döngüsü	16
Şekil 2.20. MVC tasarım deseni	18
Şekil 2.21. İlk 10 web uygulama açıkları	19
Şekil 2.22. Docker Mimarisi	21
Şekil 3.1. İlçe/mahalle varlık ilişki diyagramı	25
Şekil 3.2. İlçe/mahalle örnek haritasal veriler	25
Şekil 3.3. Çalışma içinde oluşturulan örnek veri setlerinin veri modeli	26
Şekil 3.4. Kurutma kanalı, sulama göleti, taşkın seddi örnek verisi	27

Şekil 3.5. Otobüs durağı, bisiklet yolu, bisiklet kiralama noktası örnek verisi	28
Şekil 3.6. CBS için Açık kaynak ve katmanlı mimari yapısı	30
Şekil 3.7. Kullanıcı, kişi, yedek, çalışma, rol tabloları varlık ilişki diyagramı	31
Şekil 3.8. Katman kaynak, katman, rol katman ve parametre tabloları varlık ilişki diyagramı	32
Şekil 3.9. Faaliyet tabloları varlık ilişki diyagramı	33
Şekil 3.10. Proje tabloları varlık ilişki diyagramı	33
Şekil 3.11. Geoserver kendi katmanlarımızın önizlemesi	34
Şekil 3.12. QGis programı ile kendi katman servislerimizin görüntülenmesi	35
Şekil 3.13. Katmanlı yazılım mimari önerisinin genel uygulama ekranı	37
Şekil 3.14. Mobil uygulama ekranı	38
Şekil 3.15. Veri tabanı katmanları yeni kayıt sekmesi	40
Şekil 3.16. Veri tabanı katmanları ekranı liste sekmesi	41
Şekil 3.17. Katman kaynak tanımlama sekmesi	42
Şekil 3.18. Katman tanımlama sekmesi	42
Şekil 3.19. Role katman atama sekmesi	43
Şekil 3.20. Katman stil yönetimi	44
Şekil 3.21. Uygulama ana ekran genel özellikleri	45
Şekil 3.22. Katman veri ekleme, silme ve güncelleme sekmesi	46
Şekil 3.23. Altlık harita seçenekleri	46
Şekil 3.24. Katman liste verileri ve filtreleme sekmesi	47
Şekil 3.25. Uygulama ana ekranı filtreleme aracı	48
Şekil 3.26. Proje yönetim ekranı	49
Şekil 3.27. Ana sayfa projeler ve örnek proje kartı	49
Şekil 3.28. Faaliyet tanımlama ekranı	50
Şekil 3.29. Faaliyet haritası ve kartı	51
Şekil 3.30. Gösterge paneli ekranı	51
Şekil 3.31. Tematik harita tanımlama ekranı	52
Şekil 3.32. Örnek tematik harita	53
Şekil 3.33. Baz istasyon katmanını tematik haritası	53
Şekil 3.34. Kullanıcı/kişi tanımlama ekranı	54
Şekil 3.35. Kullanıcı adı ve şifre tanımlama ekranı	55

Şekil 3.36. Parametre tanımlama ekranı	55
Şekil 3.37. Organizasyon şeması ekranı	56
Şekil 3.38. Rol tanımlama ekranı	57
Şekil 3.39. Rol yetki tanımlama ekranı	57
Şekil 3.40. Veri paylaşım görüntü oluşturma ekranı	58
Şekil 3.41. Servis kullanıcı oluşturma sekmesi	59
Şekil 3.42. Log ekranı	60
Şekil 3.43. Birim faaliyet tanımlama ekranı.....	60

TABLolar LİSTESİ

Tablo 3.1. Lisanslı uygulamalar ve önerilen model kıyaslaması	24
Tablo 3.2. Lisanslı uygulamalar ve önerilen model kıyaslaması	39

ÖZET

Anahtar kelimeler: CBS, OGC, yerel yönetimler, yazılım mimarisi, Inspire, TUCBS

Coğrafi Bilgi Sistemleri günümüzde her alanda karşımıza çıkmaktadır. Kullanıcıya problemlerini veya verilerini görselleştirme ve farklı açılardan bakma imkanı tanıdığı için de önemli bir araçtır. Fakat coğrafi bilgi sistemleri içeren projeleri oluşturabilmek için birçok açık/ticari yazılım, meslek dalı, insan kaynağı ve de fiziksel sunuculara ihtiyaç duyulabilmektedir. Günümüzde kullanılan yöntemler bu projelerin uzun sürede tamamlanmasına ve maliyetli olmasına sebebiyet verebilmektedir. Farklı dallarda uzman personellerin bulunamaması da coğrafi bilgi sistemleri için yine problem teşkil edebilmektedir. Tüm bu problem ve riskleri azaltabilmek için yerel yönetimlerde coğrafi veri tabanı, yazılım, harita sunucusu, veri toplama ve analizini içeren açık kaynak ve katmanlı bir yazılım mimari önerisi sunulmuştur.

Yerel yönetimlerin bilmek, uygulamak ve kullanmak zorunda olduğu OGC, Inspire, TUCBS gibi standartlar bu yazılım mimari önerisi ile daha anlaşılır ve uygulanabilir hale getirilmeye çalışılmıştır. Yerel yönetimlerin bu standartlardan mümkün olduğunca soyutlanarak CBS veri modellerini oluşturmaya odaklanabilmeleri hedeflenmiştir.

Günümüz yazılım sistemlerinin kalitesini belirleyen birçok faktör bulunmaktadır. Bunlardan güvenlik ve temiz kod yazma kavramı çok önemli görülen faktörlerden ikisidir. Çalışmamızda, bu iki faktör için dünyada en çok kabul gören ve evrensel yazılım mimari, standart ve yöntemlerden faydalanılmaya çalışılmıştır.

GEOGRAPHIC INFORMATION SYSTEM PORTAL ON MAPS

SUMMARY

Keywords: CBS, OGC, local governments, software architecture, Inspire, TUCBS

Geographical Information Systems nowadays appear in all areas. It is also an important tool because it allows the user to visualize Gis problems or data from different angles. However, many open/commercial software, occupational branches, human resources and physical servers may be needed to create projects involving geographic information systems. The methods used today can cause these projects to be completed in a long time and costly. The lack of expert personnel in different branches may be a problem for geographical information systems. The methods used today can cause these projects to be completed in a long time and costly. The lack of expert personnel in different branches may be a problem for geographical information systems. In order to reduce all these problems and risks, an open-source and layered software proposal including geographic data base, software, map server, data collection and analysis was presented in local governments.

Standards such as OGC, Inspire, TUCBS which local governments have to know, apply and use have been tried to be made more understandable and applicable with this software architectural proposal. It is aimed that local governments can focus on creating GIS data models from these standards as much as possible.

There are many factors that determine the quality of today's software systems. The concept of security and clean code writing are two of the most important factors. In our study, the most accepted and universal software architecture, standards and methods are tried to be used for these two factors.

BÖLÜM 1. GİRİŞ

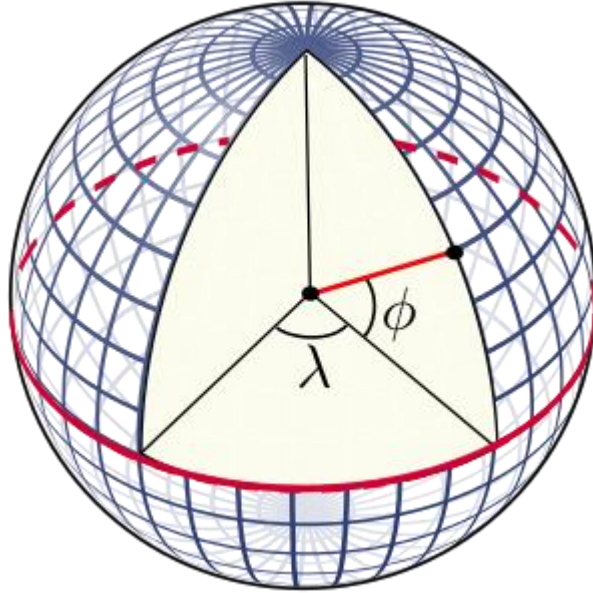
Günümüzde şehirlerdeki nüfus hızla büyümektedir. Yerel yönetimler çöp, çevre temizliği, şehir içi ulaşım, sosyal etkinlikler, hobi ve meslek edindirme kursları, yol bakım ve asfalt işleri, itfaiye hizmetleri, defin işleri ve numarataj işleri gibi hizmet vermekle zorunlu olduğu konularda daha büyük iş yükü ile karşılaşmaktadırlar. Tüm bu hizmetlerin yönetiminin kolaylaşması, daha hızlı ve efektif çalışabilmesi için bilgi sistemlerinden faydalanma zorunluluk haline gelmektedir. Hepsi kendi konularında uzmanlaşmış yerel yönetim birimlerinin verdiği hizmetlerin hedef kitlesi büyüdükçe yerel yönetim bilgi sistemlerinde oluşan verilerin boyutunun büyüdüğü görülmektedir. Tüm bu veriler yerel yönetimlerin sınırları kapsamında yapılan çalışmalarla üretildiği için bir mekan/adres tanımı da içerebilmektedir. Bu konumsal verilerin depolanması, işlenmesi ve anlaşılması Coğrafi Bilgi Sistemlerini kullanmayı zorunlu hale getirmektedir [1].

CBS sistemlerini kendi yetki alanlarında kurma zorunluluğu olan yerel yönetimlerin, CBS sistemlerini oluştururken birçok teknolojik ve uzmanlık gerektiren konuda sıkıntı yaşaması muhtemel bir durumdur. Yerel yönetimler CBS çalışmaları kapsamında evrensel ve ülkesel birçok CBS ve yazılım standartlarını uygulamak durumundadır. Birlikte çalışabilirlik ve güvenlik gibi konularda birçok kıstas bulunmaktadır. Yerel yönetimlerin tüm bu kritik ve sorunlu konular ile daha kolay mücadele edebilmesi ve kendi CBS modelini oluşturabilmesi için katmanlı bir CBS yazılım mimari önerisi sunulacaktır. Sunulan öneri katmanlı bir yazılım mimarisinden uygulama ile de gösterilecektir.

BÖLÜM 2. COĞRAFİ BİLGİ SİSTEMLERİ

2.1. Giriş

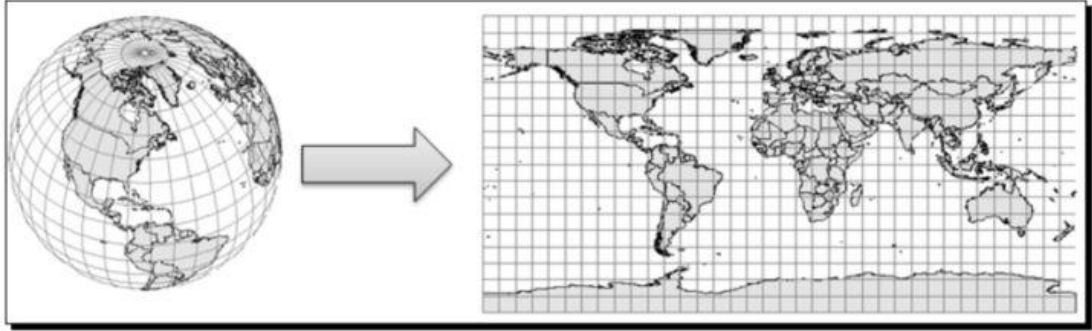
Yeryüzünde belli bir konumu olan tüm nesnelere coğrafik varlıklardır. Tüm bu varlıkların toplanması, işlenmesi, standardize edilmesi, analiz edilmesi coğrafi bilgi sistemlerini oluşturmaktadır. İlk CBS çalışmaları Kanada'da 1960'lı yıllarda başlamıştır. Coğrafi veri, dünyanın üzerinde ki herhangi bir sayısal konumdur. Bu sayısal konumu bulmak için küre şeklinde olan dünyanın merkezinden açılardan olarak enlem ve boylamlara ayrılmıştır. Bu şekilde herhangi bir coğrafi konumu " Lat. 30° 25' 18" N, Long. 30° 5' 20" W" şeklinde derece, dakika, saniye cinsinden ifade edebilmektedir [1,2].



Şekil 2.1. Dünyanın enlem ve boylamlara bölünmüş şekli [3].

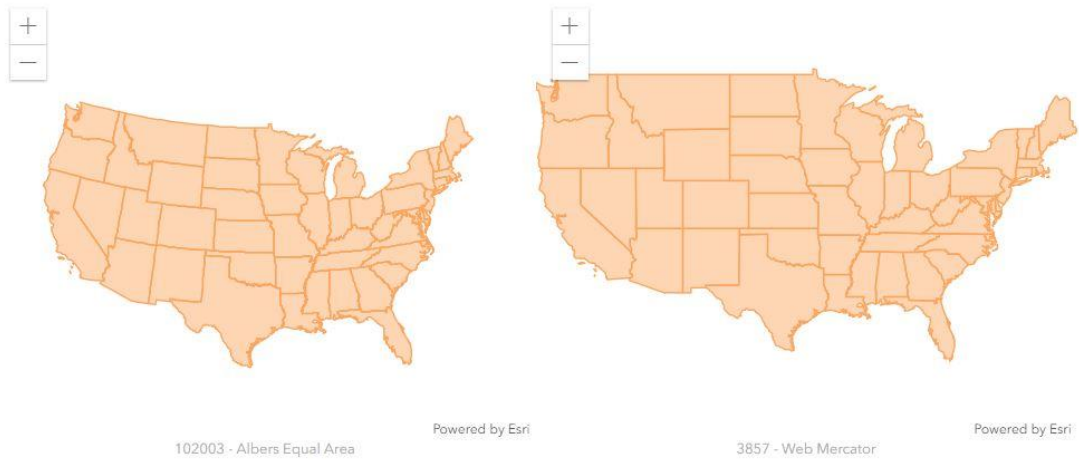
Dünyamız kutuplardan hafifçe basık ve ekvatorundan şişkin bir şekle sahiptir. Yeryüzü birçok farklı yapıya sahiptir. Bazı yapılar dağlar gibi daha yüksek, bazıları ise daha

alçaktır. Coğrafi Haritalarsa genelde iki boyutlu ve düzlemseldir. Dünyanın şekli ve yükselti göz önünde bulundurularak iki boyutlu ve düzlemsel sisteme taşımak gerekliliği vardır. Bu taşıma işlemi de hangi bölgede yapılacaksa ona uygun seçilen projeksiyon sistemleri ile yapılmaktadır. Dönüşüm işlemi hiçbir zaman kusursuz değildir ancak dönüşümü yapan kişi haritanın kullanım hedeflerine ve dönüşüm bölge özelliklerine bakarak en uygun projeksiyonu seçmeye çalışır [2,4].



Şekil 2.2. Dünyanın iki boyutlu düzleme dönüştürülmesi [2].

Coğrafi varlıklar yerel, bölgesel veya genel bir şekilde koordinat tabanlı evrensel bir şekilde ifade edebilmek için “SRS” sistemleri oluşturulmuştur. Bu sistemler “OGC” temelli “WKT” olarak tanımlanmış olup; coğrafi veri, farklı bir koordinat sistemine dönüştürülmek istendiğinde kullanılmaktadır. Tüm dünyada evrensel olabilmesi içinde “SRID” olarak adlandırılan id, formül değerleri içeren bir yapıda kullanılmaktadır [5].

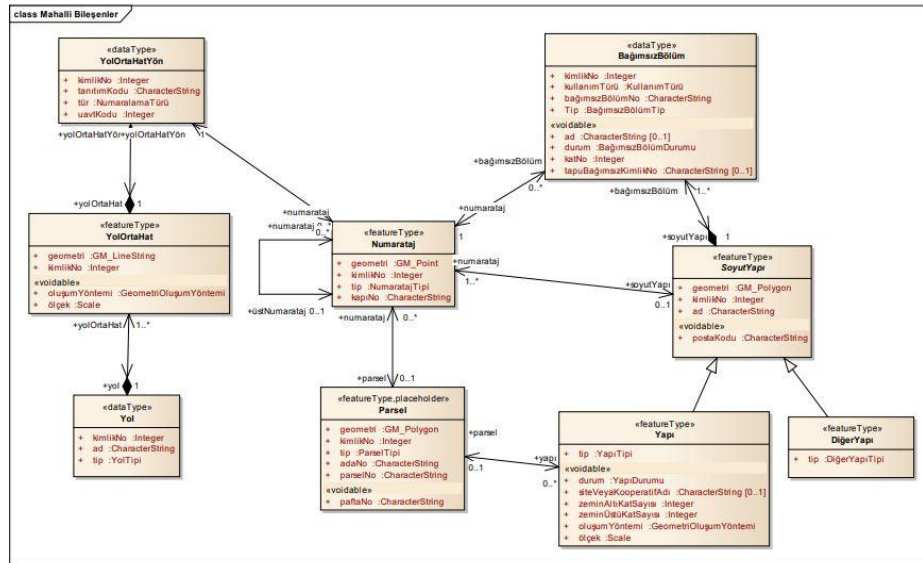


Şekil 2.3. Veriyi aynı kaynaktan alan farklı projeksiyon dönüşümlü haritalar [6].

2.1.1. Coğrafi bilgi sistemlerinin Türkiye'deki örnekleri

Ülkemizde bakanlıkların, yerel yönetimlerin birçok CBS projesi bulunmaktadır. Bu projelerin genelde OGC ve Inspire standartlarında gerçekleştirilmektedir. Türkiye'de birçok devlet kurumu idari birimler, bina, adres, orta görüntü, jeoloji, arazi örtüsü, hidroğrafya gibi birçok konuda veri toplamak veya üretmek zorundadır. Inspire, 2001 yılında Avrupa Birliği tarafından başlatılmış ve birlikte çalışabilirlik esaslarına göre servis mimarisi ve içeriğini tanımlayan bir projedir. Ülkemizde bu esaslara uygun bir şekilde çalışmalarını yapmaktadır. Bu proje ile Türkiye'deki tüm coğrafi adres çalışmaları tek bir veri tabanında tutulup, ihtiyaç duyan birçok kuruma servis edilebilecektir [7,8].

MAKS projesi İç İşleri Bakanlığı tarafından yönetilmektedir. 24 Haziran 2011 tarihinde başlamıştır. 2012 yılında tamamlanmıştır. Hala yaygınlaştırma projeleri devam etmektedir. Bu proje ile Adres Kayıt Sisteminde sözel olarak tutulan verilere coğrafi koordinatların eklenmesi hedeflenmiştir. Belediyelerin numarataj ve adres bilgilerini coğrafi olarak üretecekleri altyapı oluşturulmuştur. Türkiye Cumhuriyeti vatandaşlarının yapı belge başvurularını takip edebilecekleri ve Muhtarlar için modüller bulunmaktadır [9].



Şekil 2.4. Mahalli bileşenler uygulama şeması [9].

TUCBS projesi Çevre ve Şehircilik Bakanlığı tarafından yürütülmektedir. Bu kapsamda bakanlık veri temalarını oluşturmaktadır. Kurumlar arası çalışabilirlik ve veri paylaşım standartlarını oluşturmaktadırlar. Bakanlık bunun yanında diğer kurumlar kullanımını için gerçek ortofoto ve 3 boyutlu veri üretimi yapmaktadır. Diğer kurumların kendi verilerini servis edebilecekleri hem de ihtiyaç duydukları verileri servis olarak alabilecekleri GeoPortal uygulamasını yine bu kapsamda geliştirmeye devam etmektedir. Şu anda on iki veri temasının katman ve öznitelik belirleme çalışmaları tamamlanmıştır [8].

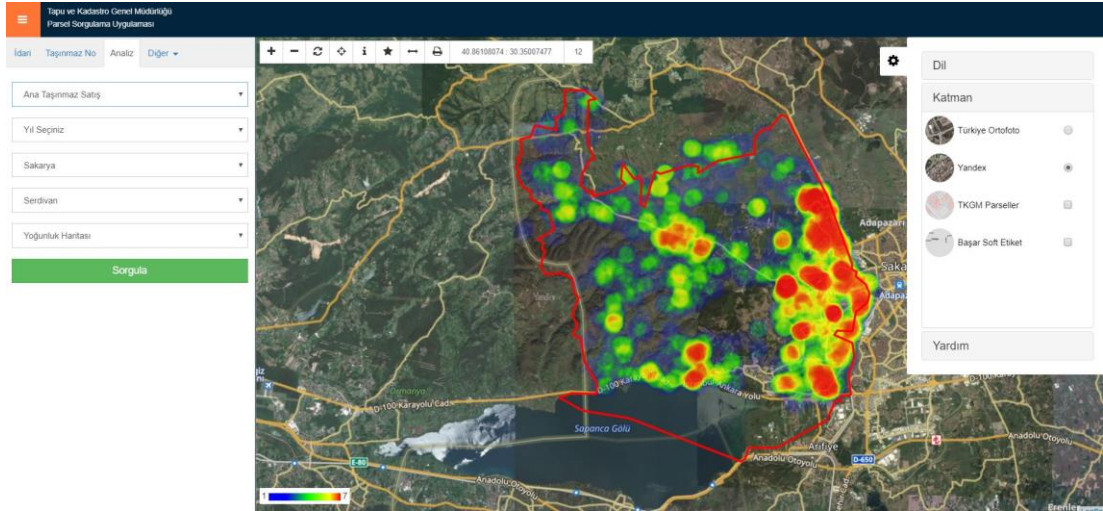
ULUSAL COĞRAFI VERİ TEMALARI

1	KOORDİNAT REFERANS SİSTEMLERİ VE COĞRAFI GRID SİSTEMLERİ	2018 Yılı Tamamlanan
2	İDARİ BİRİMLER	2018 Yılı Tamamlanan
3	COĞRAFI YER ADLARI	2018 Yılı Tamamlanan
4	KADASTRO	2018 Yılı Tamamlanan
5	BİNA	2018 Yılı Tamamlanan
6	ADRES	2018 Yılı Tamamlanan
7	YÜKSEKLİK	2018 Yılı Tamamlanan
8	ORTOGÖRÜNTÜ	2018 Yılı Tamamlanan
9	ULAŞIM AĞLARI	2018 Yılı Tamamlanan
10	HİDROĞRAFYA	2018 Yılı Tamamlanan
11	JEOLJİ	2018 Yılı Tamamlanan
12	ARAZİ ÖRTÜSÜ	2018 Yılı Tamamlanan
13	ARAZİ KULLANIMI	2019 Yılı Planlanan
14	TOPRAK	2019 Yılı Planlanan
15	KORUMA BÖLGELERİ	2019 Yılı Planlanan
16	DOĞAL AFET RİSKİ BÖLGELERİ	2019 Yılı Planlanan
17	ALTYAPI	2019 Yılı Planlanan
18	ENERJİ KAYNAKLARI	2019 Yılı Planlanan
19	MADENLER	2019 Yılı Planlanan
20	İNSAN SAĞLIĞI VE GÜVENLİĞİ	2019 Yılı Planlanan
21	NÜFUS DAĞILIMI – DEMOGRAFI	2019 Yılı Planlanan
22	ÇEVRE İZLEME TESİSLERİ	2019 Yılı Planlanan
23	SANAYİ TESİSLERİ	2019 Yılı Planlanan
24	TARIM TESİSLERİ	2019 Yılı Planlanan
25	KAMU YÖNETİM BÖLGELERİ	2019 Yılı Planlanan
26	TUR DAĞILIMI	2019 Yılı Planlanan
27	HABİTAT BÖLGELERİ	2019 Yılı Planlanan
28	BİYOCOĞRAFYA BÖLGELERİ	2019 Yılı Planlanan
29	DENİZ VE TUZLU SU ALANLARI	2019 Yılı Planlanan
30	ATMOSFER VERİLERİ	2019 Yılı Planlanan
31	METEOROLOJİ VERİLERİ	2019 Yılı Planlanan
32	İSTATİSTİKSEL RAPORLAMA BÖLGELERİ	2019 Yılı Planlanan

Şekil 2.5. TUCBS coğrafi veri temaları [8].

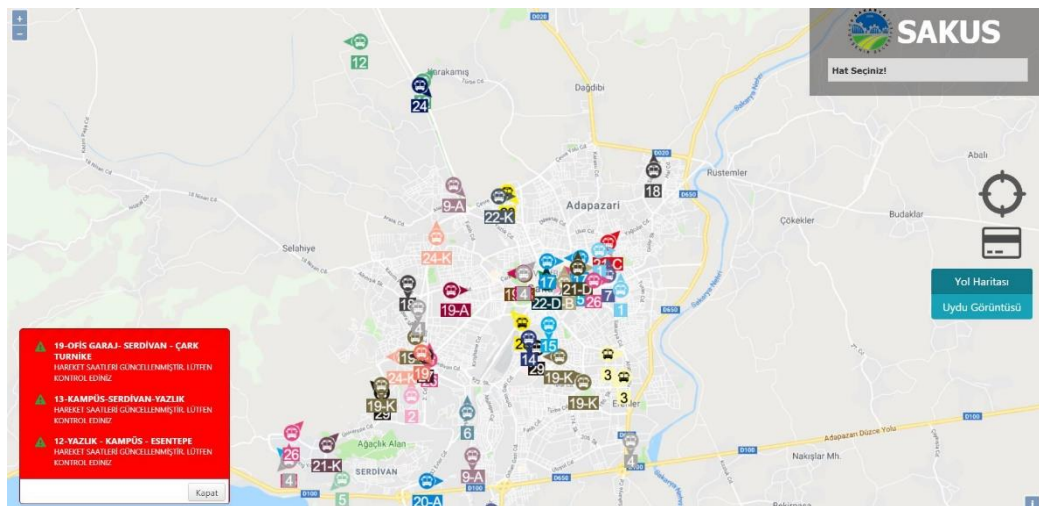
TKGM parsel sorgulama uygulaması; Tapu ve Kadastro Genel Müdürlüğüne geliştirilmektedir. Bu uygulama ile vatandaşlar ada, parsel numarasından tapu

kayıtlarını görüntüleyebilmektedir. Bina, bağımsız bölüm ipotekli/ipoteksiz satış, alım satım yoğunluğu gibi tematik haritalara ulaşılabilir [10].



Şekil 2.6. TKGM uygulama ekranı [10].

SAKUS uygulaması Sakarya Büyükşehir Belediyesi tarafından geliştirilmiştir. Bu uygulama; otobüslerin anlık konumları, güzergahları, ulaşım haberleri, hareket saatleri, kart dolun noktaları gibi bilgileri sunmaktadır. Mobil uyumlu olarak açık kaynak kodlu teknolojiler ile geliştirilmiştir. Kullanıcıların yakınından geçen hatları göstermektedir. Kullanıcılar favori hatlarını kaydedip, toplu bir şekilde görebilmektedirler.



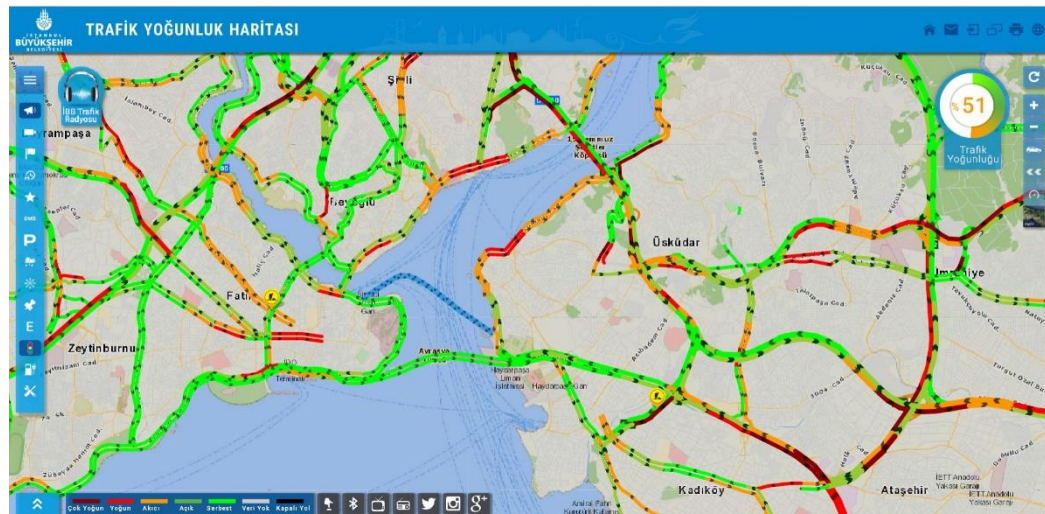
Şekil 2.7. Sakus uygulaması [11].

Sakarya Büyükşehir Belediyesi proje harita uygulaması; planlanan, devam eden, bitmiş projeler, kategorilere ayrılmış bir şekilde harita üzerinde görüntülenebilmektedir. Konum üzerine tıklandığında proje bilgisi ve fotoğrafları gözükmetedir [12].



Şekil 2.8. Proje Haritası [12].

İstanbul Büyükşehir Belediyesi trafik yoğunluk haritası ile İstanbul ili trafik yoğunluğuna ulaşılabilmektedir. Bunun yanında trafik lambaları, nöbetçi eczaneler, otopark noktaları, yol çalışmaları, elektrikli araç şarj noktaları, önemli yerler, iskeleler de harita üzerinde görüntülenebilmektedir. Harita üzerinden seçilen şehir kamerası ile trafik yoğunluğu bilgisi gözükürken yolun gerçek görüntüsünde ulaşılabilmektedir.



Şekil 2.9. İstanbul Büyükşehir Belediyesi trafik yoğunluk haritası [13].

2.1.2. Coğrafi bilgi sistemlerinin alt bileşenleri

Coğrafi bilgi sistemlerini gerçekleştirebilmek için; farklı problemlere çözüm getiren birçok yazılımsal araca ihtiyaç duyulmaktadır. Bunlar coğrafi veri tabanları, harita servis yazılımları, masaüstü, web ve mobil uygulamalar olarak gösterilebilirler. Her bir kategori için açık kaynak kodlu veya lisanlı birçok farklı firma veya kişilerce geliştirilmiş araçlar bulunmaktadır. Aynı kategorilerde bulunan yazılımsal CBS araçları benzer sorunlara çözüm üretmektedirler. Bu araçlardan bazıları kategorik olarak sınıflandırılarak, iyi ve eksik yönleri incelenecektir.



Şekil 2.10. Coğrafi bilgi sistemleri bileşenleri.

2.1.2.1. Coğrafi veri tabanları

“Google”, “Yandex” gibi firmaların harita uygulamaları; insanların herhangi bir şeyin konumunun nerede olduğu sorusuna cevap bulmasını sağlamaktadırlar. Kullanıcıya marketin yanından sağa dön ve aradığınız yer yüz metre ilerde gibi sonuçlar ile bilgilendirmektedirler. Sağladıkları interaktif harita deneyimi ile kullanıcının bir çok yeri keşfetmesine yardımcı olmakta ve kullanıcının kendi konum tanımlamalarına

imkan tanıyarak, veri toplama işlemini kolaylaştırmaktadırlar. Tüm bu işlemler ve kolaylıklar bize coğrafi veri tabanları tarafından sağlanmaktadır. Coğrafi veri tabanları ile geometrik verilerimizi depolama ve analiz etme imkanına sahip olmaktayız. Örneğin pizza yemek istediğimizde coğrafi veri tabanları yardımı ile analiz yapan programlar en çok tercih edilen ve beğenilen pizzacıları bize gösterebilmektedirler. Kullanıcılara tutulan geometrik veri ile yalnızca harita üzerinde konum bilgisi göstermenin yanında, mesafe analizleri yapılabilmekte, bir kurumda çalışan personeller adreslerine göre coğrafi olarak kategorize edilebilmektedir. İki boyutlu coğrafi bilgi sistemleri için üç basit geometri tipi ile konumlarımızı tanımlayabilmekteyiz. Bu geometrik veri tipleri; nokta, çizgi ve kapalı alandır. Bir yol bilgisi çizgi veri tipi ile bir bina bilgisi kapalı alan veri tipi ile gösterilebilmektedir [14].



Şekil 2.11. Coğrafi veri tipleri.

Açık kaynak kodlu ve bedava veri tabanlarından olan “PostgreSQL” veri tabanı ile birçok basit ve karmaşık nesneyi ilişkisel veri modeli ile depolayabilmekteyiz. Yine açık kaynak kodlu “PostGIS” eklentisi ile PostgreSQL veri tabanına coğrafi özellikler kazandırılmaktadır. PostgreSQL dışında bir çok ilişkisel veri tabanı coğrafi özellikleri sunmaktadır. Bunlardan açık kaynak kodlu olanlarına örnek olarak “SQLite” ve lisanslı ürünlerde “SqlServer”, “DB2” ve “Oracle” veri tabalarının da çözümleri vardır. Oracle daha küçük ölçekli coğrafi veriler için “Locator”, daha büyük ölçekli veriler içinse “Spatial” adından iki ürüne sahiptir [14].

2016 yılında aynı veri setleri ve coğrafi sorgulamalar üzerinden, Oracle Spatial ve Postgis eklentili PostgreSQL üzerinde yapılan bir çalışmaya göre iki coğrafi veri tabanının performans ve kaynak kullanımı kıyaslanmıştır. En yakın sonuçla, PostgreSQL üzerinde çalışan sorguların yüzde 50 daha hızlı sonuçlandığı görülmüştür. En iyi sonuçla da PostgreSQL üzerinde çalışan sorguların yüzde 450 daha hızlı çalıştığı

görülmüştür. PostgreSQL veri tabanı bu kıyaslamada Oracle veri tabanına göre biraz daha fazla sistem kaynağı kullanmıştır [15].

2.1.2.2. Harita web servisleri

Veri tabanlarında tutulan coğrafi verinin herhangi bir platformdan sorunsuz bir şekilde ulaşılabilmesinin en kolay yollarından biri web servisleridir. Coğrafi web servisleri tarafına baktığımızda standartlar OGC tarafından belirlenmektedir. OGC kar amacı gütmeyen, sunucu ve istemci arasındaki coğrafi veri transferinin yapılabilmesi için “WFS”, “WMS”, “WCS” gibi veri tanımlama standartlarını oluşturan kuruluştur [16]. Bu standartlara uygun şekilde yazılmış tüm servisler evrensel bir formata sahip olduğu için aynı standarda uygun geliştirilmiş web, mobil ve masaüstü yazılımlarınca sorunsuzca kullanılabilir.

Web servisleri için Restful mimariden faydalanmak günümüzde çok popülerdir. Restful servisler “HTTP” protokolünün “GET”, “POST”, “PUT”, “DELETE” metotları ile oluşturma, okuma, güncelleme, silme işlemlerini yapabilmektedir [17].

Coğrafi servis oluşturma işlemini kolaylaştırmak ve OGC standartlarını sağlayabilmek için geliştirilmiş birçok araç mevcuttur. Bunlardan açık kaynak koldu olanlara “Geoserver”, “Mapserver” örnek olarak gösterilebilir. Lisanslı ürün tarafında ise “Arcgisserver” ürünü mevcuttur. Geoserver “Java” programlama dili ile yazılmış, platform bağımsız bir harita servislerini yayınlama aracıdır. Restful servis mimarisini gerçeklemektedir. Geoserver birçok veri tabanına bağlanabilmektedir. Veri tabanındaki bilgileri okuyup vektör, resim gibi birçok farklı formatta servis edebilmektedir. İçerisindeki kullanıcı, rol, yetki sistemi ile servislere erişim seviyeleri oluşturulabilmektedir. Performans artışı için bellekten getirme seçenekleri vardır. Sistem yöneticilerinin, kullanıcıların karşılaştığı hataları detaylı bir şekilde görüntüleme ve analiz etme imkanı vardır. Geoserver uygulamasında tüm bu hizmetlerden faydalanmamızı sağlayacak web yönetici ekranları vardır [18].

Sunucu Durumu
Sunucu yapılandırma ve durum özeti

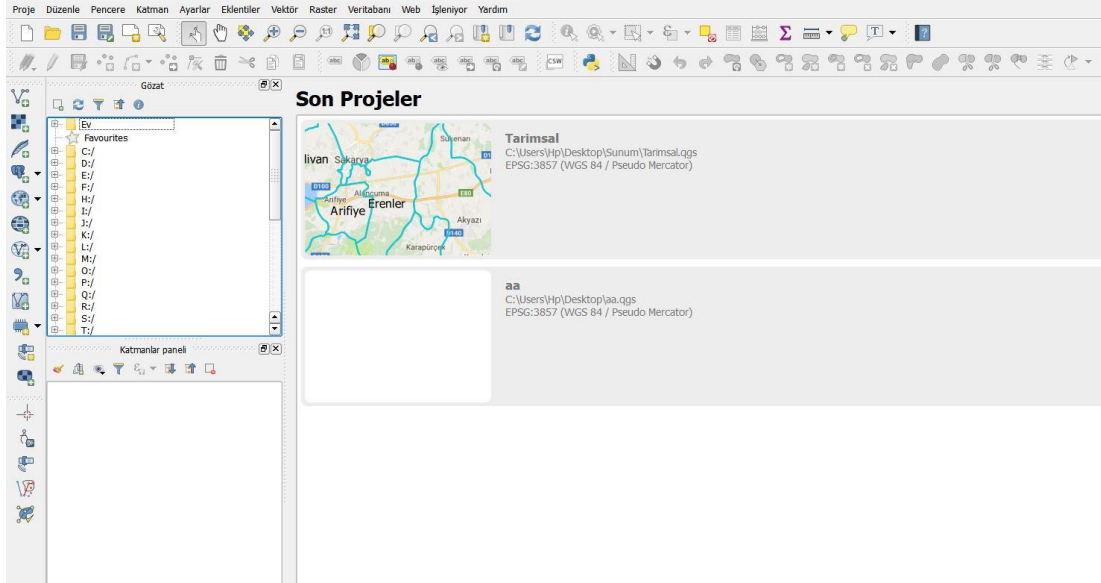
Status	Modules	Eylem
Veri klasörü	C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\geoserver\data	
Kilitler	0	Boş kilitler
Bağlantılar	21	
Hafıza Kullanımı	151 MB / 910 MB	Boş hafıza
JVM Versiyonu	Oracle Corporation: 1.8.0_144 (Java HotSpot(TM) 64-Bit Server VM)	
Java Rendering Engine	sun.dc.DuctusRenderingEngine	
Mevcut Fontlar	GeoServer 389 farklı font dosyasına ulaşabiliyor.Mevcut fontların tam listesi	
Yerel JAI	false	
Yerel JAI ImageIO	false	
JAI Maksimum Hafıza	455 MB	
JAI Hafıza Kullanımı	0 KB	Boş hafıza
JAI Hafıza Eşliği	75%	
JAI Kare Parçaları Sayısı	7	
JAI Kare Parça Önceliği	5	
ThreadPoolExecutor Çekirdek Havuz Boyutu	5	
ThreadPoolExecutor Maksimum Çekirdek Havuz Boyutu	10	

Şekil 2.12. Geoserver web uygulama arayüzü.

2.1.2.3. Masaüstü uygulamalar

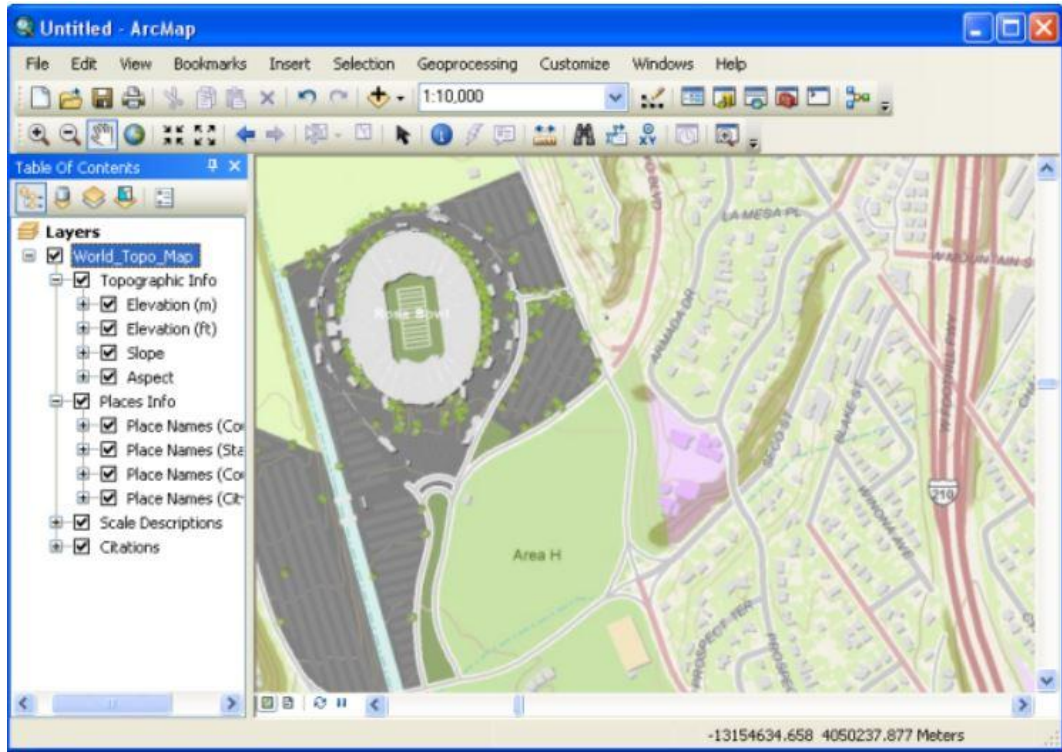
Coğrafi bilgi sistemlerinde verinin hassas bir şekilde üretilmesi en önemli konulardan biridir. Günümüzde en hassas çizim imkanına sadece masaüstü uygulamalar sahiptir. Birçok lisanslı veya açık kaynak kodlu ve bedava masaüstü CBS yazılımı bulunmaktadır. Bu masaüstü yazılımlarından bazıları “Quantum Gis”, “ArcGis”, “uDig”, “Map Info”, “Netcad” dir. Bu yazılımların birbirlerine karşı bir çok avantaj ve dezavantajları mevcuttur. Bunlardan açık kaynak kod dünyasında çok meşhur Quantum Gis ile lisanslı ürünler arasında ön plana çıkan ArcGis yazılımı çok fazla kıyaslanmaktadır.

Quantum Gis çok fazla veri tipini okuyabilmesi, bedava olması, bir çok eklentisinin bulunması, gelişmiş semboloji paketinin olması, dinamik etiket stillendirme yapısı, tematik harita yardımcıları ile çok tercih edilen bir masaüstü CBS uygulamasıdır [19].



Şekil 2.13. Quantum Gis CBS uygulaması.

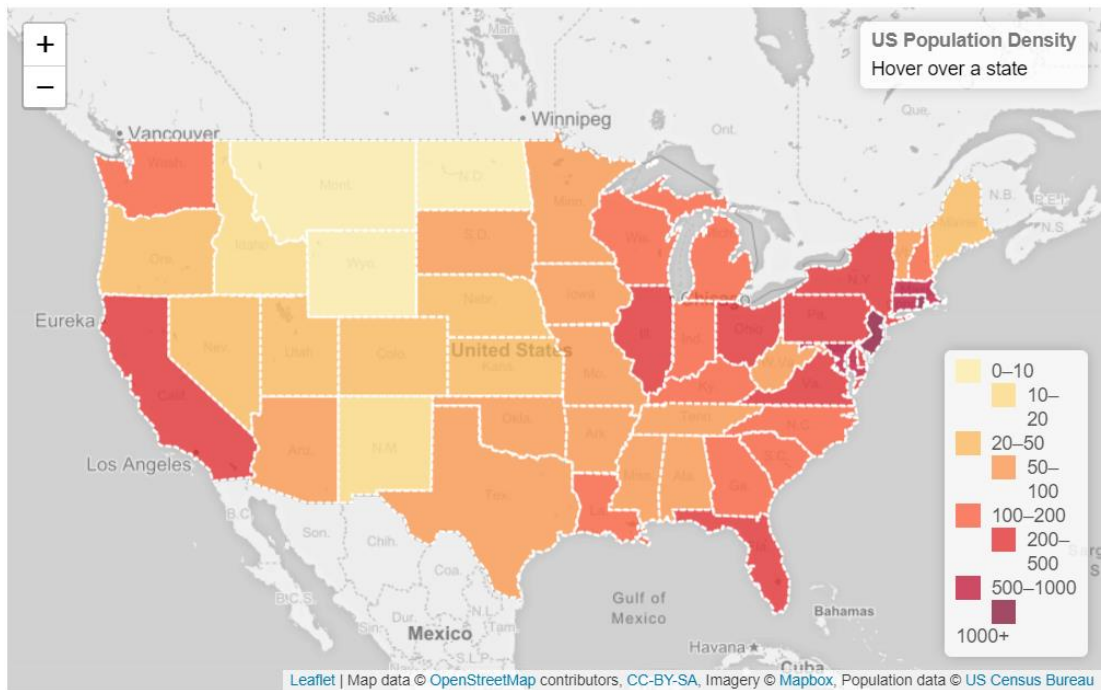
ArcGis ise gelişmiş dökümanyasyonu, kullanıcı dostu arayüzü, istatistiksel analiz araçları, hataları hemen bildiren gelişmiş çizim araçları ile ön plana çıkan lisanslı bir masaüstü CBS uygulamasıdır [19].



Şekil 2.14. ArcGis masaüstü harita uygulaması [20].

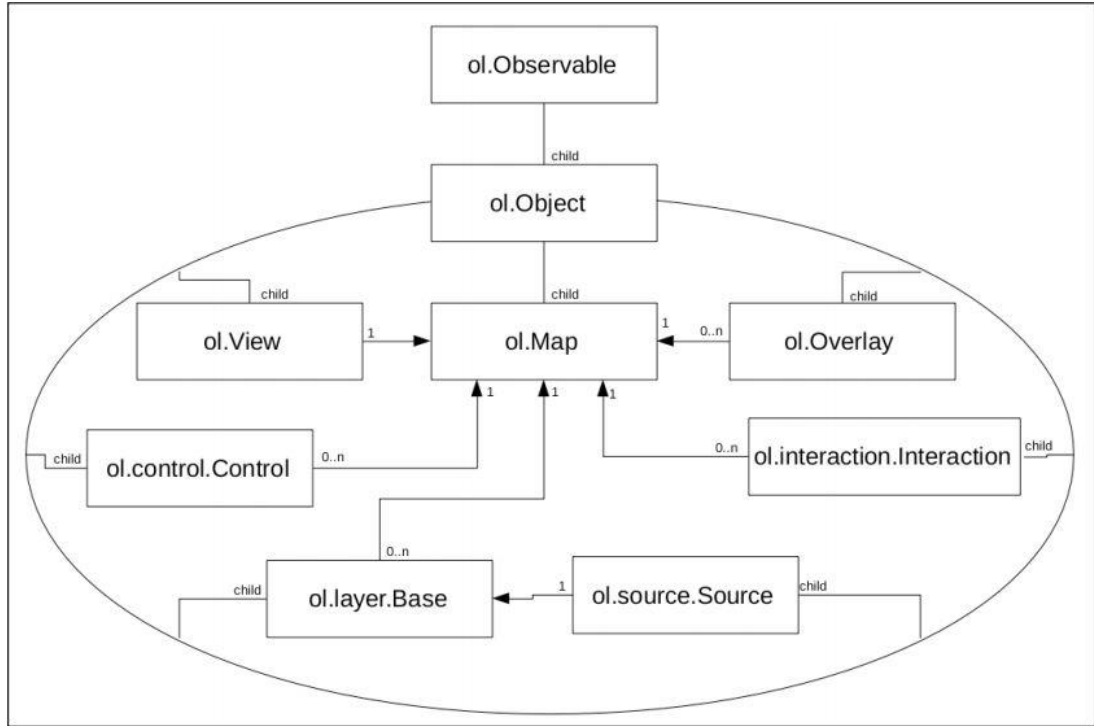
2.1.2.4. Web/Mobil uygulamalar

Web/mobil CBS uygulamaları geliřtirmek için birok aık kaynak kodlu kütüphane ve teknoloji bulunmaktadır. Hem web ortamı hem de mobil ortamdaki seenekleri ile ‘‘Leaflet’’ ve ‘‘Openlayers’’ javascript kütüphaneleri mevcuttur. Leaflet basit ve anlaşılır yapısı, az ‘‘Javascript’’ kodu ile uygulama geliřtirme, mobil dostu, daha küçük kütüphane boyutları ile öne ıkmaktadır. Openlayers ise daha eski ve geliřmiş özellikleri, daha iyi dökümantasyonu ile ön plana ıkmaktadır [21].



Şekil 2.15. Örnek Leaflet ekranı [22].

Openlayers kütüphanesi ile OGC formatındaki vector, resim servislerini kullanıp web ve mobil ekranlar geliřtirilebilmektedir. Bu kütüphane ile harita zoom seviyeleri, kuş bakışı projeksiyon deęiřtirmeleri, geometrik veri stillendirmeleri, açılır kutular, tematik haritalar kolayca yapılabilir. Bir takım eklentiler ile üçüncü boyut desteęi bulunmaktadır. Bir ok farklı teknoloji ile beraber kullanılabilir [23].



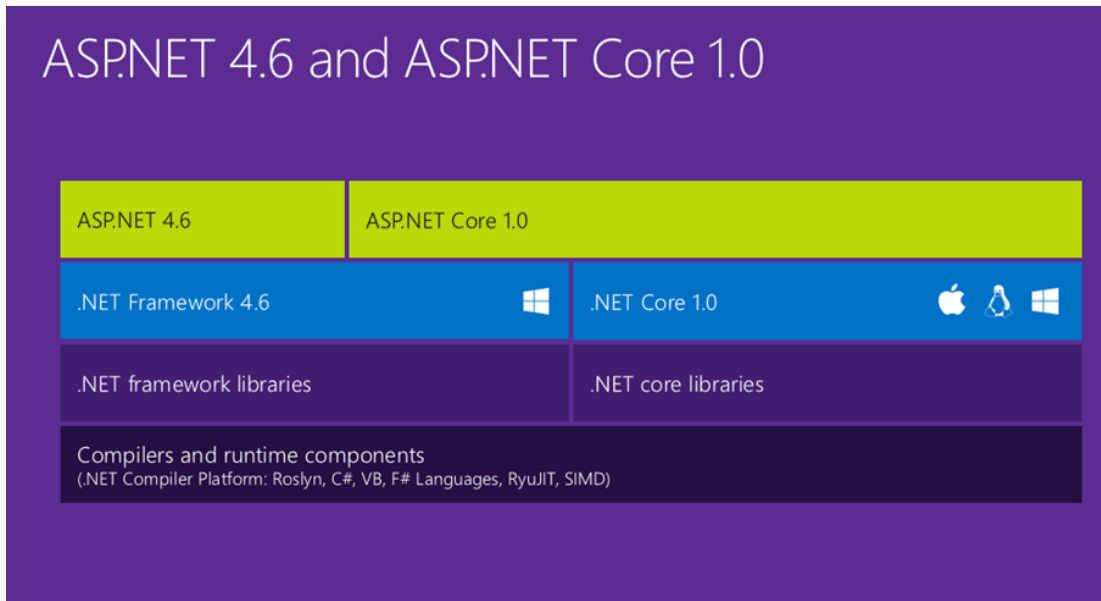
Şekil 2.16. Openlayers kütüphanesi UML diagramı [23].



Şekil 2.17. Openlayers harita ekranı [24].

Web/mobil CBS uygulamaları geliştirirken Openlayers gibi kütüphaneler her türlü ihtiyacımızı karşılayamamaktadır. Kullanıcı, rol, yetki işlemleri, oturum yönetimi, form da girilen verilerin doğrulanması, link yönlendirmesi gibi konular için bir çok

teknoloji bulunmaktadır. Bunlardan açık kaynak kodlu ve bedava olarak “Asp.net Core MVC” teknolojisi yeni bir teknoloji olmasına rağmen oldukça popülerlik kazanmıştır. Platform bağımsızdır. Visual studio ve visual studio code gibi birden fazla geliştirme ortamına sahiptir. Uygulamanın hem kullanıcı arayüzleri hemde arka tarafını geliştirmek için birçok kolaylık sunmaktadır. Açık kaynak kodlu olduğu için kullanıcı kaynak kodu indirip kendine has bir kütüphane oluşturma imkanı da vardır. Asp.net MVC teknolojiye göre yenilenmiş özellikleri ile daha az yer kaplamakta ve daha hızlı çalışmaktadır. Düzenlenmiş yapılandırma dosyaları ile uygulama yönetimi daha da basitleşmiştir. İçerindeki Html etiket ve bileşen yardımcıları sayesinde az kodla çok iş mantığını desteklemektedir. Kod yeniden kullanılabilirliği çok yüksektir. Html 5 desteği ile birçok tarayıcıda sorunsuz çalışan bileşenlere sahip olmaktadır. Restful servis mimarisinide içerisinde barındırmaktadır [25].

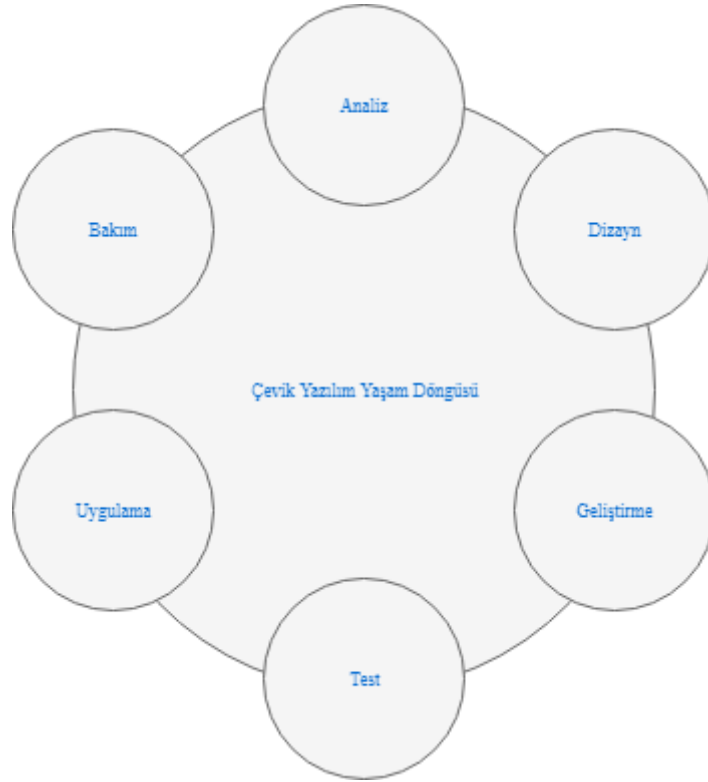


Şekil 2.18. Asp.net Core platformu [26].

2.2. Yazılım Mimarisi

Yazılım mimarilerinin temel hedefi sistemin yaşam döngüsünü desteklemektir. İyi yazılım mimarisi ile sistemin daha kolay anlaşılmasını, geliştirilmesini, bakımını ve dağıtımını sağlamaktadır. Bu şekilde sistemin geliştirme maliyetinin azaltılması amaçlanmıştır. Sistem gerçekleştirilmeden önce; seçilecek yazılım mimarisi

argümanları ile problemin en efektif bir şekilde çözüme kavuşturacak araç ve teknolojiler seçilmiştir. Nesneye dayalı programlama ile geliştirilecek uygulamanın problemleri modellenmiştir. Nesneye dayalı programlama gerçek dünyaya daha yakın bir yapıda olduğu için gerçekleştirilen uygulamanın kaynak kodunun daha anlaşılır olmasını sağlamaktadır [27]. Projeye yönetiminde çevik yöntemlerden faydalanılmıştır.



Şekil 2.19. Çevik Yazılım Yaşam Döngüsü.

2.2.1. Katmanlı mimari

Katmanlı mimari ile yazılım kendi içinde bağımsız ve birbiri ile ilişkili fonksiyonlardan oluşan parçalara bölünmektedir. Tüm bu parçalar yatay bir şekilde sıralı ve önceki parça sonraki parçaya bağlı bir şekilde yazılımı oluşturmaktadır. Uygulamanın karmaşıklığına göre katman sayısı belirlenmektedir. İki katmanlı mimari en basit ve hızlıca kullanılabilen katmanlı mimari örneğidir. Birinci katman kullanıcı arayüzü, ikinci katman ise veri erişim katmanıdır. Kullanıcı arayüz katmanı; kullanıcının uygulama ile etkileşime geçeceği bir yöntem sunmaktadır. Bu bir web sayfası, masa üstü uygulaması veya mobil uygulama olabilmektedir. Kullanıcıya istediği veriyi

göstermektedir. Kullanıcının isteklerini alıp, gerekli doğrulama işlemlerini yaptıktan sonra veri erişim katmanına iletmektedir. Veri erişim katmanı da isteklere uygun veri sorgulama işlemlerini yapıp, kullanıcı arayüz katmanının anladığı formata çevirmektedir. İki katmanlı mimari iş kurallarının az olduğu durumlarda kullanılmaktadır. Üç katmanlı mimaride ise fazladan; iş kurallarının ve süreçlerin yönetimi için iş katmanı bulunmaktadır. İş katmanı kullanıcı arayüz katmanı ile veri erişim katmanının arasında konumlandırılmaktadır. Katmanlı mimari ile uygulamanın daha anlaşılır ve test edilebilir olması sağlanmaktadır [28].

2.2.2. Tasarım desenleri

Günümüzde yazılımı oluşturan kodlar esnek, kolayca bakımı yapılabilir ve yeniden kullanılabilir olmalıdır. Kodların kalitesini ölçmek için tasarım desenlerinden ne kadar faydalandığına bakılabilir. Yazılım geliştirirken çözülmesi gereken birçok problemle karşılaşmaktayız. Bu problemler genelde yazılım dünyasında genel kabul görmüş ve en etkili çözümü sunan tasarım desenleri ile kolayca çözülebilmektedir. Tasarım desenleri ile uygulama geliştirirken problem yaşayabileceğimiz birçok konuyu da önceden bertaraf etmektedir.

Uygulamamızda kullandığımız tasarım desenleri şu şekildedir;

- Singleton tasarım deseni: Eğer bir nesnenin tek bir merkezi örneğinin oluşturulmasını istiyorsak, bu tasarım deseni ile yapabilmekteyiz. Bu şekilde hem bellek kullanımını optimize etmiş oluruz hem de ortak bir nesne başvuru örneği tanımlamış oluruz [29].
- Fabrika tasarım deseni: Uygulama içindeki nesnelerin üretimi için kurallar tanımlanıp, çok biçimlilik desteklenmiştir [29].
- MVC tasarım deseni: Bu tasarım deseni ile kullanıcı arayüzlerini geliştirirken uygulamayı birbiri ile ilişkili üç parçaya bölmekteyiz. Kod yeniden kullanılabilirliğini ve paralel geliştirme yapmayı desteklemektedir. Birinci

parça olan “view” ile kullanıcı arayüzlerini tanımlayabiliriz. İkinci parça olan “model” ile uygulama verisini ve iş kurallarını yönetebilmekteyiz. Sonuncu ise ilk iki parçanın arasındaki iletişimi ve koordinasyonu sağlayan “controller” parçasıdır. Son parçada istenirse kullanıcının girdiği verilere gerekli doğrulama işlemleri yapılabilmektedir [29].



Şekil 2.20. MVC tasarım deseni.

- Dependency injection deseni: Kodlar ve modüller arasındaki bağımlılıklar yönetilebilmektedir. Uygulamanın içerisindeki bu yapıların birbirlerine az bağımlı olması sağlanarak; bir yerde yapılan değişikliğin başka bir yeri daha az etkilemesi hedeflenmektedir [28].
- Repository tasarım deseni: Kod tekrarlarından kaçınılmaya kolaylaştırmaktadır. Bu tasarım deseni ile veri tabanı işlemleri belli bir formatta ve kuralla yönetilebilmektedir.

2.2.3. Yazılım güvenliği

Yazılım güvenliği yazılım geliştirme yaşam döngüsünün her alanında olması gereken multi-disipliner bir alandır. Yazılım mimarisi oluşturulurken dikkate alınması gerekmektedir. Bu alan için kullanılacak birçok otomatik çalışan test araçları mevcuttur. Güvenlik kavramı yazılımın her alanında bulunması gerektiği için standardizasyona ihtiyaç vardır. Bu standarsizasyonu sağlayan kuruluşlar vardır. En önemlilerinden biri OWASP’dir. OWASP 21 Nisan 2004 tarihinde kar amacı gütmeyen bir organizasyon olarak kurulmuştur. Dünyanın her yerinde seminer, çalıştay ve konferanslarını yaparak katılımcıların da desteğiyle güvenlik standartlarını devamlı geliştirmektedirler. Tüm OWASP dökümanlarına açık ve bedava olarak ulaşılabilir. Son güncel halinde uygulama güvenliği on başlık altında toplanmıştır. Bu on başlık ile sql sızıntıları, betik atakları, şifre güvenliği, yetki ve

oturum yönetimi, loglama ve monitörleme, servis kullanımı gibi konularda standart çözümler sunulmaktadır. Bu başlıklar bir çok uygulamada güvenliği daha yönetilebilir ve izlenebilir hale getirmiştir [30,31,32].

A1 – Siteler Arası Betik Yazma (XSS)	XSS açıkları uygulama kullanıcıdan veri alıp, bunları herhangi bir kodlama ya da doğrulama işlemine tabi tutmadan sayfaya göndermesi ile oluşur. XSS saldırısının kurbanın tarayıcısında kullanıcı oturumları bilgilerin çalınmasına, web sitesinin tahrip edilmesine veya solucan yüklenmesine sebep olan betik çalıştırmasına izin verir .
A2 – Enjeksiyon Tasarım Hatası	Enjeksiyon saldırılarına , özellikle SQL enjeksiyonu, web sitelerinde sıkça rastlanmaktadır . Enjeksiyon, kullanıcı tarafından alınan verinin yorumlayıcıya (interpreter) komut ya da sorgunun bir parçası olarak gönderilmesi durumunda oluşur. Saldırının düşmanca gönderdiği veriler yorumlayıcının istenmeyen komutları çalıştırmasına veya veriyi değiştirmesine sebep olur.
A3 – Zararlı Dosya Çalıştırma	Uzaktan Dosya Ekleme (RFI) ataklarına karşı zayıflıklan olan kod, saldırının sunucunun güvenliğini tehlikeye düşürebilecek kadar tehlikeli düşmanca kodlar ve veriler yüklemesine yol açar. Kötü amaçlı dosya çalıştırılması kullanıcıdan dosya adı veya dosya kabul eden PHP, XML veya herhangi bir çerçeveyi (framework) etkileyebilir.
A4 – Güvensiz Doğrudan Nesne Başvurusu	Doğrudan Nesne Referanslama; geliştirici dosya, dizin, veritabanı kaydı gibi bir bilgiyi URL veya form parametresi olarak alıp bunu uygulamaya referans olarak tanımladığı zaman oluşur. Böylece saldırın referansı manipule ederek yetkisi olmayan nesnelere erişebilir.
A5 – Siteler Ötesi İstek Sahtecilliği (CSRF)	CSRF saldırıları; sisteme girişi yapmış mağdura ait tarayıcının, bir web uygulamasına sonradan saldırının yararına olacak, düşmanca hazırlanmış ve önceden doğrulanmış bir istek göndermesine sebep olur. CSRF, saldırdığı web uygulaması kadar güçlü olabilir.
A6 – Bilgi Sızıntısı ve Uygunsuz Hata İşleme	Uygulamalar istemeden de olsa yapılandırılmaları, iç işleyişleri hakkında bilgi sızdırabilir veya uygulama sorunlarından dolayı güvenlik ihlallerine yol açabilir. Saldırınlar bu zayıflıkları daha ciddi saldırılar gerçekleştirmek veya önemli bilgileri çalmak için kullanabilir.
A7 – İhlal Edilmiş Kimlik Doğrulama ve Oturum Yönelimi	Hesap Bilgileri ve oturum anahtarları çoğu zaman düzgün olarak korunmamaktadır. Saldırınlar şifreleri ve kimlik denetimi anahtarlarını kullanıcının diğer bilgilerini elde etmek için kullanabilirler.
A8 – Güvensiz Kriptografik Depolama	Web uygulamaları verilerin ve bilgilerin güvenliğini sağlamak için nadiren kriptografik fonksiyonları kullanırlar. Saldırınlar zayıfça korunan veriyi, kimlik hırsızlığı ve kredi kartı dolandırıcılığı gibi diğer suçları işlemek için kullanırlar.
A9 – Güvensiz İletişimler	Uygulamalar önemli iletişim bilgilerini ve veri alışverişlerini korumaları gerekirken, çoğu zaman ağ trafiğini şifrelemekte yetersiz kalmaktadırlar.
A10 – URL Erişimini Kısıtlamada Bozukluk	Çoğu zaman uygulamalar, bağlantı ve URL satırlarını yetkili olmayan kullanıcılara sadece göstermeyerek hassas işlevselliği korumaktadırlar. Saldırınlar bu URL'lere doğrudan erişerek, yetkisi olmayan işlemleri gerçekleştirmek için bu zayıflığı kullanabilirler.

Şekil 2.21. İlk 10 web uygulama açıkları [32].

2.3. Yazılım Çalışma Ortamı ve Konteyner Yapısı

CBS yazılımları birçok parçadan oluştuğu ikinci bölümde anlatılmıştı. Tüm bu parçalar; coğrafi veri tabanı, harita web seviyesi ve uygulama, çalışabilmek için

sunuculara ihtiyaç duyarlar. Tüm bu parçalar bir sunucuda çalışabileceği gibi, birden fazla sunucuda kurularak da çalıştırılabilmektedir. Geliştirilecek CBS sisteminin büyüklüğüne göre farklı ölçekte modeller seçilebilmektedir. Yük dengeleyici sistemler ile daha büyük ve fazla isteğe cevap verebilecek ortamlar hazırlanabilmektedir. CBS yazılımları için öncelikli olarak seçilmesi gereken sistemin kaç fiziksel sunucudan oluşması gerektiğidir. Ardından her sunucu için gerekli programların kurulması ve ayarlamaların yapılması gerekmektedir. CBS uygulamaları birçok farklı teknolojinin birleşiminden oluştuğu için zahmetli bir kurulum ve yapılandırma sürecine sahip olabilmektedir.

Sunucu ölçeklemesi, ilgili program yüklemesi, yapılandırma ayarları için günümüzde konteyner yapısından çok fazla faydalanılmaktadır. Bu yapı ile tüm bu kurulumları ve yapılandırma işlemlerini tek bir seferde ayarlayıp, basit ve toplu bir kurulum dosyası oluşturulabilmektedir. Konteyner yapısını oluşturmak için birçok teknoloji bulunmaktadır. Bunlardan en popülerlerinden biri “Docker” programıdır. Docker programı ile yüklenecek programlar, o programların hangi yapılandırma ile çalışacağı, dışarıdan veri alışverişi yapılan başka bir program ile nasıl iletişim kurulacağı gibi bilgileri ile “Dockerfile” denen bir formatta dosyalar oluşturulabilmektedir. Bu dosya tüm yapılandırmanın şablon bilgilerini ve yol haritasını içermektedir. Bu Dockerfile’ı DockerEngin’e göndererek şablonun uygulanmış halini imaj olarak elde edebiliriz. Bu imajı alan herkes basit bir çalıştırma işlemi ile karmaşık kurulum işlemlerinden soyutlanarak imaj içeriğindeki programları kullanmaya başlayabilmektedir. Daha karmaşık yapılar için birden fazla imaj oluşturulmaktadır. Docker ile bir sunucuya kurulum yapabileceğimiz gibi birden fazla sunucuyada kurulum yapabilmekteyiz. Bunun için Docker imajlarını doğru bir şekilde oluşturmak gerekmektedir. Eğer yük dengeleyici sistemler ile bir programın birden fazla örneği seri ve paralel çalıştırılacak ise yine ilgili dosyalar oluşturulurken gerekli işlemlerin yapılması gerekmektedir. Docker ile tüm bu konteynirlama işlemleri yürütülebilmektedir. Kodu geliştiren yazılım ekibi ile kodun çalışma ortamında docker ile aynı ortam olması sağlanabilir. Bu sayede yazılım geliştiricinin bilgisayarında çalışan ama sunucuda çalışmayan kod problemlerinden de kurtulmuş olunabilmektedir [33].

BÖLÜM 3. CBS YAZILIM MİMARİ GELİŞTİRMESİ

3.1. Problemin Tanımı ve Önerilen Model

Coğrafi bilgi sistemleri asgari olarak yazılım, donanım, veri ve bunların yönetimi içiren multi disiplinler bir çalışma alanıdır [35]. Coğrafi bilgi sistemlerinin birçok çalışma alanından oluşması beraberinde farklı problemleri açığa çıkarmaktadır. Özellikle yerel yönetimlerin karşılaştığı insan kaynağı ile ilgili problemlerin başında; kurumdaki yetişmiş çalışanların sayısal olarak az olması, uzman personel bulma ve istihdam etme zorluğu ve yöneticilerin konusunda yeterince uzman olmaması gösterilebilir [36]. Yerel yönetimlerin insan gücü haricinde karşılaştığı bir diğer problem ise meta veri formatını belirlemektir. Meta veri formatı ile coğrafi verinin sahibinin, üretim zamanının ve ne amaçla üretildiği gibi künye bilgilerine ulaşabilmek mümkündür [37]. Meta verinin belirlenmesi ve gerektiğinde ilgili diğer kurumlar ile hangi protokoller ile paylaşılabilceği “Türkiye Ulusal Coğrafi Bilgi Sistemi (TUCBS)” olarak isimlendirilen Ulusal Konumsal Veri Altyapısı ile standardize edilme çalışmaları devletimiz tarafından devam ettirilmektedir. Bu çalışmalarda coğrafi portalın oluşturulması için gereken mimari ve web servis ihtiyaçları “OGC (Open Geospatial Consortium)” ve Avrupa Birliği tarafından yürütülen “Avrupa Birliği Konumsal Veri Altyapısı (INSPIRE)” projesi kapsamında yapılan çalışmalarla belirlenmiştir [7].

Yerel yönetimlerin kurumlarında uygun bir coğrafi bilgi sistemleri modeli kurmak için yukarıda sayılan standart ve çalışmaları takip etmeleri gerekmektedir. Çok fazla bileşenin bir arada bulunması ve yetkin personel azlığı sebebi ile bazı yerel yönetimlerin coğrafi bilgi sistemleri projelerini gerçekleştirebilmek için lisanlı ürünler kullandıkları görülmektedir. Lisanlı CBS ürünleri genelde yüksek maliyetli olmalarının yanı sıra diğer servislere entegrasyonları için de ayrıca uygulama yazılımlarına ihtiyaç

duyarlar. Bu uygulama yazılımları da yerel yönetimler için ekstra maliyet ve zaman kaybına sebep olabilmektedir.

Yukarıda bahsedilen sebeplerden ötürü, bu çalışmada açık kaynak kodlu ürünler ile coğrafi veri tabanı, yazılım, harita sunucusu, veri toplama ve analizini içeren OGC standartları ile uyumlu katmanlı bir yazılım mimarisi önerilmiştir. Lisanslı ürün, teknik personel ve fiziksel ekipmandan yoksun yerel yönetimlerin, geliştirilen katmanlı ve açık kaynak kodlu CBS yazılım mimari önerisini kullanarak kendi coğrafi bilgi sistemlerini hayata geçirebilmeleri hedeflenmektedir. Bir çok farklı açık kaynak kodlu yazılım (GeoServer, OpenLayers vb.) önerilen mimari sayesinde birbirleriyle entegre bir şekilde çalıştırılarak, yerel yönetimler için gerekli CBS ihtiyaçları karşılanacaktır. Bu şekilde açık kaynak kodlu ve her biri kendi konusunda uzman yazılımlardan genel bir coğrafi bilgi sistemleri motoru oluşturulacaktır. Bu CBS motoru ile yerel yönetimler birçok teknolojiye habersiz bir şekilde kendi CBS modellerini oluşturabileceklerdir. Veri modellerini yasa, yönetmelik ve protokollerle belirlenmiş çerçevede bakanlık veya daha üst kurumların istediği şekilde oluşturup, diğer kurumların kullanımı için servis edebileceklerdir. Uygulama ve veri güvenliğine, evrensel standartlara uygun bir şekilde daha en baştan sahip olabileceklerdir. Diğer kurumların kendileri için sunduğu servisleri; uygulamaya yüklemeleri halinde görüntüleyebileceklerdir. Dışardan alınan bu servisler ile kendi verilerini bir arada görüp, görsel analizler yapabileceklerdir. Çalışmada evrensel standartlar ile geliştirileceği için daha önceden alınmış ve ellerinde olan uygulamalarda bu motor ile entegre çalışabileceklerdir. Ayrıca coğrafi süreçler bu CBS motoru ile görselleştirilerek, karar destek mekanizması oluşturulacaktır. Bu şekilde birden fazla lisans parası ödenerek oluşturulabilecek bir model, açık kaynak kodlu ve bedava ürünlerle bütünleşik olarak sunulacaktır. Farklı görev ve özelliklerin yerine getirilebilmesi için birçok farklı ekran dizayn edilmiştir. Bu ekranların kullanım tarzı birbirine benzer şekilde tasarlanmıştır. Hata ve bilgi mesajları her sayfa için aynı tasarımda oluşturulmuştur. Kullanıcının gelecekte birçok yeni özelliğe ihtiyaç duyabileceği hesap edilip, uygulamada gelecekteki geliştirmelere eklemeye uygun olacak şekilde geliştirilmiştir. Gerçekleştirilen katmanlı mimari CBS önerisi ile esnek, kolay kullanılabilir ve güvenli bir CBS uygulaması hedeflenmiştir.

Tablo 3.1. Lisanslı uygulamalar ve önerilen model kıyaslaması

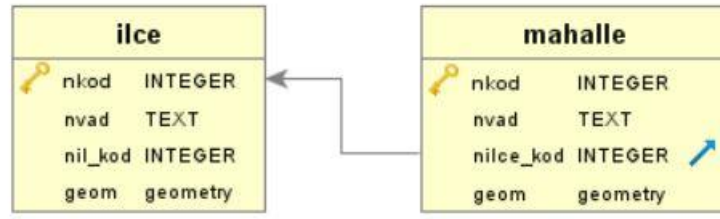
	Lisanslı Uygulamalar	Bizim Çalışmamız
Lisanslama	Ücretlidir	Açık kaynak kodludur
Meta Veri	Ekstra ücretli uygulamalar ile yapılabilmektedir.	Otomatik olarak verinin künye bilgilerini arka planda saklamaktadır.
Tematik Harita	Ekstra ücretli uygulamalar ile yapılabilmektedir.	Tüm katman özneliklerine göre yapılabilmektedir.
İnsan Kaynağı	Birçok meslek dalından uzman personele ihtiyaç duyulmaktadır.	Uzman personel sayısını minimuma indirmektedir.
Faaliyet ve Proje	Ekstra ücretli uygulamalar ile yapılabilmektedir.	Uygulama bu özelliklere sahiptir.
Verilerin formatlanıp güvenlik bir şekilde servis edilmesi	Uzman personellerce yapılabilmektedir.	Uzman olmayan personeller hızlıca yapılabilmektedir.
Mobil Uygulama	Ekstra ücretli uygulamalar ile yapılabilmektedir.	Mobil uygulamada web uygulamasının tüm yeteneklerine sahiptir.
Katman Stillendirme	Genelde uygulamaya özgü veya evrensel çözümler mevcuttur.	Sadece evrensel ve diğer programlar tarafından okunabilecek formattadır.

3.2. Kullanılan Materyaller

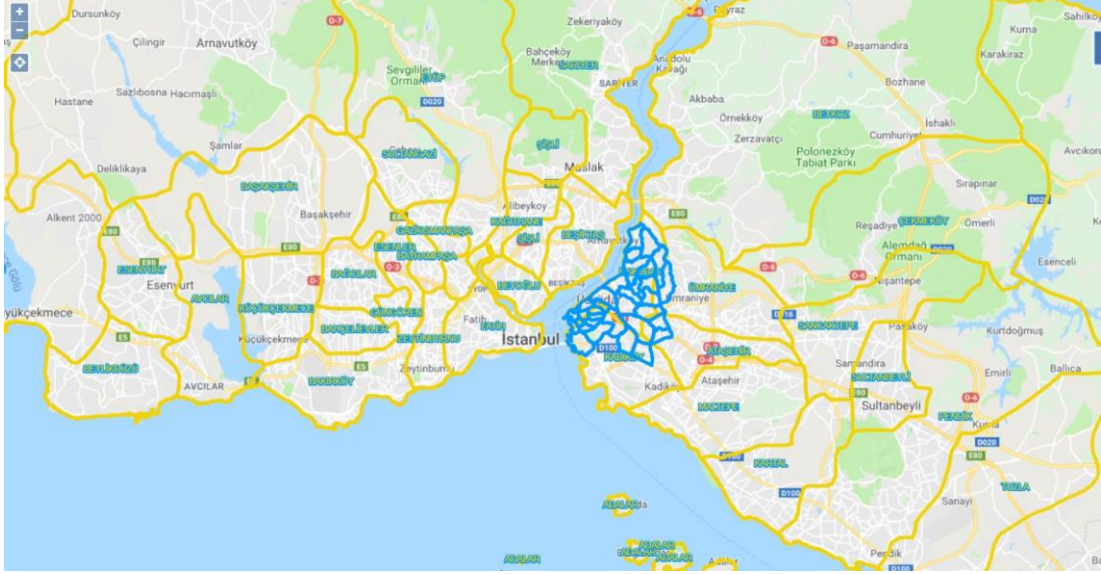
CBS çalışmamızın hem test edilebilmesi hem de yukarıda sayılan özelliklerin gösterilebilmesi için ön tanımlı veya çalışma içinden dinamik olarak tanımlanmış iki farklı tipte veri setleri kullanılmıştır.

3.2.1. Ön tanımlı veri setleri

Çalışmada, İstanbul ili baz alınmıştır. İstanbul ili ilçe haritası ve Üsküdar ilçesi mahalle haritası kullanılmıştır. İl ve ilçe kodları ulusal adres veri tabanında kullanılan kodlardır. Bu şekilde uygulamanın ulusal adres veri tabanı ve Maks verileri ile uyumlu olması hedeflenmiştir. Uygulamaya servis yolu ile bağlanabilecek ve ulusal adres veri tabanına uygun diğer sistemler ile birlikte çalışılabilecektir.



Şekil 3.1. İlçe/mahalle varlık ilişki diyagramı.



Şekil 3.2. İlçe/mahalle örnek haritalar veriler.

3.2.2. Çalışma içinde oluşturulan veri setleri

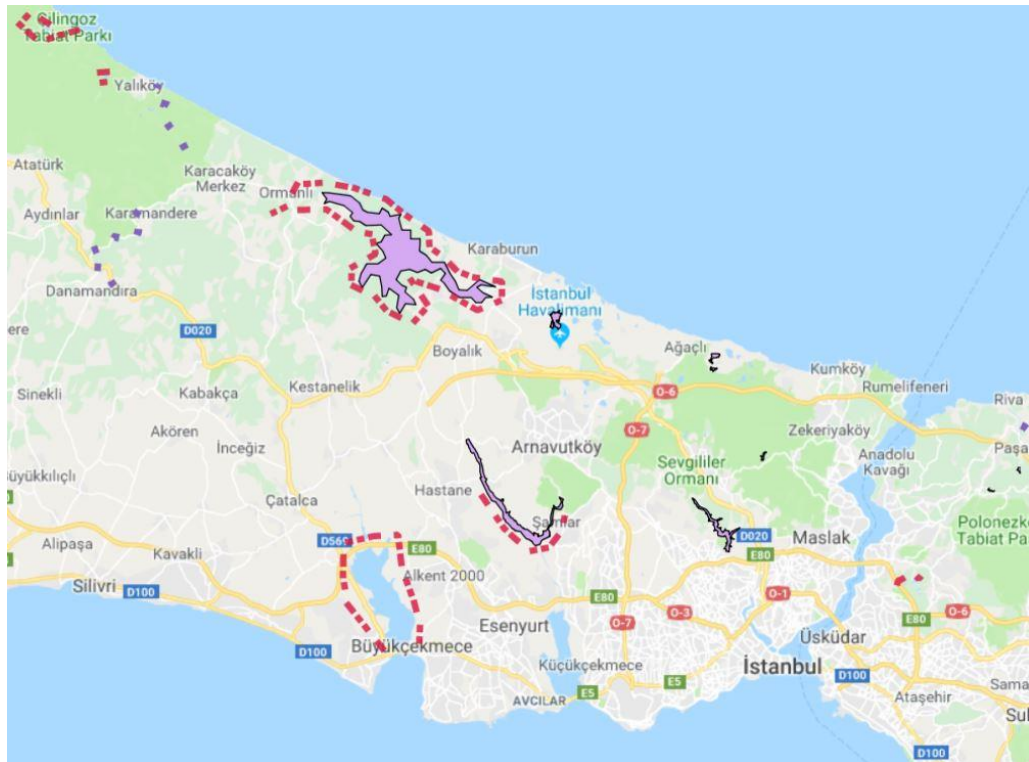
Çalışmada İstanbul ili sınırları içinde bazı coğrafi veriler toplanmış ve geliştirilen katmanlı CBS motoru ile veri modelleri oluşturulmuş ve verileri yüklenmiştir. Veri modelinde bulunan ekleme tarihi, son güncelleme tarihi, pasif, onay, onaylama tarihi, kullanıcı id, son güncelleyen kullanıcı id, kullanıcı konumu gibi alanlar uygulama ekranından kullanıcı tarafından seçildiği takdirde otomatik olarak uygulama tarafından oluşturulmaktadır. Toplanan verilerin bazılarının orjinal halleri temin edilemediği için harita üzerinden üretilmiştir. Uygulamanın sahip olduğu özelliklerin daha kolay kullanılabilmesi için bu veri modeli üretilmiş ve verileri toplanmıştır. Kullanıcılar, isterlerse farklı veri modelleri oluşturup, verilerini yükleyebilmektedirler. Bu CBS katmanları aşağıdaki şekilde gösterilmiştir. Veri modeli; nokta, kapalı alan, çizgi geometri tiplerinde geometrik alanlara sahiptir.

<p>kiralik_bisiklet_noktaları</p> <p>123 nkiralik_bisiklet_noktaları_id</p> <p>geom</p> <p>ABC nvad</p> <p>ABC nvadres</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p>	<p>cop_kutulari</p> <p>123 ncop_kutulari_id</p> <p>geom</p> <p>ABC nvkodu</p> <p>123 p25_cop_kutusu_turu</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p>	<p>kent_mobilyasi</p> <p>123 nkent_mobilyasi_id</p> <p>geom</p> <p>ABC nvadi</p> <p>123 p20_kent_mobilyasi_tipi</p> <p>dkullanima_alinis_tarihi</p> <p>btarihi_ozelligi_var_mi</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p> <p>ddeneme_saat</p> <p>obonay</p> <p>odonay_tarih</p> <p>123 ononaylayan_kullanici_id</p>	<p>otobus_duragi</p> <p>123 notobus_duragi_id</p> <p>geom</p> <p>ABC nvkodu</p> <p>ABC nvadi</p> <p>123 p22_otobus_duragi_turu</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p> <p>obonay</p> <p>odonay_tarih</p> <p>123 ononaylayan_kullanici_id</p>
<p>park</p> <p>123 npark_id</p> <p>geom</p> <p>123 nbuyukluk_m2</p> <p>boyunu_grubu_var_mi</p> <p>123 nagac_sayi</p> <p>123 p21_park_tipi</p> <p>ABC nvad</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p>	<p>bisiklet_yollari</p> <p>123 nbisiklet_yollari_id</p> <p>geom</p> <p>ABC nvad</p> <p>bkoruma_duvari</p> <p>obonay</p> <p>odonay_tarih</p> <p>123 ononaylayan_kullanici_id</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p> <p>123 p13_proje_asamasi</p>	<p>taskin_seddi</p> <p>123 ntaskin_seddi_id</p> <p>geom</p> <p>ABC nvadi</p> <p>123 xnkullanici_id</p> <p>123 xnson_guncelleyen_kullanici_id</p> <p>xdeklenme_tarih</p> <p>xdson_guncelleme_tarih</p> <p>xbpasif</p> <p>ABC xkullanici_konumu_wkt</p>	<p>sulama_goleti</p> <p>123 nsulama_goleti_id</p> <p>geom</p> <p>ABC nvadi</p> <p>123 nhacim</p> <p>dyapim_yili</p>
			<p>kurutma_kanali</p> <p>123 nkurutma_kanali_id</p> <p>geom</p> <p>ABC nvadi</p> <p>123 nuzunluk</p>

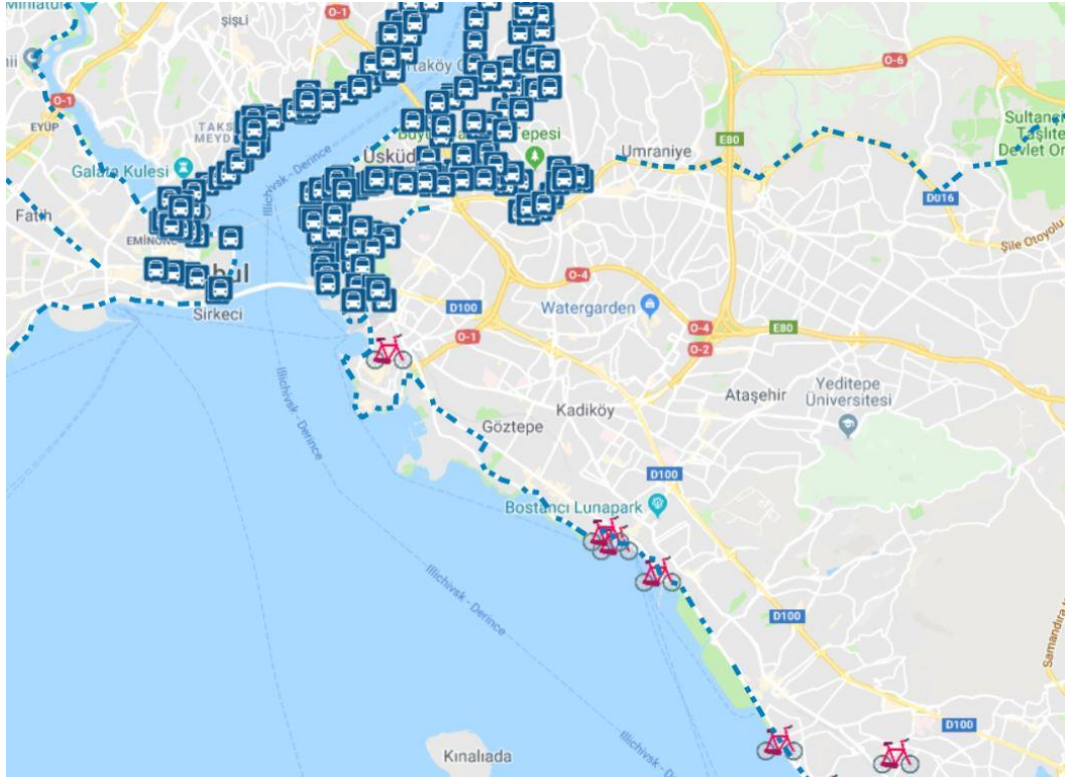
Şekil 3.3. Çalışma içinde oluşturulan örnek veri setlerinin veri modeli.

- Kiralık bisiklet noktaları: İstanbul ilindeki bazı bisiklet kiralanabilecek yerlerin noktaları. Noktasal konum, ad, adres alanlarını içermektedir. Bu katmanın verileri harita üzerinden toplanmıştır.
- Parklar: İstanbul ilindeki bazı parkların yerlerini içermektedir. Noktasal konum, metre kare cinsinden büyüklük, ad, ağaç sayısı, oyun grubu var mı alanlarını içermektedir.
- Çöp Kutuları: Metal, plastik mavi ve plastik yeşil tipindeki çöp kutularının konumsal verilerini içermektedir. Çöp kutusu tiplerine göre tematik harita stili de oluşturulmuştur.
- Bisiklet yolları: Bitmiş, planlanan veya yapım aşamada olan bisiklet yollarının çizgi türünde çizimlerini içermektedir. Koruma duvarı var mı bilgisinide tutmaktadır. Çizgi veri tipindedir. bu katmanın verileri harita üzerinden kuşbakışı bir şekilde çizilerek oluşturulmuştur.

- Kent mobilyası: Ahşap saksı, bank, heykel, çeşme, saat kulesi tipindeki kent mobilyalarının noktasal konumlarını içermektedir. Adı, kullanıma alım tarihi gibi bilgileri de tutulmaktadır.
- Taşkın seddi: Nehirlerin, derelerin kenarlarında taşkını önleyen yapıların çizimlerini ve adlarını içermektedir.
- Otobüs durağı: Açık durak, akıllı durak veya kapalı, korumalı durak tipindeki otobüs duraklarını noktasal konum olarak içermektedir.
- Sulama göleti: Ad, hacim ve yapım yılı bilgilerini kapalı alan olarak içermektedir. Harita üzerinden gölet sınırları göz önünde bulundurularak çizilmiştir.
- Kurutma Kanalı: Ad, uzunluk bilgileri ile su taşkınlarını engelleyecek kanalların çizimleri bulunmaktadır.



Şekil 3.4. Kurutma kanalı, sulama göleti, taşkın seddi örnek verisi.



Şekil 3.5. Otobüs durağı, bisiklet yolu, bisiklet kiralama noktası örnek verisi.

3.3. Yöntem

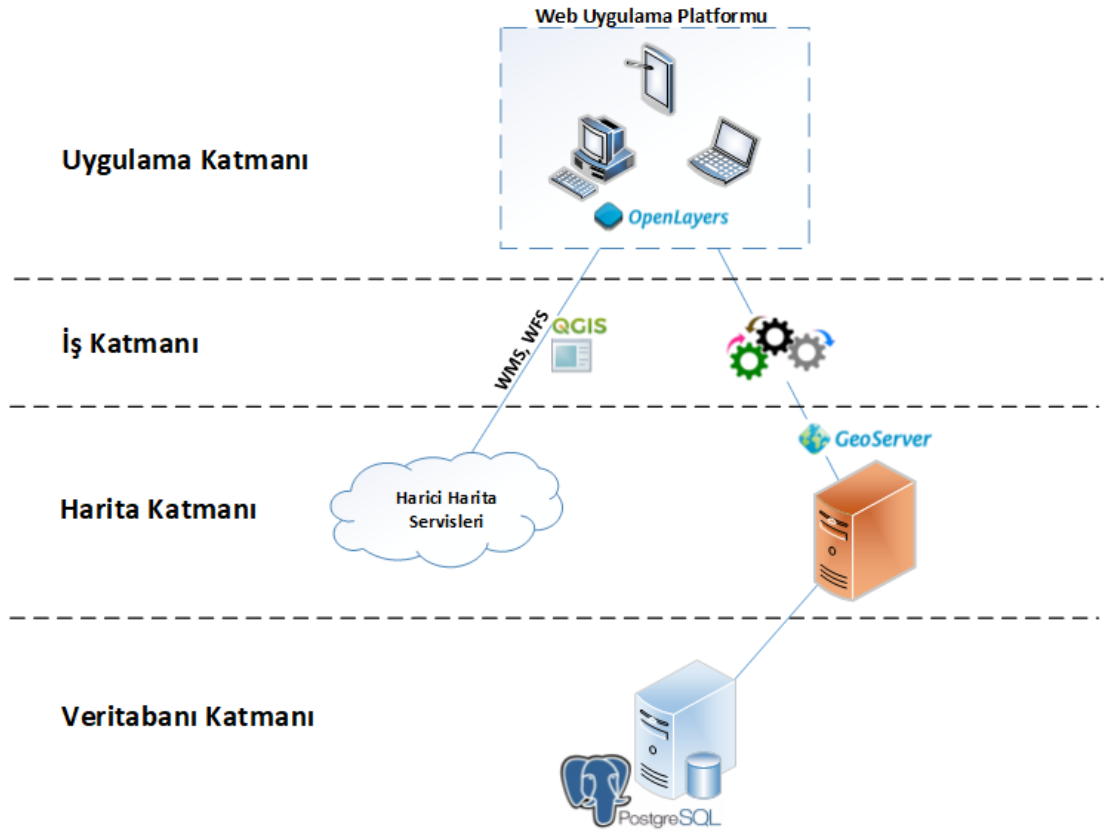
Geliştirilen CBS aracı web ve mobil olmak üzere iki uygulamadan oluşmaktadır. Web uygulaması tablet, telefon ve kişisel bilgisayarlara uyumlu şekilde geliştirilmiştir. Uygulama veri tabanı katmanı, coğrafi harita katmanı, servis katmanı, web ve mobil uygulama katmanlarını içeren çok katmanlı yazılım mimarisi kullanılarak geliştirilmiştir. Kullanıcılar web ve mobil uygulamalardan istek yaparak harita servisleri vasıtası ile veri tabanındaki coğrafi verileri görüntüleyebilirler.

CBS projeleri oluşturulurken mevcut lisanslı uygulamalarda kullanılan yöntemlerden biri öncelikle masaüstü harita uygulamaları veya veri tabanı yönetim programları ile coğrafi veri modelinin oluşturulmasıdır. Ardından oluşturulan coğrafi veri tabanı katmanı, bir harita servis sunucusu ile OGC formatında servis olarak kullanıma açılmaktadır. Bu servisleri kullanabilmek için gerekli yetkilendirme ve güvenlik protokolleri tanımlandıktan sonra geliştirilen web/mobil uygulamalar vasıtası ile son kullanıcının kullanımına açılmaktadır. Bu çalışmada ise son kullanıcının veri tabanı ve

harita servisi ile ilgili işlemlerden soyutlanarak, projesine ve coğrafi modeline odaklanması hedeflenmektedir. Geliştirilen Web/Mobil uygulama ile kullanıcı basit bir ara yüzden ihtiyaç duyduğu coğrafi veri modelini tanımlayabilmektedir. Veri tabanı, harita servisi işlemleri otomatik olarak uygulama tarafından gerçekleştirilmektedir.

Kullanıcı elindeki hazır “.json”, “.kml” veya “.xls” formatındaki verilerini uygulama yardımı ile veri tabanına yükleyebilmektedir. Başka kullanıcılarla okuma/yazma yetkilerini ayarlayarak harita katmanını paylaşabilmektedir. Uygulama üzerinden yetki dahilinde yeni kayıt ekleme, güncelleme, silme veya güncelliği geçmiş ama silinmesini istemediği verileri pasif ederek saklayabilmektedir. SLD formatında katman stillerini ayarlayabilmektedir. Coğrafi veri modelinin içerdiği verilere göre dinamik kurallar tanımlayıp, tematik haritalar tanımlayabilmektedir. Tematik haritalar ile analiz konusunda yeni bakış açıları kazandırılmıştır. Kullanıcılar yetkisi dahilindeki katmanlardan bazılarını açıp, istediği zoom seviyesinde ayarlamalarında yaptıktan sonra, bunu bir çalışma olarak kaydedip daha sonra sunum veya başka bir amaçla tekrar kullanabilmektedir. Sistem yöneticisi isterse sahadan toplanan verilerin gerçekte hangi konumdan, hangi tarihte kim tarafından toplandığını görebilmektedir. Yerel yönetimler faaliyet ve projelerini tanımlayabilmektedir. Faaliyetlerin ölçü birimleri tanımlanabilmektedir. Dinamik oluşturulan katmanlar için de gerçekleştirilen faaliyetler tanımlanabilmektedir. Örneğin durak katmanı içindeki her bir durak için bakım ve onarım faaliyeti yapılabilmektedir. Yine bu faaliyeti hangi personellerin gerçekleştirdiği tanımlanabilmektedir. Yerel yönetimler CBS modelini oluşturduktan sonra istediği kurum ile verilerini paylaşabilmektedir. Bunu için servis isteyen kurum için özelleşmiş katmanlarını ayarlar ardından yine aynı kurum için kullanıcı adı ve şifresini oluşturduktan sonra otomatik oluşturulan web servisleri ile verilerini başka kurumlar ile paylaşabilmektedir.

Yerel yönetimler, lisanlı ürünlerde olduğu gibi birçok CBS probleminden ve uzmanlığından soyutlanıp, CBS projelerini yapmaya odaklanabileceklerdir. Bu çalışmada yerel yönetimler için en uygun açık kaynak kodlu teknolojilerin seçilmesi hedeflenmiştir. Bu çalışmada seçilen açık kaynak kodlu ürünlerin bir arada çalışabileceği katmanların birlikte çalışma yapıları Şekil 3.6.'da gösterilmiştir.



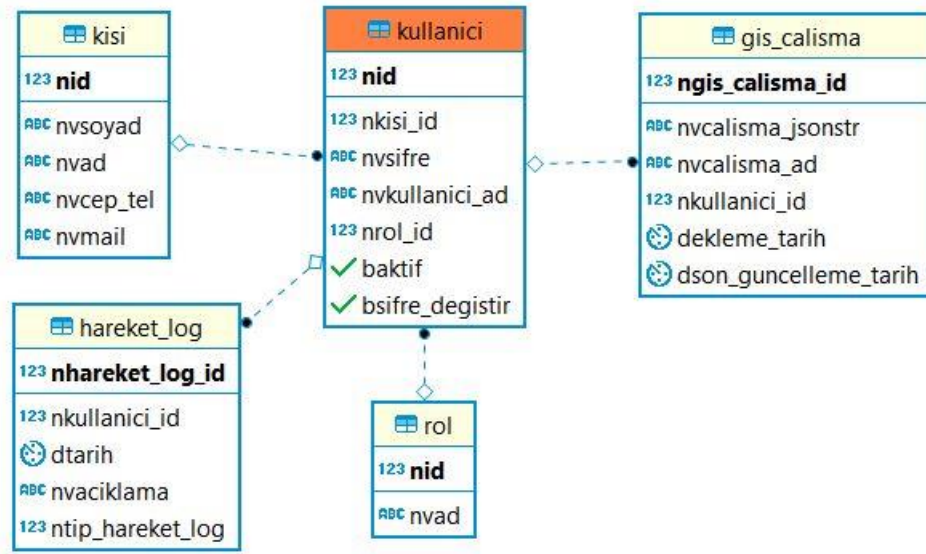
Şekil 3.6. CBS için Açık kaynak ve katmanlı mimari yapısı.

3.3.1. Veri tabanı katmanı

İlişkisel veri tabanı prensipleri ile PostgreSQL veri tabanı üzerinde gerçekleştirilmiştir. PostgreSQL veri tabanı üzerinde PostGis eklentisi kullanılarak CBS özellikleri kazandırılmıştır. Veri tabanı katmanı iki farklı şemadan oluşmaktadır. Bunlar;

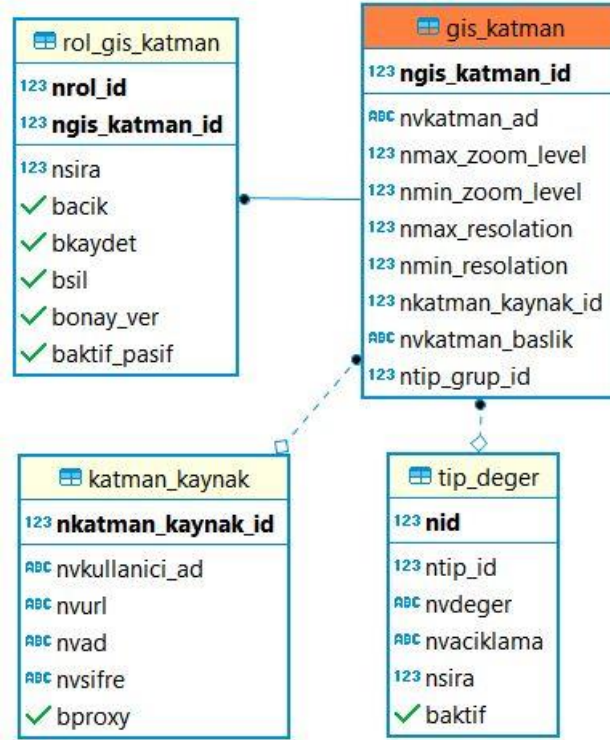
- “GisAna” şeması: Kullanıcı, kisi, rol, yetki, parametre, organizasyon şeması, katman kaynağı, log, link, tematik, proje, faaliyet gibi ön tanımlı, sistem tablolarının bulunduğu şemadır.
- “GisKatman” şeması: Tüm kullanıcı tanımlı katmanların tabloları bu şemada bulunmaktadır. Ayrıca bu şemadaki tablolar üzerinden başka kurumlara özel açılan web servislerinin veri ihtiyacını karşılayan görüntüler burada tutulmaktadır. Yetkili kullanıcı uygulama ara yüzünden buradaki tablo, tablo alanlarını ve veri tiplerini oluşturabilmektedir.

Kişi tablosunda uygulamadaki yerel yönetim personellerinin kişisel bilgileri bulunmaktadır. Bu kişilerden kullanıcısı olan kişilerin kullanıcı adı, şifre, rol gibi bilgileri kullanıcı tablosunda bulunmaktadır. Kullanıcı rolleri rol tablosunda tanımlanmıştır. Rol yetki tablosunda da, o rolün hangi yetkileri gerçekleştirebileceği tanımlanmaktadır. Organizasyon şeması tablosunda hiyerarşik bir şekilde birimler tanımlanabilmektedir. Gis çalışma tablosunda kullanıcının kendi çalışmaları kaydedilmektedir.



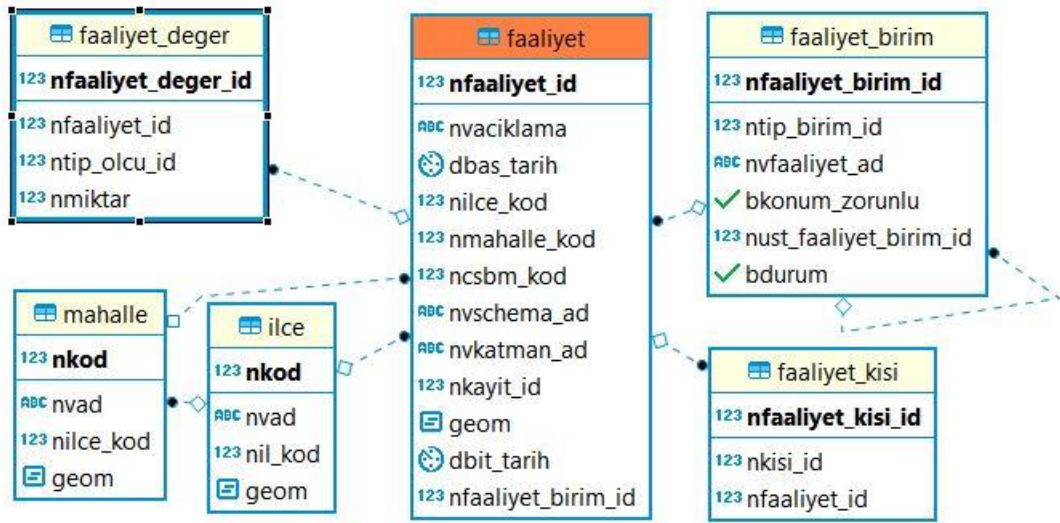
Şekil 3.7. Kullanıcı, kişi, yedek, çalışma, rol tabloları varlık ilişki diyagramı.

Katman kaynak tablosunda ön tanımlı uygulamanın kendi vektör ve harita servis bilgileri tanımlanmıştır. Bunun haricinde dışardan alınan servislerin genel servis adresleri, vekil sunucu üzerinden bağlantı kurulacaksa da servis kullanıcı adı ve şifreleri bulunmaktadır. Gis katman tablosunda ise ilgili kaynaktan kullanılan servislerin bilgileri mevcuttur. Bu tabloda katman hangi zoom seviyelerinde gözükeceği ayarlanabilmektedir. Bu tablolarda hem dışarıdan alınan servis bilgileri, hem de uygulama içinden oluşturulan katmanların bilgileri tutulmaktadır. Uygulama buradaki katmanın iç veya dış katman olduğunu otomatik olarak algılamaktadır. Katmanlar farklı kategoriler altına tanımlanabilmektedir. Rol gis katman tablosunda ise hangi rolün hangi katmanı görüntüleyebileceği tutulmaktadır. Rol için veri ekleme, silme, güncelleme, onaylama, pasif etme yetkileri yönetilebilmektedir. Tip değer tablosunda diğer tabloların ihtiyaç duyduğu parametrik veriler tutulmaktadır.



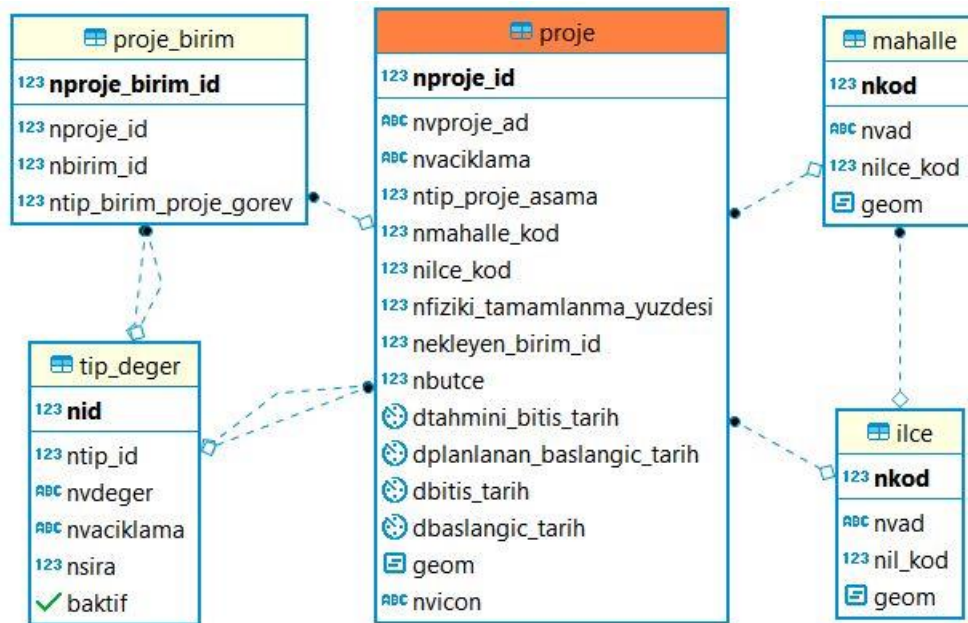
Şekil 3.8. Katman kaynak, katman, rol katman ve parametre tabloları varlık ilişki diyagramı.

Faaliyet tablosunda hangi tarihte, nerede, hangi birimin, hangi çalışmayı yaptığı tutulmaktadır. Eğer kullanıcı tanımlı katmanlar ile ilişkili bir faaliyet gerçekleştirilecekse o katmanın tablo bilgileri de burada tutulmaktadır. Örneğin bir bisiklet yolunun boyama faaliyeti bir çizgi olarak, bisiklet yolu boyama faaliyeti seçildikten sonra saklanabilmektedir. Faaliyet kişi tablosunda bu faaliyeti gerçekleştiren personeller tutulmaktadır. İlçe ve mahalle tablosundan sözel konum alınmaktadır. Faaliyet birim tablosunda; birimlerin ne tür faaliyetleri var ve konum zorunluluğu var mı bilgileri tutulmaktadır. Faaliyet değer tablosunda ise faaliyet gerçekleştirilirken kullanılan malzeme miktarının ne olduğu tutulmaktadır. Örneğin, asfalt dökümünde kaç ton sıcak asfalt kullanılmış gibi. Kullanıcılar faaliyet tanımlarken, yetkili oldukları birimi seçip o birime ait faaliyet başlıklarından birini seçmeleri gerekmektedir. Her faaliyet başlığının kendine özgü nicelik tanımları bulunmaktadır. Eğer uygulama içerisinde oluşturulan bir katman için faaliyet tanımı yapılacaksa tablodaki şema adı, katman adı ve kayıt id alanlarının doldurulması gerekmektedir. Uygulamanın kendi katmanları için faaliyetin çizimi eklenirken referans alınabilmesi için ilgili katman verisinin çizimide gösterilmektedir.



Şekil 3.9. Faaliyet tabloları varlık ilişki diyagramı.

Proje tablosunda ad, açıklama, fiziki tamamlanma yüzdesi, başlama/bitiş tarihi gibi bilgiler tutulmaktadır. Projeler bitmiş, planlanan veya devam etmekte olan tiplere sahiptir. Proje birim tablosunda projenin hangi birime ait olduğu tutulmaktadır. Projelerin sözel olarak hangi ilçe ve mahallelere ait olduğu tutulmaktadır. Proje birim tablosunda ise projedeki görevlerine göre yerel yönetimlerdeki proje birimleri tutulmaktadır.



Şekil 3.10. Proje tabloları varlık ilişki diyagramı.

3.3.2. Harita katmanı

Bu katmanda dışarıdan başka kurumlar tarafından sağlanan OGC formatındaki web servisleri ile uygulama tarafından otomatik olarak açılan web servisleri bulunmaktadır. Dışarıdan alınan servisler için eğer ulaşıma her yerden açık bir servis ise servis yolunu girmek yeterli gelmektedir. Şifreli bir servis ise uygulamaya kullanıcı adı ve şifreyi kaydederek, vekil sunucu üzerinden ulaşılabilmektedir.

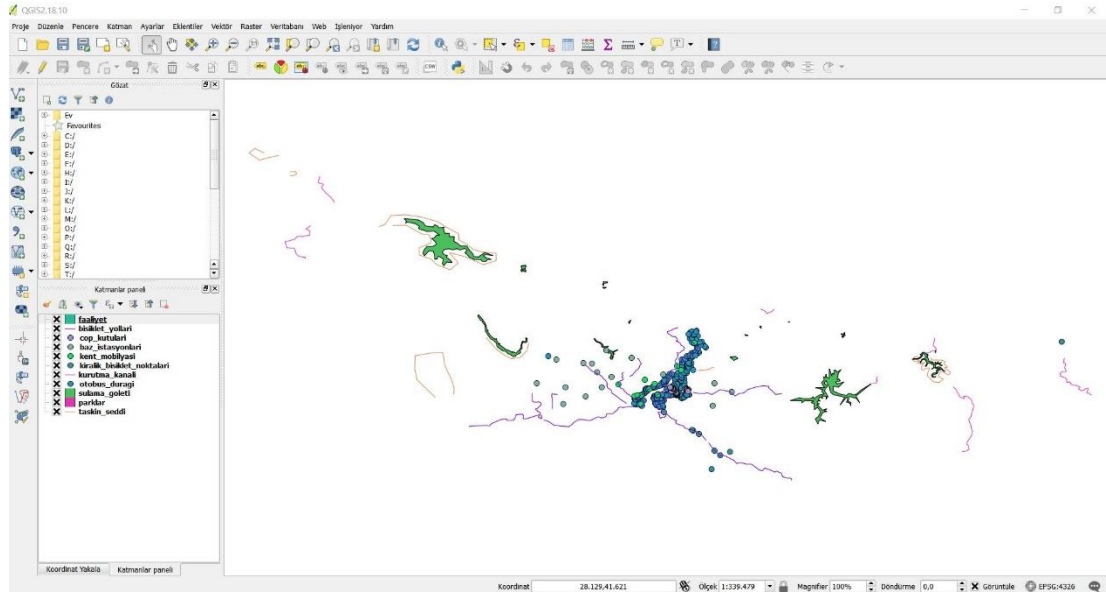
Uygulama katmanındaki servisler, son kullanıcının haberi olmadan bir katman oluşturduğu anda otomatik olarak oluşturulmaktadır. Arka plandaki servisler Geoserver uygulaması ile oluşturulmaktadır. Bu servislerin güvenliği yine Geoserver uygulaması tarafından otomatik olarak yapılmaktadır. Hazırladığımız uygulamadan dışarıya bir servis vermek için oluşturduğumuz kullanıcı adı, şifre, rol ve yetkiler yine Geoserver uygulamasında tutulmaktadır. Kullanıcı bu işlemlerde de Geoserver uygulamasından soyutlanmıştır. Yetkin personel isterse herhangi bir işlemi Geoserver uygulamasının web ara yüzünü kullanarak ta yapabilmektedir. Geoserverda bu uygulamaya özel öntanımlı iki adet depo bulunmaktadır. Bunlardan biri kullanıcı tarafından oluşturulan veri modeli için kullanılmaktadır. Diğeri ise paylaşım için oluşturulan görüntüler için kullanılmaktadır.

The screenshot shows the GeoServer web interface. The main content area is titled 'Katman Önizleme' (Layer Preview). Below the title, there is a search bar and a table listing layers. The table has five columns: Tip, Başlık, İsim, Yaygın Formatlar, and Tüm Formatlar. The layers listed are:

Tip	Başlık	İsim	Yaygın Formatlar	Tüm Formatlar
	nokta	GIS_SHARE:nokta	OpenLayers KML GML	Seçiniz
	v_bisiklet_yollari	GIS_SHARE:v_bisiklet_yollari	OpenLayers KML GML	Seçiniz
	Baz İstasyonları	GIS:baz_istasyonlari	OpenLayers KML GML	Seçiniz
	Bisiklet Yolları	GIS:bisiklet_yollari	OpenLayers KML GML	Seçiniz
	Çöp kutuları	GIS:cop_kutulari	OpenLayers KML GML	Seçiniz
	deneme nokta	GIS:deneme_nokta	OpenLayers KML GML	Seçiniz
	E Sarj Noktalar	GIS:e_sarj_noktalar	OpenLayers KML GML	Seçiniz
	faaliyet	GIS:faaliyet	OpenLayers KML GML	Seçiniz
	v_faaliyet_cluster	GIS:faaliyet_cluster	OpenLayers KML GML	Seçiniz
	Hali Saha	GIS:hali_saha	OpenLayers KML GML	Seçiniz
	Havuzlar	GIS:havuzlar	OpenLayers KML GML	Seçiniz
	ilçe	GIS:ilce	OpenLayers KML GML	Seçiniz
	Kent mobilyası	GIS:kent_mobilyasi	OpenLayers KML GML	Seçiniz

Şekil 3.11. Geoserver kendi katmanlarımızın önizlemesi.

Uygulama tarafından oluşturulan tüm bu katmanlara QGis, Udig, ArcMap gibi OGC formatını destekleyen programlar ile de bağlanılıp, birçok işlem kolayca yapılabilmektedir. SLD formatındaki stiller açılabilir. Hassas CBS çizimleri için bu programlara ihtiyaç duyulmaktadır.



Şekil 3.12. QGis programı ile kendi katman servislerimizin görüntülenmesi.

Geoserver uygulamasının web arayüzüne ihtiyaç duyulmadan kendi uygulamamızın arka planında yapılan tüm işlemler Geoserver uygulamasının yönetici restful web servisleri üzerinden yapılmaktadır. Tüm bu servisleri kullanabilmek için web servislerini okuyan kütüphane geliştirilmiştir.

3.3.3. İş katmanı

Uygulamanın iş kuralları ve harita motoru altyapısını içermektedir. Uygulama katmanı ve harita servisleri katmanının birbiri ile uyum içerisinde çalışmasından sorumludur. Uygulama katmanında oluşturulan istekleri karşılayarak, iş kurallarını ve süreçlerini yönetmektedir. Haritasal olmayan istekler içinse ORM kütüphanesinden faydalanarak, uygulama katmanının isteklerini cevaplamaktadır. Bu katman, diğer katmanlar üzerinde bir yönetici rolü üstlendiği için; birçok nesneye dayalı tasarım prensibinden faydalanılmıştır. Karmaşık mimarisel problemler tasarım desenleri kullanılarak

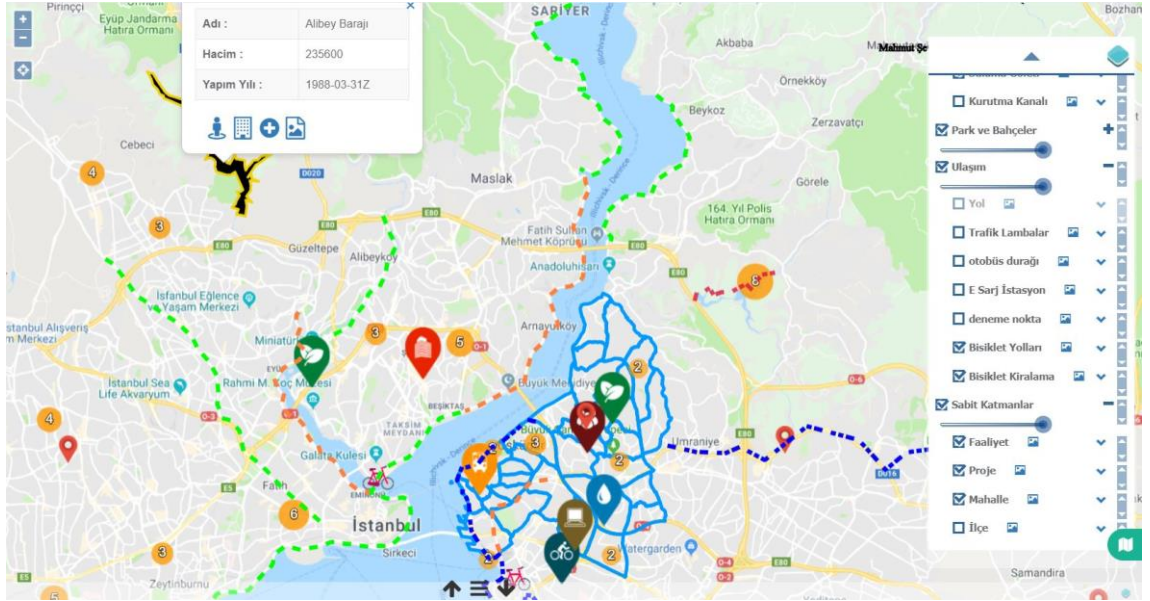
basitleştirilmek istenmiştir. Çalışmada birçok katmanda kullanılan oturum kontrolü, yetki kontrolü, hata kaydetme ve bellek kullanımı gibi ihtiyaçlar yine bu katmanda AOP 'ten faydalanılarak tanımlanmış olup, ihtiyaç duyulan her yerde AOP'ye uygun şekilde kullanılmıştır. Dependency injection tasarım deseninden faydalanılarak yazılan kodlar ve modüller arasındaki bağımlılıklar yönetilmiştir. Gelecekte uygulamada ihtiyaç duyulacak yeni süreç ve iş kuralları eklenirken diğer katmanların minimum seviyede etkilenmesi amaçlanmıştır. Bu katmanda çok fazla süreç ve iş kuralı yönetildiği için SOLID prensiplerin faydalanılmıştır. Bu prensipler ile uygulamanın daha okunabilir ve evrensel bir mimariye sahip olması hedeflenmiştir. Değişime daha toleranslı bir uygulama ve yeniden kullanılabilirlik amaçlanmıştır. Repository Tasarım Deseninden faydalanılarak kod tekrarlarından kaçınılmaya çalışılmıştır. Bu tasarım deseni ile veri tabanı işlemleri belli bir formatta ve kuralla yönetilmiştir. Standart ekleme, silme, güncelleme, listeleme ve sayfali listeleme işlemleri için herhangi bir kod yazılmadan Repository tasarım deseni yardımı ile dinamik olarak sorgu üretilebilmektedir. Bu şekilde iş katmanı geliştirilirken veri tabanı işlemleri için daha az kod yazılmıştır. Kodlarda daha sade ve okunabilir şekilde geliştirilebilmiştir.

3.3.4. Uygulama katmanı

Web/Mobil uygulama katmanı “MODEL-VIEW-CONTROLLER” Tasarım deseni kullanılarak geliştirilmiştir. Uygulama bu tasarım desenine göre parçalara ayrılarak geliştirilmiştir. Her bir parça mümkün olduğunca birbirine en az bağımlı şekilde geliştirilmiştir. Uygulama içerisinde ihtiyaç duyulan parametrik değerler mümkün olduğunca kod içine gömülmemiş, konfigürasyon dosyalarında tutulmuştur. Bu parametreler program çalıştığında sunucu belleğine yüklenerek ihtiyaç duyulan her yerde yetki dahilinde kullanılmıştır. Kullanıcılar tüm işlemlerini bu uygulamalar vasıtası ile yapabilmektedirler.

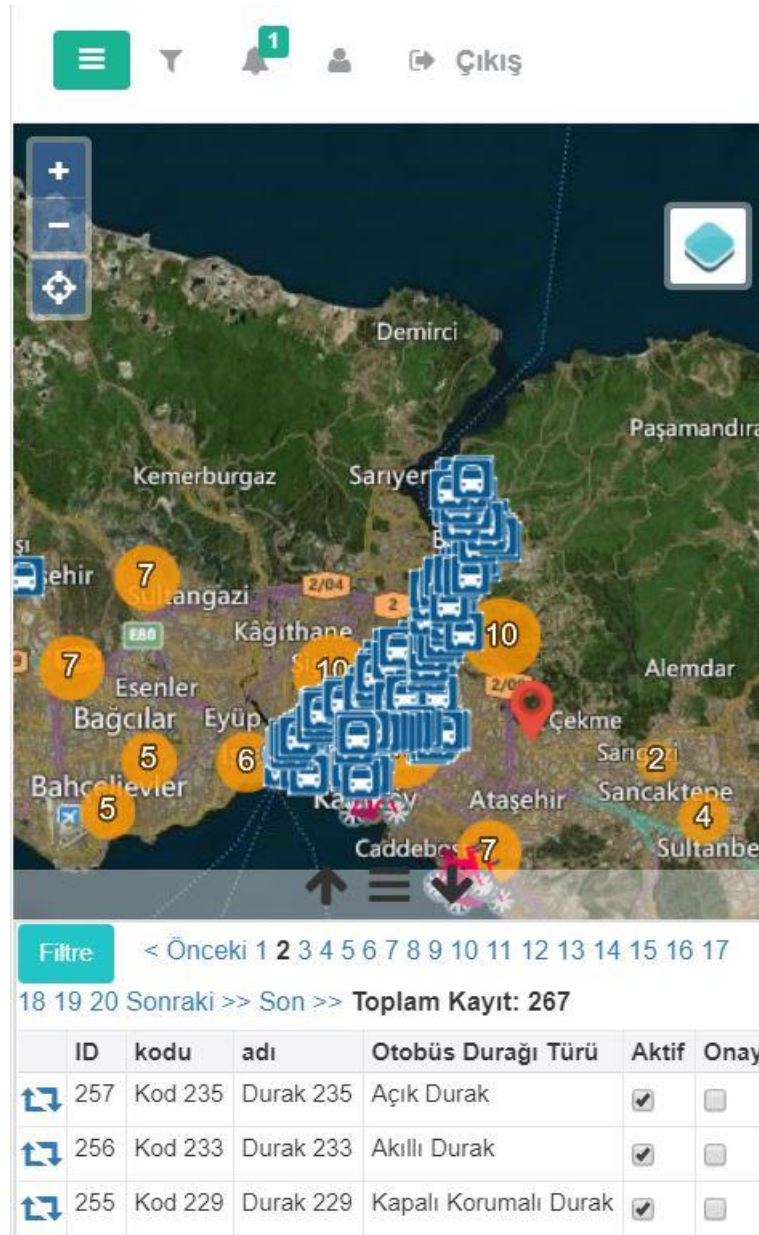
Web uygulaması Asp.net Core teknolojisi kullanılarak geliştirilmiştir. Bootstrap kütüphanesi kullanılarak uygulamanın tüm güncel tarayıcı ve cihazlarda da uyumlu çalışması sağlanmıştır. Tüm harita işlemleri Openlayers javascript kütüphanesi kullanılarak yapılmıştır. Uygulama mekansal olarak İstanbul ili, Üsküdar ilçesi temel

alınarak geliştirilmiştir. Kullanıcı dinamik olarak tanımladığı tüm katman özneliklerine göre filtreleme yapabilmektedir. Yerel yönetimdeki birimlerin gerçekleştirdiği faaliyetleri ve projeleri tanımlayıp, harita üzerinde gösterebilmektedir. Katman verileri için varsa sokak görünümünde ulaşabilmektedir. Uygulama katmanı herhangi bir işletim sisteminde çalışabilmektedir.



Şekil 3.13. Katmanlı yazılım mimari önerisinin genel uygulama ekranı.

Mobil uygulama Android Kütüphanesi kullanılarak Eclipse üzerinde Webview kullanılarak geliştirilmiştir. Web uygulamasının tüm özelliklerini içermektedir. Gösterge paneli ile yapılan çizimlerin, faaliyetlerin ve projelerin özet ve grafiksel raporlarına ulaşabilmektedir. Kurum organizasyon şeması temelli filtreleme imkanı sunulmuştur. Cihaz konumunu kapalı ise açmak için, açık ise konuma ulaşmak için Android kütüphanesinin özellikleri kullanılmıştır. Android kütüphanesi ile geliştirilen uygulama herhangi bir markette bulunmamaktadır. Özel olarak dağıtmakta ve Web ara yüzünden kullanılan kullanıcı adı ve şifre ile kullanılabilir. Web uygulaması mobil cihazlarda uyumlu bir şekilde geliştirildiği için mobil uygulama tasarım olarak sorunsuz bir şekilde çalışabilmektedir. Mobil uygulamanın performansı, kullanıldığı mobil cihazın özelliklerine göre değişiklik gösterebilmekle beraber web uygulamasının performans özelliklerine benzer şekilde sonuçlar üretmektedir. Mobil uygulama sadece android cihazlar için geliştirilmiştir.



Şekil 3.14. Mobil uygulama ekranı.

Katmanlı yazılım mimari önerisi geliştirilirken, kullanıcının farklı işlemleri gerçekleştirebilmesi için birçok ekran geliştirilmiştir. Tüm bu ekranlar benzer kullanım şekillerine, hata ve uyarı mekanizmalarına sahiptir. Tüm ekranlar mobil uyumlu bir şekilde geliştirilmiştir. Tablet, telefon ve farklı büyüklüklerdeki bilgisayar ekranlarında en uygun görüntü ilkesi doğrultusunda geliştirilmiştir. Uygulama menüsü iki ana parçadan oluşmaktadır. Birinci parçada uygulamanın ana ekranları, ikinci parçada ise

yardımcı ekranlar bulunmaktadır. Kullanıcının işlemleri ve uygulama girişleri veri tabanında saklanmaktadır. Bu ekranlar;

Tablo 3.2. Lisanslı uygulamalar ve önerilen model kıyaslaması

Uygulama Ekranları
Uygulama ana ekranı
Katman tanımlama ekranı
Veri tabanı katmanları ekranı
Projeler ekranı
Faaliyetler Ekranı
Gösterge Paneli
Tematik harita tanımlama ekranı
Kullanıcı-kişi tanımlama ekranı
Kullanıcı profil ekranı
Parametre tanımlama ekranı
Organizasyon şeması ekranı
Rol tanımlama ekranı
Rol yetkilendirme ekranı
Veri paylaşım ekranı
Log izleme ekranı
Birim faaliyetleri yönetim ekranı

Uygulamanın en temel ekranı, veri tabanı katmanları ekranıdır. Bu ekran ile kurum kendi CBS veri modelini oluşturabilmektedir. Öncelikle yeni kayıt butonuna bastıktan sonra açılan pencerede katmanın adı yazılır, projeksiyon sistemi seçilir, geometri tipi seçilir ve öznitelik alanları da doldurulduktan sonra kaydet butonuna basılarak katman hem veri tabanında hem de harita servisinde otomatik olarak oluşturulabilmektedir. Projeksiyon sistemleri PostgreSQL veri tabanından gelmektedir ve güncel bir çok projeksiyon sistemi burada bulunmaktadır. Geometri tipi; çizgi, kapalı alan, çoklu çizgi, çoklu kapalı alan veya nokta olarak seçilebilmektedir. Öznitelikler oluşturulurken metin, sayı, virgüllü sayı, tarih gibi birçok veri tipi seçilebilmektedir. Parametre ekranı kullanılarak önceden tanımlanmış alanlarda tek seçmeli bir yapıda öznitelik olarak kaydedilebilmektedir. Öznitelikler zorunlu alan olarak işaretlenebilmektedir. Eğer katman için versiyonlama yapılsın seçeneği seçilirse, arka plan da o katman için toplanan her veri için; kim tarafından, ne zaman ve nerede oluşturulduğu otomatik

olarak tutulmaktadır. Katman verilerine pasif yapma imkanı getirilebilmektedir. Eğer katman için onay mekanizması seçeneği seçilirse; sahadan toplanan her verinin kabul edilmiş veri olması için toplandıktan sonra sistem yöneticisi tarafından onaylanması gerekmektedir.

Yeni Katman

Katman Adı: Otobüs Durakları

Projeksiyon: 4326

EPSG:4326 WGS 84

Geometri Tipi: Nokta Çizgi Çoklu Çizgi Kapalı Alan Çoklu Kapalı Alan

Öznitelikler:

Öznitelik Adı	Veri Tipi	Zorunlu Mu?	Sil
Kodu	Sayı	<input checked="" type="checkbox"/>	<input type="button" value="X"/>
Otobüs Durakları Türü	Parametre	<input checked="" type="checkbox"/>	<input type="button" value="X"/>
Adı	Yazı	<input type="checkbox"/>	<input type="button" value="X"/>

Ek Özellikler:

Versiyonlama yapılınsın mı? Onay Mekanizması?

Kaydet

Şekil 3.15. Veri tabanı katmanları yeni kayıt sekmesi.

Aynı ekrandan daha önce oluşturulmuş tüm CBS katmanlarına ulaşılabilir. Katman listesinde o katmana ait öznitelikler gözükmemektedir. Eğer istenirse bu liste üzerinden katmana yeni bir öznitelik eklenebilir. Tabloya, ilgili butona basılarak; .xlsx, .kml veya .json formatında dışarıdan veri yüklemek mümkündür. Tablonun harita servisinin oluşup, oluşmadığı yine buradan takip edilip, gerekli durumda yeniden servis oluştur denilebilir. Eğer ilk katman oluşturulurken versiyonlama yapılınsın seçeneği veya onay mekanizması seçilmemişse buradan sonradan da seçilebilir. Tablo listesinde katmanın projeksiyonu ve geometri tipi görüntülenmektedir. Liste üzerinden katmana yeni öznitelikler eklenebilir. Katman adının üzerine tıklanarak katman içindeki kayıt sayıları görüntülenebilir ve yine buradan katman silinebilir. Eğer tablonun herhangi bir verisine faaliyet tanımlandı ise silme işlemi gerçekleştirilememektedir. Tablonun buradan silinebilmesi için yine tablonun katman olarak oluşturulmamış olması ve katman olarak bir role yetkilendirilmemiş olması gerekmektedir. Tablo listesinde öznitelik alanı üzerine tıkladığında öznitelik veri tipi ve zorunlu olup olmadığı görüntülenebilir.

Son olarak bir sonraki ekran olan katman tanımla ekranına buradaki ilgili buton üzerinden geçiş yapılabilmektedir. Bu ekran, yerel yönetimlerin kendi CBS veri modellerini oluşturup, yönetebilecekleri en temel ekrandır. Öznitelik listesinde, kullanıcı seçtiği takdirde uygulama tarafından onay ve versiyonlama için otomatik olarak oluşturulan alanlar görüntülenmemektedir. Sadece kullanıcının oluşturduğu öznitelik alanları görüntülenerek karmaşıklık ortadan kaldırılmış, sadelik hedeflenmiştir.

Katman Adı	Projeksiyon	Geometri Tipi	Öznitelikler
Baz İstasyonları	4326	POINT	Başvuru Tarihi, Adres, Açıklama, Operatör Tipi, Sertifika Tarihi
Bisiklet Kiralama	4326	POINT	Ad, Adres
Bisiklet Yolları	4326	LINESTRING	ad, koruma duvarı varmı, Proje Aşaması
Çöp kutuları	4326	POINT	kodu, çöp kutusu türü
deneme nokta	4326	POINT	kodu, adı, Otobüs Durağı Türü
E Sarj Noktaları	4326	POINT	Ad, Adres, Adres Tarihi, Soket Say
Halı Saha	4326	POLYGON	Türü, Kişi Sayısı, Ölçü (m)
Havuzlar	4326	POLYGON	Max Derinlik, Su Kapasitesi(m3)
Kent mobilyası	4326	POINT	Adı, Kent mobilyası tipi, Kullanıma alınış tarihi, Tarihi özelliği var mı, deneme saat
Kurutma Kanalı	4326	LINESTRING	Adı, Uzunluk

Şekil 3.16. Veri tabanı katmanları ekranı liste sekmesi.

Katman tanımlama sayfasında 4 sekme bulunmaktadır. Bunlardan ilk aşamada kullanılan katman kaynak sekmesidir. Katman kaynak sekmesinde ön tanımlı olarak programın kendi harita servis yolları tanımlanmıştır. Bunun haricinde dışarıdan alınan harita servisleri için her seferinde servis yolu tanımlanmak yerine burada adı ve gerekiyorsa vekil sunucu kullan seçilerek kullanıcı adı ve şifresi ile tanımlanabilmektedir. Sonrasında o servis yolundan kullanılacak her katman için aynı katman kaynak kaydı kullanılabilir. Gerektiğinde buradan ilgili katman kaynak kaydı güncellenebilmektedir. Javasript kodları üzerinden şifreli CBS servislerine ulaşmak mümkün değildir. Genelde tüm bu tipteki şifreli servisler için güvenlik sebebi ile aynı durum söz konusu olmaktadır. Bu yüzden bu tip şifreli servislerin kullanılabilmesi için uygulamamızda bulunan tek yok vekil sunucu kullanma yoludur. Vekil sunucu için ayrı bir fiziksel sunucu kullanılmamıştır. Uygulamanın bulunduğu

sunucu aynı zamanda vekil sunucu olarak ta kullanılmaktadır. Uygulamanın kendi katmanları görüntülenirken yine aynı vekil sunucu üzerinden işlemler yürütülmektedir.

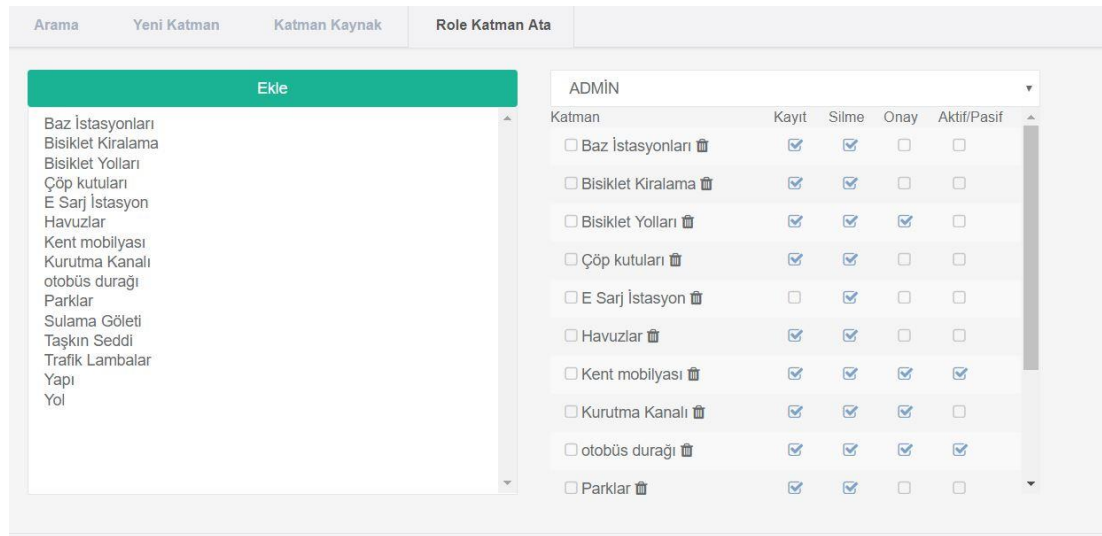
Şekil 3.17. Katman kaynak tanımlama sekmesi.

Katman tanımlama sayfasındaki yeni katman sekmesinden dış ve iç katmanlarımız eklenebilmektedir. Dış katmanlar için katman kaynağı eklendikten sonra bu sekmede kaynak olarak seçilip, biliniyorsa katman adı yazılır, bilinmiyorsa katman adı etiketinin yanındaki düğmeye tıklanarak o kaynaktaki mevcut katmanlar listelenerek seçilip yetkilendirme ve stil işleri için katman oluşturulmuş olur. Bu sekmede ayrıca katmanın grubu, başlığı ve hangi zoom seviyesinde görünür olacağı seçilebilmektedir. İstenirse tüm bilgilerde güncelleme işlemi veya silme işlemi yapılabilmektedir.

ID	Katman Başlık	Katman Adı	Kaynak	Grup	Min Resolution	Max Resolution
181	Baz İstasyonları	Geoserver Wfs	GIS_baz_istasyonlari	Bilgi İşlem		
139	Bisiklet Kiralama	Geoserver Wfs	GIS_kiralik_bisiklet_noktalari	Ulaşım		
101	Bisiklet Yolları	Geoserver Wfs	GIS_bisiklet_yollari	Ulaşım		
183	Çöp kutuları	Geoserver Wfs	GIS_cop_kutulari	Temizlik İşleri		
140	E Sarj İstasyon	Geoserver Wfs	GIS_e_sarj_noktalari	Ulaşım		
184	Havuzlar	Geoserver Wfs	GIS_havuzlar	Park ve Bahçeler		
145	Kent mobilyası	Geoserver Wfs	GIS_kent_mobilyasi	Park ve Bahçeler		
143	Kurutma Kanalı	Geoserver Wfs	GIS_kurutma_kanali	Tanımsal İşler		
161	otobüs durağı	Geoserver Wfs	GIS_otobus_duragi	Ulaşım		
182	Parklar	Geoserver Wfs	GIS_parklar	Park ve Bahçeler		

Şekil 3.18. Katman tanımlama sekmesi.

Katman tanımlama sayfası, role katman ata sekmesi ile sistemdeki rollerin hangi katmanı görebileceği seçilmektedir. Bu katmanları görme yetkisinin yanında veri ekleme, güncelleme, silme, onay verme ve pasif etme yetkileri de belirlenebilmektedir. Rolün görme yetkisine sahip olduğu katmanlar tüm bu yetkiler ile listelenmektedir. Gerektiğinde yine bu sekmeden yetki kaldırma ve katman silme işlemleri yapılabilmektedir.



Şekil 3.19. Role katman atama sekmesi.

Katman tanımlama sayfası SLD aç sekmesi ile ilgili katmanın tüm stil düzenlemeleri yapılıp, saklanabilmektedir. Bu sekme hangi katman için açıldıysa, o katmanın geometri tipine göre özelleşmiş stiller gözükmemektedir. Eğer geometri tipi nokta ise noktanın büyüklüğü, nokta yerine resim yükleme gibi özellikler bulunmaktadır. Geometri tipi çizgi ya da çoklu çizgi ise çizgi kalınlığı ve tipi şeklinde özellikler bulunmaktadır. Geometri tipi kapalı alan ya da çoklu kapalı alan ise yine çizgi kalınlığı ve tipi şeklinde özellikler bulunmaktadır. Tüm geometri tipleri için ortak olarak renk, çerçeve rengi, çerçeve kalınlığı, şeffaflık ve etiket gibi stil özellikleri bulunmaktadır. Etiket için bir ya da daha fazla katman öznitelik değeri gösterilebilmektedir. Etiket geometri üzerindeki konumu ayarlanabilmektedir. Etikete renk, büyüklük, şeffaflık ve hangi zoom seviyesinde gözüksün gibi stil ayarları yapılabilmektedir. Etiketler evrensel bir format olan SLD formatında veri tabanında tutulmaktadır. Bu yüzden sadece

geliştirdiğimiz uygulama değil, birçok farklı evrensel uygulama tarafından da görüntülenebilmektedir. İstenildiği takdirde aynı sekmeden stil güncelleme işlemi de yapılabilmektedir.

SLD isim: otobus_duragi_3

Geometri tipi: point

Point size: 5

Dolgu

Renk: [Black color swatch]

Şeffaflık: 1

Çerçeve

Renk: [Black color swatch]

Kalınlık: 2

Stil: Çizgi

Şeffaflık: 1

Ek yeri: Bevel

Cap: Kare

Resim: [Image icon]

[Dosya Seç] [Dosya...edi]

Etiket

Öznitelikler: Seçiniz

Yazı biçim: Times New Roman

Align: Center

Max Resolution: 1,200

Etiket: [Text box]

Yazı büyüklük: 12

Baseline: Middle

Çerçeve Renk: [Black color swatch]

Yazı stil: Normal

Offset X: 0

Çerçeve Genişlik: 0

Dolgu Renk: [Black color swatch]

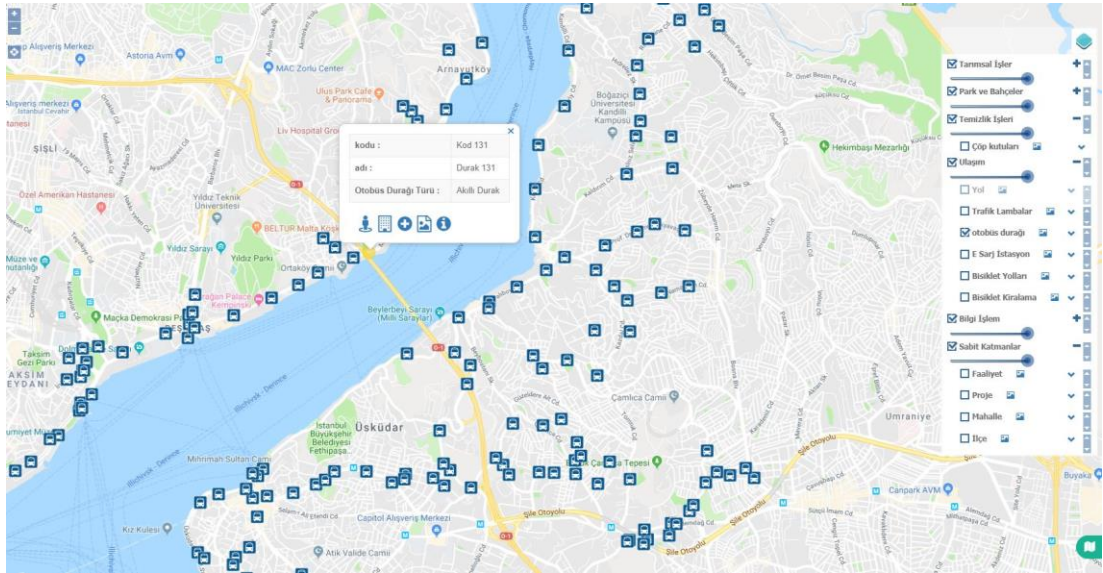
Yazı kalınlık: Normal

Offset Y: 0

Şekil 3.20. Katman stil yönetimi.

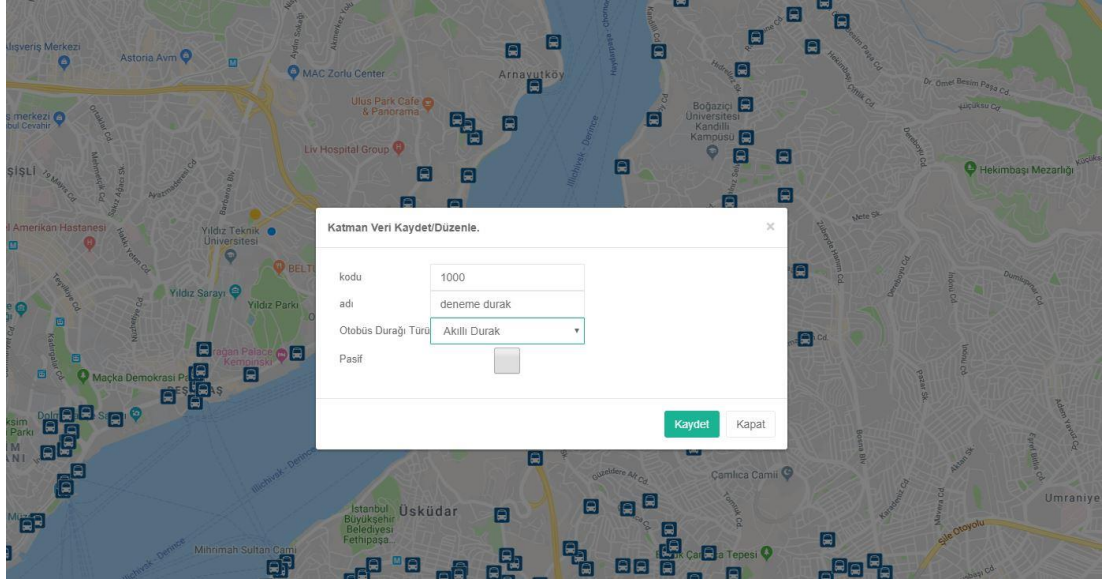
Uygulama ana ekranı birçok özellik ile donatılmıştır. Ekranın sağ tarafında katman yönetim araç çubuğu bulunmaktadır. Bu araç çubuğu içerisinde yetkimiz dahilindeki tüm katmanlar, katman tanımlamada seçilen gruplara göre kategorik bir şekilde listelenmektedir. Her katman veya herhangi bir gruptaki tüm katmanlar için şeffaflık

ve açma, kapatma özellikleri bulunmaktadır. Yine her katman veya herhangi bir gruptaki tüm katman verileri için harita üzerinde daha üstte veya altta gözükmeleri sağlanabilmektedir. Harita üzerindeki katman verilerinin detay bilgisine ulaşmak için üzerine basmamız gerekmektedir. Detay bilgisi açılan bir pencerede gözükmektedir. Bu pencereden sürüm ve onay bilgisine ulaşmak mümkündür. Eğer varsa katman verisinin Google Street View görüntüsüne de ulaşmak mümkündür. Katman verisinin bulunduğu yere odaklanabilmek için bir kısa yol düğmesi bulunmaktadır. Kullanıcı katman yönetim araç çubuğu ile tüm bu özellikleri kullanarak farklı bir zamanda sunum yapmak veya değişik bir amaç ile çalışmalarını kaydedebilmekte ve daha sonra ulaşabilmektedir.



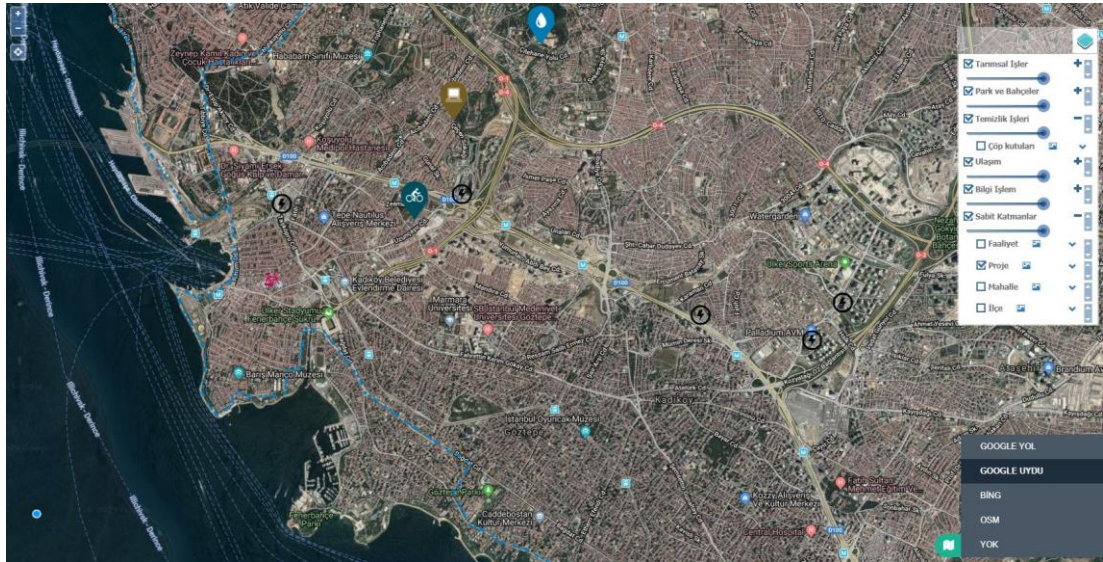
Şekil 3.21. Uygulama ana ekran genel özellikleri.

Yetki çerçevesinde her katman için geometrik veri eklenebilmekte, silinebilmekte ve güncellenebilmektedir. Katman adının altında bulunan veri düzenleme düğmesine basıldığında silme işlemi için haritadan ilgili elemanın seçilmesi gerekmektedir. Güncelleme ve ekleme işlemi içinse öncelikle çizim işleminin yapılması gerekmektedir. Ardından açılan pencerede ilgili katmanın öznelik verileri, veri tipi ve zorunluluk kurallarına göre doldurulduktan sonra işlem tamamlanmaktadır.



Şekil 3.22. Katman veri ekleme, silme ve güncelleme sekmesi.

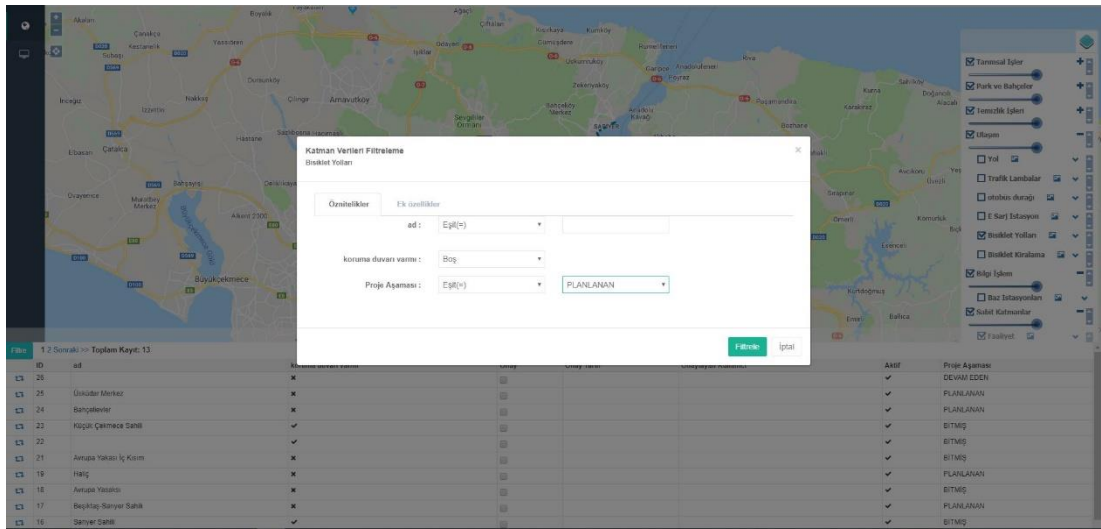
Altık harita ekranı farklı seçeneklere sahiptir. Altık harita istenmiyorsa düz beyaz zemin üzerinde de çalışma yapılabilmektedir. Katman ile altık haritanın projeksiyon sistemi farklı ise katman kuşbakışı bir şekilde altık haritanın koordinat sistemine dönüştürülerek gösterilmektedir.



Şekil 3.23. Altık harita seçenekleri.

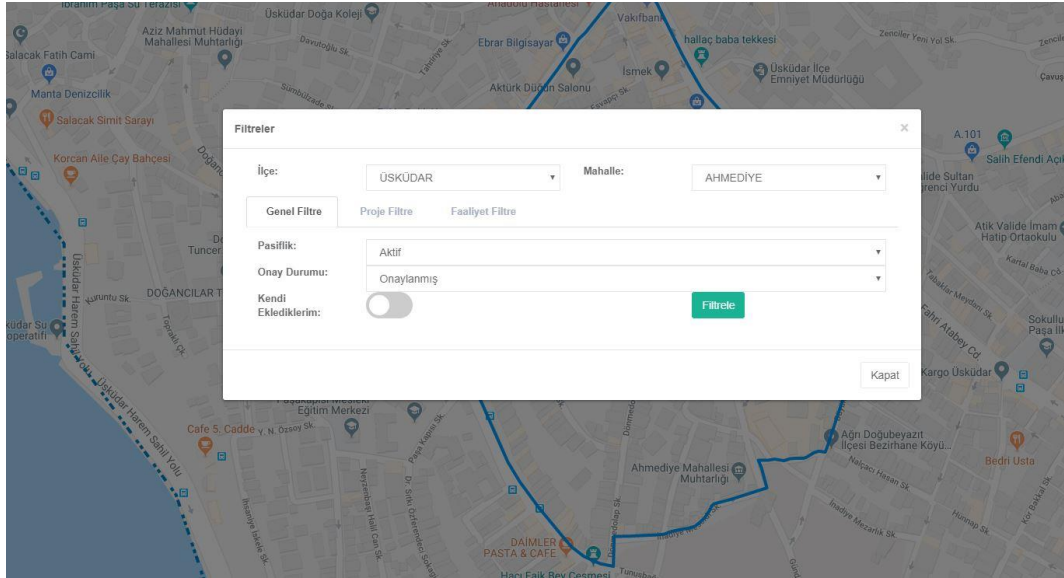
Katman adı altında bulunan katman verileri butonuna basarak; haritada geometrik olarak gösterilen kayıtlara bir liste içerisinde sözel bir şekilde de ulaşabilmektedir. Bu

listeden herhangi bir verinin konumuna odaklanılabilmektedir. Yetki çerçevesinde, katman verisi aktif/pasif yapılabilmektedir. Dinamik sorgu aracı ile tüm katman öznetelikleri üzerinden her türlü filtreleme işlemi yapılabilmektedir. Listede performans açısından sayfalama yapılmaktadır. Toplam kayıt sayısı gösterilmektedir. Liste genel ekran içerisinde dinamik bir şekilde büyütülebilmektedir. Katmandaki dinamik parametrelerde bu sekmede seçmeli menü olarak görülmektedir. Seçmeli menülere görede filtreleme işlemi yapılabilmektedir. Aynı anda birden fazla kriter içeren filtreleme işlemleri yapılabilmektedir.



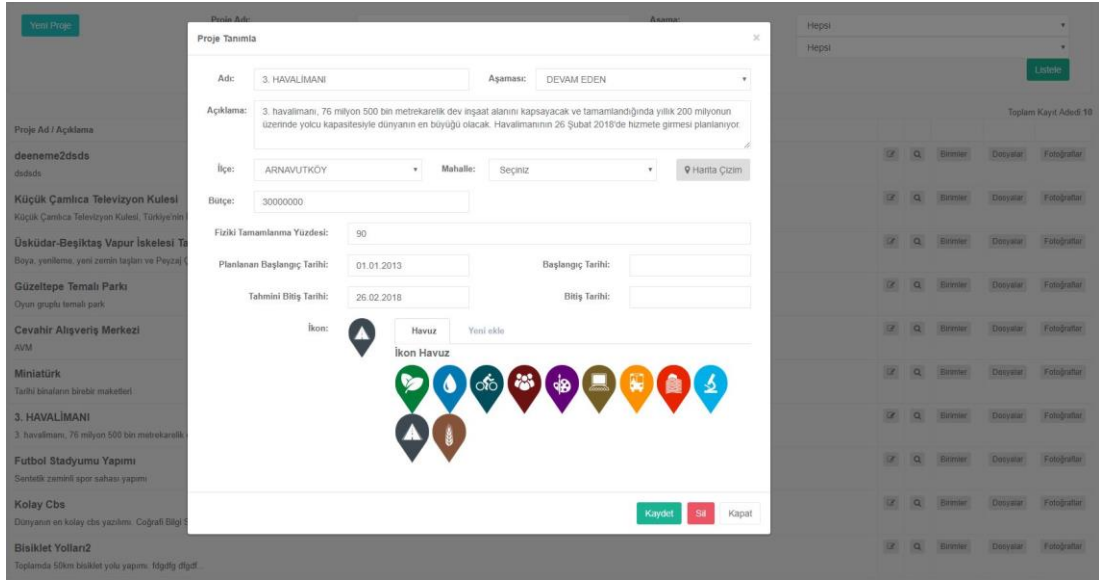
Şekil 3.24. Katman liste verileri ve filtreleme sekmesi.

Uygulama ana ekranındaki filtre düğmesine tıklanırsa filtre araç çubuğu açılmaktadır. Bu araç çubuğu içerisinde üç filtreleme sekmesi bulunmaktadır. Üç filtreleme sekmesi içinde ilçe ve mahalle araması standart olarak çalışmaktadır. Harita ekranı seçilen ilçe veya mahalleye odaklanmakta ve onların sınırlarına göre filtreleme işlemlerini yapmaktadır. Birinci sekmede pasiflik durumu, onay durumu ve kullanıcının kendi eklediği kayıtlara göre filtreleme yapılmaktadır. İkinci sekmede yerel yönetimin organizasyon şemasındaki birim bazlı ve tarih bazlı projeler filtrelenebilmektedir. Üçüncü sekmede ise yerel yönetimin organizasyon şemasındaki birim bazlı ve tarih bazlı faaliyetler filtrelenebilmektedir. Filtreleme işlemlerinde deki standart tarih aralıkları ayar dosyalarında tutulmakta ve buradan değiştirilebilmektedir. Faaliyet filtreleme işleminde faaliyet başlığında seçilebilmektedir.



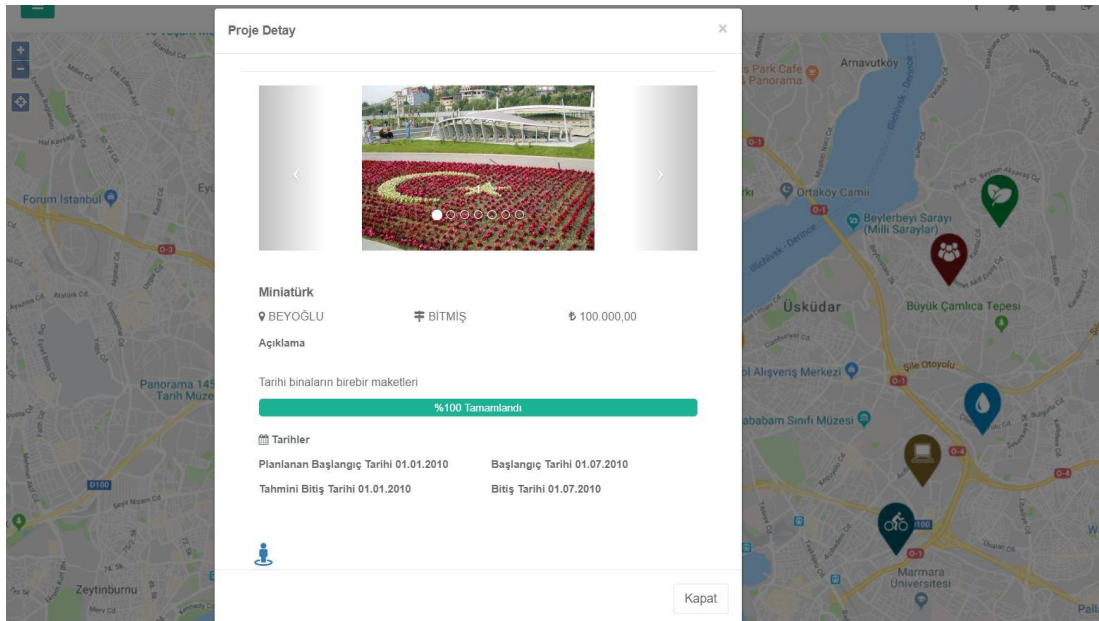
Şekil 3.25. Uygulama ana ekranı filtreleme aracı.

Projeler sayfası açıldığında daha önce girilmiş projelerin listesi bulunmaktadır. Proje adı, ilçe, mahalle, proje aşaması ve birim bilgilerine göre filtreleme yapılabilmektedir. Birim kendi yetkisine göre sadece kendi birimi, üst birimi veya tüm birimleri proje bilgilerini bu ekrandan görebilmektedir. İsterse proje bilgilerini güncelleyebilmektedir. Yeni proje kaydı yapabilmektedir. Projenin başlangıç-bitiş tarihleri, tamamlanma yüzdesi, bütçesi gibi bilgileri bu ekrandan tanımlanabilmektedir. Projelerin dosya ve fotoğrafları yüklenebilmektedir. İkon havuzundan bir resim seçilirse ana uygulama ekranında seçilen resim gözükmektedir. İkon havuzundaki resimler yeterli gelmezse kullanıcı yeni resimler ekleyebilmektedir. Bu ekrandan girilen tüm bilgiler proje kartında oluşturmaktadır. Proje aşamaları planlanan, yapım aşamasında veya bitmiş olabilmektedir. Gerekli durumlarda planlanan giriş tarihi ve tahmini bitiş tarihi bilgileride girilebilmektedir. Projeye birim sekmesinden seçilen birimler o proje için sorumlu veya yardımcı yetkisinde olabilmektedir. Proje ikonları genelde yerel yönetimlerin gerçekleştirdiği projeleri ifade edecek şekilde tasarlanmıştır. Harita çizim düğmesine tıkladığında açılan sekmede proje geometri bilgisi oluşturulabilmektedir. Projenin fiziki tamamlanma yüzdesi sıfır ile yüz arasında bir sayı olmak zorundadır. Sabit katman kategorisinde olan proje katmanının tüm bilgileri proje katmanından yönetilebilmektedir. Deneme amaçlı İstanbul odaklı projeler tanımlanmış, bilgi ve verileri girilmiştir.



Şekil 3.26. Proje yönetim ekranı

Uygulama ana sayfa ekranında projeler harita üzerinde görüntülenebilmektedir. Proje ikonu üzerinde tıkladığında proje kartı gözükmektedir. Bu kart içerisinde proje ile ilgili tüm bilgilere ulaşılabilmektedir. Google sokak görünümü düğmesine basılarak varsa sokak görünümü gösterilebilmektedir. Proje kartının en üstünde ilgili proje için yüklenen resimler görüntülenmektedir. Resimlerin üzerine tıkladığında resimlerin büyük halde görüntülenebilmektedir.



Şekil 3.27. Ana sayfa projeler ve örnek proje kartı.

Faaliyet ekranından daha önce girilmiş faaliyetler listelenmektedir. Birim, faaliyet başlığı, başlangıç-bitiş tarihleri, açıklaması girilebilmektedir. Faaliyetin geometri tipi nokta, çizgi veya kapalı alan olabilmektedir. Faaliyetin ölçü değerleri, gerçekleştiren kişileri, dosya ve fotoğrafları yüklenebilmektedir. Faaliyetin ilçesi, mahallesi, caddesi seçilebilmektedir.

Faaliyet

Ara Tanımla

Birim: TRAFİK VE SINYALIZASYON ŞUBE MÜDÜRLÜĞÜ Faaliyet: TEMİZLİK

Başlangıç tarihi: 5.02.2019 Bitiş tarihi: 5.02.2019 Harita Çizim

Açıklama: deneme

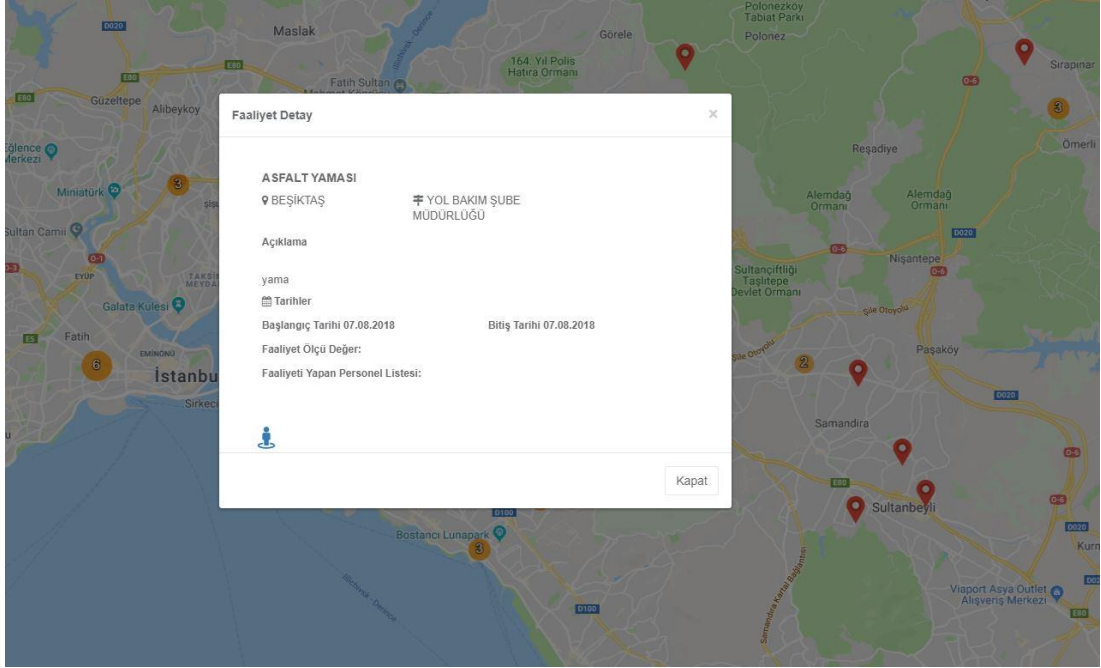
İlçe: Hepsisi Mahalle: Hepsisi Cadde: Hepsisi

Güncelle Temizle

Birim	Faaliyet	Başlangıç tarihi	Bitiş tarihi	Açıklama						
TRAFİK VE SINYALIZASYON ŞUBE MÜDÜRLÜĞÜ	TEMİZLİK	5.02.2019	5.02.2019	deneme	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
YOL BAKIM ŞUBE MÜDÜRLÜĞÜ	ASFALT YAMASI	15.01.2019	15.01.2019	iftayo	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
TRAFİK VE SINYALIZASYON ŞUBE MÜDÜRLÜĞÜ	BİSİKLET YOLU BOYAMA	15.01.2019	15.01.2019	denemeqıwıççğÜŞğ	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
YOL BAKIM ŞUBE MÜDÜRLÜĞÜ	ASFALT YAMASI	25.12.2018	25.12.2018	yama	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
TRAFİK VE SINYALIZASYON ŞUBE MÜDÜRLÜĞÜ	BİSİKLET YOLU BOYAMA	25.12.2018	25.12.2018	a	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
TARIMSAL İŞLER ŞUBE MÜDÜRLÜĞÜ	SU ANALİZİ	4.12.2018	4.12.2018	Rutin su analizi yapıldı	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
YOL BAKIM ŞUBE MÜDÜRLÜĞÜ	YENİ ASFALT	19.09.2018	19.09.2018	Sıcak asfalt döküm işi yapıldı.	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
TARIMSAL İŞLER ŞUBE MÜDÜRLÜĞÜ	TAŞKIN SEDDİ BAKIMI	10.08.2018	10.08.2018	bakım	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar
TARIMSAL İŞLER ŞUBE MÜDÜRLÜĞÜ	TAŞKIN SEDDİ BAKIMI	10.08.2018	10.08.2018	bakım	CF	X	Ölçü değeri	Kipi	Doz yatar	Fotoğraflar

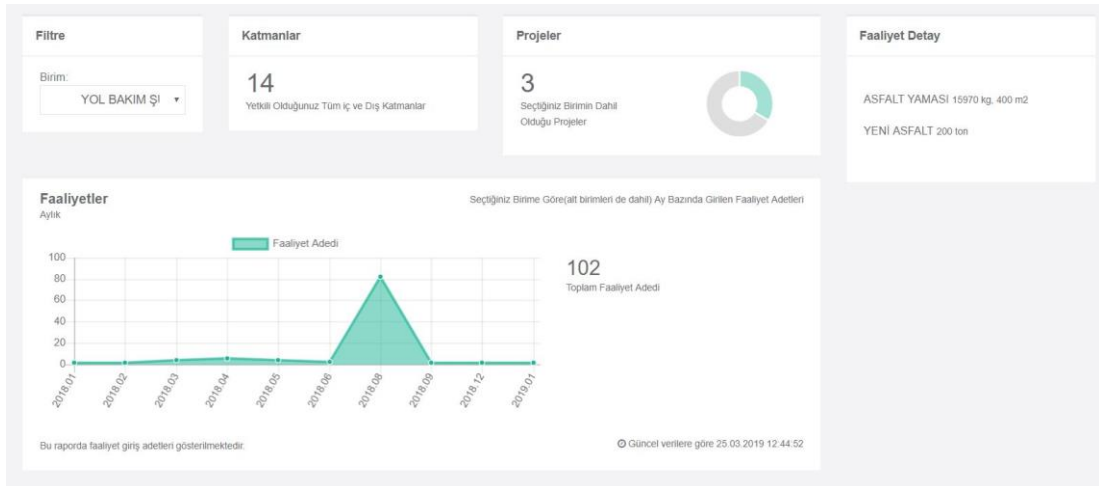
Şekil 3.28. Faaliyet tanımlama ekranı.

Uygulama ana sayfa ekranında faaliyetler harita üzerinde görüntülenebilmektedir. Faaliyet ikonu üzerinde tıklandığında faaliyet kartı gözükmetedir. Bu kart içerisinde faaliyet ile ilgili tüm bilgilere ulaşılabilir. Google sokak görünümü düğmesine basılarak varsa sokak görünümü gösterilebilmektedir. Katman verilerine tıklanarak, o veri ile alakalı faaliyetlerde tanımlanabilmektedir. Örneğin yaya geçidi katmanı için bir yaya geçidi kaydına bir yaya geçidi boyama faaliyeti buradan girilebilmektedir. Faaliyet katman haritası yoğunluğundan dolayı yüksek zoom seviyelerinde nokta bulutu şeklinde gösterilmektedir. Haritada yaklaştıkça faaliyetler ayrı ayrı gösterilmektedir. Yakın zoom seviyesinde faaliyetin geometrik şekli neyse o gözükmetedir. Faaliyet kartı içerisinde faaliyetin konusu, açıklaması, birimi, ilçesi, mahallesi, faaliyet konusu, başlangıç tarihi, bitiş tarihi, o faaliyeti gerçekleştiren personel isimleri, faaliyet ölçü birimi ve miktarı bulunmaktadır. Faaliyet alakalı genel işlemler bu ekrandan yönetilmektedir.



Şekil 3.29. Faaliyet haritası ve kartı.

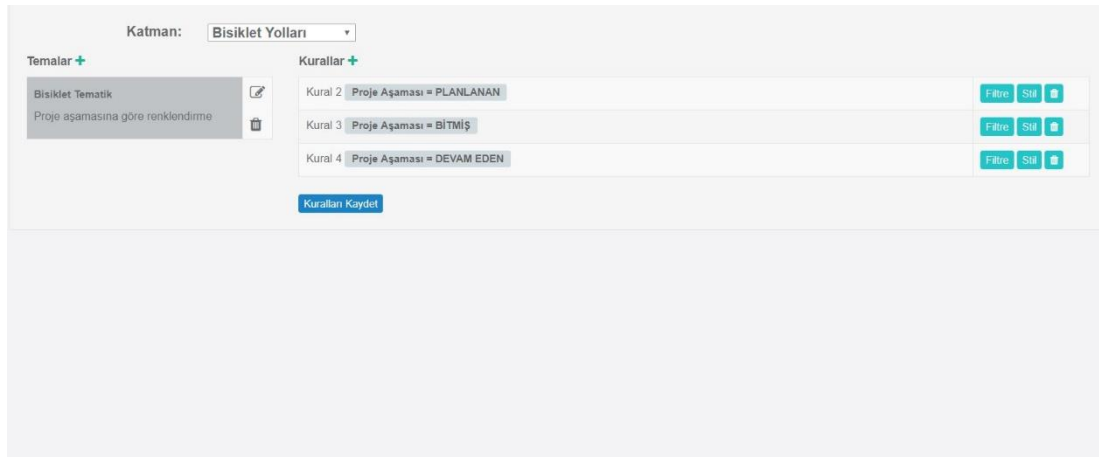
Gösterge paneli ekranında katmanlar, projeler, faaliyetler ve faaliyet detayı grafiksel olarak raporlanabilmektedir. Organizasyon şeması temelli filtreleme işlemi yapılabilmektedir.



Şekil 3.30. Gösterge paneli ekranı.

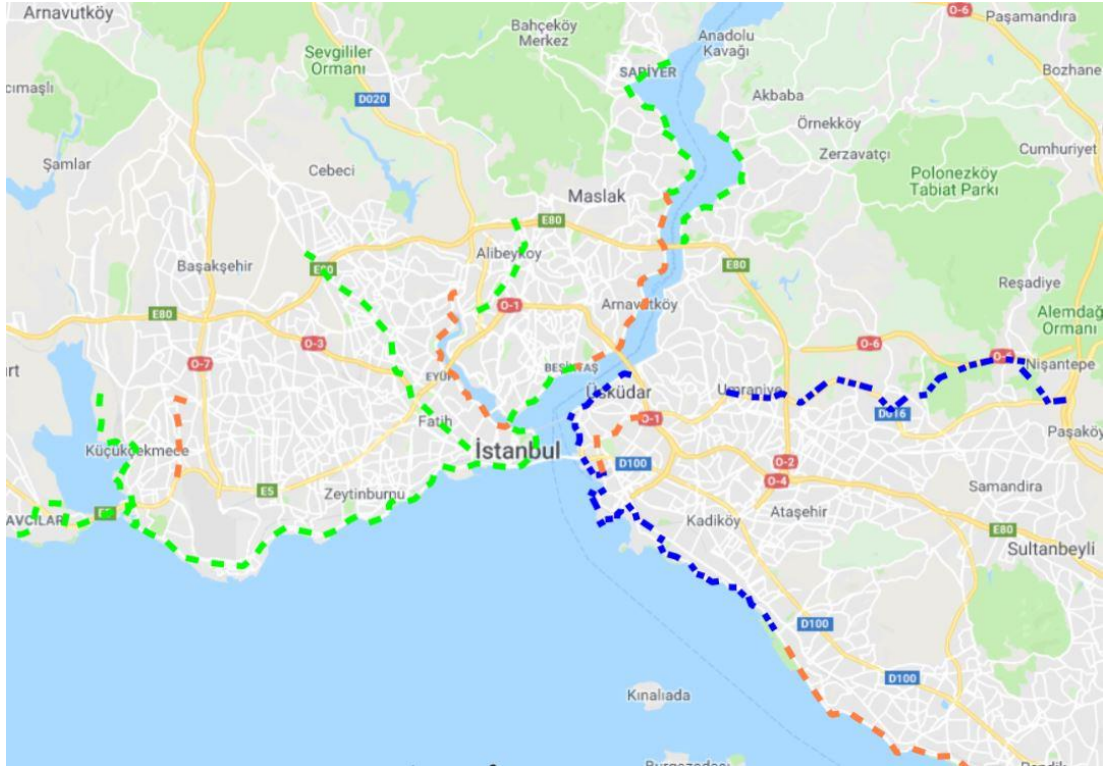
Tematik stiller ekranı ile herhangi bir katman için istenildiği kadar tematik harita tanımlanabilmektedir. Katmanın tüm sahip olduğu öznelik bilgileri için bir birinden farklı kurallar tanımlanabilmektedir. Her kural için yapılan stil temaları ile tematik

haritalar basitçe oluşturulabilmektedir. Her bir kural içerisinde bir veya daha fazla kural tanımlanabilmektedir. SQL dilinin arama yapısına benzer bir şekle sahiptir. Kural tanımlarken parametrik alanlar için seçmeli arama yapısı bulunmaktadır. Diğer alanlar için değer yazılabilecek metin kutuları bulunmaktadır. Karşılaştırma işlemleri için eşittir, büyüktür, küçüktür gibi bir çok kriter seçilebilmektedir. Eğer kural oluşturulduktan sonra stil tanımlanmazsa ön tanımlı bir stil gözükecektir. Birden fazla kurala stil tanımı yapılmazsa, o tematik harita için renksel ayırım yapılamayacaktır. Bu stil tanımları yine SLD formatına uygun bir şekilde tanımlanıp, diğer uygulamalar tarafından da okunabilecek şekilde veri tabanında saklanmaktadır.



Şekil 3.31. Tematik harita tanımlama ekranı.

Uygulama ana ekranından katmanın tematik stilleri katman isminin hemen yanındaki düğmeye basılarak görüntülenebilmektedir. Bu tematik harita örneğinde yapımı bitmiş bisiklet yolları yeşil kesikli çizgi, yapımı devam eden bisiklet yolları mavi kesikli çizgi ve yapımı planlanan bisiklet yolları turuncu kesikli çizgi olarak gösterilmiştir. Tematik harita desteği ile katman özneliklerinin stil olarak görselleştirilmesi ve katman verilerinin görsel olarak kategorize edilmesi hedeflemiştir. Yöneticiler için görsel karar destek yapılarının oluşturulabilmesi hedeflenmiştir. Sahadan veri toplayan personeller oluşturulan bu tematik haritaları seçerek daha görsel ve kolay veri toplayabilmektedirler. Örneğin kent mobilyasının ana temasında tek resim kullanılmıştır. Kent mobilyası için oluşturulan bir tematik harita; bank, çeşme, çöp kutusu gibi türler için kendi resimlerini içerebilmektedir. Eğer kullanıcı sahadan veri toplarken bu temayı seçerse, görsel olarak işi kolaylaşmaktadır.



Şekil 3.32. Örnek tematik harita.

Bu tematik harita örneği ise baz istasyonları katmanı için hazırlanmıştır. Baz istasyonunun sahibi olan operatör şirketine göre kategorize edilmiştir. Stil içinse bazı operatör şirketlerinin logoları yüklenmiştir.



Şekil 3.33. Baz istasyon katmanı tematik haritası.

Kullanıcı ve kişi tanımı aynı ekranda yönetilmektedir. Eğer sadece kişi kaydı girilecekse kullanıcı pasif olarak seçilip; ad, soyadı, telefon, mail, unvan ve birim alanlarının doldurulması yeterlidir. Kişinin sisteme kullanıcı girişi yapacaksa da kullanıcı aktif olarak seçilip; kullanıcı adı, şifre ve rol alanlarının doldurulması gerekmektedir. Tüm kullanıcı ekranları bu sayfadan yönetilebilmektedir. Kullanıcıya şifre verildikten sonra ilk girişte şifre değiştirmesi gerekiyorsa bu ekrandan ilk giriş şifre değiştirsin seçilebilmesi gerekmektedir. Kullanıcının yetkisel olarak hangi birimlerin faaliyet ve projelerinin görüntüleyebileceğinin anlaşılması için birim alanının doldurulması gerekmektedir. Ünvan bilgisi ile de kişilerin yapıları görev belirtilebilmektedir. Şifre bilgisi yazılırken konfigürasyon dosyalarındaki kurallara uymazsa kullanıcı oluşturulamamakta ve şifrenin hangi formatta olması gerektiği gösterilmektedir. Kullanıcı adı seçilirken, sistem otomatik olarak o kullanıcı adı kullanımda mı değil mi kontrolünü yapmakta ve bilgi mesajı vermektedir. Eğer o kullanıcı adı kullanımda ise işleme devam etmek için farklı bir kullanıcı adının seçilmesi gerekmektedir.

Arama
Yeni Kayıt

Ad

Soyad

Kullanıcı Adı

Şifre

Telefon

E-Mail

Rol

Birim

Ünvan

Kullanıcı Aktif/Pasif İlk giriş şifre değiştirsin

Ekle
Temizle

	Ad	Soyad	Kullanıcı Ad	Cep Tel.	E-Mail	Rol	Birim	Ünvan
<input checked="" type="checkbox"/>	AHMET1	YILMAZ	osmany			ADMIN	BİLGİ İŞLEM DAİRESİ BAŞKANLIĞI	Bilgisayar Mühendisi
<input checked="" type="checkbox"/>	ALI	VELI	mahmut	05344444444		KULLANICI	TRAFİK VE SINYALİZASYON ŞUBE MÜDÜRLÜĞÜ	Yok
<input checked="" type="checkbox"/>	DE	DE	mmmm			KULLANICI	Yok	Yok
<input checked="" type="checkbox"/>	DENEME1	KULL	deneme			KULLANICI	FEN İŞLERİ DAİRESİ BAŞKANLIĞI	Yok
<input checked="" type="checkbox"/>	HAMDİ	YILMAZ	mmmm			ADMIN	Yok	Bilgisayar Mühendisi
<input checked="" type="checkbox"/>	MAHMUT	YILMAZ	ersin	05555555555	ersin@ersin.com	ADMIN	BELEDİYE	Yok
<input checked="" type="checkbox"/>	MEHMET	YILMAZ	seb			ADMIN	CBS VE YAZILIM ŞUBE MÜDÜRLÜĞÜ	Yok
<input checked="" type="checkbox"/>	MUSA	BALTA	musa			ADMIN	ULAŞIM DAİRESİ BAŞKANLIĞI	Bilgisayar Mühendisi
<input checked="" type="checkbox"/>	MUSTAFA	YILMAZ	can			ADMIN	YOL BAKIM ŞUBE MÜDÜRLÜĞÜ	Yok

Şekil 3.34. Kullanıcı/kişi tanımlama ekranı.

Kullanıcı, uygulamadan izin verildiği takdirde kullanıcı adı veya şifresini değiştirebilmektedir. Şifre değiştirmek istediğinde mevcut şifresini ve kullanmak istediği yeni şifresini iki kez girerek şifresini güncelleyebilmektedir. Şifreler geri döndürülemez şekilde şifrelenerek veri tabanında tutulmaktadır.

Şekil 3.35. Kullanıcı adı ve şifre tanımlama ekranı.

Uygulama içindeki tüm parametrik alanlar ekstra bir veri tabanı tablosu yapılmadan parametre tablosunda tutulmaktadır. Bu parametrik alan ve değerleri parametre ekranından yapılabilmektedir. Parametrik alan tanımlanırken uygulamadan ekleme seçilmişse, parametre alanları da aynı ekrandan yüklenebilmektedir. Eğer seçilmedi ise sadece veri tabanından eklenebilmektedir. Cinsiyet, faaliyet ölçü birimi, proje aşaması ve durak tipi gibi birçok parametre ön tanımlı olarak parametre tablosunda bulunmaktadır.

ID	Değer	Aktif/Pasif
107	Açık Durak	Aktif
109	Akıllı Durak	Aktif
108	Kapalı Korumalı Durak	Aktif

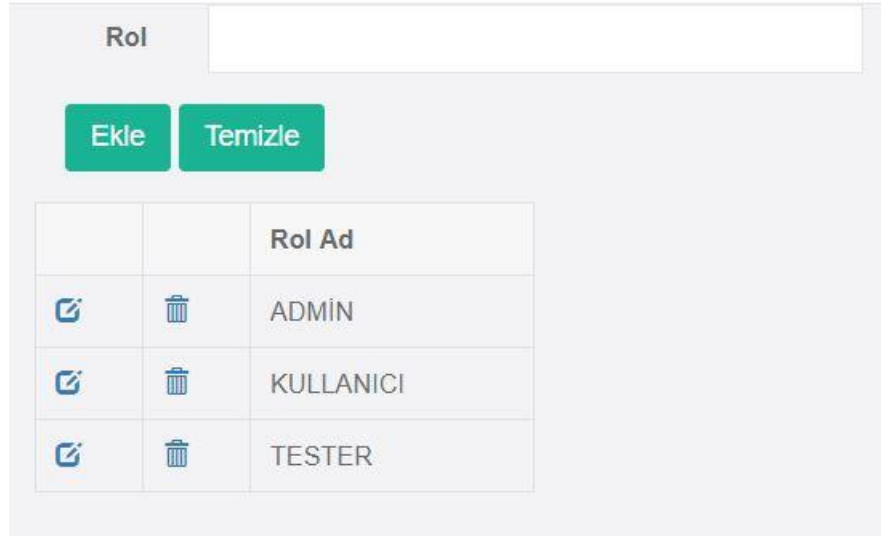
Şekil 3.36. Parametre tanımlama ekranı.

Organizasyon şeması ekranı ile yerel yönetimler kurum müdürlükleri, daire başkanlıkları ve diğer birimleri hiyerarşik olarak tanımlayabilmektedirler. Bu hiyerarşi yetkilendirme işlemlerinde de kullanılmaktadır. İsmi güncellenen kaydın eski bilgileri bu tabloda tutulmamaktadır. Eğer bir müdürlük başka bir daire başkanlığının altına taşınır veya bir daire başkanlığı başka bir genel sekreter yardımcılığının altına taşınırsa; otomatik olarak o birimin altındaki tüm personeller yeni hiyerarşide görüntülenecektir. Kullanıcılar ekranları açarken rol yetki kontrolü yapılmaktadır. Rol yetki kontrolü içerisinde kendi birimi, üst birimi, tüm birimler gibi hiyerarşik veri kontrolü de yapılmaktadır. Bu hiyerarşi organizasyon şeması ekranında tanımlanmaktadır. Eğer birim altında personeller tanımlanmış ise birim silme işlemi esnasında hata mesajı verilecek ve silme işlemi yapılmayacaktır. Kullanıcının birim silme işlemini tamamlayabilmesi için öncelikle kullanıcı/kişi tanımlama ekranından o birimdeki personelleri listelemesi ve birimlerini değiştirmesi gerekmektedir. Ardından sorunsuz bir şekilde birim silme işlemini gerçekleştirebilecektir. Organizasyon şeması oluşturulurken herhangi bir hiyerarşi limiti bulunmamaktadır.



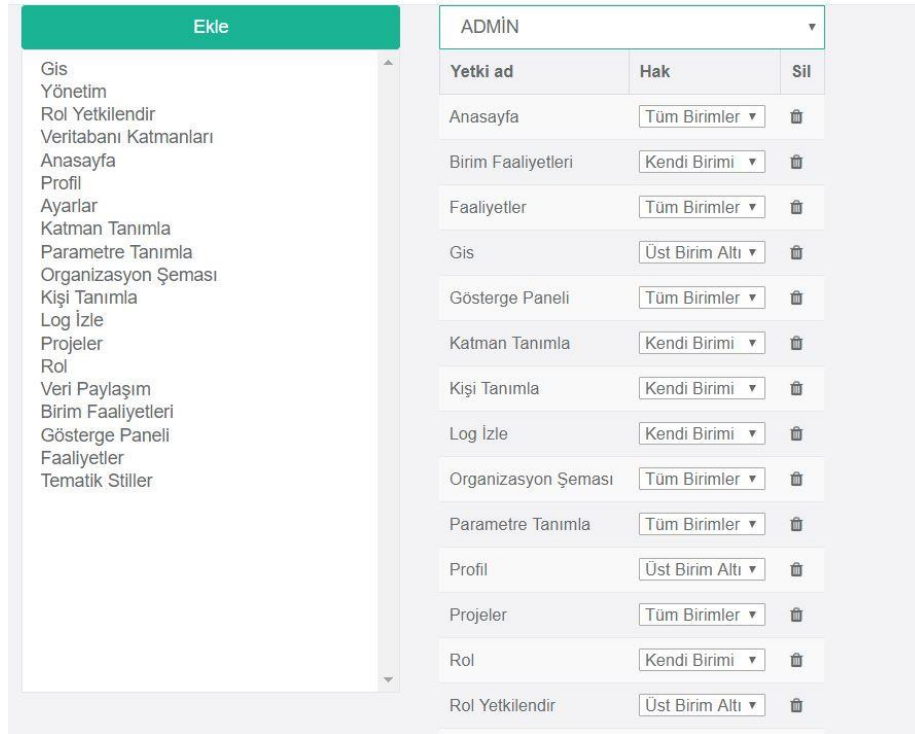
Şekil 3.37. Organizasyon şeması ekranı.

Uygulamadaki farklı kullanıcı tiplerini gruplamak için rol tanımlama ekranından rol tanımları yapılabilmektedir. Güncelleme ve silme işlemleri yine bu ekrandan yapılabilmektedir. Liste halinde tüm rollere ulaşılabilir.



Şekil 3.38. Rol tanımlama ekranı.

Rol yetki ekranı ile rollerin sahip olduğu yetkiler yönetilebilmektedir. Rolün eklenecek yetki için organizasyon şeması temelli kullanıcının kendi birimi temelli mi, üst birimi temelli mi yoksa tüm birimleri görecektir seçimi yapılabilmektedir. Bu özellik ile yerel yönetimlerde birden fazla birimin işini yapan ya da raporlamasını yapan kullanıcılar hedeflenmiştir.



Şekil 3.39. Rol yetki tanımlama ekranı.

Yerel yönetimler, CBS modelini bu uygulama ile oluşturduktan sonra başka kurumlar ile servisler üzerinden veri paylaşımına ihtiyaç duymaktadırlar. Bu paylaşım işlemini yapabilmeleri için birçok yöntem ve teknoloji bulunmaktadır. Kullanıcı veri paylaşım ekranı ile bu teknoloji ve yöntemlerden soyutlanarak verilerini diğer kurumlar ile paylaşabilmektedir. Bunun için öncelikle hangi katmanları ve o katmanın hangi özniteliklerini paylaşılacağına karar verilerek, arayüz yardımı ile bir görüntü oluşturulması gerekmektedir. Eğer seçtiğimiz katman içinde parametrik öznitelikler var ise uygulama, otomatik olarak o öznitelikleri parametre tablosu ile birleştirebilmektedir. Görüntüyü oluşturan kişinin basit anlamda SQL bilgisi varsa birçok farklı özellikte görüntü oluşturması mümkündür. Oluşturulmak istenen görüntünün nasıl sonuçlar ürettiği ekrandaki yıldırım düğmesine basılarak görülebilmektedir. Görüntü oluştuğu anda resim veya vektör servisleri de otomatik olarak uygulama tarafından oluşturulmaktadır.

The screenshot displays the 'View Oluştur' (View Creation) interface. On the left, there is a list of fields including 'deneme_nokta', 'bisiklet_yollari', 'e_sarj_noktaları', 'hali_saha', 'havuzlar', 'kiralik_bisiklet_noktaları', 'kurutma_kanali', 'park', 'sulama_goleti', 'taskin_seddi', 'trafik_jambalari', 'yapi', 'yol', and 'otobus_duragi'. The central area shows a SQL query editor with the following query:

```
select
t0.nbisiklet_yollari_id, t0.nvad, t0.bkoruma_duvari, t0.p13_proje_asamasi, t1.nvdeger as p13_proje_asamasi_adi, t0.geom
from
giskatman.bisiklet_yollari t0
left join gisana.tip_deger t1 on t1.nid = t0.p13_proje_asamasi
```

Below the query editor, a table displays the query results. The table has 5 columns: 'nbisiklet_yollari_id', 'nvad', 'bkoruma_duvari', 'p13_proje_asamasi', and 'p13_proje_asamasi_adi'. The results are as follows:

nbisiklet_yollari_id	nvad	bkoruma_duvari	p13_proje_asamasi	p13_proje_asamasi_adi
18	Avrupa Yasaksız	false	64	BITMİŞ
23	Küçük Çekmece Sahili	true	64	BITMİŞ
22	null	true	64	BITMİŞ
21	Avrupa Yakası İç Kısım	false	64	BITMİŞ
15	Beykoz Sahili	true	64	BITMİŞ
16	Sarıyer Sahili	true	64	BITMİŞ

Şekil 3.40. Veri paylaşım görüntü oluşturma ekranı.

Bir önceki sekmede servis edilmek istenen görüntü oluşturulduktan sonra bu sekmeden hangi kuruma servis açılacaksa onun için bir kullanıcı adı ve şifresi oluşturulması gerekmektedir. Kurum bilgisi oluşturulduktan sonra daha önce oluşturulan görüntüler bu ekrandan yetki olarak o kuruma verilebilmektedir. İstenildiğinde ilgili kurumun kullanıcısı pasif edilebilmektedir. Yeni oluşturulan yetkiler yine bu ekrandan sonradan

verilebilmektedir. Bir kurumun sahip olduğu görüntü yetkileri yine bu sekmede liste olarak bulunmaktadır. Karşı kurumun açılan bu katman görüntülerine ulaşabileceği servis yolları sekmenin en üstünde, otomatik kopyalama düğmesi ile bulunmaktadır. Bu ekranla oluşturulan tüm rol, kullanıcı, şifre ve yetkiler Geoserver uygulama arayüzlerinden de yönetilebilmektedir. Katman görüntüleri ise PostgreSQL veri tabanında görüntü sekmesinde bulunmaktadır. Herhangi bir veri tabanı yönetim programı ile değişiklik yapılabilir. Eğer görüntü üzerindeki değişiklikler geliştirdiğimiz uygulama yerine başka bir uygulamadan yapılırsa, aynı görüntünün Geoserver üzerinden harita servisinin de güncellenmesi gerekmektedir. Yerel yönetimler bu ekran ile kendisinden servis isteyen kurumlara hem vektör hem de resim servislerini oluşturabilmektedir.

Kullanıcı Adı	Aktif?	Yetkiler
K_CEVREBAKANLIGI	<input checked="" type="checkbox"/>	Yetkiler
K_DENEME	<input checked="" type="checkbox"/>	Yetkiler
K_YETKISIZ_KULLANICI	<input checked="" type="checkbox"/>	Yetkiler

Şekil 3.41. Servis kullanıcı oluşturma sekmesi.

Uygulamanın tüm hareket ve giriş, çıkış kayıtlarına bu ekrandan ulaşılabilir. Hatalı ve kötü niyetli giriş denemeleri esnasında kullanılan kullanıcı adı ve şifreler bu ekranda görüntülenebilir.

Log

Hareket Log Giriş Log

Baş.Tarih: 22.03.2019 Bit.Tarih: 25.03.2019

Giriş Listele

Tarih	Ad	Soyad	Ip	Kul.Ad	Kul.Şifre
25.03.2019 17:04:37	MUSTAFA	YILMAZ	::ffff:172.19.0.2		
25.03.2019 16:45:08	MUSTAFA	YILMAZ	:::1		
25.03.2019 16:24:58	MUSTAFA	YILMAZ	:::1		
25.03.2019 16:03:14	MUSTAFA	YILMAZ	::ffff:172.19.0.2		
25.03.2019 13:43:00	MUSTAFA	YILMAZ	:::1		
25.03.2019 13:27:41	MUSTAFA	YILMAZ	::ffff:172.19.0.2		
25.03.2019 12:44:32	MUSTAFA	YILMAZ	::ffff:172.19.0.2		
25.03.2019 10:55:04	MUSTAFA	YILMAZ	::ffff:172.19.0.2		
25.03.2019 10:23:05	MUSTAFA	YILMAZ	::ffff:172.19.0.2		
25.03.2019 06:18:17	MUSTAFA	YILMAZ	::ffff:172.19.0.2		

« < 1 2 > » 10 ▾

Şekil 3.42. Log ekranı.

Birimlerin faaliyetleri bu ekrandan tanımlanabilmektedir. Birimlere daha önce tanımlanan faaliyetler liste olarak görüntülenmektedir. Faaliyetler üst/alt faaliyet şeklinde hiyerarşik olarak tanımlanabilmektedir. Bu şekilde yerel yönetimlerin birimlerin yaptığı işleri belli üst başlıklar ile gruplandırabilmeleri hedeflenmiştir. Faaliyetler gerektiğinde pasif yapılabilir. Çizimi yapılmayan faaliyet başlıkları için konum zorunlu değil alanı seçilmelidir. Bu ekran, yerel yönetimlerin stratejik planlarına uygun bir şekilde faaliyetlerini tanımlayabilmelerine göre geliştirilmiştir. Birim faaliyetleri tanımlanırken sadece 2 kademe hiyerarşi olabilmektedir. Fazlası yapılmak istendiğinde hata vermektedir. Üst faaliyet başlığının silinebilmesi için alt faaliyetlerinin olmaması gerekmektedir. Aksi takdirde uygulama hata verecek ve silme işlemi başarısız olacaktır.

Arama Yeni Kayıt

Birim: YOL BAKIM ŞUBE MÜDÜRLÜĞÜ

Listele

Faaliyet Ad	Üst Faaliyet	Durum	Konum Zorunlu mu			
ASFALT YAMASI	ASFALT İŞLERİ	True	True	🔍	🔗	🗑️
YENİ ASFALT	ASFALT İŞLERİ	True	True	🔍	🔗	🗑️
ASFALT İŞLERİ		True	True	🔍	🔗	🗑️

Şekil 3.43. Birim faaliyet tanımlama ekranı.

3.3.5. Uygulama çalışma ortamı

Yerel yönetimler için geliştirdiğimiz CBS uygulaması yukarıdaki kısımlarda anlatıldığı gibi birçok katmandan oluşmaktadır. Her katmanın içinde bir veya daha fazla açık kaynak kodlu yazılım bulunmaktadır. CBS uygulamamızın çalışabilmesi için tüm bu yazılımların sunucu ortamına kurulup, gerekli yapılandırma ayarlarının yapılması gerekmektedir. Bu kurulum ve yapılandırma işlemleri hem zaman almakta hem de uzmanlık gerektirmektedir. Bu katmanlı yazılım mimari önerisini kullanmak isteyen tüm yerel yönetimleri, kurulum ve yapılandırma işlemleri ile uğraştırmamak için konteynır yapısından faydalanılmıştır. Konteynır yapısını da sağlamak için Docker uygulaması kullanılmıştır. Örnek sunucu sistemi bir adet ve iki çekirdekli işlemcili bir sunucudur. Tüm kurulumlar bu bir adet sunucu üzerine yapılarak çalıştırılmış ve test edilmiştir. Docker uygulaması ile veri tabanı katmanı içindeki PostgreSQL için bir adet konteynır oluşturulmuştur. Harita katmanı içindeki Geoserver için bir adet konteynır oluşturulmuştur. Uygulama ve iş katmanı için geliştirdiğimiz Net Core uygulaması için bir adet konteynır oluşturulmuştur. Uygulamanın gelen istekleri yönetebilmesi ve doğru uygulama yoluna yönlendirilebilmesi için de bir adet konteynır oluşturulmuştur. Tüm bu dört adet konteynırın tek bir seferde ve uygun sırada kullanılabilmesi için gerekli yapılandırma dosyaları oluşturulmuştur. Bu uygulamayı kullanmak isteyen yerel yönetimlerin Linux işletim sistemli bir makinaya Docker uygulamasını kurması ve oluşturulan yapılandırma dosyasını çalıştırması yeterli olacaktır. Katmanlı yazılım mimari önerimiz birkaç dakika içerisinde kurulum ve yapılandırma işlemlerini tamamlayıp, kullanıma hazır hale gelecektir.

BÖLÜM 4. TARTIŞMA VE SONUÇ

Çalışmamızda; lisanlı uygulamaların paket halinde sunduğu coğrafi bilgi sistemi özelliklerini, açık kaynak kodlu muadil kütüphane ve programları bir araya getirerek sağlayan bir uygulama geliştirilmiştir. Yüksek lisans ücretlerini ödemek istemeyen yerel yönetimler geliştirdiğimiz uygulama ile kendi coğrafi bilgi sistemleri modelini oluşturabilirler. Lisanlı ürünlerden farklı olarak birçok programla sunulan özellikler ve fonksiyonellik bizim çalışmamızda tek bir uygulama ile sunulmuştur. Hem web hem de mobil uygulama ile kullanıcıya birçok ortamdan uygulamayı kullanma esnekliği sağlanmıştır. Yerel yönetimlerin en çok ihtiyaç duyacağı; konumsal proje, faaliyet tanımlama özellikleri uygulamaya eklenmiştir.

Bu çalışma nesneye dayalı programlama ile geliştirilmiştir. Geliştirme aşamasında birçok tasarım deseninden faydalanılarak; esnek, basit ve yeni ihtiyaçları daha kolay entegre etme hedeflenmiştir. Bu çalışma ile yerel yönetimlerin CBS konusundaki ihtiyaçlarını karşılarken, bilgi sahibi olmak zorunda olduğu; OGC, Restful web servisleri, Inspire, coğrafi veri tabanları ve meta veri gibi konularda kılavuzluk etmek amaçlanmıştır.

Yerel yönetimler bu çalışma ile oluşturdukları CBS veri modelini diğer kurumların kullanabilmesi için esnek bir şekilde servis oluşturabileceklerdir. Bu servisler için gereken görüntü ve web servisleri, uygulama tarafından teknolojik olarak kullanıcıdan soyutlanarak otomatik olarak oluşturulacaktır.

Bu çalışmayı kullanmak isteyen yerel yönetimler konteynır kavramı ile kurulum ve yapılandırma işlemlerinden soyutlanarak, otomatik kurulum ve yapılandırma ile katmanlı yazılım mimari önerimizi kullanabilmeleri hedeflenmiştir.

Çalışma ile üretilen CBS modeli ve verileri, evrensel kurallar ve formatlar göz önünde bulundurularak hazırlandığı için; bu format ve kurallara uygun diğer tüm programlarca da okunabilecektir.

Katmanlı mimari ile üretilen çalışmamız temel CBS metod ve araçlarını içeren bir harita motoru görevindedir. Gelecekte ihtiyaç duyulan birçok farklı özellik bu çekirdek konumundaki harita motoruna eklenebilecektir. Örneğin katman verisine tıklandığında son kullanıcı tarafından o katmana göre tasarlanmış açılır menüler gözükebilecektir. Şu anda Maks, tapu kadastro gibi servislere bağlanılması mümkündür. Ancak bu sistemler çok kapsamlı ve özelliştirmiş tasarım ve ekranlar ile daha kolay kullanılabilmesi sağlanabilmektedir. Bu ekranların geliştirilmesi yapılabilir. Katmanlar birbirinden bağımsız bir şekilde tanımlanmaktadır. Ancak bazı katmanların birbirleri ile ilişkileri olabilmektedir. Örneğin otobüs durağı katmanı, hat çizimi ve otobüs konum katmanı birbirleri ile ilişkili katmanlardır. Bu katmanlar arasında ilişki kurulması ve birbirleri ile gruplanması sağlanabilir. Faaliyeti gerçekleştirilen personeller faaliyet işlemi bitirildikten sonra girilebilmektedir. Sahada personel görevlendirme, online iş emri ve takibi gibi özellikler eklenebilir. Katman verileri toplanırken geometrik şekil harici, herhangi coğrafi bir kontrol yoktur. Nokta veri tipi için iki nokta arası mesafe kontrolü konulabilir. Çizgi ve kapalı alanlar arası mesafe ve birbirlerini kesmeme gibi kontroller konulabilir. Sahadan toplanan trafik verilerine göre bulanık mantık temelli sinyalizasyon süreleri belirlenebilir. Bu tip yeni özellikleri gerçekleştiren sistemler geliştirmek normalde çok zahmetli olabilmektedir. Geliştirdiğimiz harita motoru ile çok daha kolay bir şekilde yerel yönetimlerin kullanımına açılacaktır.

KAYNAKLAR

- [1] Sönmez, N.K., Sarı, M., Coğrafi Bilgi Sistemleri Temel Esasları ve Uygulama Alanları. Derim, 21(1): 54-68, 2004.
- [2] Lacovella, S., Youngblood, L.S., GeoServer Beginner's Guide. İçinde: GIS Fundamentals. 1. Baskı, Packt Yayınları, Birmingham, 7-15, 2013.
- [3] https://en.wikipedia.org/wiki/Spatial_reference_system, Erişim Tarihi: 23.02.2019.
- [4] Yılmaz, İ., Uygun Harita Projeksiyonu Seçiminde Bazı Temel Esaslar. Harita Teknolojileri Elektronik Dergisi, 1(2): 31-42, 2009.
- [5] <https://en.wikipedia.org/wiki/Latitude>, Erişim Tarihi: 23.02.2019.
- [6] <https://developers.arcgis.com/documentation/core-concepts/spatial-references/>, Erişim Tarihi: 23.02.2019.
- [7] Akıncı, H., Cömert, C., TUCBS ve INSPIRE Teknik Mimarisi. 12. Türkiye Harita Bilimsel ve Teknik Kurultayı, Ankara, 2019
- [8] <https://cbs.csb.gov.tr/veri-temalari-hakkinda-i-86103>, Erişim Tarihi: 23.02.2019.
- [9] <https://maks.nvi.gov.tr/>, Erişim Tarihi: 23.02.2019.
- [10] <https://parselsorgu.tkgm.gov.tr>, Erişim Tarihi: 23.02.2019.
- [11] <https://sakus.sakarya.bel.tr/>, Erişim Tarihi: 18.03.2019.
- [12] <http://proje.sakarya.bel.tr:86/ItsProjeWeb/Harita>, Erişim Tarihi: 18.03.2019.
- [13] https://uym.ibb.gov.tr/YHarita/Harita_tr.aspx, Erişim Tarihi: 18.03.2019.
- [14] Obe, R.O., Hsu, L.S., PostGIS in Action, Manning Yayınları, 1-481, 2011.

- [15] Shukla, D., Shivnani, C., Shah, D., Comparing Oracle Spatial and Postgres PostGIS. IJCSC, 7(2): 95-100, 2016.
- [16] Castronova, A.M., Goodall, J.L., Elag, M.M., Models as web services using the Open Geospatial Consortium(OGC) Web Processing Service (WPS) standard. ELSEVIER, 41: 72-83, 2016.
- [17] <https://www.ibm.com/developerworks/webservices/library/ws-restful/>, Erişim Tarihi: 23.02.2019.
- [18] Henderson, C., Mastering Geoserver, Packt Yayınları, 1-399, 2014.
- [19] <https://gisgeography.com/qgis-arcgis-differences/>, Erişim Tarihi: 23.02.2019.
- [20] <http://desktop.arcgis.com/en/arcmap/latest/map/main/mapping-and-visualization-in-arcgis-for-desktop.htm>, Erişim Tarihi: 23.02.2019.
- [21] <http://blog.thinkgeo.com/2015/09/08/leaflet-vs-openlayers-3-which-is-the-better-client-side-javascript-mapping-library/>, Erişim Tarihi: 23.02.2019.
- [22] <https://leafletjs.com/examples/choropleth/>, Erişim Tarihi: 23.02.2019.
- [23] Farkas, G., Mastering OpenLayers 3, Packt Yayınları, 1-279, 2016.
- [24] <https://openlayers.org/en/latest/examples/heatmap-earthquakes.html?q=>, Erişim Tarihi: 23.02.2019.
- [25] Freeman, A., Pro Asp.net Core Mvc 2, Apress Yayınları, 1-1011, 2017.
- [26] <https://www.hanselman.com/blog/ASPNET5IsDeadIntroducingASPNETCore10AndNETCore10.aspx>, Erişim Tarihi: 23.02.2019.
- [27] Martin, R.C., Clean Architecture: A Craftsman's Guide to Software Structure and Design, Prentice Hall, 1-404, 2018.
- [28] Hall, G.M., Adaptive Code: Agile coding with design patterns and SOLID principles, Apress Yayınları, 1-418, 2017.
- [29] Sarcar, V., Design Patterns in C#, Apress Yayınları, 1-455, 2018.
- [30] McGraw, G., Software Security. IEEE, 2(2): 80-83, 2004.
- [31] https://www.owasp.org/index.php/About_The_Open_Web_Application_Security_Project, Erişim Tarihi: 23.02.2019.

- [32] http://csirt.ulakbim.gov.tr/dokumanlar/Ceviri_OWASP_ilk10_2007.pdf, Erişim Tarihi: 23.02.2019.
- [33] Yıldırım, G., Konteynerler ve Docker, Abaküs Yayınları, 1-167, 2017.
- [34] <https://www.aquasec.com/wiki/display/containers/Docker+Containers>, Erişim Tarihi: 23.02.2019.
- [35] Savaş, Ö., Özcan, G., İl Özel İdareleri için Yeni Bir Açık Kaynak Coğrafya Bilgi Sistemi Yazılımı. DEUFMD, 21(61): 113-123, 2016.
- [36] Bensghir, T.K., Akay, A., Bir Kamu Politika Aracı Olarak Coğrafi Bilgi Uygulamalarının Değerlendirilmesi. Çağdaş Yerel Yönetimler, 15(1): 31-46, 2006.
- [37] Duval, E., Hodgins, W., Sutto, S., Weibel, S.L., Metadata Principles and Practicalities. Çağdaş Yerel Yönetimler, 8(4), 2002.

ÖZGEÇMİŞ

Can Alper ÖZTÜRK, 31.05.1986'da İzmir'de doğdu. İlk, orta ve lise eğitimini İzmir'de tamamladı. 2004 yılında 60.Yıl Anadolu Lisesi'nden mezun oldu. 2005 yılında başladığı Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü'nü 2009 yılında bitirdi. 2010 yılında Sakarya Üniversitesi Bilgisayar ve Bilişim Mühendisliği Bölümü'nde yüksek lisans eğitimine başladı. 2009 yılında Sakarya Büyükşehir Belediyesinde'sinde yazılım geliştirici olarak çalışmaya başladı. Halen Sakarya Büyükşehir Belediyesinde'sinde yazılım geliştirici olarak görev yapmaktadır.