

# Yerel Yönetimlerdeki Coğrafi Bilgi Sistemleri İçin Yeni Bir Yazılım Mimari Önerisi

<sup>1</sup>Can Alper Öztürk, <sup>\*2</sup>Musa Balta ve <sup>2</sup>Cemil Öz

<sup>1</sup>Sakarya Büyükşehir Belediyesi Yazılım Birimi Sakarya, Türkiye

<sup>\*2</sup>Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilgisayar Mühendisliği, Sakarya Üniversitesi, Türkiye

## Özet

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

**Anahtar Kelimeler:** Yazılım mimarileri, Yerel yönetimler, OGC, TUCBS, INSPIRE

## Abstract

Nowadays, Geographic Information Systems (GIS) are available in every field. It is also an important tool because of allowing the user to visualize their problems or data and also look at them from different perspective. Today many open/commercial GIS projects may need human resources and physical servers. Existing methods cause these projects to be completed in long period and also costly. Lack of expert staff in different branches may also be another problem for GIS. In order to reduce all these problems and risks, in this study, an open source and layered software architecture proposal including geographical database, software, map server, data collection and analysis in local governments is presented.

**Key words:** Software architectures, local governments, OGC, TUCBS, INSPIRE

## 1. Giriş

Günümüzde şehirlerdeki nüfus hızla artmaktadır. 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 sosyal ve teknik hizmet vermekle zorunlu olduğu konularda daha büyük iş yükü ile karşılaşmaktadırlar. Tüm bu hizmetlerin yönetiminin kolaylaşması, daha hızlı ve efektif çalışabilmesi için bilgi ve iletişim teknolojilerinden faydalanmaları zorunluluk haline gelmiştir. Hepsi kendi konularında uzmanlaşmış yerel yönetim birimlerinin verdiği hizmetlerin hedef kitlesi büyüdükçe yerel yönetim bilgi sistemlerinde oluşan verilerin boyutunun arttığı görülmektedir. Tüm bu veriler yerel yönetimlerin sınırları kapsamında

\*Corresponding author: Musa BALTA Adres: Bilgisayar ve Bilişim Bilimleri Fakültesi, Bilg. Müh.lığı, Sakarya Üniversitesi, 54187, Sakarya TÜRKİYE. E-mail adresi: mbalta@sakarya.edu.tr, Phone: +902642955646

yapılan çalışmalarla üretildiği için bir mekan/adres tanımı da içerebilmektedir. Bu konumsal verilerin depolanması, işlenmesi, anlaşılması için CBS'nin tüm yerel ve kamu kuruluşlarında kullanılması zorunlu hale gelmiştir [1].

CBS sistemleri yazılım, donanım, veri ve bunların yönetimi açısından çok alanlı bir çalışma konusudur [2]. Coğrafi bilgi sistemlerinin birçok çalışma alanından oluşması beraberinde farklı problemleri de açığa çıkarmaktadır. Özellikle yerel yönetimlerin karşılaştığı insan kaynağı ile ilgili problemlerin başında; kurumdaki yetişmiş personelin sayıca az olması, uzman personel bulma zorluğu ve yöneticilerin konusunda uzman olmaması gösterilebilir [3]. 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 [4]. Meta verinin belirlenmesi ve gerektiğinde ilgili diğer kurumlar ile hangi protokoller ile paylaşılacağı "Türkiye Ulusal Coğrafi Bilgi Sistemi (TUCBS)" olarak isimlendirilen "Ulusal Konumsal Veri Altyapısı" ile standardize edilme çalışmaları devletimiz tarafından devam ettirilmektedir. Bu çalışmalarda coğrafi portalın teknik mimarisi ve servis gereksinimleri "Open Geospatial Consortium (OGC)" ve Avrupa Birliği tarafından yürütülen "Avrupa Birliği Konumsal Veri Altyapısı (INSPIRE)" projesi kapsamında yapılan çalışmalarla belirlenmiştir [5].

Yerel yönetimlerin kurumlarında uygun bir CBS 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 CBS projelerini gerçekleştirebilmek için lisanslı ürünler kullandıkları görülmektedir. Lisanslı 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 olmaktadır. Tablo 1'de lisanslı ürünler ile geliştirilen CBS sistemleri ile bu çalışmada önerilen CBS mimarisinin karşılaştırma tablosu verilmiştir.

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. "OGC", sunucu ve istemci arasındaki coğrafi veri transferinin yapılabilmesi için "Web Feature Service (WFS)", "Web Map Service (WMS)", "Web Coverage Service (WCS)" gibi veri tanımlama standartlarını oluşturan kuruluştur [6,7]. Lisanslı ürün, teknik personel ve fiziksel ekipmandan yoksun yerel yönetimlerin, geliştirilen katmanlı ve açık kaynak kodlu CBS yazılım mimari önerisini kullanarak kendi CBS'lerini hayata geçirebilmeleri hedeflenmektedir. Birçok farklı açık kaynak kodlu yazılım (GeoServer, OpenLayers vb.), önerilen mimari sayesinde birbirleriyle bütünleşmiş bir şekilde çalıştırılarak, yerel yönetimler için gerekli CBS ihtiyaçları karşılamaktadır. Bu şekilde açık kaynak kodlu ve her biri kendi konusunda uzman yazılımlardan oluşan genel bir CBS motoru oluşturulmuştur.

Önerilen/gerçekleştirilen bu CBS motoru ile evrensel standartlara göre geliştirilmiş açık kaynak kodlu ürünler için ilgili kurumlara destek maliyeti getirmeden, birbirleri ile en uygun şekilde çalışabilecekleri yapılandırma ayarları ile sunulmuştur. Kurumların bu CBS motorunu kullanmak istemeleri halinde, birçok program ve uygun işletim sistemi kurulumu ve yapılandırma ayarları ile uğraşmamaları için "Konteynır" teknolojisinden faydalanılmıştır. Kurum tercihlerine, kullanım

amacı ve performans beklentisine göre bu CBS motorunun sunucu ölçeklemesini dinamik olarak yapmak mümkündür. Önerilen CBS mimarisi; tek bir sunucuda çalışabileceği gibi, her bir katman için ihtiyaca göre birden fazla sunucuda da çalışabilmektedir.

Tablo 1. Lisanslı Uygulamalar ile gerçekleştirilen çalışmanın kıyaslanması

	Lisanslı Uygulamalar	Gerçekleştirilen Çalışma
<b>Lisanslama</b>	Ücretlidir.	Açık kaynak kodludur.
<b>Meta Veri</b>	Ekstra ücretli uygulamalar kullanılarak yapılabilmektedir.	Otomatik olarak verinin künye bilgilerini arka planda saklamaktadır.
<b>Tematik Harita</b>	Ekstra ücretli uygulamalar kullanılarak yapılabilmektedir.	Tüm katman özneliklerine göre yapılabilmektedir.
<b>İnsan Kaynağı</b>	Birçok meslek dalından uzman personele ihtiyaç duyulmaktadır.	Uzman personel sayısını minimuma indirmektedir.
<b>Faaliyet/Proje</b>	Ekstra ücretli uygulamalar kullanılarak yapılabilmektedir.	Uygulama bu özelliklere sahiptir.
<b>Verilerin formatlanıp güvenlik bir şekilde servis edilmesi</b>	Uzman personellerce yapılabilmektedir.	Uzman olmayan personeller hızlıca yapılabilmektedir.
<b>Mobil Uygulama</b>	Ekstra ücretli uygulamalar kullanılarak yapılabilmektedir.	Mobil uygulamada web uygulamasının tüm yeteneklerine sahiptir.

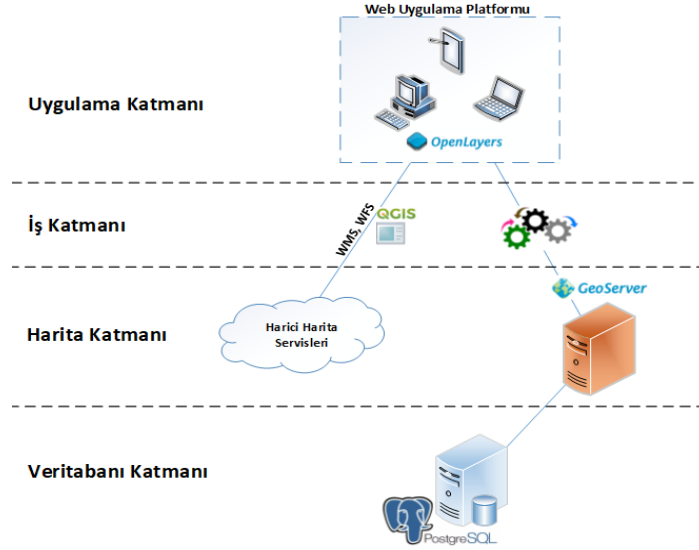
## 2. Önerilen Yöntem

Önerilen CBS motoru 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 görüntüleyebilir, işlemleri yapabilirler. Önerilen mimari ayrıca “OWASP TOP 10” güvenlik kriterleri göz önünde bulundurularak da geliştirilmiştir [8].

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 literatürdeki çalışmalardan farklı olarak, son kullanıcının veri tabanı ve harita servisi ile ilgili işlemlerden soyutlanarak, kendi 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 gibi işlemler 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 dâhilinde yeni kayıt ekleme, güncelleme, silme veya güncelliği geçmiş ama silinmesini istemediği verileri pasif ederek saklayabilmektedir. “Styled Layer Descriptor(SLD)” formatında katman stillerini ayarlayabilmektedir. “SLD” ile harita stil tanımlamaları yapılabilmektedir [9]. Coğrafi veri modelinin içerdiği verilere göre dinamik kurallar tanımlayıp tematik haritalar tanımlayabilmektedir. Tematik haritalar ile analiz konusunda yeni bakış açıları kazandırılmıştır [10]. Lisanslı ürünlerde karşılaşılan birçok CBS probleminden ve uzmanlığından soyutlanıp, CBS projelerinin yerel yönetimlerin diğer birimleriyle bütünleşmiş olarak çalışması kolaylaştırılmıştır. Bu çalışmada seçilen açık kaynak kodlu ürünlerin bir arada çalışabileceği katmanların birlikte çalışma yapıları Şekil 1’de gösterilmiştir. Şekilde gösterilen bu katmanlı yapıların hangi teknolojiler ile gerçekleştirildiği aşağıda hiyerarşik bir şekilde anlatılmaktadır.



Şekil 1. CBS için Açık kaynak ve katmanlı mimari yapısı

## 2.1. Veri Tabanı Katmanı

Veri tabanı katman tasarımları için literatürde iki farklı yöntem bulunmaktadır. Bunlardan ilki daha klasik ve yaygın olan ilişkisel veri tabanları, ikincisi ise “NOSQL” veri tabanlarıdır. Hızlı veri okuma ve yazma kabiliyetleri ile “NOSQL” veri tabanları birçok ticari uygulamada kullanılmaktadır [11]. Bu çalışmada hem ilişkisel hem de “NOSQL” yetenekleri ile ön plana çıkan ve açık kaynak kodlu “POSTGRESQL” [11] veri tabanı “POSTGIS” eklentisi ile birlikte kullanılmaktadır. “POSTGIS”; “POSTGRESQL” veri tabanına coğrafi özellikleri kazandıran açık kaynak kodlu bir eklentidir [13].

Çalışmada önerilen açık kaynak kodlu katmansal CBS mimarisinin veri tabanı yönetimsel ve icra olmak üzere iki kısım olarak tasarlanmıştır. Veri tabanının ilk kısmında (Şekil 2) uygulamanın

çalışabilmesi için kullanıcı, rol, yetki, mahalle, ilçe, organizasyon şeması, birim tabloları gibi yönetsel genel özellikler bulunmaktadır. Önerilen mimari modelindeki veri tabanının ikinci kısmında ise kullanıcı tarafından dinamik olarak tanımlanan ve kurumun coğrafi veri modelini içeren tablolar bulunmaktadır. Uygulamanın çalışabilmesi için veri tabanının ilk kısmındaki ön tanımlı olan tabloların veri tabanı sorgulama işlemleri için açık kaynak kodlu olarak geliştirilen “ENTITY FRAMEWORK CORE” [14] teknolojisi kullanılmıştır. İkinci kısımdaki tablolar, kullanıcı tarafından sonradan tanımlandıkları için “NPGSQL” kütüphanesi ile dinamik sorgu üreten bir kütüphane geliştirilmiştir [15]. Dinamik sorgu oluşturan bu kütüphanede “Sql” açıklarına karşı tüm sorgular parametrik olarak üretilmektedir. Tüm bu dinamik sorgular çalıştırılmadan önce kötü niyetli saldırılara karşı güvenlik kontrolleri de yapılmaktadır. Gerçekleştirilen CBS motoru içinde tanımlı ve bu sorguları çalıştıran veri tabanı kullanıcısı basit veri tabanı yetkilerine sahip ve tablo oluşturma, silme, veri tabanı kullanışı oluşturma, silme, yetkilendirme gibi yönetsel yetkileri bulunmamaktadır. Tüm dinamik sorgu işlemleri kaydedilmekte ve geliştirilen CBS motoru ekranlarından ulaşılabilir.

## **2.2. Harita Servisleri Katmanı**

Harita servisleri katmanında “Geoserver” uygulaması kullanılmıştır. “Geoserver” açık kaynak kodla geliştirilmekte olan “OGC” standartlarında “WFS/WMS” harita katman servislerinin oluşturabileceği, java programlama dili ile yazılmış bir uygulamadır [16]. “Geoserver” ile harita servislerini oluşturmanın yanında uygulamanın harita servis kullanıcı, rol ve yetkilendirme işlemleri de yönetilmektedir. Kullanıcıların direk “Geoserver” uygulamasına ihtiyaçları yoktur. “Geoserver” ile ilgili tüm işlemler; geliştirilen uygulama tarafından arka plan da “Geoserver”ın “Restful” servisleri kullanarak yapılmaktadır. “Restful” servisler Http protokolünün “get”, “post”, “put”, “delete” metodları ile oluşturma, okuma, güncelleme, silme işlemlerini yapmaktadır [17].

## **2.3. İş Katmanı**

Bu katman, 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 “Object/Relational Mapper (ORM)” kütüphanesinden faydalanarak, uygulama katmanının isteklerini cevaplamaktadır. Bu katman, diğer katmanlar üzerinde bir yönetici rolü üstlendiği için; solid, dry, yagni gibi birçok nesneye dayalı tasarım prensibinden faydalanılmıştır. Farklı tasarım mimarilerinin kullanılmasının amacı, önerilen CBS mimarisindeki farklı teknoloji ve platformların birbirleriyle entegre çalışabilmesidir. Çalışmada birçok katmanda kullanılan oturum kontrolü, yetki kontrolü, hata loglama ve cache kullanımı gibi ihtiyaçlar yine bu katmanda Aspect-oriented programming (AOP) ‘ten faydalanılarak tanımlanmış olup ihtiyaç duyulan her yerde AOP’ye uygun bir şekilde kullanılmıştır. Yazılan kodlar ve modüller arasındaki bağımlılıklar ise “Dependency injection” tasarım deseninden faydalanılarak yönetilmiştir. Gelecekte uygulamada ihtiyaç duyulabilecek yeni süreç ve iş kuralları eklenebilmesi için önerilen mimarinin katmanları esnek ve bağımsız bir şekilde tasarlanmıştır [18].

Bu katmanda çok fazla süreç ve iş kuralı yönetildiği için ayrıca SOLID prensiplerinden de faydalanılmıştır. Bu prensipler ile uygulamanın daha okunabilir ve evrensel bir mimariye sahip

olması hedeflenmiş, ayrıca değişime daha toleranslı bir uygulama ve yeniden kullanılabilirlik amaçlanmıştır [19]. “Factory” tasarım deseni ile ise uygulama içindeki nesnelerin üretimi için kurallar tanımlanmış ve çok biçimlilik desteklenmiştir [20]. Son olarak ise “Repository” Tasarım Deseniyle de kod tekrarlarından kaçınılmaya çalışılmıştır. Bu tasarım deseni ile veri tabanı işlemleri belli bir formatta ve kuralla yönetilmiştir.

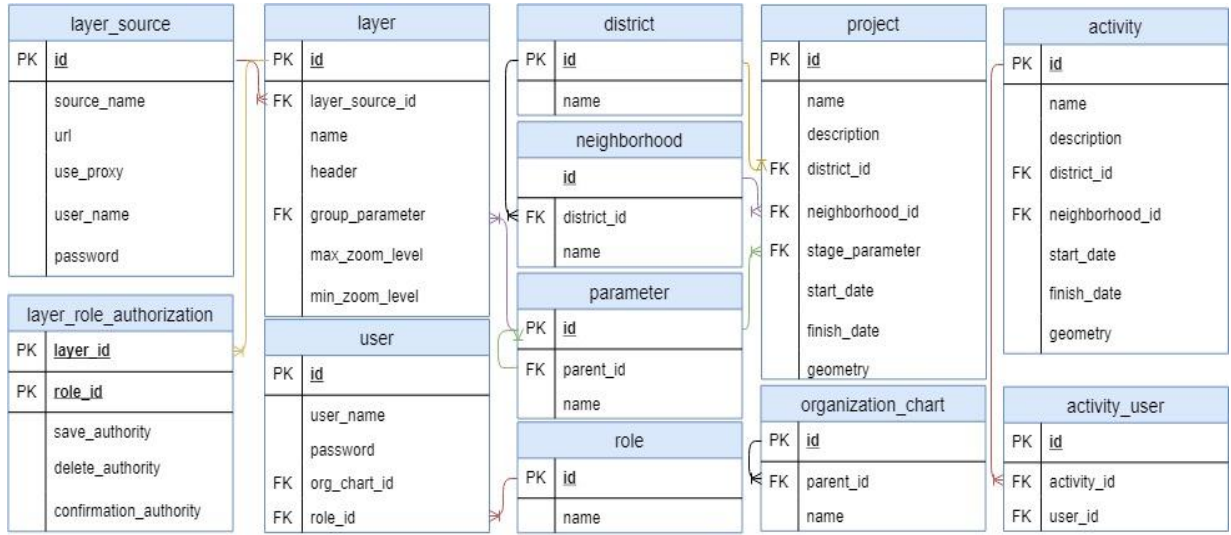
#### **2.4. Uygulama Katmanı**

Web/Mobil uygulama katmanı “Model-View-Controller” tasarım deseni kullanılarak geliştirilmiştir. Uygulama bu tasarım desenine göre parçalara ayrılarak geliştirilmiştir. Bu şekilde uygulamamın daha açık, anlaşılabilir ve evrensel formatlarda geliştirilebilmesi hedeflenmiştir [21]. Kullanıcılar tüm işlemlerini bu uygulamalar vasıtası ile yapabilmektedirler. Kullanıcılar yetkileri dâhilinde önerilen CBS mimarisinin ana ekranlarına ulaşabilmekte, kendi coğrafi veri modelini üretebilmektedirler. Yönetici ekranları ile de kullanıcı tanımı, rol tanımlama, kullanıcı yetkilendirme, organizasyon şeması yönetimi, faaliyet ve proje tanımlama ekranlarına ulaşabilmektedirler.

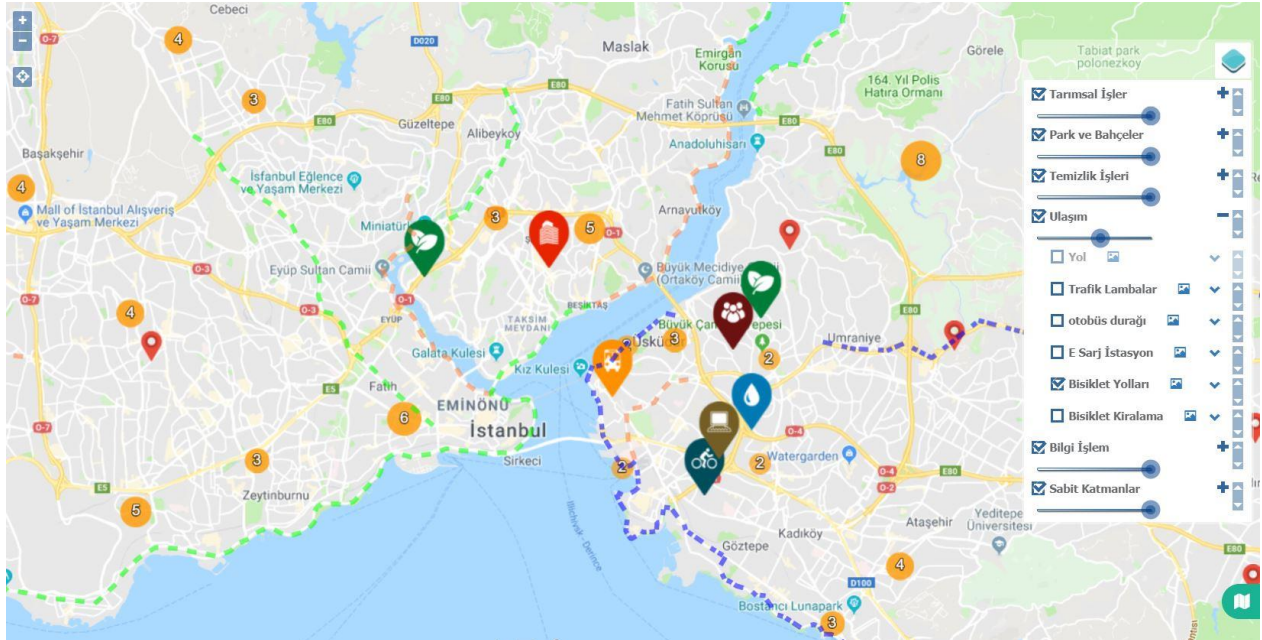
Önerilen CBS motorunun web uygulaması “Asp.NET Core” [22] teknolojisi kullanılarak geliştirilmiştir. “Bootstrap” kütüphanesi kullanılarak uygulamanın tüm güncel tarayıcı ve cihazlarda da uyumlu çalışması sağlanmıştır. “Ajax” teknolojisi kullanılarak daha hızlı ve daha az veri transferi yapan sayfalar geliştirilerek, kullanıcıların işlemlerini daha hızlı yapmaları amaçlanmıştır. Tüm harita işlemleri “Openlayers” javascript kütüphanesi kullanılarak yapılmıştır. Uygulama mekânsal olarak İstanbul ili, Üsküdar ilçesi temel alınarak geliştirilmiştir. Kullanıcı dinamik olarak tanımladığı tüm katman özniteliklerine göre filtreleme yapabilmektedir. Yerel yönetimdeki birimlerin gerçekleştirdiği faaliyetleri ve projeleri tanımlayıp, harita üzerinde gösterebilmektedir. Kullanıcı tarafından dinamik olarak tanımlanan katmanlar içinde faaliyet tanımlanabilmektedir.

Önerilen CBS motorunun mobil uygulama katmanı “Android” kütüphanesi kullanılarak “Eclipse” platformu üzerinde “Webview” [23] teknolojisi ile geliştirilmiştir. Web uygulamasının tüm özelliklerini içermektedir. Gösterge paneli ile yapılan çizimlerin, faaliyetlerin ve projelerin özet ve grafiksel raporlarına ulaşabilmektedirler. Kurum organizasyon şeması temelli filtreleme imkânı sunulmuştur.

Önerilen CBS motorunun web ve mobil ana ekran görüntüleri Şekil 3 ve Şekil 4’de gösterilmiştir. Hem web hem de mobil tasarım, günümüz ekran tasarım ilkelerine göre tasarlanmıştır. Günümüz web ve mobil uygulamalarını kullanmaya alışık kullanıcıların kolay uyum sağlayabilecekleri bir tasarım dili kullanılmıştır. Web ve mobil uygulamalarının her ikisinde de kullanıcının manuel olarak harita odaklama ve görüntü ayarı yapabilmesine olanak sağlanmıştır. Web ve mobil ana ekranından katmanların harita üzerinde gözükmemesinin yanı sıra katman verilerine liste görünümünde ulaşılabilir. Ayrıca donanım gücü açısından daha zayıf mobil cihazlarda, önerilen CBS motorunun daha hızlı çalışabilmesi için gerekli optimizasyon çalışmaları yapılmıştır.



Şekil 2. Önerilen CBS mimari-veri tabanının yönetsel kısmının varlık ilişki diyagramı



Şekil 3. Önerilen CBS katmanlı mimari uygulamasının ana sayfa görüntüsü



Şekil 4. Önerilen CBS katmanlı mimari uygulamasının mobil görüntüsü

### 3. Sonuç

Çalışmamızda; lisanslı uygulamaların paket halinde sunduğu coğrafi bilgi sistemi özelliklerini, açık kaynak kodlu muadil kütüphane ve programları bir araya getirerek sağlayan bir CBS motor uygulaması önerilmiş ve geliştirilmiştir. Yüksek lisans ücretlerini ödemek istemeyen yerel yönetimler önerilen/geliştirilen bu CBS uygulaması ile kendi coğrafi bilgi sistem modellerini kolayca oluşturabileceklerdir. Lisanslı ürünlerden farklı olarak birçok programla sunulan özellikler ve fonksiyonellik önerilen bu çalışmamızda tek bir uygulama çatısı altında 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 gibi bir çok özelliğin desteklendiği bu CBS motorunun geliştirme aşamasında birçok tasarım deseninden faydalanılarak; esnek, basit ve yeni ihtiyaçları daha kolay entegre etme hedeflenmiştir. Bu çalışma ile yerel yönetimlerin CBS konusundaki ihtiyaçları karşılanırken, kullanıcıları ve teknik personeli OGC, “Restful” web servisleri, “Inspire”, coğrafi veri tabanları, meta veri gibi bilgi sahibi olmaları gerektiği CBS konularından soyutlayarak, bu konularda onlara kılavuzluk etmek amaçlanmıştır.



## Referanslar

- [1] N. K. Sönmez and M. Sarı.” Princibles and applications of geographic information systems.” Derim 21/1 (2004): 54-68
- [2] Savaş, Ömer, Gıyasettin Özcan. An Open Source Geographical Information System Tool for Province Administration. DEUFMD 21/61 (2019): 113-123
- [3] Bensghir T K., A. Akay. Bir Kamu Politika Aracı Olarak Coğrafi Bilgi Uygulamalarının Değerlendirilmesi. Çağdaş Yerel Yönetimler 15/1 (2006): 31-46
- [4] Duval, Erik, Wayne Hodgins, Stuart Sutto and Stuart L. Weibel. ”Metadata Principles and Practicalities.” D-Lib Magazine 8/4 (2002)
- [5] Akıncı, Halil, Çetin Cömert. ”TNGIS AND INSPIRE TECHNICAL ARCHITECTURE.” 12. Türkiye Harita Bilimsel ve Teknik Kurultay’ında sunulan bildiri, TMMOB Harita ve Kadastro Mühendisleri Odası, Ankara, Mayıs 11-15, 2019.
- [6] Bonaccorsi, Andrea, Cristina Rossi.” Why Open Source software can succeed.” ELSEVIER 32/7 (2003): 1243-1258
- [7] Castronova, Anthony M., Jonathan L. Goodall, and Mostafa M. Elag.” Models as web services using the Open Geospatial Consortium(OGC) Web Processing Service (WPS) standard.” ELSEVIER 41 (2013): 72-83
- [8] OWASP. “The Open Web Application Security Project.” Last Updated 23 February, 2019. [https://www.owasp.org/index.php/Main\\_Page](https://www.owasp.org/index.php/Main_Page)
- [9] Zipf, Alexander. ” Using Styled Layer Descriptor (SLD) for the Dynamic Generation of User- and Context-Adaptive Mobile Maps – A Technical Framework” 5th International Workshop on Web and Wireless Geographical Information Systems (W2GIS), University of Applied Sciences FH Mainz, Lausanne, December 15-16, 2005
- [10] Yalçın, Cihan, Engin Bişar, Seda Yalçın and Levent Sabah.” Tematik Coğrafi Haritaların Verimlilik Açısından Değerlendirilmesi.” 5. Ulusal Verimlilik Kongresi’nde sunulan bildiri, Ankara, October 6-7, 2015.
- [11] Jing, Han, Haihong E, Guan Le, and Jian Du.” Survey on NoSQL database.” 6th International Conference on Pervasive Computing and Applications’nde sunulan bildiri, Port Elizabeth, October 26-28, 2011
- [12] Postgresql. “The World's Most Advanced Open Source Relational Database.” Last Updated 23 February, 2019. <https://www.postgresql.org/>
- [13] Obe, Regina, Leo Hsu. PostGIS in action. Stamford: Manning Publications, 2011
- [14] Entity Framework Core. “Open Source ORM Tool.” Last Updated 23 February, 2019. <https://docs.microsoft.com/tr-tr/ef/core/>
- [15] Npgsql. “.NET Access to PostgreSQL.” Last Updated 23 February, 2019. <https://www.npgsql.org/>
- [16] Iacovella, Stefano, Brian Youngblood. PostGIS in Action. Birmingham: Packt Publishing, 2013
- [17] Rodriguez, Alex. “RESTful Web services: The basics.” Last Updated 23 February, 2019 <https://www.ibm.com/developerworks/webservices/library/ws-restful/>
- [18] Gary McLean Hall, Adaptive Code: Agile coding with design patterns and SOLID principles. Redmond, Washington: Apress Publications, 2017.
- [19] Robert C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2018

- [20] Vaskaran Sarcar, Design Patterns in C#. Whitefield, Bagalore, Karnataka, India: Apress Publications, 2018.
- [21] Leff, Avraham, James T. Rayfield.” Web-Application Development Using the ModelNiewlController Design Pattern.” Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference ’ında sunulan bildiri, IBM T. J. Watson Research Center, Seattle, September 4-7, 2001
- [22] Adam Freeman, Pro ASP.NET Core MVC 2. London: Apress Publications, 2011.
- [23] Android. “Building web apps in WebView.” Last Updated 23 February, 2019 <https://developer.android.com/guide/webapps/webview>