**Gözlem İstasyonları Değerlendirmelerinin Yazılım ve Coğrafi Bilgi Sistemleri ile Buluşması**

**Begüm Günel1,\*, Kemal Seyrek2, Nesibe Gülşah Güreşçi3**

*1Yazılım Mühendisi, Devlet Su İşleri Genel Müdürlüğü, 61000, Ankara.*

*2Şehir ve Bölge Plancısı, Devlet Su İşleri Genel Müdürlüğü, 61000, Ankara.*

*3Harita Yüksek Mühendisi, Devlet Su İşleri Genel Müdürlüğü, 61000, Ankara.*

**Özet**

*Su kaynaklarının tükenme tehlikesine karşın su kaynakları yönetiminin teknoloji ile birleştirilerek en etkin şekilde sağlatılması gerekmektedir. Sayısal ortamlarda yapılan analizler ile doğru planlama yapılabilmekte ve kaynakların nesiller boyunca en optimum şekilde kullanılma imkanı sağlatılabilmektedir. Bu kapsamda, Coğrafi Bilgi Sistemleri (CBS) ile su kaynakları haritalar üzerinde gösterilebilmekte olup havzalar, akarsu ağı, sulama alanları, depolama tesisleri, pompa gibi su yapılarını konumsal olarak gösterilebilmektedir. Diğer taraftan baraj işletme programı kapsamında, depolama tesislerinin mevcut depolama hacmi ile planlanan durumu aynı anda görüntülenerek işletme açısından planlanan duruma hangi ölçüde uyulduğu anlık izlenebilmektedir ve planlama yeniden revize edilmektedir.*

*Gözlem İstasyonları Yönetim Sistemi web uygulaması yukarıda bahsedilen teknik gelişmeler ve gereksinimler doğrultusunda hazırlanmıştır. Bu uygulama ile rasat istasyonları; Akım, Göl, Kar, Meteoroloji ve Müteferrik Gözlem İstasyonları olarak sınıflandırılmış “Açık/Kapalı” durumlarına göre renklendirilerek sunumu güçlendirilmiştir. Tüm Gözlem İstasyonlarının “Değerlendirilmiş Yıllar” bilgisinin tablosal gösterimi yapılmış bu sayede hangi yıllarda istasyonların etkin kullanıldığı tespit edilmiştir. Akım Gözlem İstasyonlarının ” Uzun Yıllar Aylık Ortalama Debi” ölçümleri grafiksel gösterimle sunulmuş ayrıca “Sediment Bilgisi” verileri tablosal olarak gösterilmiştir. Göl Gözlem İstasyonlarının “Seviye Grafik” verileri ile gözlem süresi boyunca aylık su seviyelerinin grafiksel gösterimi yapılmıştır. Gözlem İstasyonları Yönetim Sistemi ile kurumumuzda tutulan verileri diğer kurumlar ve vatandaşlar ile paylaşımı sağlanmıştır. Rasat istasyonlarının verileri veri tabanımıza gerek çevrimiçi, gerek çevrimdışı yöntemler ile veri tabanımıza belirlediğimiz formatta giriş yapmakta bu aşamadan sonra hazırlanan RESTful Web Servisler ile gerek konumsal veriler gerekse de ölçüm verileri web uygulamasına aktarılmaktadır. RESTful olarak hazırlanıp yayımlanan web servisi ise JavaScript, HTML5, jQuery ve OpenLayers gibi değişik yazılım kütüphaneleri kullanılarak bir WEB uygulaması hazırlanmıştır.*

*Sonuç olarak Gözlem İstasyonları Yönetim Sistemi uygulaması ile mevcut akarsu debilerinin aylık ortalaması CBS-WEB ortamına aktarılmış ve tüm rasat verilerinin tek platformdan incelenebilir olması sağlanmıştır. Bu uygulama ile su kaynakları yönetiminde en güncel teknolojilerin aktif olarak kullanılması hedeflenmiştir.*

Anahtar Sözcükler

Rasatlar, REST Web Servisleri, Debi, JavaScript, jQuery

**1. Giriş**

İnsanın doğası gereği taşıdığı merak duygusu teknoloji sayesinde giderebilmekte ve hayatı kolaylaştıran yenilikleri yine bu merak duygusu sayesinde geliştirebilmektedir. Önceleri sadece askeri uygulamalar ile bilimsel araştırmalarda hızlı bilgi paylaşımı için tasarlanan internet ağı, günümüzde kişilere veya değişik kurumlara hitap eden, günlük yaşantımızı kolaylaştıran değişik uygulamalar sayesinde hayatın vazgeçilmez bir parçası haline gelmiştir. İnternet ile tüm dünya elimizin altında olup zaman ve mekân farkı olmaksızın bilgiye ulaşım ve bilgiyi güncelleme imkânına sahip bulunulmaktadır. Web uygulamaları sayesinde bu nitelikler her geçen gün geliştirebilmekte ve kazanılan kolaylıklar bir adım daha ileriye götürebilmektedir.

“Gözlem İstasyonları Web Uygulaması”, DSİ Genel Müdürlüğü tarafından bu amaçlar ışığında tasarlanarak hayata geçirilmiş bir projedir. Coğrafi Bilgi Sistemleri (CBS) ortamında Genel Müdürlüğümüz tarafından işletilmekte olan tüm gözlem istasyonlarına ait verileri ilişkisel bir veritabanı yönetim sisteminde (RDBMS) toplanarak, hazırlanan web veya masaüstü uygulamaları ile farklı platformlarda çalışacak şekilde gerek kurum içinde gerekse de kurum dışındaki kullanıcılara farklı detaylarda sunulabilmektedir. CBS’ nin günümüzde artan kullanım alanları sayesinde son kullanıcıya yönelik mekânsal veriyi saklama, yönetme ve sorgulama fonksiyonları efektif bir şekilde sunulabilmektedir. “Gözlem İstasyonları Web Uygulaması” ile amaçlanan kurumsal veritabanın da tutulan verilere son kullanıcı tarafından etkin bir şekilde erişilebilmesi ve değişik sorguların yapılarak sorgu sonucu elde edilen verilerin hızlı bir şekilde sunulmasının sağlanılmasıdır. Ayrıca, hazırlanan uygulama ile uzun yıllardır değişik ortamlarda saklanılan gözlem verileri, çeşitli sorgulama ve analiz yöntemleri kullanılarak son kullanıcıya iletilebilmekte, web üzerinde çalışması kullanıcıya zaman ve mekân farkı olmadan erişim sağlayabilmektedir.

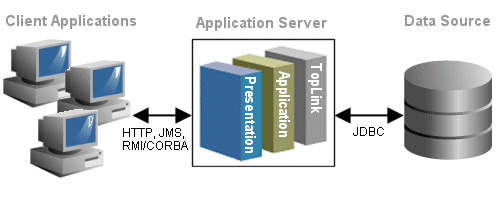
“Gözlem İstasyonları Web Uygulaması” na ait tasarım, yazılım mimarisi seçimi ve uygulamanın kaynak kodlarının yazılması sırasında mümkün olduğunca güncel yazılım mimarisi göz önüne alınmış olup çok katmanlı servis mimarisi kullanılmış olup kullanılan yazılım katmanların büyük bir kısmı kurum içinde hazırlanarak uygulama geliştirme aşamasında kullanılmıştır. Uygulama geliştirme sırasında iki ana veri kaynağından veriler elde edilmiştir. Bu veri kaynaklarından ilki daha önceki yıllarda kullanıma alınmış olan “Devlet Su İşleri Genel Müdürlüğü Su Veri Tabanı (DSİ-SVT)” dır. “DSİ-SVT” de depolanmakta olan istasyon ölçüm verilerini uygulamaya da kullanabilmek için REST (Representational State Transfer) Web Servis teknolojisinden faydalanılmıştır. İkinci olarak ise kurumumuzda günümüze kadar toplanmış olan verilerin saklanıldığı DSİ-CBS Veritabanıdır. Hâlihazırda, Genel Müdürlüğümüz tarafından yürütülmekte olan baraj, gölet, sulama, içme suyu, taşkın vb... tüm faaliyetlere ait mekânsal veriler DSİ-CBS veritabanı olarak “Oracle İlişkisel Veritabanı Yönetim Sisteminde” saklanılmaktadır. Söz konusu CBS veritabanından elde edilen mekânsal verilerde ayrıca Genel Müdürlüğümüz tarafından hazırlanmış olan RESTful Web Servisi aracılığıyla “Gözlem İstasyonları Web Uygulaması” içinde kullanımı sağlatılmıştır. Sunum katmanının yazılımı aşamasında ise, Rest Web servisleri ile uygulamaya taşınan veriler “JavaScript”, “HTML5”, “jQuery” ve “Openlayers” gibi yazılım kütüphaneleri kullanılarak hazırlanan web uygulaması ile esnek ve kolay kullanılabilir bir şekilde sunumu gerçekleştirilmiştir.

Bu uygulama ile istasyon verilerinin yönetimi efektif bir şekilde gerçekleştirilmesi hedeflenmektedir.

**2. Yazılım Mimarisi ve Uygulamada Kullanılan Teknolojiler**

Bir yazılım projesinde ki ihtiyaçların tanımlanması ve öngörülen problemlerin analiz edilmesi yazılım mimarisi sürecidir. Geliştirilen ve geliştirilmekte olan uygulamalar zamanla tek başlarına çalışmaktansa ihtiyaçlar doğrultusunda birbirleri ile etkileşim haline girmektedirler ve bu etkileşim ile uygulamalar hazırlanırken bazı temel prensipler ışığında geliştirilmesi zamanla zorunlu hale gelmiştir. Bu süreçte temel karar verme objeleri kalite, performans, sürdürülebilirlik ve kullanıcı etkileşimidir. Yazılım mimarisini önemi projelerin fonksiyonelliklerinin artmasıyla daha da önem kazanmıştır. Her yeni fonksiyonellik projenin zamanla karmaşık yapısını arttırmaktadır. Bu karmaşıklığın sonucu ile yönetilebilirlik azaldığı için proje sahiplerine zaman ve maliyet acısından zarar vermektedir. Zararların önlenebilmesi açısından çok katmanlı mimari yaklaşımı geliştirilmiştir. Çok katmanlı mimari yaklaşımı ile yazılım projeleri fonksiyonelliklerine göre sınıflar halinde mantıksal katmanlara ayrılmaktadır. İhtiyaç dâhilinde söz konusu uygulama fiziksel olarakta katmanlara ayrılmakta ve veri katmanı, iş katmanı ile sunum katmanı farklı sunucularda çalıştırılabilmektedir. Böylece proje basit bileşenlere ayrılmış ve doğru zaman ve doğru yerde koda müdahale edilebilirlik şansı arttırılmıştır.

Web uygulamaları genel olarak istemci-sunucu mimari yaklaşımı ile hazırlanmaktadır. İstemcinin talep ettiği bilgiyi sunucunun sağlaması prensibine dayalı yaklaşım ile hazırlanan web uygulamaları bileşenlerine ayrılmaktadır. İki katmanlı mimari de istemci ve sunucu arasında kurulan bağ sayesinde verilerin farklı uygulamalar tarafından erişime açık olması sağlanmaktadır. Ancak, ağ üzerinde oluşan yoğun trafikten dolayı zaman içerisinde isteklerin hızlı olarak cevap alamamasına neden olmaktadır. Bu sorunu önlemek adına veri katmanı oluşturularak veri tabanı ile ilgili tüm işlemleri bu katman sahiplenir. Üç katmanlı mimari yapı ile performans sorunları çözülmesinin yanı sıra daha güvenilir bir sistem elde edilir. Sunucu tarafında kullanılan ve web uygulamaları için vazgeçilmez olan web sunucuları projenin ara yüz bileşenlerini ve kullanılan programlama dilinde kaynak kodunu kapsamaktadır. İstemci, web uygulamasından ilk isteğini yaptığında dolayısıyla web sunucusu ile karşılaşır. Web uygulamasından istenilen veriler içinse veritabanı sunucusuna bağlanılır. Veritabanı sunucusunda isteğe bağlı olarak sorgulama, ekleme, güncelleme ve silme işlemlerine cevap verir. Son yıllarda veri katmanları arasındaki bağlantının hızlı ve güvenilir bir şekilde yönetimi amacıyla web servislerinden yoğun bir şekilde faydalanılmaya başlanılmıştır.



*Şekil 1: 3 Katmanlı Mimari Yapı*

Web servislerin birincil amaçları farklı katmanlar arasında uyum sağlamak olduğu gibi iletişimi sağlamak içinde web servislerine başvurulmaktadır. Web servisleri kullanılarak uygulama yazılımından bağımsız bir şekilde tüm projenin yönetimi kolaylaştırılarak, aynı verilere ihtiyaç duyan diğer uygulamalar ile de iletişim gerçekleştirilebilecektir. Ayrıca iletişim istemci-sunucu mimari yapı düzeyine indirgenmiş olacaktır. Bu özelliklerin yanı sıra proje bir grup tarafından hazırlanıyorsa kişiler arasında görev dağılımı ayrımı yapılabilecek ve platform bağımsız çalışmalar gerçekleşecektir. Elde edilen kazanç kişilerin farklı programlama dillerini uygulamada bir uyum içerisinde kullanabilecek olmasıdır.

Gözlem İstasyonları Web Uygulaması hazırlanma aşamasında üç katmanlı mimari yapı yaklaşımından faydalanarak aslında istenilen istasyon analiz sonuçlarına ulaşmada performans artışını gerçekleştirmiştir. Hazırlanan web servisleri ile veriye kolaylıkla erişimi sağlanmakta ayrıca güvenlik acısından büyük bir önlem alınabilmektedir. Mekânsal veritabanı olarak “PostGreSQL/PostGIS” kullanılmış olup veritabanı ile iletişime geçen web servisler ise “.Net Framework” teknolojisi ile hazırlanmıştır. Uygulama ara yüz ve fonksiyonelliği için javaScript programlama dilinin esnek yapısı kullanılmıştır. Ayrıca jQuery ve OpenLayers kütüphaneleri de uygulamaya esneklik katmıştır.

**2.1 Mekânsal Veritabanı**

Günümüzde gerek ticari olsun gerekse de açık kaynak olsun hemen hemen tüm ilişkisel veritabanı yönetim sistemleri (RDBMS) mekânsal verileri saklayacak ve yönetimi gerçekleştirecek araçlara sahip bulunmaktadır. Oracle, Microsoft SQL Server, PostGreSQL, MySQL, gibi büyük ölçekli kurumsal veritabanı sistemleri ile SQLlite gibi dosya tabanlı veritabanı sistemleri mekânsal verilerin saklanılması ve değişik SQL sorgulamalarının yapılabilmesine imkân veren araçlara sahip bulunmaktadır. Söz konusu veritabanı sistemleri tamamen “Open GeoSpatial Consortium-OGC” tarafından belirlenen standartlara uygun olarak çalışmakta ve OGC tarafından tanımlanan kurallar çerçevesinde mekânsal verilerde klasik diğer verilere benzer şekilde ilişkisel olarak saklanabilmekte ve sorgulanarak raporlanabilmektedir.

RDBMS veri tabanları, nümerik ve karakter veri tiplerini barındırırken, son yıllarda eklenen mekânsal (Spatial) araçlar ile söz konusu veri tabanları klasik veri tiplerine ek olarak mekânsal veri tiplerini de barındırmaktadır. Mekânsal veri tipleri nokta, çizgi ve poligon olarak veritabanında tutulmaktadır. Bu projede, ölçüm verileri Oracle veritabanı içerisinde yönetilmek de, mekânsal veriler ise PostGreSQL veritabanı içerisinde yönetimi yapılmaktadır.

PostGreSQL veritabanları için ilişkisel modeli kullanan ve SQL standart sorgu dilini destekleyen açık kaynaklı bir veritabanı yönetim sistemi olup günümüzde birçok CBS yazılımı tarafından desteklenmekte ve veritabanı kaynağı olarak “PostGIS” aracılığıyla kullanılabilmektedir. “PostGIS” ise “PostGreSQL” veritabanı içerisinde mekânsal verileri OGC standartlarına uygun bir şekilde yönetmek üzere geliştirilmiş bir eklenti olarak çalışan veritabanı aracıdır. Temel fonksiyonlara ek olarak topoloji ve raster desteği de PostGIS 2.0 versiyonu ile birlikte gelmiş bulunmaktadır.

PostGreSQL Veritabanına erişmek için sözkonusu RDBMS’nin işleyiş mekanizmasından bağımsız olarak veritabanına dışarıdan bağlanacak uygulamalar için bir ara katman ile uygulama arayüzleri ayrıca bulunmaktadır.

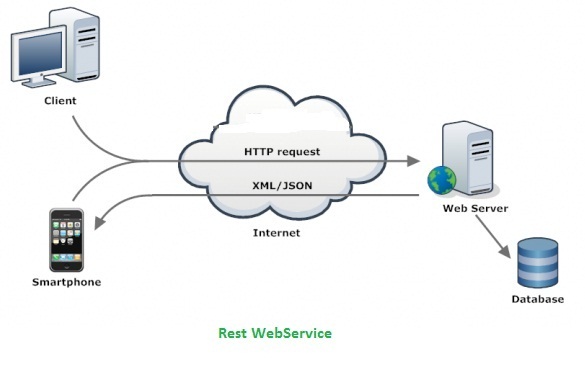
**2.2 Web Servisler ve REST Web Servisleri**

Son yıllarda yaşadığımız teknoloji, ilerde elde edebileceklerimizin öngörüsünü kazandırmış olup internet çağı hızlı iletişimi, bilgiye hızlı erişimi sağlamıştır. Ancak teknolojik ilerlemenim hayatımıza kattığı sayısız yenilik ve olumlu değişikliklerle birlikte bu hızlı gelişim bir takım sorunları da beraberinde getirmiş ve daha önceden karşılaşılmayan sorunlarda ortaya çıkmaya başlamıştır. Bilişim dünyasındaki bu sorunların en mühim olanı kuşkusuz güvenlik açıklarıdır. Kişisel ya da kurumsal bilgiler sadece ilgili kişiler tarafından erişilebilir ve kullanılabilir nitelikte olan bilgilerdir. Ancak yaşanılan değişik güvenlik açıkları nedeniyle gerek birey bazında gerekse de kurumsal anlamda gizli kalması gereken birçok veri kötü amaçlı olarak ele geçirilmiş ve art niyetli olarak söz konusu bireylere zarar verecek mahiyette kullanılmıştır. Bu sorunun önüne geçilebilmek için geliştirilen en önemli adımlardan biri Web Servilerinin kullanılmaya başlanmasıdır. Daha önceki yıllarda sadece tek bir sistemde hayata geçirilen uygulamalar amaçlarına ve fonksiyonelliklerine uygun olarak değişik bileşenlere ayrılarak farklı sistemler arasında bütünleşmiş bir şekilde yönetilmeye başlanmıştır ve bu sistemler arasında birbirleri ile iletişime geçecek başarılı bir bağ yapısı kurulmuştur. Böylece, günümüzde en çok tercih edilen web uygulamalarında güvenlik açıklarının önlemi alınmış bulunmaktadır. Tanımlanan değişik prosedürler sayesinde veritabanında tutulan bilgilere sadece ilgili kişilerce ulaşılması sağlanmış ve geliştirilen çeşitli standartlar sayesinde dış güçler tarafında sisteme erişildiğinde güvenlik duvarları veriye ulaşılmasını engelleyecek bir altyapı kurulmuştur.

Web Servislerinin temel yapısı “HTTP” protokolü üzerinde olmaktadır. “HTTP” protokolü sayesinde tüm “HTTP” protokolü üzerinde bulunan uygulamaların servis ile iletişime geçebileceği anlamına gelmektedir. Platform bağımsızlığı ile yazılımcıların uzman olduğu programlama dillerini kullanmasını sağlamıştır. Örneğin uygulama tarafında çalışan yazılımcı uygulamayı “JAVA” dili ile hazırlarken ve “UNIX” platform üzerinde çalışırken iletişime geçeceği web servis “.Net” ile geliştirilerek Windows işletim sistemi üzerinde çalışabilmektedir. Bu iki farklı sistem ise “Extensible Markup Language-XML” iletişim standartları ile birbirleri ile uyum içerisinde çalışabilmektedir.

Web servislerinin en büyük avantajı “XML” tabanlı olmaları olup “XML” aslında bir meta dildir. Bu meta dil ile yazılımcının kendi kurallarını koyması sağlatılmakta ayrıca bu kurallara göre hiyerarşik yapı oluşturabilmektedir. Söz konusu bu hiyerarşik yapı yazılımcı tarafından istenildiği zaman değiştirebilecek esnek bir yapıdır. “XML” ayrıca uluslararası bir format olup text tabanlı olduğu için güvenlik duvarlarına takılmaz. Web servis fonksiyonelleri yazıldıktan sonra uygulama projesine referans olarak eklenilmektedir. XML web servisleri çeşitli protokoller üzerinden tüketilmektedir. Bu protokoller SOAP (Simple Object Access Protocol), WSDL (Web Services Description Languagage) ve UDDI (Universal Description, Discovery and Integration) olarak adlandırılır.

“XML” Web Servislerinin yanı sıra REST (Representional State Transfer) Web Servisleri de modern web mimarilerinde yaygın bir şekilde kullanılmaktadır. REST web servisleri “HTTP” protokolü yanı sıra WWW (World Wide Web) sisteminde de kullanılmaktadır. REST servislerine ihtiyaç duyulma sebebi istemci-sunucu arasında ki alışverişin SOAP, RPC gibi karmaşık mimariler ile sağlamak yerine direk “HTTP” protokolü üzerinden alışverişi tamamlamaktır. Ayrıca SOAP tabanlı web servislerin yazımı sırasında çıkan güçlüklerle ve keskin standartlarla karşılaşılmamaktadır. Bir diğer önemli özellik ise Proxy generatorlere de ihtiyaç duyulmamaktadır. ”REST” servisleri “HTTP” metotları ile işlerini tamamlar. Bu metotlar veri üzerinde güncelleme, ekleme, silme ve sorgulamadır. REST web servislerinin bir çok cevap verme yöntemi bulunmaktadır. Bunlardan en popüleri olarak “JavaScript Object Notation -JSON” kullanılmaktadır. REST web servisleri kullanımı sayesinde SOAP web servislerinden farklı olarak bant genişliği ile XML servislerin kullandığı veri boyutu küçülmekte ve ek olarak bir ara yazılıma ihtiyacı bulunmamaktadır.



*Şekil 2: REST Web Servis Yapısı*

Gözlem istasyonları web uygulamasının tükettiği web servisler REST mimarisi ile tasarlanmıştır. REST web servisleri seçilmesinin nedeni, SOAP Servis yapısına göre kullanımının esnek olması, multi-platform olarak kullanılması ve ilave kütüphanelere ihtiyaç duyulmamasıdır. REST web servis yapısının hazırlanma aşaması aşağıdaki başlıkta detaylı olarak açıklanmaktadır.

**2.3. Uygulamada Sunum Katmanı Geliştirilmesi için Kullanılan Kütüphaneler**

Web uygulamalarında dinamik harita ile çalışmayı kolaylaştıran gerek ticari gerekse de açık kaynak kodlu birçok kütüphane bulunmaktadır ve her geçen gün söz konusu bu kütüphanelerin hem sayısı hem de içeriği artmakta olup daha yaygın bir şekilde kullanılmaktadır. Konumsal verilerin sunulması ve işlenmesi için büyük kolaylık sağlayan bu kütüphaneler sayesinde gerek masaüstü gerekse de web tabanlı uygulamaların geliştirilmesi daha kolay bir hale gelmiş bulunmakta olup CBS birçok alanda kullanılabilir bir forma dönüştürmüştür.

Google, Microsoft, Yahoo aracılığı ile sunulan haritalar, birçok uygulamanın temel ihtiyaçlarına cevap verebilmektedir. Fakat bu uygulamaların internet bağlantısına, harita sunucularına erişim konusunda bağımlılıkları, lisans konusunda yükümlükleri bulunmaktadır. Diğer taraftan ihtiyaç duyulan bileşenleri kullanmak üzere geliştirilmiş değişik açık kaynak kütüphaneleri de bulunmaktadır. Bu çalışmada, açık kaynak olarak erişimi mümkün olan OpenLayers javaScript Kütüphanesi kullanılmaktadır.

**2.3.1 OpenLayers Kütüphanesi**

OpenLayers javaScript tabanlı açık kaynak kodlu bir kütüphane olup harita verilerinin web tarayıcılarında sunulmasını sağlamaktadır. Ayrıca harita verilerinin gösterimini sağlarken herhangi bir ilave sunucu desteğine de ihtiyaç duyulmamaktadır. OpenLayers sahip olduğu farklı “Application Programming Interface-API” destekleri ile zengin coğrafik harita uygulamaları geliştirmemize olanak sağlamaktadır. “Google Maps API”, “Bing Map API” ve “MSN Virtual Earth API” gibi web tabanlı coğrafi uygulamalar için bir “Javascript API” çalıştırır. OpenLayers, harita sağlayıcısı olarak “OpenStreetMap” haritalarına da erişebilmekte ve sunulmaktadır. OpenStreetMap ile haritaların hem uydu görüntüsünü hem de sokak görünümünü alabilmekteyiz. Bunun yanı sıra Google, Bing, Yahoo ve kendi hazırladığınız haritaları da desteklemektedir. “Harita” ve “katman” yapısı, “OpenLayers” için temel kavramlardır.

**2.3.2 JQuery Kütüphanesi ve JQuery UI**

JavaScript ile uygulama geliştirirken hem ara yüz hem de kullanım alanları bakımından JQuery Kütüphanesi büyük kolaylık sağlamaktadır. JavaScript kullanımını hızlandırmak ve kolaylaştırmak amacıyla kullanılmaktadır. JQuery kütüphanesi açık kaynak olup sürekli güncellenmektedir. HTML içerisinde animasyon yapımını kolaylaştırdığı gibi nesne seçimlerini ve tepkimelerini hızlandırır. Uygulamalarda yer alan animasyonlar için Flash alternatifi olarak kullanılmaktadır. Geliştirilmiş kütüphaneler ve fonksiyonlar jQuery UI adı altında toplanmıştır. Kullanışlı ara yüz örnekleri sayesinde slide ve hover efektlerini, resim galerilerini ve bir uygulamada olabilecek tüm animasyonları kapsamaktadır.

**3. “Gözlem İstasyonları Web Uygulamasının” Geliştirilmesi**

Yukarıda “Gözlem İstasyonları Web Uygulamasının geliştirilmesinde kullanılan temel yazılım mimari yapısı ile kullanılan yazılım kütüphaneleri hakkında kısa bir bilgilendirme yapılmış olup bu bölümde web tabanlı olarak geliştirilen “Gözlem istasyonları web uygulamasının” hazırlanışı hakkında bilgi verilecektir.

**3.1. Uygulamada kullanılacak REST Web Servislerinin Hazırlanması**

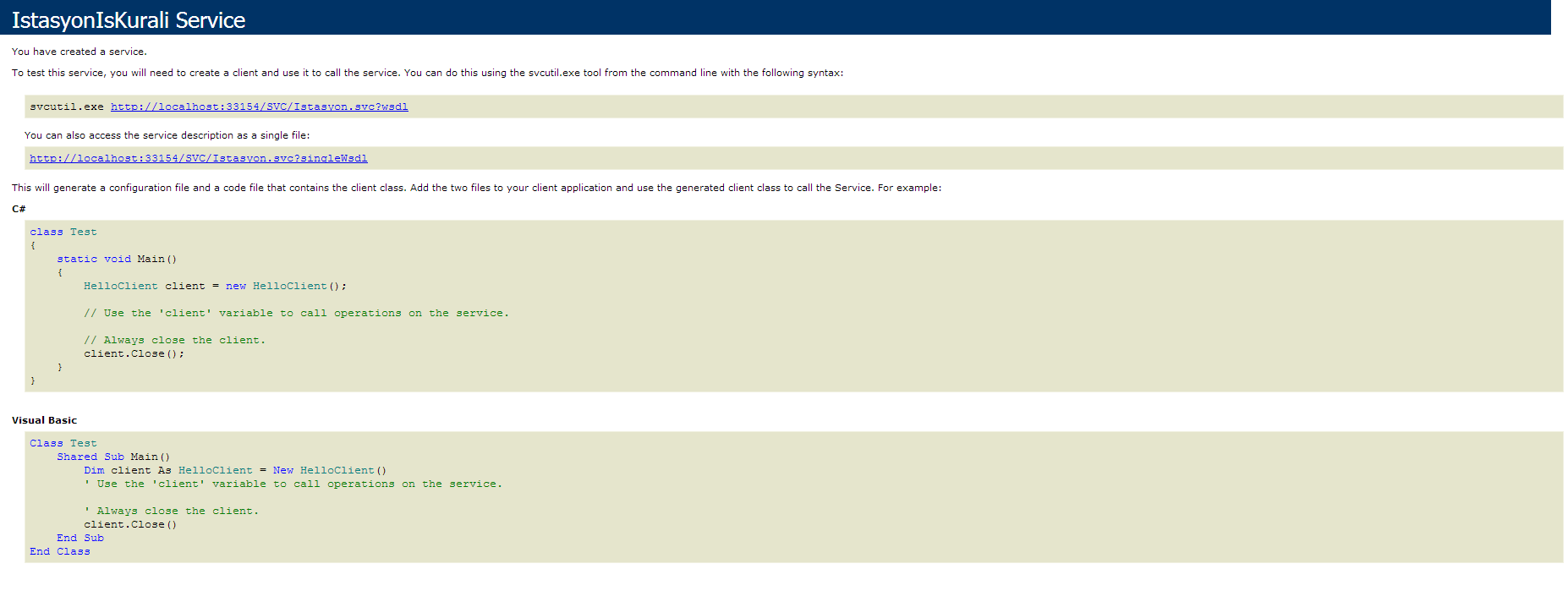
DSI veri tabanında tutulan verilere gerek kurum içi gerek kurum dışı uygulamalarda ihtiyaç duyulmakta ve farklı uygulamalarda kullanılmaktadır. Uygulama çeşitliliği arttıkça ve bu uygulamaların dış IP’lere erişimi açıldıkça veri paylaşımının hem performans hem de güvenlik açısından bazı standartlar çerçevesinde yer alması gerekmektedir. Bu kapsamda yapılan çalışmaların esnek bir yapıda olması ve çeşitli platformlarda çalışan uygulamalara kolaylıkla adapte olması için REST web servis mimarisinin kullanılması tercih edilmiştir. Bir diğer önemli tercih sebebi de JSON response tipinin kullanılmasıdır. JSON tipinin veri alışverişi yaparken XML’e göre daha küçük boyutlar ile veri alışverişi yapabilmektedir. Ayrıca jQuery’de olduğu üzere değişik JavaScript kütüphaneleri ile kullanımda çok basittir. Bu çalışmada veritabanında saklanan verilerin sunulması amacıyla Microsoft Visual Studio kullanılarak REST Web Servisleri hazırlanmış olup WCF teknolojisi kullanılmıştır. Aşağıda, Genel Müdürlüğümüz tarafından hazırlanarak geliştirilen değişik uygulamalarda kullanılmakta olan REST Web Servisi örnek olarak verilmektedir. Söz konusu web servisi ile “Gözlem İstasyonları Web Uygulaması” uygulamasında kullanılmakta olan “İstasyon” katmanına ait verilerin PostGreSQL/PostGIS veritabanında okunarak REST Web servisi olarak yayımlanması tablo-1’ de gösterilmektedir.

*Tablo 1: REST Web Servisi Kod Bloğu*

|  |  |
| --- | --- |
| İstasyonArabirim.cs | [OperationContract]  [WebGet(BodyStyle = WebMessageBodyStyle.Bare,  UriTemplate = "Istasyon/IstasyonGenelBilgi/identify?",  RequestFormat = WebMessageFormat.Json,  ResponseFormat = WebMessageFormat.Json)]  Stream IstasyonGenelBilgi(); |
| İstasyonIsKurali.cs | public Stream IstasyonGenelBilgiData()  {  string sSQL= " Select ID,ISTASYON\_ADI,st\_asgeojson(st\_simplify(geom, 500))  as GEOM from ISTASYON " ;  SqlDataAdapter cmd = new SqlDataAdapter (sSQL,baglanti);  DataSet ds = new DataSet();  cmd.Fill(ds);  return islem.GeoJsonFormat(ds);  } |
| web.config | <system.serviceModel>  <services>  <service name="IsKurali.Istasyon.IstasyonIsKurali" behaviorConfiguration="GISBehavior">  <endpoint name="IstasyonGISEndPoint"  address=""  binding="webHttpBinding"  contract="AraBirim.Istasyon.IstasyonAraBirim">  </endpoint>  </service>  </services>  </system.serviceModel> |

“Gözlem İstasyonları Web Uygulaması” için hazırlanan REST Web Servisinde, “İstasyonArabirim.cs” ile arabirim oluşturulmaktadır. “OperationContract” ile “Request” ve “Response” formatlar sisteme tanıtılmaktadır. Mesajlaşma formatı ise “JSON” olarak tanıtılmıştır. “İstasyonIsKurali.cs “ ile iş kuralı tanımlaması yapılmış olup SQL sorgu ile alınan verinin dataset’e aktarımı sağlanmıştır. Alınan veriler son olarak return işleminde GeoJson formatına dönüştürülmüştür. Uygulamamızda “IstasyonArabirim.cs” arabirimi ile “IstasyonIsKurali.cs” iş kuralını tanımladıktan sonra web servisi yayınlamak için yapılacak ayarları “web.config” dosyasında yapılmıştır. “web.config” dosyasında yukarıdaki düzenlemeleri yaptıktan sonra istemciden web servise ulaşarak veri sağlayıcı olarak kullanabiliriz.

İstasyon verilerinin REST servis aracılığı ile ilgili yukarıda verilen örnekle veritabanında tutulan mekânsal ve öznitelik verilerini kolaylıkla ulaşılabilmek ve yayınlayabilmektedir. Veritabanındaki tabloya “HTTP” protokolü üzerinden “GET”, “POST”, “PUT” veya “DELETE” response tiplerini kullanarak tablo üzerinde güncelleme, ekleme, düzeltme ve silme işlemlerinin yapılması da mümkün bulunmaktadır. İstasyon iş kuralı servis kullanımı resim-3 de gösterilmektedir.



*Şekil 3: IstasyonIsKurali Servis Kullanımı*

Sonuç olarak “Spatial” formatında bulunan verilere hazırladığımız REST Web servisleri ile ulaşarak hem kendi uygulamamızda kullanılmak üzere hem de bu verilere ihtiyaç duyulan diğer uygulamalarda kullanılmak üzere herhangi bir farklı araç kullanmadan yayınlanabilmesi mümkün bulunmaktadır.

**3.2. Uygulamada Sunum Katmanının Geliştirilmesi**

JavaScript, “HTML” kod bloklarının içine “**<script>...</script>**”etiketleri arasında kalan kod bloğu ile bildirilir. Esnek yapısı sayesinde HTML’ in zayıf kaldığı alanlarda devreye girer. JavaScript bir olay halinde devreye girer yani web sayfalarına bir iş yaptırma ve durumlara duyarlı hale getirme imkânı sunar.

Bu projede genel işleyişi sağlamak amacıyla kullanılan metotlar “javaScript” scriptleri ile hayata geçirilmiştir. İstasyonların koordinat sistemlerine göre harita üzerinde yerleştirmekten detay bilgisine kadar tüm metotlar “JavaScript” ile tetiklenmiştir.

**3.2.1. “OpenLayers API” Kullanılarak Temel Altlık Haritaların Uygulamaya Eklenmesi**

Uygulamaya OpenLayers Kütüphanesi yüklendikten sonra harita oluşturulur ve katmanlar bu haritaya eklenir. Aşağıda bulunan örnek ile OpenLayers Kütüphanesi ile harita oluşturma uygulamalı olarak tablo-2’de aktarılmıştır.

*Tablo 2: OpenLayers Kütüphanesi ile Harita Oluşturma*

|  |
| --- |
| // harita tanımlaması  DsiApp.Globals.map = new OpenLayers.Map('map', options);  //Katmanlar  var dsiBase = new OpenLayers.Layer.XYZ(  "DSI Altlık",  [  "http://xxx/basemap/${z}/${x}/${y}.png",  "http://xxx/basemap/${z}/${x}/${y}.png"  ],  {  attribution: "DSİ",  sphericalMercator: true,  transitionEffect: "resize",  buffer: 1,  numZoomLevels: 13  }  ); |

Çalışmamızda temel altlık olarak DSI için hazırlanan haritaları kullanmaktadır. Oluşturulan harita üzerine dsiBase adı verilen vektörel haritaların linkleri tanıtılmaktadır. Tablo-3 ile harita katmanının sisteme ekleniş kod bloğu verilmektedir.

*Tablo 3: OpenLayers Kütüphanesi ile Katman Ekleme*

|  |
| --- |
| DsiApp.Globals.map.addLayers([dsiBase]); |

OpenLayers kütüphanesinin “addLayers” metottu ile oluşturulan haritaya dsiBase katmanı eklenmektedir. Uygulamaya harita eklendikten sonra görünümü resim-4 de gösterilmektedir.



*Şekil 4: Harita Görünümü*

**3.2.2. “OpenLayers API” Kullanılarak “İstasyon” Katmanının Servis Olarak Okunması ve Haritaya Eklenmesi**

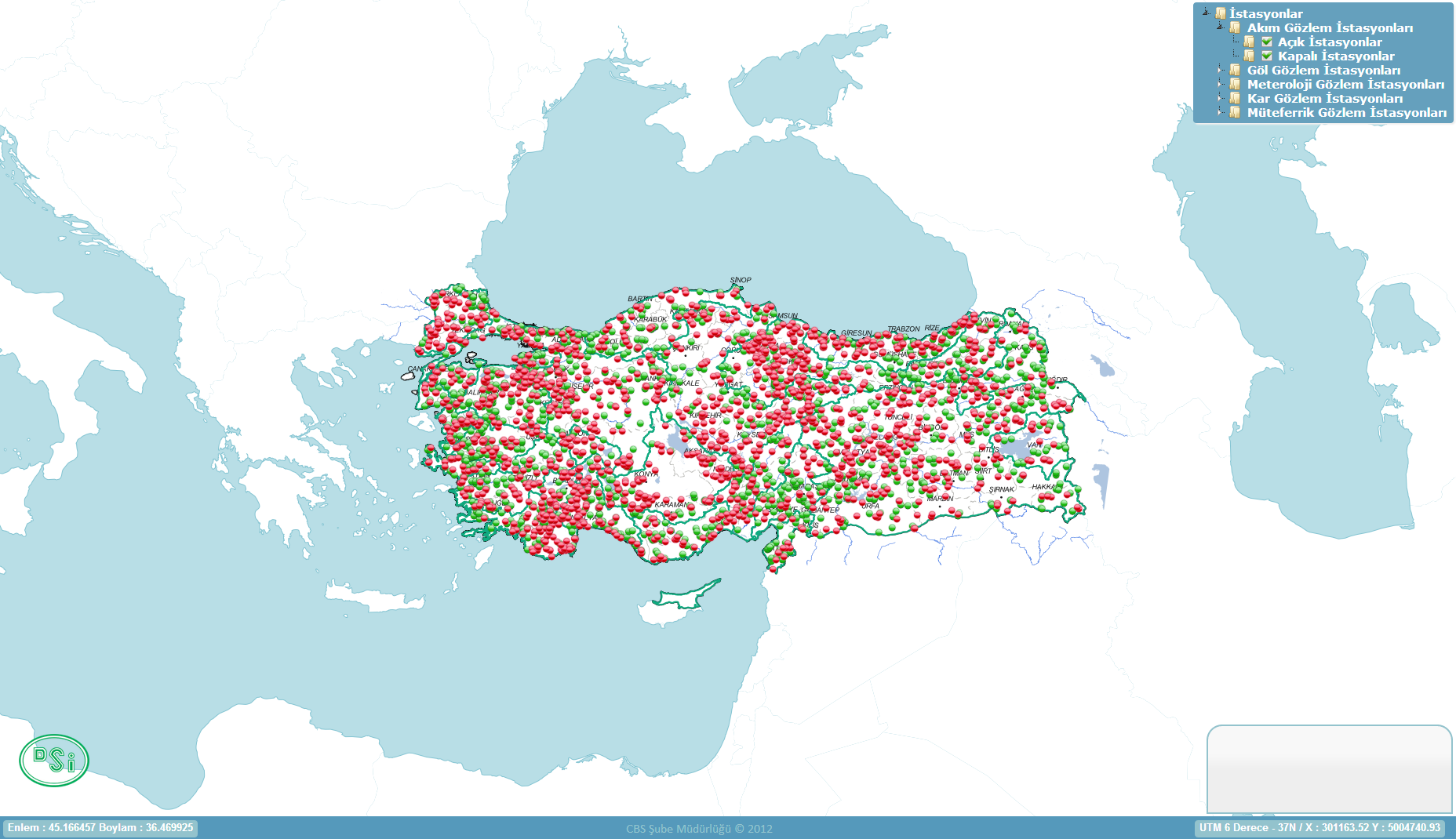
Aşağıda verilen tablo-4’de bulunan kod bloğu ile istasyonların harita üzerin de yerleştirilmesi örneklenerek gösterilmiştir.

*Tablo 4: İstasyonGosterim Fonksiyonu Kod Bloğu*

|  |
| --- |
| function istasyonGosterim() {      $.ajax({          url: DsiApp.Globals.ajaxUrl,//Web Servis Url          dataType: "jsonp",          success: function (data) {              gelenData = data;                in\_options = {                  'internalProjection': DsiApp.Globals.map.baseLayer.projection,                  'externalProjection': new OpenLayers.Projection("EPSG:4326")              };              format = new OpenLayers.Format.GeoJSON(in\_options);                if (id == "AkimIst") {                  AkimIst = format.read(gelenData);                    vectordata.addFeatures(AkimIst);                }              else if (id == "GolIst" ) {                  GolIst= format.read(gelenData);                    vectordata1.addFeatures(GolIst);                 }    /\*  …………………………...  \*/          },          error: function (data) {           alert("Exceptions");     }  });  } |

Web servisten gelen verileri ajax ile belirlediğimiz projeksiyonda uygulamaya eklenebilmektedir. Yukarıda örnek olarak paylaşılan kod bloğu “*Akım Gözlem İstasyonları*” ve “*Göl Gözlem İstasyonları*” ağaç görünümünden faydalanarak If-Else koşuluyla haritaya eklenmektedir.

Hazırlanan yapıya OpenLayers kütüphanesinde bulunan stil tanımlamaları eklenerek varsayılan stiller ile çalışmak yerine yeni semboller ile çalışılması tercih edilmektedir. Sonuç olarak elde edilen görünüm istasyonların net olarak bulundukları noktaları göstermektedir. Aşağıda bulunan resim-5 istasyonların haritaya eklenmiş halini göstermektedir.



*Şekil 5: Haritada İstasyon Gösterimi*

**2.2.3 “JQuery” ve “JQuery UI” Kütüphanelerinin Uygulama İçinde Kullanımı**

Gözlem İstasyonları Web Uygulamasında kullanılan tüm efektler “jQuery UI” dan faydalanılarak oluşturulmuştur. İstasyon detay bilgisi ve grafik gösterimleri “jQuery UI” ile güçlendirilmiştir. Resim-6 ve resim-7 de ekran çıktıları gösterilmektedir.



*Şekil 6: İstasyon Detay Gösterimi*

****

*Şekil 7: İstasyon Akım Grafiği*

**2.3.4. Uygulamada Kullanılan Diğer Fonksiyonlar**

CBS için hazırlanmış her proje belirli fonksiyonellikte olmaktadır. Gözlem İstasyonları Web uygulaması da bu fonksiyonellikleri barındırmaktadır. Haritaya belirli bir ölçekte yakınlaşma, uzaklaşma, harita altlığını değiştirme vb. Böylece temel harita işlemleri yapılabilmektedir. Uygulamada harita üzerine araç çubuğu yerleştirilerek, tüm fonksiyonlara erişim sağlanmaktadır.

Uygulama harita altlıkları için Devlet Su İşleri Genel Müdürlüğü CBS Şube Müdürlüğü tarafından DSI uygulamalarının kullanması için harita servisleri hazırlamaktadır. Projede kendi harita servislerini tüketmenin yanı sıra Google haritaları da altlık olarak kullanılmaktadır.

Harita üzerinde ölçüm aracı eklenilerek Jeodezik ve Yaklaşık ölçüm metotları ile mesafe ve alan ölçümü yapılmaktadır. İstasyonlar arası mesafe ölçümü ile birbirlerine olan uzaklık ve yakınlık dereceleri belirlenebilmektedir.

Harita uygulamalarının asıl amacı veriyi sorgulama işlemi de DSI’ nin belirlediği havza ve bölge sınırlarına göre sorgulanabilmektedir. Bu amaçla DSI Bölge Müdürlükleri kendi alanlarında kalan istasyonlar hakkında detay bilgisi alabilmektedir. Ayrıca hazırlanan arama çubuğu ile bilinen istasyon numarası ve ya su adı ile detayına ulaşılmak istenen istasyona kolay erişim sağlanmıştır.

Bilgisayar faresi sayesinde istasyonun konumu hem *Enlem-Boylam* hem de *UTM* cinsinden gösterilerek, koordinatlarına hızlı erişim sağlanmıştır.

**4. Sonuç**

“Gözlem İstasyonları Web Uygulaması” ile asıl amaçlanan veriyi olduğu noktada inceleyebilmek ve yapılan analiz sonuçlarını son kullanıcı tarafından kolaylıkla kullanabileceği uygulamalar ile paylaşmaktır. Geliştirilen uygulamaların Web uygulaması olarak hazırlanması, herhangi bir yer kısıtlaması olmadan ve farklı platformlarda kullanılması imkânını beraberinde getirecektir. Hazırlanacak değişik uygulamalar ile mevcut verilere işyerinde, evde erişme imkânı sağlatıldığı gibi araziye gidildiğinde de söz konusu verilere erişim sağlanacak ilerleyen süreçte mobil cihazlar vasıtasıyla veri girişi, güncellenmesi de sağlanacaktır. Mevcut akarsu debilerinin aylık ortalaması, sediment bilgileri, değerlendirilmiş yıllara göre ölçüm bilgileri ve istasyon sınıflandırılması yapılarak incelenecek istasyonun durumunun renkler ve sembollerle ifade edilmesi ve tüm rasat verilerinin tek bir platformda incelenebilir olması bu uygulama ile sağlanmıştır. Uygulamanın ilk açılışı ile istasyon hakkında basit bilgiye ulaşabilmek hızlandırılmış ve öngörü de bulunulması sağlanmıştır.

**Teşekkür**

Devlet Su İşleri Genel Müdürlüğü, Etüt Plan Dairesi ve Teknoloji Dairesi Başkanlığına projenin hazırlanışından bitişine kadar gösterdikleri ilgi ve yardımlarından dolayı teşekkür ederim. Ayrıca CBS Şube Müdürü Sayın Kemal Seyrek’e önerileri ve projeyi hep bir adım ileriye taşımam doğrultusunda ki katkılarından dolayı, teknolojik alt yapı ve projelendirme aşamasında yardım ve katkılarından dolayı Sayın Alper Dinçer’e ve veri altlıkları hazırlanması ile projeye bilgi ve katkılarını ekleyen Sayın N. Gülşah Güreşçi’ ye teşekkürü bir borç bilirim.

**Kaynaklar**

Dinçer, A.; Uraz, B.; Seyrek, K., 2012, CBS Web Uygulamalarında HTML5 ve Gelecek, UZAL-CBS 2012, Bülent Ecevit Üniversitesi, Zonguldak.

Erik Hazzard, (2011), OpenLayers 2.10 Beginner’s Guide Publisher, Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK.

Oracle® Application Server Web Services Developer's Guide 10g Release 3 (10.1.3) http://www.oracle.com/technetwork/articles [Erişim 12 Şubat 2013].

OsGeo Project PostGIS Web Sitesi, 2012, PostGIS 2.0.1 Release, http://postgis.refractions.net/news/20120622/, [Erişim 26 Mart 2013].

Sefer Algan, (2010), Her Yönüyle C# Yayınevi, Pusula Yayıncılık ve İletişim Tic. Ltd.

Terry McNavage, (2010), JavaScripts for Absolute Beginners Publisher, Appres

Uraz, B.; Dinçer, A.; Seyrek, K., 2012, Mekansal Veritabanlarında Masaüstü ve CBS Yazılımlarının Birlikte Çalışabilirliği, UZAL-CBS 2012, Bülent Ecevit Üniversitesi, Zonguldak.

http://jquery.com/ [Erişim 12 Aralık 2012].