

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**WEB TABANLI GPS/GNSS PROJE VERİ YÖNETİM SİSTEMİ TASARIMI
VE UYGULAMASI**

YÜKSEK LİSANS TEZİ

Deniz BAŞAR

Geomatik Mühendisliği Anabilim Dalı

Geomatik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Rahmi Nurhan ÇELİK

MAYIS 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**WEB TABANLI GPS/GNSS PROJE VERİ YÖNETİM SİSTEMİ TASARIMI
VE UYGULAMASI**

YÜKSEK LİSANS TEZİ

**Deniz BAŞAR
(501111605)**

Geomatik Mühendisliği Anabilim Dalı

Geomatik Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Rahmi Nurhan ÇELİK

MAYIS 2014

İTÜ, Fen Bilimleri Enstitüsü'nün 50111605 numaralı Yüksek Lisans Öğrencisi **Deniz BAŞAR** ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**WEB TABANLI GPS/GNSS PROJE VERİ YÖNETİM SİSTEMİ TASARIMI VE UYGULAMASI**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Rahmi Nurhan ÇELİK**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Hakan KUTOĞLU**
Bülent Ecevit Üniversitesi

Yrd. Doç. Dr. Caner GÜNEY
İstanbul Teknik Üniversitesi

Teslim Tarihi : **05 Mayıs 2014**
Savunma Tarihi : **30 Mayıs 2014**

dönüşümünü bekleyen Potansiyel Enerji'ye...

ÖNSÖZ

John Lennon’a göre “Hayat biz planlar yaparken başımıza gelenlerdir”. Bazen görmeden yaşarız hayatı zaman geçer biz nefes alırken, bazen de nefes almadan koşarız tek gördüğümüz planlarımızken yaşamayı unutup. Umut ışığıdır tek gereken... Bir başkasının elindeki fenerden çıkar kimi zaman ve gösterir bize her zaman var ve içimizde olan umudu, kendi fenerimizdir çoğu zaman da kaldırışımızda havaya gerçekten bir onur varsa içinde yanan alevi bizden olmayan, Cibran’ın söylediği gibi.

Herşey bir ve tek, herkes aynı, farkındalıklardan gelen farkla kendini bilmek herşeyi bilmek ve hiçken, elbiseyi yırtmak ve kendinden vermek gerçek erdem, kesin belirsizlikler gerçek hayat, düşlerimiz tek özgürlükken, kendi fenerimizdendir bizim bize okutulduğumuz, yeterince şanslıysak hayatın kalbine dokunduğumuz, büyüdüğümüz yolların ışığı ve yalnızca zihinlerinin eşiğine taşır başkalarını. Sevgi, sabır, hoşgörü, merhamet ve herşeye rağmen iyi kalabilmenin önemini görebilsek ne mutlu...

Ruhumun dümeni ve yelkeni, kalbimin mevsimleri arasında gölgemdeki yabancıya inat yol arkadaşım Cemil’e,

İdrakımın şafağında öğle saatlerime vurduğum zincirleri kırıp varlıkları ve fenerleriyle gündüzlerime ve gecelerimin ötesine giden yollara umut ışığı olan Abdullah Başar, Musa Doğaner, “Komutanım” Mevlüt Şevki Öncü, Aygül Aral, Ayşe ve Mustafa Şahin, Şefika ve Turan Özyıldırım, Celalettin Şimşir, Rahmi Nurhan Çelik ve Ailem’e fenerlerinin her daim doldurulup yakılması dileklerim ve sonsuz teşekkürlerim, varlıkları ve destekleri ile gönlümü şenlendiren yolgezer ve çileklere teşekkürlerim, yitirilenlerin gölgelerinin kıvanması dileklerimle...

Desteklerini esirgemeyen Hocalarım Caner Güney’e, Bihter Erol’a ve arkadaşlarıma teşekkürler.

Mayıs 2014

Deniz BAŞAR
Geomatik Mühendisi

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
1. GİRİŞ	1
1.1 ALAGÖZ Projesi ve ALAGÖZ Web Uygulaması	1
1.2 Sistem Mimarisi	2
1.2.1 JeoVT	3
2. PROGRAMLAMA DİLLERİ ve VERİTABANI	5
2.1 PHP.....	5
2.1.1 Tarihçesi ve gelişimi	5
2.1.2 Çalışma prensipleri	9
2.2 HTML.....	10
2.3 CSS.....	12
2.4 Veritabanı	12
2.4.1 PostgreSQL	15
2.4.1.1 Gelişimi ve tarihçesi.....	16
2.4.1.2 PHP ve postgresQL	17
3. ALAGÖZ WEB UYGULAMASI	19
3.1 Teknik Özellikler.....	19
3.2 Veritabanı Tasarımı.....	21
3.3 Web Yazılım	31
3.3.1 Laravel framework	33
3.3.1.1 Framework’ün özellikleri.....	34
3.3.2 Formlar.....	36
4. SONUÇ VE ÖNERİLER.....	39
KAYNAKLAR	41
EKLER.....	43
ÖZGEÇMİŞ.....	47

KISALTMALAR

ACID	: Atomicity, Consistency, Isolation, Durability
BLOB	: Binary Large Object
CERN	: European Council for Nuclear Research (Avrupa Nükleer Araştırma Merkezi)
CGI	: Common Gateway Interface
CK	: Composite Key(Birleşik Anahtar)
COMPASS	: BeiDou Navigation Satellite System, China
CVS	: Concurrent Versions System
DBM	: Database Management (Veri Tabanı Yönetimi)
DBMS	: Database Management System (Veri Tabanı Yönetim Sistemi)
DML	: Data Manipulation Processing
FIFO	: First In First Out
FIG	: Framework Interoperability Group
GALILEO	: European Satellite Navigational System
GLONASS	: GLObalnaya NAVigatsionnaya Sputnikovaya Sistema
GNSS	: Global Navigational Satellite System
GPS	: Global Positioning System
FK	: Foreign Key (Yabancı Anahtar)
HTML	: HyperText Markup Language (Zenginleştirilmiş İşaretleme Dili)
IBM	: International Business Machines Corporation (Uluslararası İş Makineleri Kurumu)
İVTYS	: İlişkisel Veri Tabanı Yönetim Sistemi
MIT	: Massachusetts Institute of Technology (Massachusetts Teknoloji Enstitüsü)
MVC	: Model View Controller
ORDBMS	: Object Relational Database Management System (Nesne İlişkisel Veri Tabanı Yönetim Sistemi - oivtys)
OLTP	: Online Transaction Processing
ORM	: Object Relatioanl Mapping
PK	: Primary Key (İkincil Anahtar)
RDBMS	: Relational Database Management System (İlişkisel Veri Tabanı Yönetim Sitemi - İVTYS)
RINEX	: Receiver Independent Exchange Format (Alıcı Bağımsız Değişim Formatı)
SEQUEL	: Structured English Query Language
SMTP	: Simple Mail Transfer Protocol
SQL	: Structured Query Language
TUTGA	: Türkiye Ulusal Temel GPS Ağı
TUSAGA-AKTİF	: Türkiye Ulusal Sabit GNSS Ağı-Aktif
URI	: Uniform Resource Identifier

URL	: Uniform Resource Locator
VTYS	: Veri Tabanı Yönetim Sistemi
XML	: Extensible Markup Language (Genişletilebilir İşaretleme Dili)

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: PHP/FI kod örneği (Lerdorf ve diğ, 2006).	6
Şekil 2. 2: PHP Sürüm Lejantı (URI-6).	7
Şekil 2. 3: PHP Sürümleri (URI-6).	8
Şekil 2. 4: PHP Kullanım İstatistikleri Ocak 2013 (URI-7).	8
Şekil 2. 5: PostgreSQL Sürümleri (URI-13).	16
Şekil 3. 1: Geometrik veri tipleri (URI-17).	20
Şekil 3. 2: Kampanya Tablosu.	22
Şekil 3. 3: İstasyon Tablosu.	23
Şekil 3. 4: Kampanya_İstasyon Tablosu.	24
Şekil 3. 5: Katılımcı Tablosu.	24
Şekil 3. 6: Kampanya_Katılımcı Tablosu.	25
Şekil 3. 7: GNSS_Oturum Tablosu.	25
Şekil 3. 8: Katılımcı_GNSS_Oturum Tablosu.	26
Şekil 3. 9: GNSS_Ölçü Tablosu.	26
Şekil 3. 10: Alıcı Tablosu.	27
Şekil 3. 11: Uydu Sistemleri Tablosu.	27
Şekil 3. 12: Kurum Tablosu.	28
Şekil 3. 13: Birim Tablosu.	28
Şekil 3. 14: Bölge Tablosu.	28
Şekil 3. 15: İl Tablosu.	29
Şekil 3. 16: İlçe Tablosu.	29
Şekil 3. 17: Mahalle Tablosu.	29
Şekil 3. 18: Varlık İlişki Modeli.	30
Şekil 3. 19: MVC' nin çalışma biçimi (URI-16).	31
Şekil 3. 20: Model View Controller Mimari Örüntüsü, (Dangar, 2013).	33

WEB TABANLI GPS/GNSS PROJE VERİ YÖNETİM SİSTEMİ TASARIMI VE UYGULAMASI

ÖZET

Levha hareketlerinden biri olan levhaların uzaklaşması durumu sonucunda Atlantik okyanusunda oluşan okyanus sırtlarının yayılması ile Afrika ve Arabistan levhaları kuzey-kuzeydoğu yönünde itilmektedir. Arabistan levhasının kuzeye doğru itilmesi, Avrasya levhasının altına dalmasıyla sonuçlanmaktadır. İki kıtasal levhanın çarpışma alanında bulunan ülkemizin doğusunda kıta kabuğu kalınlaşmış ve hala kalınlaşmaya devam etmektedir. Arabistan levhasının kuzeye doğru ilerlemesi sonucu anadolu levhası batıya doğru hareket etmeye başlamıştır (URI-1).

Türkiye ve çevresinde tektonik hareketler nedeniyle oluşan deformasyon ve depremsellik nokta koordinatlarında zamana bağlı değişimler meydana getirmektedir. Bu değişimlerin tespit edilebilmesi ve Türkiye Ulusal Temel GPS Ağı koordinatlarının güncellenebilmesi için mevcut istasyonlarda tekrar gözlem yapılması ve gerekli görülecek yerlerde yeni tesis edilecek istasyonlarla mevcut ağın sıklaştırılması gerekebilmektedir.

Tasarım aşamasında olan Alıcını Al Gel Ölçü Zamanı (ALAGÖZ) Projesi ile Türkiye genelinde Küresel Uydu Seyrüsefer Sistemi (Global Navigational Satellite System-GNSS) ya da Küresel Konumlama Sistemi (Global Positioning System-GPS) ölçüsü yapılarak Türkiye'nin jeodezik altyapısındaki sorunlara çözüm üretilmesi ve ülke tektoniğinin ortaya konulması amaçlanmaktadır. Proje kapsamında, mevcut olan ED50-ITRF ortak noktalarından ve TUSAGA-Aktif Projesi kapsamındaki noktalardan yararlanılması da planlanmaktadır (URI-2).

Tez çalışması kapsamında, ALAGÖZ Projesi katılımcılarının yapacağı GNSS gözlemleriyle elde edilecek Alıcıdan Bağımsız Değişim (RINEX) formatındaki verilerin değerlendirilmek üzere depolanması ve platform kullanıcıları arasında paylaşılabilesinin sağlanması amaçlanmaktadır. Bu amaçla, ALAGÖZ Projesinde kullanılmak üzere bir veri yönetimi sistemi tasarlanmış ve site kullanıcılarının GPS/GNSS verilerini projenin ihtiyaçları doğrultusunda tasarlanan veritabanına web formları aracılığıyla, lokal sunucuda, kaydedebilmesini sağlayan bir platform geliştirilmiştir .

Çalışma kapsamında geliştirilen platform için tasarlanan sistem mimarisi ile öncelikli olarak ALAGÖZ Projesinin ve proje katılımcılarının gereksinimleri tespit edilmiş, yazılımın geliştirileceği mimari desen bileşenleri belirlenmiş, tanımlanan sistem ana bileşenlerine uygun olarak veritabanı tablo yapıları ve web form yapıları tasarlanmıştır.

Platform, paket program olan ve PHP, PostgreSQL - PostGIS ile Apache sanal sunucusunun kurulumunu hazır olarak sunan BITNAMI programında Laravel Framework'ü üzerinde geliştirilmiştir.

Web yazılım için PHP, JS, JQuery dillerinden, tasarım için HTML, CSS dillerinden yararlanılmış, GNSS verisinin depolanması ve ilerleyen süreçlerde uygulamanın geliştirilmesi aşamasında yapılabilecek CBS analizleri için uygun bir zemin oluşturan PostgreSQL veritabanı kullanılmıştır ve PostGIS ile ilişkilendirilmiştir.

DEVELOPMENT AND APPLICATION OF WEB BASED GPS/GNSS PROJECT DATA MANAGEMENT SYSTEM

SUMMARY

Because of the fact that Turkey is situated between Europe, Asia and Africa continents, it has been under the effect of crustal deformation for ages. The reason of Turkey being affected is that the plate tectonics resulting from large-scale motion of Earth's rigid outermost shell called lithosphere. Owing to the divergent tectonic plate causing Mid-Atlantic Ridge, African and Arabian plates have been moving northward. Hence, Arabian Plate have been colliding with Eurasian Plate. Therefore, Anatolian plate has been moving westward and rotating counterclockwise due to being pushed west by Arabian Plate, (URI-1).

The reasons such as seismicity and deformation resulting from plate tectonics in and around Turkey result in the changes of point coordinate in time. In order to detect these changes and update the coordinates of TUTGA (Turkish National Fundamental GPS Network), new settlements on the existing stations and densification of the existing network should be required.

Due to fulfill the requirement a project called as ALAGOZ has been designed. In order to keep TUTGA coordinates up to date, to repeat the settlements on the existing stations and network densification, with ALAGOZ Project it is aimed that to plan GPS/GNSS campaigns all around the Turkey. Moreover, the project plan includes serving a platform on which project participants allowed to upload the GPS/GNSS data gathered from the field within the scope of the campaigns planned by the ALAGOZ Project managers. In the scope of the project, not only the newly established stations but also existing common ED50-ITRFyy stations being used is foreseen. Besides, it is planned to satisfy the requirement of an active system that is going to store and manage the GNSS data gathered for different purposes such as academic, commercial and governmental works.

In addition, it is a well known fact that there are more than 4000 receivers all around the Turkey. Therefore, in case of 4000 receiver's being used during the GNSS campaigns of the project point densification is being estimated as 20km² on the other hand, in case of 2000 receiver's being used during the campaigns of the project point densification is being estimated as 40km² (URL-2).

In this study, in order to use in ALAGOZ Project, a web based data management system has been developed. The web application has been developed as a platform to serve ALAGOZ Projects requirements. This platform serve the project as letting the project's participants uploading, the GNSS data that they gathered from the field with GNSS observations, into the specifically created database with the form structures through the server. These observed GNSS data have a standard data format which is being called RINEX format. This format provides a standart data upload type and facilitates the data processing, and data sharing.

In scope of this thesis project with the aim of creating the platform, a software system architecture has been planned first. According to being main step of system architecture, ALAGOZ Project's and the participant's requirements are determined. Afterward, architeturall pattern componenets in which the platform will be created is determined. Thereafter, according to system's main components database creation and related form structure's determination are completed.

Including PHP, Apache, and MySQL all in a packet BITNAMI WAMP Stack has been used in web development of ALAGOZ Project. PostgreSQL module is added over the WAMP Stack just in case of altering the database in the future. Moreover, PostGIS is attached to PostgreSQL because of the geometric data type support. Netbeans and PhpStorm IDEs have been used during the web development process of the thesis.

Owing to the fact that the wheel has already been invented, Laravel PHP Framework is used for the web development of ALAGOZ Project's web platform. Laravel 4 framework, which is one of the ready to use frameworks in BITNAMI wamp stack, is a composer compatible framework. Hence, Laravel framework provide ease of use due to enable user to add all Composer components to their own specific applications. In other words, Laravel 4 framework consists of dependencies that are available on GitHub. Laravel differ from other PHP Frameworks with it's Composer compatibility, interoperability, Eloquent ORM, database seeding, artisan, queues features, and etc.

ALAGOZ Project's web application is developed with an open source programming language PHP that is provided by BITNAMI. PHP, used to mean Personal Home Page, means PHP Hypertext Preprocessor and is used for creating dynamic websites. Because of being a server side programming language, PHP allows users to interact with the site. In another word, instead of only being able to gather information from the website, users can post data into the site with forms, comment on a forum, or even be a part of a survey. PHP language is a script language; hence, it does not require a compiler. However, it requires a web server, because of this reason Apache web server is preferred in this application. Additionally, HTML (Hypertext Markup Language), and CSS (Cascading Style Sheets) have been used for the visualization of the website.

Furthermore, application needs a database to be able to store the data that is posted by the users. PostgreSQL database, which is an open source relational database management system (RDBMS), is preferred in ALAGOZ Project. RDBMS generally has client-server architecture that allows remote connection and multi usage at the same time via internet.

Forms are vital structures of the web application because of informing the campaign participants about the news of project. Moreover, it is the only way to obtain the information about participants, metadata and the data which is gathered from the field with the purpose of updating TUTGA point coordinates, or densification of networks such as National Geodetic Network. The forms will send the data posted by the participant in the database. RINEX formatted GPS/GNSS datafiles including navigation, meteorology and observation files are stored inside the application environment which is created in Model View Controller (MVC) Pattern.

To sum up, the web platform of the ALAGOZ Project consists of membership, session management, form processing, database handling, moreover RINEX formatted GPS/GNSS data analysis and visualization process are planned to be accomplished in the following developments.

Finally, being a web platform developed in the localhost the thesis project required to be developed in order to deal with the real data and metadata supplied by the participants of the ALAGOZ Project. These real data provided by the project's participants would be stored appropriately for the standards such as ISO TC211, OGC, INSPIRE etc. Therefore, interoperability feature could be provided for the platform.

Consequently, GIS applications, visualization of the data on maps, and creation of an android version of the ALAGOZ web application is planned to be achieved in the continuing steps of this study.

1. GİRİŞ

1.1 ALAGÖZ Projesi ve ALAGÖZ Web Uygulaması

Tasarım aşamasında olan ALıcını Al Gel Ölçü Zamanı (ALAGÖZ) Projesi ile farklı kurum, kuruluş ve üniversitelerce ortaklaşa Küresel Uydu Seyrüsefer Sistemi (Global Navigational Satellite System-GNSS) ya da Küresel Konumlama Sistemi (Global Positioning System-GPS) ölçüsü yapılarak Türkiye'nin jeodezik altyapısındaki sorunlara çözüm üretilmesi ve ülke tektoniğinin ortaya konulması amaçlanmaktadır. Proje kapsamında, mevcut olan ED50-ITRFyy ortak noktalarından ve TUSAGA-Aktif Projesi kapsamındaki noktalardan yararlanılması da öngörülmektedir (URI-2). ALAGÖZ Projesi ile Türkiye ve çevresindeki tektonik hareketler nedeniyle oluşan nokta koordinatlarında meydana gelen zamansal değişimler tespit edilebilecek, Türkiye Ulusal Temel GPS Ağı (TUTGA) koordinatlarının güncelliği korunabilecek ve gerekli görülecek yerlerde mevcut jeodezik ağın sıklaştırılması yapılabilecektir.

Ayrıca proje ile ülkemizde; akademik, ticari ve kamusal amaçlarla elde edilen GPS/GNSS verilerinin tek bir ortamda standart bir formatta saklanıp yönetilebileceği aktif bir sistemin mevcut olmamasından doğan ihtiyacın giderilmesi de amaçlanmaktadır.

Projede, Türkiye genelinden katılımcıların desteğiyle gerçekleştirilecek GNSS kampanyaları ile ülke genelinde mevcut olan 4000'in üzerinde GNSS alıcısının katılımı halinde 20 km²'ye bir nokta, 2000 GNSS alıcının katılımı durumunda 40 km²'ye bir nokta sıklığı sağlanması öngörülmüştür. Proje dâhilinde toplanan verilerin değerlendirilebilmesi ve paylaşılabilmesi için bir internet platformuna yüklenmesi planlanmıştır (URI-2).

Tez çalışmasının amacı, ALAGÖZ Projesi dâhilinde toplanacak GPS/GNSS verilerinin Alıcıdan Bağımsız Değişim Formatı'nda (Receiver Independent Exchange Format-RINEX) depolanıp, ihtiyaçlar doğrultusunda sorgulanabilmesi için verilerin yönetimini içeren bir sistemin tasarlanmasıdır. Bu sistem ile projenin gereksinimlerini

karşılacak bir veritabanı tasarımının yapılması ve verilerin sanal sunucu üzerinden manipüle edilebilirliğinin sağlanması hedeflenmektedir.

Tasarlanan sistem ile ALAGÖZ Projesi katılımcılarının GNSS alıcıları kullanarak GPS, Rusya'ya ait konumlandırma sistemi (GLObalnaya NAvigatsionnaya Sputnikovaya Sistema, GLONASS) ve tamamlanması halinde Avrupa birliği tarafından GPS ve GLONASS' a alternatif olarak tasarlanan konumlandırma sistemi (European Satellite Navigational System, GALILEO) ile Çin' e ait konumlandırma sisteminden (BeiDou Navigation Satllite System, COMPASS) de yararlanarak elde edecekleri verileri ya da mevcut verileri Proje için tasarlanan veritabanına doğrudan yükleyebilmelerine olanak sağlanması ve bu verilerin world wide web (web) ortamında paylaşılabilmesi planlanmaktadır (URI-3).

1.2 Sistem Mimarisi

Tez çalışması kapsamında, ALAGÖZ Projesi için tasarlanan sistemde, öncelikli olarak yazılım sistem mimarisinin modellenmesi için projenin ana bileşenleri ile bu bileşenlerin birbirleriyle etkileşimlerinin belirlenmesini ve -projeye dahil olarak tasarlanacak sistemi kullanacak- kullanıcıların gereksinimlerinin tespit edilmesini içeren üst düzey tasarım gerçekleştirilmiştir (URI-4). Bu tasarım kapsamında ALAGÖZ Projesi dahilinde gerçekleştirilecek kampanyalar, kampanyaların gerçekleştirileceği tarihler ve bölgeler, bu kampanyalarda görev alacak katılımcılar ve katılımcılara ait öznitelik bilgileri, GNSS ölçmesi yapılacak istasyonlar ve bu istasyonlara ait konum bilgisi (*navigasyon*), istasyon adı, tesis tarihi ve varsa mevcut olan istasyonlara ait veri ve metadata'lar, istasyonlardan elde edilecek koordinat bilgileri temel gereksinimler olarak belirlenmiştir

Mimari desen olarak, temel gereksinimleri karşılayacak web tabanlı form yapılarının gerçekleştirilebilmesi için sanal Apache sunucusu, geometrik veri tiplerini de destekleyen açık kaynak kodlu ve ücretsiz PostgreSQL ilişkisel veritabanı, form yapılarının ve veritabanı ile ilişkilendirilmesini sağlayacak programama dili olarak PHP dili- uygulamanın geliştirileceği ortam bileşenleri olarak- belirlenmiştir. Bu ortam bileşenlerini paket olarak sunan ve farklı eklentilere kolaylıkla erişim olanağı sağlayan Bitnami WappStack tercih edilmiştir.

Sistem ana bileşenleri olarak, proje kapsamında düzenlenecek kampanyalar ile sahadan toplanacak GNSS verilerinin saklanması ve sorgulanabilmesi için oluşturulacak ilişkisel veritabanı tasarlanırken proje için gerekli olan temel veriler; kampanyalar, katılımcılar, istasyon ve GNSS verileri olarak tespit edilmiştir.

Bu gereksinimleri karşılayacak web tabanlı form yapılarının tasarlanmasında ve temel verilerin saklanıp ilişkilendirileceği tablo yapılarını içeren veritabanının tasarlanmasında, daha önce Dr. Özgür ÖZASLAN tarafından geliştirilen “Jeodezik amaçlı web tabanlı mekansal veritabanı JeoVT”nin veritabanı ve form yapılarından yararlanılmış ve her türlü jeodezik ölçme verisinin depolanıp sorgulanabileceği yapının ALAGÖZ Projesinin gereksinimlerini karşılamak üzere sadece GPS/GNSS verileri için özelleştirilmesi kararlaştırılmıştır.

1.2.1 JeoVT

JeoVT Projesinde, ilişkisel veritabanı modelinden yararlanılarak jeodezik verilerin tutulacağı kavramsal bir veritabanı tasarlanmış, bu tasarım üzerinden fiziksel bir model oluşturulmuş ve fiziksel model MySQL veritabanı yönetim sistemine aktararak gerçekleştirilmiştir. Ayrıca bu veritabanının yönetimi ve kullanıcıların veri girişi, güncellemesi ve veritabanında arama yapabilmesini ve dengeleme sonuçları ve baz çözümlerinin veritabanına kayıt edebilmeleri için, PHP dili kullanılarak, web tabanlı kullanıcı dostu bir ara yüz hazırlanmıştır. Proje ile aynı bölgeye ait farklı zamanlı, farklı özellikli ölçmelerin birlikte sorgulama birliği ve analizi, statik form yapısı ile veri kaydının standartlaştırılması, ölçme ve değerlendirme sonuçlarının web’e açılması, kurumların ölçme ve değerlendirme verilerinin tek bir veritabanında toplanması amaçlanmıştır (Özaslan, 2003).

Projenin veritabanı doğrultu, düşey açı, uzunluk, yükseklik, GPS ölçmeleri için ayrı ayrı oluşturulmuş tabloların yanı sıra proje, istasyon, kurum, personel, alet ve meteoroloji ortak tablolarından oluşmaktadır.

Veritabanı oluşturulurken, tutulması gereken veriler belirlenmiş, niteliklerine göre sınıflandırılmış, kavramsal veri modelleri ile tablolar arasındaki ilişkisel yapı belirlenmiş, Qdesigner yazılımı ile fiziksel veri modeli oluşturulmuş ve tasarım MySQL veritabanında gerçekleştirilmiştir (Özaslan, 2003). Oluşturulan veritabanındaki ortak tabloların sütun isimleri şu şekillerde belirlenmiştir;

Proje tablosu; proje ID, proje adı, uygulanacağı il, ilçe, bölge, projenin başlangıç tarihi, bitiş tarihi

İstasyon tablosu; istasyon ID, istasyon numarası, istasyon adı, istasyonun koordinatları, gravite değeri, topografik düzeltme değeri, koordinatların kesinlik bilgisi, hesaplanma yöntemi, datum, tesis tarihi, tesis türü, pafta adı, ulaşım bilgisi, kroki, röper

Kurum tablosu; kurum ID, kurum adı, adresi, telefonu, faks, e-posta, ilgili kişi

Personel tablosu; personel ID, kurum ID, personel ad, soyadı, adres, e-posta, telefon, faks

Alet tablosu; alet ID, seri no, ismi, türü, üretici firma,

Meteoroloji tablosu; istasyon ID, tarih, basınç, kuru sıcaklık, ıslak sıcaklık, nem. GPS ölçmeleri için oluşturan tablo yapıları ise şu şekildedir;

GPS oturum tablosu; GPS oturum ID, proje ID, gps oturum adı başlangıç tarihi, bitiş tarihi, kayıt çizelgesi, veri kayıt aralığı, yükseklik açısı (eğim açısı)

GPS ölçme tablosu; GPS ölçü ID, değerlendirme ID, gps oturum ID, istasyon ID, alet yüksekliği, ölçü dosyası, navigasyon dosyası, meteoroloji dosyası

GPS değerlendirme tablosu; GPS değerlendirme ID, dengeleme sonuçları, baz çözümleri

GPS oturum - alet ara tablosu; GPS oturum ID, istasyon ID, alet ID

GPS oturum - personel ara tablosu; GPS oturum ID, istasyon ID, personel ID

GPD değerlendirme-personel ara tablosu; personelID, GPS değerlendirmeID

(Özaslan, 2003).

ALAGÖZ Projesi için tasarlanan veritabanında JeoVT'nin GPS oturum, GPS ölçme, Alet, İstasyon, Proje, Kurum, Birim tablo yapılarından yararlanılmıştır.

2. PROGRAMLAMA DİLLERİ VE VERİTABANI

2.1 PHP

Dinamik web içeriklerinin oluşturulabilmesi ve web sitelerinin kullanıcı ile aktif etkileşiminin sağlanabilmesi için Sunucu Taraflı (*Server-Side*) programlama dillerinin kullanılması gerekir. Sunucu taraflı programlama dilleri ile kullanıcının siteye yorum eklemesi, sitedeki anketlere katılması, mevcut bilgileri istediği gibi sorgulaması sağlanabilir. Personal Home Page (*PHP*), günümüzde sunucu taraflı teknolojileri arasında en yaygın olarak kullanılan programlama dillerinden birisidir (Şamlı, 2011).

2.1.1 Tarihçesi ve gelişimi

PHP, Web geliştiriciler tarafından dinamik web uygulamaları geliştirmek amacıyla tasarlanan ve geliştirilen betik (*script*) bir dildir. İlk olarak 1994 yılında, Rasmus Lerdorf tarafından kişisel web sayfasına yerleştirdiği özgeçmişinin görüntülenme sayısını belirlemek ve kullanıcı ile site arasındaki veri akışını ve miktarını kaydetmek amacıyla kullandığı bir dizi Perl/CGI scriptinin yerini almak üzere C dilinde tasarlanmış ve PHP Tools sürüm 1.0 ismini almıştır (Doyle, 2010). Rasmus Lerdorf'un PHP Tools' u ilk bildirim metni EK A' da yer almaktadır (Lerdorf ve diğ, 2006).

Takip eden süreçte, Lerdorf, PHP'nin işlevselliğini arttırmak ve uygulamaları zenginleştirmek için PHP Tools'u yeniden yazmış ve yeni modele veri tabanları ile etkileşim özelliği ve dinamik web sayfası yaratmayı kolaylaştıracak bir taslak (*framework*) eklemiştir. 1995 yılının Haziran ayında PHP Tools' un kaynak kodlarının açık olarak yayınlanması, PHP'nin bilinirliğinin artmasını sağlamış ve kullanıcıların da hataları ayıklayabilmesi gelişim sürecini hızlandırmıştır (URI-5). Dilin olumlu gelişim süreci Lerdorf'u verimliliği arttırmak için form yapısını da destekleyen ve kısa bir dönem sadece FI (*Form Interpreter*) olarak anılan, PHP 2.0 ya da Personal Home Page-Form Interpreter (*PHP-FI*) script motorunu oluşturmaya yöneltmiş ve dile bugün de bilinen temel özelliklerini kazandırmıştır. FI'ye kazandırılan, Perl benzeri

değişkenler, '\$' işareti ile tanımlanıp otomatik olarak yorumlanan form değişkenleri ve HTML içine gömülebilen söz dizimi (*sentaksı*) sayesinde FI hızlı gelişimini sürdürmüştür. Henüz bir programlama dili özelliği kazanamasa da *Hyper Text Markup Language (HTML)* içine gömülerek statik web yapısına dinamik bir özellik kazandırması sayesinde *Computer Gateway Interface (CGI)* aracı olarak kabul görmüştür. Aslında bir program olan CGI, Web Servisleri ile bu servislerin dışındaki programlar arasında etkileşim -ortak çalışma- platformu oluşturmak için geliştirilmiş bir standarttır. 1995 yılının sonlarına doğru Lerdorf, tamamen/tamamını yeniden yazdığı kodları yayınlamış ve PHP ismini tekrar kullanmaya başlamıştır ve artık PHP ileri betik ara yüzü (*Advance Scripting Interface*) olarak anılmaya başlanmıştır. 1996 yılının ortalarına doğru kodlarda yapılan yeni güncellemelerle ikinci sürüm, ayrı bir programlama dili olarak PHP/FI nihai ismini almıştır. PHP/FI artık DBM, mSQL ve Postgres95 veri tabanları, çerez (*cookie*) ve oturum (*session*) için gömülü (*built-in*) destek içermektedir. 2.0 sürümü ile PHP'ye kazandırılan betik motor yapısı uygulama hızında artış, veri transferi ve denetiminde istikrar, dış tehditlere karşı farklı güvenlik düzeyleri ve C/C++ ve Javascript dillerine benzerliğinden dolayı basitlik gibi özellikleri beraberinde getirmiştir. PHP/FI kod örneği Şekil2.1 de gösterilmektedir.

```
<!--include /text/header.html-->

<!--getenv HTTP_USER_AGENT-->
<!--ifsubstr $exec_result Mozilla-->
  Hey, you are using Netscape!<p>
<!--endif-->

<!--sql database select * from table where user='$username'-->
<!--ifless $numentries 1-->
  Sorry, that record does not exist<p>
<!--endif exit-->
  Welcome <!--$user-->!<p>
  You have <!--$index:0--> credits left in your account.<p>

<!--include /text/footer.html-->
```

Şekil 2.1: PHP/FI kod örneği (Lerdorf ve diğ, 2006).

1997 yılının ortalarında, binlerce kullanıcıya ulaşan dilin popüleritesindeki hızlı artış sonucu, projenin tek geliştiricisi olan Lerdorf, dilde süregelen ve temel ayrıştırma motorundan (*Underlying Parsing Engine*) kaynaklanan ayrışmazlık (*stability*) problemlerinin giderilmesi için gönüllü olan Andi Gutmans ve Zeev Suraski ile anlaşmış ve birlikte yeni ve bağımsız bir programlama dilinin yazılması için işbirliği yapmıştır. 1998 yılında, oluşturulan bu yeni dil PHP 3.0 sürümü olarak yayınlanmış,

ancak PHP kısaltmasının anlamı PHP: Hypertext Preprocessor olarak değiştirilmiştir. PHP-FI de olmayan pek çok özelliğin bu yeni oluşturulan sürüme eklenmesiyle PHP tam olarak bir sunucu tarafı teknolojiye dönüştürülmüş ve 50.000 den fazla kullanıcı tarafından kişisel web sayfalarında kullanılmıştır.

PHP 3.0 sürümünün yayınlanmasından kısa bir süre sonra Andi Gutmans ve Zeev Suraski karmaşık uygulamalarda performansı arttırmak ve dilin modülerliğini arttırmak için PHP çekirdeğini yeniden yazma çalışmalarına başladı. 1999 yılının ortalarında tanıtılan ve PHP mimarisindeki tasarım sorunlarını başarılı bir şekilde ortadan kaldıran yeni çekirdek, Zeev ve Andi isimlerinin birleşimi olan “Zend-Motoru” olarak adlandırıldı. Netcraft internet araştırma firması, Ocak 1999 da yaptığı gerçekçi tahminler ile PHP’ nin o yıl bir milyondan fazla kullanıcıya ulaşarak dünyanın en yaygın betik dillerinden biri olacağını öngörmüştü (Ledorf ve diğ, 2006). 2000 yılında çok sayıda geliştiricinin de katkısı ile geliştirilmiş Zend-Motoru’nu temel alan PHP 4.0 yayınlandı ve beklentilerin de üstünde bir popülerlik yakaladı. Zaman içerisinde PHP 4.0 sürümünün yüksek performans özelliğine yeni web sunucular ı için destek, HTTP sessions, farklı programlama dilleri ile ilişkilendirme gibi pek çok yenilik eklenmiştir.

2000 yılında popülerliği beklentileri aşan PHP 4.0’ın hayata geçmesi dili tek kişinin üretimi olmaktan çıkarıp dünya çapında yaygın olan UNIX ve çok geliştiricili bir açık kaynak koda dönüştürmüştür (Lerdorf ve diğ, 2006).

Şekil 2.2 de PHP sürümlerini geçerliliklerini gösteren lejant, Şekil 2.3 de PHP programlama dilinin sürümleri ve yayınlanma tarihleri ile gelişimi gösterilmektedir.

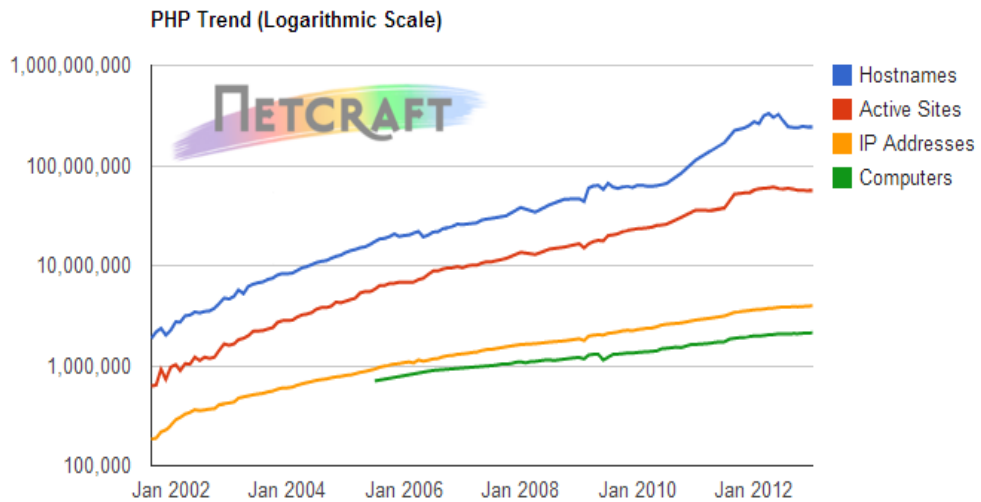
	Anlamı
Kırmızı	Eski sürüm, desteklenmiyor
Sarı	Eski sürüm, hala destekleniyor
Yeşil	Mevcut sürüm
Mavi	Gelecek sürüm ya da BETA, ALFA sürüm

Şekil 2. 2: PHP Sürüm Lejantı (URI-6).

Ana Sürüm	Ek Sürüm	Yayın Tarihi	Notlar
1	1.0.0	1995-06-08	"Personal Home Page Tools(Kişisel Anasayfa Araçları)". olarak anılıyordu ve kısaltımı "PHP" idi.
2	2.0.0	1997-11-01	Yaratıcısı ve geliştiricisi "en hızlı ve en sade araç" olarak, web sayfaları için geliştirdi.
3	3.0.0	1998-06-06	Sürüm kişiselikten, kurumsallığa geçti. Zeev Suraski ve Andi Gutmans tekrar basit dil bilgisini yazdı.
4	4.0.0	2000-05-22	Daha ileri seviye dil bilgisi eklendi/Zend Motoru denilen dil bilgisi denetleyicisi getirildi.
	4.1.0	2001-12-10	'Süper globaler' tanıtıldı. (\$_GET, \$_POST, \$_SESSION, \$_SERVER vb...)
	4.2.0	2002-04-22	register_globals öntanımlı olarak engellendi.
	4.3.0	2002-12-27	CLI tanıtıldı, CGI ekleri yapıldı.
	4.4.0	2005-07-11	phpize ve php-config betikleri için eklenmeler yapıldı.
	4.4.8	2008-01-03	Çoğu güvenlik sorunları ve hatalar düzeltildi. Son kalıcı PHP 4 sürümü oldu. Gerekli ise, güvenlik güncellemeleri 2008-08-08 e kadar sürdü.>
5	4.4.9	2008-08-07	Daha fazla açık onarıldı. PHP 4 serileri için son sürüm oldu.
	5.0.0	2004-07-13	Zend Motoru II ile yeni obje modelleri geliştirildi.
	5.1.0	2005-11-24	Eklenen değişkenler ile performans artışı gözlemlendi.
	5.2.0	2006-11-02	Filtreleme öntanımlı olarak iptal edildi.
	5.2.11	2009-09-16	Hata ve güvenlik sorunları çözüldü.
	5.2.12	2009-02-25	Güvenlik ve veri tabanı iletişimi güçlendirildi. 60 adet hata düzeltildi.
	5.2.13	2010-07-22	Hatalar düzeltildi.
	5.2.14	2010-12-17	Hatalar düzeltildi. PHP 5.2.x sürümleri geliştirilmesi durduruldu.
	5.3.0	2009-06-30	Namespace desteği, Atlama kodları (goto ile limitlendi), Yerli PHP arşivlemesi (phar), Windows desteği artırıldı, sqlite3 geliştirildi, fileinfo, mime_magic yerine daha iyi bir MIME desteği için getirildi, uluslararası olarak "ereg" kod yapısı iptal edildi.
	5.3.1	2009-11-19	100'den fazla hata düzeltildi, bazı küçük sorunlar da düzeltildi.
	5.3.2	2010-03-04	Çok geniş bir hata düzeltimi yapıldı.
6	5.3.3	2010-07-22	Başlıca hata ve güvenlik sorunları onarıldı. FPM
	6.0.0	Şu an güncellenmedi, gelecek sürüm	'register_globals' iptal edildi, Unicode, 'magic_quotes' ve 'safe_mode' desteği geldi.

Şekil 2. 3: PHP Sürümleri (URI-6).

İlk sürümden itibaren GPL (*GNU General Public License*) Açık Kaynak Kod lisansı ile dağıtılmaya başlanan dilin kullanıcı sayısı son on yıl içinde yetmiş milyonu aşmıştır. Şekil 2.4 Netcraft internet araştırma firmasının tarihli PHP dilinin kullanım yaygınlığını gösteren Ocak 2013 istatistiklerini içermektedir.



Şekil 2. 4: PHP Kullanım İstatistikleri Ocak 2013 (URI-7).

2.1.2 Çalışma prensipleri

Betik bir dil olduğu için, kodları düz yazı dosyaları halinde kaydedilen PHP, .exe veya çalıştırılabilir başka bir dosya oluşturmadığı için derleyiciye ihtiyaç duymaz ve kullanılacağı ortamdaki yorumlayıcı tarafından yorumlanarak çalıştırılır. Ancak PHP scriptlerini çalıştırabilmek için kullanılacak yorumlayıcı programın komut istemci penceresinde veya terminal ekranında çalışması mümkün olduğu gibi bir web sunucusu üzerinde de çalışabiliyor olması gerekir. Web sunucusu üzerinde çalışan PHP programları, istek üzerine web sayfalarını ziyaretçiye sunar. Web sunucusu hazırlanan web sayfalarının yayınlandığı yazılımdır. Web sunucularının çalışma mantığı şu şekildedir;

1. Tarayıcı, arama çubuğuna yazılan URL'i mesaj olarak Web Sunucusu'na yollar.
2. Sunucu mesaj ile talep edilen HTML dosyasını taracıya gönderir.
3. Tarayıcı sunucudan gönderilen HTML dosyasını okuyarak, HTML etiketleri ile tanımlanmış web sayfasını görüntüler.

Bu işlem döngüsü, web sayfaları içinde yer alan bağlantılar için de geçerlidir.

Eğer tarayıcı “.php” uzantılı bir dosya talep ederse, sunucu talep edilen dosyayı doğrudan tarayıcıya göndermek yerine sayfa içerisindeki PHP bölümlerine bakarak onları işler. Bu süreç şu şekilde gerçekleşir;

Web Sunucusu talep edilen dosyayı HTML dosyası olduğunu varsayarak HTML Modunda tarar ve tarayıcıya geri gönderir. HTML modunda gerçekleşen bu tarama, sunucu PHP'nin açılış etiketi olan ‘<?php’ ile karşılaşınca kadar devam eder. Sunucu ‘<?php’ etiketine rastladığı zaman PHP Moduna geçer ve tüm ifadelerin PHP Modunda olduğunu varsayarak PHP ifadelerini çalıştırır. Kodların ‘echo "çıkış değeri";’ komutu ile çıkış değeri (output) ürettiği yerlerde, çıkış değerini tarayıcıya gönderir ve PHP'nin kapanış etiketi olan ‘?>’ ile karşılaşınca kadar kodları yorumlamaya devam eder. Kapanış etiketi ile karşılaştığı noktada HTML Moduna geri dönerek dosyayı -tarayıcıya göndermek üzere- taramaya devam eder, döngü bu şekilde tamamlanır.

PHP kodları, sunucu tarafından çıkış değeri olarak tarayıcıya gönderilmez. Yani, PHP kodları güvenli ve kullanıcıya görünmezdir. Örneğin; sunucu

```
<?php echo"<p>Hello World!</p>"; ?>
```

kodunu tarayıcıya `<p>Hello World!</p>` metni olarak gönderecek, tarayıcı da HTML paragraf etiketi olan `<p></p>` ifadelerini/ ikiliklerini/ arasındaki metni yeni bir paragraf açarak web sayfasında görüntüleyecektir.

PHP, birçoklu platform yazılımı olduğu için Windows, Mac OS X, Linux, FreeBSD, ve Solaris gibi birden fazla işletim sisteminde çalışabilmektedir. Ayrıca Apache, *Internet Information Server (IIS)*, Zeus ve lighttpd gibi web sunucuları PHP Motoru için yerleşik destek sunmaktadır. Apache Software Foundation'ın bir projesi olarak gelişimini sürdüren PHP, en yüksek performansı Apache sunucu ile yakalamaktadır. Apache Server Windows, Linux, Unix, Mac OS işletim sistemlerinde çalışabilen açık kaynak kodlu, ücretsiz, bir yazılımdır.

PHP, tüm web sunucuları ile bütünleşik yapıda olmasa da popüler pek çok web sunucusu ile çalışmaktadır (Valade ve Ballad, 2008).

2.2 HTML

Hypertext Markup Language (HTML), statik web sayfalarını hazırlamak için kullanılan ve web tarayıcıları tarafından yorumlanarak görüntülenen bir işaretleme dilidir.

Avrupa Nükleer Araştırma Merkezi' nin (*Conseil Européen pour la Recherche Nucléaire-CERN*) bilgi işlem bölümünde görevli olan fizikçi Tim Berners-Lee, CERN'de çalışan araştırmacıların belgelerini birbirleriyle ve dünya üzerindeki meslektaşları ile paylaşabilmelerini ve birlikte çalışabilmelerini sağlamak amacıyla kullanılacak bir sistemin gerekliliğini fark etmiş ve 1980 yılında ENQUIRE olarak anılan sistemi tasarlamıştır. Bu sistemde önce çok fazla belgenin dosya yapısında paylaşılması ve kullanıcıların bilgisayarına indirilebilir olması sağlanmış, daha sonra bu belgelerin birbirleri ile ilişkilendirilmesi fikri takip eden iki yıl içinde de önce İnternet - Tabanlı Hypertext yapısının gelişmesini sağlamıştır. Bu yapı ile kullanıcı bilgisayarına indirilebilir belge ve resimler birbirlerine bağlanarak, bir belgeyi içeren

sayfada konu ile doğrudan ilişkili belge ve çizelgelerin, bir form yapısı içinde, görüntülenebilmesi ve bağlantı düğmeleri ile seçilen belgeye geçiş sağlanmıştır. Daha sonra Berners-Lee, *Standard Generalized Mark-up Language (SGML)*'i temel alan HTML'i oluşturarak, tarayıcı ve sunucu yazılımlarını geliştirmiştir. SGML bağımsız bir dildir yani görüntüleneceği yazılıma ya da tarayıcıya bağlı değildir ve metin formatındaki metnin ekranda görüntülenmesini sağlar. HTML, <title> Sayfa Başlığı </title> gibi etiket çiftleri özelliğini ve <p>Paragraf</p>; H1'den H6'ya doğru azalan <h1> Ana Başlık </h1> ... <h6>Alt Başlık</h6>; Liste elemanları; Sırasız liste; Sıralı Liste etiketlerini SGML'den miras almıştır (URI-8). On sekiz element içeren ilk açık kaynak "HTML Tags" adı ile 1991 yılının sonlarında yayınlanmıştır. HTML5 ise son yayınlanan sürümüdür. En yaygın kullanılan sürümlerden birisi, 1999 yılında yayınlanan HTML 4.01'dir (URI-9).

Bir işaretleme dili olan HTML, işaretleme etiketleri ile oluşturulur ve etiketler ile belgenin içeriği tanımlanır. HTML uzantılı bir belge, HTML etiketlerinin ve düz metin yazısını içerir ve web sayfasını oluşturur.

HTML etiketleri açılı parantezler "< >" içerisine yazılan anahtar kelimelerden oluşur ve genellikle başlangıç etiketi ve bitiş etiketi olmak üzere çiftler halinde kullanılırlar. Bitiş etiketi, başlangıç etiketine benzer yapıdadır ancak açılı parantez ile anahtar kelime arasına eğik çizgi konularak oluşturulur. Bir HTML web sayfası oluşturulurken, tarayıcının HTML sayfası olduğunu anlayabilmesi için, belge tipini tarayıcıya anlatan bir bildirim yapılması gerekir. Bu bildirim, sürümlere göre şu şekillerde yapılabilir.

HTML5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML sayfası , <html>, <head> ve <body> etiket çiftlerinden oluşur. <!DOCTYPE html> ile tarayıcıya HTML5 sayfası olduğu bildirilen belgenin <html></html> etiketleri arasında kalan kısmı tarayıcı tarafından yorumlanacak kısımdır. <head></head> etiketleri arasında kalan kısım web sayfasında görüntülenmeyen

tarayıcıya sayfa hakkında başlık bilgisi, sayfada kullanılacak karakter setleri ve sayfa tasarımında kullanılacak stil gibi öznitelik bilgilerini sağlayan kısımdır. `<body></body>` etiketleri arasında kalan kısım web sayfasının içeriğini oluşturan kısımdır. Bu etiketler arasına yazılan her şey web sayfasında görüntülenir.

2.3 CSS

Cascading Style Sheets (CSS) HTML'e ek olarak metin ve format biçimlendirme alanında fazladan olanaklar sunan bir stil şablonudur.

İnternet sayfaları için genel geçer şablonlar hazırlama olanağı verdiği gibi, bağımsız olarak harflerin stilini, yani renk, yazı tipi, büyüklük gibi özelliklerini değiştirmek için de kullanılabilir. Bu tekniğin en önemli özelliği kullanımındaki esnekliktir.

Bir Web sayfası içerisinde birbiriyle uyumlu birkaç renk ve birkaç yazı tipi kullanılır ve bunları her sayfada ayrı ayrı tekrar belirtmek yerine CSS yardımıyla bir sefer tanımlayıp bütün Web sayfalarında ortak olarak kullanılabilir. Böylece sayfaların hafızadaki boyutu epey küçüldüğü gibi güncelleme yapmak da kolaylaşır. Web sayfalarının görselliğini iyileştirmek ve anlaşılabilirliğini arttırmak için HTML sayfalarında yer alan nesneler **id**'ler ve **class**'larla temsil edilir. Bu şekilde yorumlayıcının sayfa içindeki her elemente ayrı ayrı ulaşabilmesi sağlanır.

Stil şablonları, HTML dosyası içinde `<head>...</head>` veya `<body>...</body>` etiketleri arasında yer alabileceği gibi, sayfa düzenin ve nesnelerin stil bilgilerini içeren '#.css' uzantılı harici bir şablon dosyası `<head></head>` etiketleri içinde `<link rel="stylesheet" href="#.css"/>` koduyla çağırılarak çalışma sayfasına eklenebilir. CSS şablonlarını ayrı dosyada veya dosyalarda saklamak, aynı şablon yapısının farklı HTML sayfalarınca kullanılabilmesini sağlar ve aynı tarayıcılardan kaynaklanabilecek farklılıklara göre ayrı yapıların oluşturulabilmesini sağlar (URI-10).

2.4 Veritabanı

Veritabanı, birbirleri ile ilişkili verilerin belirli bir formatta tutulup, yönetildiği güncellenip geri çağırılabilirdiği organize edilmiş dosyalardır. Veritabanı yapısını, bilgi depolayan diğer yazılımlardan ayıran en önemli özellik verinin veritabanında hızlı ve

verimli bir şekilde yönetilip geri çağrılabilmesidir. Veri Tabanı Yönetim Sistemleri geliştirilmeden önce veritabanının oluşturulması, kontrollerinin yapılması ve düzenli dosyalarında tutulabilmeleri için programcılar kendi kodlarını yazmaları gerekmektedir. Ancak uzmanlık alanı gerektiren bu konu yüksek maliyete ve ticari programlarda performans kaybına neden olmaktaydı. Zamanla, veritabanı kullanımının yaygınlaşması ile veritabanı işlemlerini içeren kütüphaneler oluşturuldu ve daha sonra veritabanı işlemlerini bağımsız olarak yapan ve Veri Tabanı Yönetim Sistemi (VTYS / DBMS) olarak adlandırılan ayrı yazılımlar kullanılmaya başlandı. Bilgisayarların işlemci, bellek, disk ve ağ yapılarındaki teknolojik gelişmeler veritabanlarının da boyut, yetenek ve performanslarına ve kendi VTYS'lerine önemli ölçüde yansımıştır (URI-11).

Veri tabanları, verilere erişim ve verileri depolama bakımından farklılık gösterirler ve yapılarındaki bu farklılıklara göre farklı kavramsal modeller kullanılarak oluşturulabilirler. Bu modelleri üç ana başlık altında incelemek mümkündür. Bunlar, hiyerarşik veri modeli, ağ veri modeli ve ilişkisel veri modelidir.

Hiyerarşik veritabanı modelinde veriler arasındaki hiyerarşi, bir ağacın gövdesi, dalları ve yaprakları arasındaki hiyerarşiye benzetilerek yaratılmıştır. Tablolar arasında ebeveyn-çocuk (*parent-child*) ilişkisi vardır, yani tablolar arasında bire-çok ilişki kurulabilir. Ancak, herhangi bir sorgu için aramaya parent yani kök tablodan başlanır, doğrudan çocuk tablo üzerinden arama yapmak mümkün değildir.

Ağ veritabanı modeli, hiyerarşik veritabanından esinlenerek yaratılmıştır, farklı tablolar arasında çok-açok ilişki kurulmasıdır. Bir başka deyişle, bir child tablonun, birden fazla parent tablosu olabilir (URI-12).

İlişkisel model 1969 ve 1970 yıllarında IBM araştırmacısı E. F. Codd tarafından, hiyerarşik ve ağ veritabanı modellerindeki eksiklikleri gidermek ve büyük çapta veriler üzerinde daha kolay ve esnek işlem yapabilmek için geliştirilmiştir. Günümüzde en çok kullanılan veritabanı modellerinin başında gelir. İlişkisel sözcüğü veritabanında yer alan herhangi bir kaydın tek bir konu hakkında ve sadece o konuyla ilgili bilgileri içermesi gerektiğini ifade eder. İlişkisel veri tabanlarında saklanan tüm veriler tek bir yapı olarak düzenlenir ve tablolarda saklanır. Tablolar alanlardan (*fields*) ve alanlardaki kayıtlardan (*records*) oluşur. İlişkisel veritabanı modeli 'Anahtarlar'

(keys) kullanarak farklı içerikteki tabloların birbirlerine bağlanmasına izin verir. Herhangi bir tablo, birden fazla ebeveyn (*parent*) tabloyla ve aynı şekilde birden fazla çocuk(*child*) tabloyla ilişkilendirilebilir. Bir başka deyişle tablolar arasında hiyerarşi söz konusu değildir (URI-12).

Veri Tabanı Yönetim Sistemleri, kullanıcıların aşağı seviyeli programlama dillerini bilmelerine gerek duymadan, yüksek seviyeli bir arayüz üzerinden kullanım sunarlar. Genellikle İstemci-Sunucu (*Client-Server*) Mimarisi ile tasarlanan VTYS'ler internet aracılığı ile uzak bağlantı yapmaya olanak sağlarlar ve çok sayıda kişinin talebini aynı anda gerçekleştirme özelliğine sahiptirler. Erişim yetkilendirmesi ile kişilerin veri tabanları üçüncü şahısların kontrolsüz erişiminden korunur. Yedekleme ve geri çağırma özellikleri sayesinde veri güvenliği önemli ölçüde arttırılmıştır.

VTYS'lerin hepsi Yapılandırılmış Sorgu Dili'ni (*Structured Query Language*) desteklemektedir. SQL bir programlama dili değil, motor kodları C/C++ dillerinde yazılmış olan VTYS'ye istemcinin isteklerini ileten bir ara dildir (Aslan, 2013). SQL ilk olarak Donald D. Chamberlin ve Raymond F. Boyce tarafından IBM de SEQUEL adı altında geliştirilmiş 1970'lerin sonlarında ise İngiltere menşeli bir uçak firmasının resmi markasıyla aynı ismi taşıdığı için SQL olarak değiştirilmiştir.

- ORACLE firmasının ilk olarak SEQUEL' den esinlenerek geliştirdiği ve kendi adı ile anılan SQL tabanlı IVTYS ORACLE yazılımı,
- Microsoft firmasının SQL Server ürünü,
- IBM firmasının ilk VTYS olma özelliği taşıyan DB2 İlişkisel veritabanı yönetim sistemi,
- Oracle firmasının, MySQL' i satın alan Sun Firmasını satın almasıyla sahip olduğu ve açık kaynak kodlu MySQL yazılımı,
- Corel firmasının DOS ortamında çalışan Paradox yazılımı,
- PostgreSQL Global Development Grubu tarafından geliştirilmiş olan NIVTYS olan PostgreSQL

Günümüzde en çok kullanılan VTYS'lerdendir.

İlişkisel Veri tabanı Yönetim Sistemleri (*Relational Database Management Systems* - *RDBMS*) büyük miktarlardaki verilerin güvenli bir şekilde tutulabildiği, bilgilere

hızlı erişim imkânlarının sağlayan, bilgilerin bütünlük içerisinde tutulabildiği ve birden fazla kullanıcıya aynı anda bilgiye erişim imkânının sağlayan programlardır.

2.4.1 PostgreSQL

PostgreSQL, en gelişmiş açık kaynak kodlu ilişkisel veritabanı yönetim sistemlerinden birisidir. MIT-style olarak anılan ücretsiz yazılım lisansına sahiptir ve PostgreSQL lisansı ile dağıtılmaktadır. SQL sorgu dilini destekler ve Oracle firmasının geliştirdiği ORACLE ya da Microsoft firmasının SQL SERVER veritabanı yönetim sistemine, MySQL'den daha fazla benzerlik göstermektedir. (Riggs ve Krosing, 2010).

PostgreSQL, ilk olarak 1977 ve 1985 yılları arasında Berkeley Üniversitesinde, Michael Stonebraker önderliğinde, Ingres Projesi kapsamında ilişkisel bir veritabanı olarak geliştirilmiş ve Ingres adını almıştır. Daha sonra, Ingres kodları Relational Technologies tarafından ticari bir yazılıma dönüştürülmüştür. 1986 ve 1994 yılları arasında Stonebraker ve ekibi Ingres çekirdek yazılımına nesne ilişkili özellikler ekleyip yeni sürümü Postgres olarak yayınlamışlardır. Postgres' in ticari bir yazılıma dönüştürülmesi, sonraları Informix Şirketinin bir parçası olacak olan, Illustra şirketi tarafından gerçekleştirilmiştir. Postgres'e SQL desteği, 1994-1995 yılları arasında,

Andrew Yu ve Jolly Chen tarafından eklenmiş ve proje sonucunda aktif e-posta listesi olan, Postgres95 olarak adlandırılmıştır (Momjian, 2001). 1996 yılı boyunca, yeni bir sunucuya taşınan e-posta listesindeki bine yakın aboneden gelen hata geri bildirimleri sayesinde PostgreSQL sürümündeki pek çok hata ayıklanmış ve aynı yılın sonlarına doğru database sunucusunun adı PostgreSQL olarak değiştirilmiştir. İlk açık kaynak kodlar PostgreSQL 6.0 sürümü olarak, Sürüm Kontrol Sistemi (CVS) sunucusuna yerleştirilerek/üzerinden yayınlanmış ve kullanıcıların CVS sunucusuna uzaktan erişim ile güncellemeleri takip edip ihtiyaç duydukları güncellemeleri edinebilmeleri sağlanmıştır. 1997 yılından beri, yazılımın geliştirilmesi ve kontrolleri PostgreSQL Global Development Team önderliğinde, veritabanı geliştiricilerinin ve gönüllü kullanıcıların internet üzerinden yardımlarıyla devam etmektedir. Geniş kullanıcı grubu ve kaynak koduna dünyanın her yerinden erişilebilir olması nedeniyle olası hatalar çok çabuk kapatılmaktadır. PostgreSQL sürümleri Şekil 2.5 de gösterilmiştir.

Release	First release	Latest minor version	Latest release	Additions
0.01	1995-05-01	0.03	1995-07-21	Initial release as Postgres95
1.0	1995-09-05	1.09	1996-11-04	Changed copyright to a more liberal license
6.0	1997-01-29	—		Name change from Postgres95 to PostgreSQL, unique indexes, pg_dumpall utility, ident authentication.
6.1	1997-06-08	6.1.1	1997-07-22	Multi-column indexes, sequences, money data type, GEQO (Genetic Query Optimizer).
6.2	1997-10-02	6.2.1	1997-10-17	ODBC interface, triggers, server programming interface, constraints.
6.3	1998-03-01	6.3.2	1998-04-07	SQL92 subselect capability, PL/pgTCL.
6.4	1998-10-30	6.4.2	1998-12-20	VIEWS and RULEs, PL/pgSQL.
6.5	1999-06-09	6.5.3	1999-10-13	MVCC, temporary tables, more SQL statement support (CASE, INTERSECT, and EXCEPT)
7.0	2000-05-08	7.0.3	2000-11-11	Foreign keys, SQL92 syntax for joins
7.1	2001-04-13	7.1.3	2001-08-15	Write-ahead log, Outer joins
7.2	2002-02-04	7.2.8	2005-05-09	PL/Python, OIDs no longer required, internationalization of messages
7.3	2002-11-27	7.3.21	2008-01-07	Schema, Internationalization
7.4	2003-11-17	7.4.30	2010-10-04	Optimization all-round
8.0	2005-01-19	8.0.26	2010-10-04	Native server on Microsoft Windows, savepoints, tablespaces, exception handling in functions, point-in-time recovery
8.1	2005-11-08	8.1.23	2010-12-16	Performance optimization, two-phase commit, table partitioning, index bitmap scan, shared row locking, roles
8.2	2006-12-05	8.2.23	2011-09-26	Performance optimization, online index builds, advisory locks, warm standby
8.3	2008-02-04	8.3.23	2013-02-07	Heap-only tuples, full text search, SQL/XML, ENUM types, UUID types
8.4	2009-07-01	8.4.17	2013-04-04	Windowing functions, default and variadic parameters for functions, column-level permissions, parallel database restore, per-database collation, common table expressions and recursive queries
9.0	2010-09-20	9.0.13	2013-04-04	Built-in binary streaming replication, Hot standby, 64-bit Windows, per-column triggers and conditional trigger execution, exclusion constraints, anonymous code blocks, named parameters, password rules
9.1	2011-09-12	9.1.9	2013-04-04	Synchronous replication, per-column collations, unlogged tables, K-nearest-neighbor indexing, serializable snapshot isolation, writable common table expressions, SE-Linux integration, extensions, SQL/MED attached tables (Foreign Data Wrappers), triggers on views
9.2	2012-09-10	9.2.4	2013-04-04	Cascading streaming replication, index-only scans, native JSON support, improved lock management, range types, pg_receivexlog tool, space-partitioned GIST indexes

Community-supported
Community support ended^{1,3}

Şekil 2. 5: PostgreSQL Sürümleri (URI-13).

2.4.1.1 Gelişimi ve tarihçesi

PostgreSQL Linux, FreeBSD, Mac OS X ve Windows gibi tüm yaygın işletim sistemlerinde ve C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP gibi programlama dilleri ile çalıştırılabilir. PostgreSQL, Stored Procedures olarak adlandırılan, veritabanı sunucusunun kodları bloklar halinde çalıştırmasına izin veren, prosedürel bir dildir ve kodların SQL ve C dilleri haricindeki programlama dilleri ile de yazılabilmesine olanak sağlar. Bu sayede kullanıcıların subroutine, trigger, aggregates ve window fonksiyonları gibi fonksiyonlar tanımlamasına izin verilir. Oracle'ın PL/SQL prosedürel diline benzer olan PL/pgSQL, PL/Tcl, PL/Perl ve PL/Python PostgreSQL kurulumunda standart olarak gömülü (*built-in*) gelen prosedürel dillerdir. Ayrıca PL/Java, PL/JS (*JavaScript*), PL/PHP gibi prosedürel diller kullanıcının ihtiyaçları doğrultusunda sonradan eklenebilmektedir (URI-14).

PostgreSQL'de veriye komut satırı programı psql kullanarak ya da pgAdmin III, phpPgAdmin gibi grafiksel kullanıcı ara yüzlerini (*GUI*) kullanarak ulaşmak mümkün olduğu gibi çoklu kullanımlar için, Microsoft Windows işletim sisteminin kullanıcılarına sağladığı, Açık Veri Tabanı Bağlantı (*Open Database Connectivity*) sürücülerini ile, bir ağ üzerinden, sunucu üzerindeki veriye ulaşmak da mümkündür. PostgreSQL 9.3 sürümü, SQL dili için tanımlanmış olan ISO/IEC 9075 “Database Language SQL” standartlarının en son güncellemesi olan ISO/IEC 9075:2011'i ya da

kısaca SQL:2011'i desteklemektedir. İlk olarak 7.x serisi ile SQL-92 standartlarına uygun hale getirilen PostgreSQL zaman içerisinde SQL:1999, SQL:2003, SQL:2008 standartlarını sağlayarak gelişmiştir (URI-14).

Nesne-İlişkisel özelliği sayesinde PostgreSQL de veriler sınıf yapısına benzer özellik gösteren tablolarda tutulur ve kolaylıkla ilişkilendirilebilir.

PostgreSQL' in önemli özelliklerinden birisi de transaction yapısını desteklemesidir. Transaction yapısı arka arkaya gerçekleşmesi gereken parçalı işlemlerin, bir işlem serisi olarak, tek seferde, gerçekleşmesini garanti altına alır. BEGIN ve COMMIT kodları arasına yazılan transaction bloğu içindeki işlemlerin tamamı gerçekleşinceye kadar hiçbirisi gerçekleşmemiş varsayılır. Bu şekilde her bir işlem parçası başarılı şekilde gerçekleştikten sonra işlem tamamlanmış olur. Transaction işleminin iptal edilmesi için COMMIT komutunun ROLLBACK komutu ile değiştirilmesi yeterli olmaktadır (URI-15).

2.4.1.2 PHP ve postgresQL

PHP ile PostgreSQL veritabanı üzerinde gerçekleştirilecek işlemler için PHP kurulumunda PostgreSQL desteğinin etkinleştirilmiş olması gerekir. Bu işlem Unix ortamında PHP kurulumunda ./configure işlemine verilecek --with-pgsql parametresi ile yapılabilmektedir. --with-pgsql parametresine öntanımlı olarak PostgreSQL'in kurulu olduğu /usr/local/pgsql dizini gelmektedir.

--with-pgsql = /usr/local/pgsql

İsteğe bağlı olarak bu dizin PostgreSQL' in kurulduğu adresi gösterecek şekilde değiştirilebilir. Windows ortamında pgsql modülünü etkinleştirmek için **php.ini** dosyasındaki **;extension = php_pgsql.dll** satırının başında bulunan ';' (noktalı virgül) işaretini kaldırmak yeterli olacaktır (Gilmore ve Treat, 2006). PHP'nin PostgreSQL desteği **phpinfo()** fonksiyonu ile sorgulanabilmektedir. Bunun için alt satırdaki kodu **phpinfo.php** olarak web sunucusunun htdocs klasörüne kaydedip, tarayıcıda görüntülemek yeterli olacaktır.

```
<?php
    phpinfo();
?>
```

Web sunucu olarak kurulan hazır derlenmiş paketlerde PostgreSQL modülü kurulu olarak geldiği için etkinleştirme işlemine gerek kalmamaktadır. Tez çalışması kapsamında PostgreSQL modülünü içeren tek hazır paket olan Bitnami Wapp Stack kullanılmıştır.

PHP'nin PostgreSQL modülü sistemde kurulu olan libpg kütüphane fonksiyonları ile programcı arasında bir ara yüz görevi göstermektedir. Bu modül ile PHP üzerinden PostgreSQL'in, libpg kütüphanesinde tanımlı, tüm fonksiyonları kolaylıkla kullanılabilir (Yazıcı, 2006).

3. ALAGÖZ WEB UYGULAMASI

ALAGÖZ web uygulamasının amacı ALAGÖZ Projesi'nin dâhilinde tasarlanacak kampanyalar kapsamında Türkiye genelinde yapılacak gözlemlerle toplanacak GNSS verilerinin bir veritabanında -web üzerinden- güvenli bir şekilde saklanabilmesini, değerlendirilebilmesini ve paylaşılabilmesini sağlayacak sistemin hayata geçirilebilmesidir.

Tasarlanan sistemin Uluslararası Standartlar Teşkilatı'nın Coğrafi Bilgi/Geomatik Komitesi tarafından 191XX isimlendirmesi ile dijital ortamda coğrafi veri yönetimi için geliştirilen standartlar (*International Organization for Standardization - Technical Committee211, ISO-TC211*), Açık Coğrafi Bilgi Konsorsiyumu'nun (*Open Geospatial Concoortium-OGC*) standartları, Avrupa Konumsal Bilgi Altyapısı - Coğrafi Veri Standartları'na (*Infrastructure for Spatial Information in the European Community, INSPIRE*) uygun olması gözetilerek birlikteçalışabilirliğin (*inteoperability*) sağlanması planlanmıştır (URL-16).

3.1 Teknik Özellikler

ALAGÖZ web sitesi tasarımında, PostgreSQL Modülü ya da desteği olan açık kaynak kodlu, ücretsiz, tek yazılım olması ve ve hızlı kurulumu ile zaman kazandıran Bitnami yazılımı kullanılmıştır. Bitnami'nin Windows İşletim Sisteminde çalışan iki uygulaması mevcuttur. Bunlar PostgreSQL Modülü'nün sonradan eklenebileceği ve kurulumunda Apache web sunucusunu, PHP programlama dilini, MySQL veritabanını ve phpMyAdmin'i hazır getiren Bitnami WAMP Stack ile Apache web sunucusunu, PHP programlama dilini, PostgreSQL veritabanını ve phpPgAdmin' i hazır olarak getiren Bitnami WAPP Stack'dir. Yazılımın Linux ortamında çalışabilen LAMP Stack, DOS OSX ortamında çalışabilen MAMP Stack uygulaması gibi uygulamaları da mevcuttur. Tez çalışmasında, BITNAMI'nin wampstack-5.4.26 sürümü kullanılmış ve postgresQL Modülü wampstack'in üzerine eklenip, PostGIS ile ilişkilendirilmiştir.

Modüller mevcut kurulumun üstüne eklenebileceği gibi ayrıca da kurulabilmektedir. Çalışmada PHP kodları HTML5’ in içine gömülmüş ve grafik tasarım/görselleştirme CSS ile yapılmıştır.

Ayrıca projede, geometrik veri tipi sağladığı için PostgreSQL VTYS tercih edilmiştir. PostgreSQL’de desteklenen geometrik veri tipi, iki boyutlu mekânsal nesneleri temsil etmektedir. Coğrafi Bilgi Sistemlerinde kullanılan vektörel veri tiplerine benzer özellikler gösteren geometrik veri tipleri nokta (*point*), çizgi (*line*) ve alan (*polygon*)’dan oluşmaktadır. Alan ve çizgi veri tipleri nokta tipini temel almaktadır (URI-17). Geometrik veri tipleri ve uzunluk bilgileri Şekil 3.1 de gösterilmiştir.

Name	Storage Size	Representation	Description
point	16 bytes	Point on the plane	(x,y)
line	32 bytes	Infinite line (not fully implemented)	((x1,y1),(x2,y2))
lseg	32 bytes	Finite line segment	((x1,y1),(x2,y2))
box	32 bytes	Rectangular box	((x1,y1),(x2,y2))
path	16+16n bytes	Closed path (similar to polygon)	((x1,y1),...)
path	16+16n bytes	Open path	[(x1,y1),...]
polygon	40+16n bytes	Polygon (similar to closed path)	((x1,y1),...)
circle	24 bytes	Circle	<(x,y),r> (center and radius)

Şekil 3. 1: Geometrik veri tipleri (URI-17).

Web Sitesinin üyelik sistemi, veri girişini ve çıkışını sağlayacak form yapıları, toplanan verilerin görselleştirilmesi, kampanya planlamasında görev bildiriminin yapılmasını sağlayacak e-posta gurubu ve duyuru sistemlerini içermesi planlanmıştır.

Üyelik sistemi, site yöneticisinin kontrolü ile sürdürülecektir. Katılımcı adayları hakkında bilgi toplayan başvuru formları (*registration_page*) katılımcı tarafından doldurulduktan sonra, yöneticilere posta ile bildirilecek ve site yöneticileri için oluşturulmuş yönetici sayfasında (*admin_page*) görüntülenebilecektir. Katılımcı bilgileri üzerinde düzenleme yetkisine sahip olan yöneticinin katılımcı isteğini onaylaması durumunda, katılımcının siteye üyeliği kesinleşecektir. Üyeliği onaylanan katılımcı adayına e-posta ile geri bildirim yapılacak ve siteyi tekrar ziyaret ettiğinde başvuru formunda sağladığı e-posta adresi ve şifre bilgileri ile giriş sayfasından (*login_page*) sisteme katılımcı olarak giriş yapabilmesi sağlanacaktır.

Adaylığı onaylanan katılımcının, sisteme giriş yapabilmesi için öncelikle giriş formundan alınan bilgilerin doğru formatta girilip girilmediğinin, veritabanına

kaydedilmiş olan bilgilerle eşleşip eşleşmediğinin kontrolünün yapılması gerekmektedir. Bilgilerin yanlış girilmesi durumunda, kullanıcıya bir uyarı metni döndürülerek, yeniden denemesi istenecektir. Giriş işlemi başarılı olduğunda, kullanıcı için tekil (*unique*) olan bir ID ile oturum (*session*) bilgisi oluşturularak web sitesinin yayınlandığı web sunucusuna kaydedilecek, bu sayede her bir kullanıcı için site içi faaliyetlerine göre özel bir içerik oluşturulması sağlanacaktır. Ayrıca web sitesinin kullanıcıyı tanıyabilmesi için, kullanıcıyı web sayfasına tanıtacak ve kullanıcının bilgisayarında saklanacak çerezler (*cookie*) ile web tarayıcısının kullanıcı bilgilerini siteden çıkış yapılması halinde de saklaması ve kullanıcının siteye aylar sonra yapacağı ziyaretlerde bile kullanıcıyı tanıması sağlanacaktır.

JeoVT projesinin form yapısı temel alınarak oluşturulan Proje/Kampanya, Kurum, Birim, Katılımcı, İstasyon, Alıcı ve GNSS formları ile katılımcıya veri girişi, düzenleme, silme yetkileri ile birlikte kampanyalar dâhilinde başka katılımcılar tarafından siteye eklenecek verilere erişme ve veri dosyalarını kendi ortamında saklama olanağı sağlanacaktır.

3.2 Veritabanı Tasarımı

ALAGÖZ web uygulaması kapsamında Data Manipulation Language (DML) komutlarını gerçekleştirmek üzere bir Online Transaction Processing (OLTP) veritabanı tasarlanmıştır. Proje kapsamında düzenlenecek kampanyalar ile sahadan toplanacak GNSS verilerinin saklanması ve sorgulanabilmesi için oluşturulacak ilişkisel veritabanı tasarlanırken öncelikle proje için gerekli olan veri tipleri ve bilgiler tespit edilmiştir. Gerekli veriler Kampanya, İstasyon, Katılımcı, GNSS Ölçü, GNSS Oturum ve Kurum varlıkları ile gruplandırılmıştır.

Varlıklara ait nitelikler, varlıklar arasındaki ilişkiler ve ilişkilerin nitelikleri (bire-bir, bire-çok ve çok-a-çok) belirlenerek kavramsal model oluşturulmuştur.

Kavramsal Model, VTYS bileşenlerine dönüştürülürken veri tekrarlarının önlenmesi, verilerin tutarlı olması ve bütünlüğün sağlanması için tablo yapısı normalizasyon kuralları çerçevesinde oluşturulmuş ve varlıklar arası mantıksal bağlantılar kurularak Mantıksal Model oluşturulmuştur.

Kavramsal Model ve Mantıksal Model tasarımı tamamlandıktan sonra phpPgAdmin ve pgAdminIII yönetim panellerinden yararlanılarak Fiziksel Model oluşturulmuştur. Fiziksel Model oluşturulurken tablolar arasında çoka-çok ilişki kurabilmek için birden fazla/birleşik birincil anahtar (*primary_key*) içeren (*composit_key*) bağlantı tablolarından (*junction table*) yararlanılmıştır.

Veritabanı kampanya, istasyon, kullanıcı, kampanya_istasyon, kampanya_kullanıcı, gnss_ölçü, gnss_oturum, alıcı, uydu_sistemi, kullanıcı_gnss_oturum, kurum, birim, bölge, il, ilçe, mahalle tablolarından oluşmaktadır.

Kampanya tablosu, proje kapsamında düzenlenecek kampanyalar hakkındaki bilginin tutulması için planlanmıştır. Tablo niteliklerinde kampanya_id birincil anahtar olarak atanmış ve veri tipi BIGSERIAL seçilerek otomatik artış sağlanmıştır. Tabloda kampanya adı, içeriği, amacı, başlangıç ve bitiş tarihleri, bulunduğu bölge, il, ilçe bilgisi, kampanya yöneticisi notları ve ek_dosya nitelikleri bulunmaktadır. Ayrıca kampanya tablosu, il_id, kurum_id ve birim_id yabancı anahtarları aracılığıyla il, kurum ve birim tabloları ile ilişkilendirilmiştir. Kampanya tablosu Şekil3.2 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
campaign_id	bigint	NOT NULL	nextval('campaign_campaign_id_seq'::regclass)	
campaign_name	character varying(50)	NOT NULL		
content	text	NOT NULL		
aim	text	NOT NULL		
begin_date	date	NOT NULL		
end_date	date	NOT NULL		
county_id	integer			
village_id	integer			
notes	text			
file_1	oid			
file_2	oid			
file_3	oid			
file_4	oid			
deleted	smallint	NOT NULL 0		
corporation_id	integer			
unit_id	integer			

Şekil 3. 2: Kampanya Tablosu.

İstasyon tablosu, gerçekleştirilecek kampanyalarda kullanılacak mevcut istasyonlar ve tespit edilecek yeni istasyonlar hakkındaki bilginin tutulması için planlanmıştır. İstasyon_id niteliği birincil anahtar olarak atanmış veri tipi BIGSERIAL seçilerek

otomatik olarak artış sağlanmıştır. Tablo: istasyon adı, istasyon numarası, hesap yöntemi, koordinat tipi, sahadan toplanacak yaklaşık koordinat verileri, dönüştürülmüş coğrafi koordinatlar, datum bilgisi, doğruluk, kesinlik, epok bilgisi, dilim orta meridyeni, dilim genişliği, gravite, topografik düzeltme, istasyon tipi ve tesis tarihi, pafta adı, kroki, röper, ulaşım bilgisi ile notlar niteliklerinden oluşmaktadır. İstasyon tablosu Şekil 3.3 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
station_id	bigint	NOT NULL	nextval('station_station_id_seq'::regclass)	
station_no	character varying			
station_name	character varying			
x1	character varying			
vx1	character varying			
y1	character varying			
vy1	character varying			
z1	character varying			
vz1	character varying			
coordinate_type	smallint			
datum	smallint	NOT NULL		
precision	smallint	NOT NULL		
epoch	numeric			
grid_zone_central_meridian	smallint			
grid_zone_width	smallint			
gravity	character varying(50)			
topographic_correction	character varying(50)			
establishment_date	date			
establishment_type	character varying(50)			
map_sheet_name	character varying(50)			
transportation	character varying(250)			
sketch_map	character varying			
benchmark	character varying			
calculation_method	character varying(100)			
accuracy	character varying(32)			
notes	text			
X	numeric			
Y	numeric			
Z	numeric			
deleted	smallint	NOT NULL	0	

Şekil 3. 3: İstasyon Tablosu.

Kampanya_İstasyon tablosu, çoka-çok ilişki kurabilen kampanya ve istasyon tablolarının ilişkilendirilebilmesi için oluşturulmuş bir bağlantı tablosudur. Bağlantı tablosunun her iki tablo ile ayrı ayrı bire-çok ilişki kurabilmesi için kampanya ve istasyon tablolarının birincil anahtarlarını (PK) yabancı anahtar (FK) olarak içermektedir. Tablonun birincil anahtarı ise her iki tablonun birincil anahtarlarından oluşan bir birleşik anahtardır (CK) . Kampanya-İstasyon tablosu Şekil 3.4 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
campaign_id	integer	NOT NULL		
station_id	integer	NOT NULL		
deleted	smallint	NOT NULL	0	

Şekil 3. 4: Kampanya_İstasyon Tablosu.

Katılımcı tablosu, ALAGÖZ web sitesi üyelik sistemine dâhil edilecek ve kampanyalarda görev alacak katılımcılara ait bilgilerin tutulması için tasarlanmıştır. Tabloda, aday anahtar olan T.C. numarası girişi isteğe bağlı tutulduğu için birincil anahtar olarak seçilmemiş ve kişiye özel olduğu için veri tipi unique olarak belirlenmiştir. Index işlemlerinde hız kazanmak için otomatik olarak artan kullanıcı_id niteliği birincil anahtar olarak atanmış ve veri tipi BIGSERIAL olarak seçilmiştir. Ayrıca kullanıcıya ait T.C. numarası, adı, soyadı, telefon numarası, fax, adres bilgisi, e-posta, mevcut ise web sitesi ve linkedin adresleri ve katılımcının olası notları tutulmaktadır. Tablo kurum_id, birim_id ve ilçe_id ile diğer tablolarla ilişkilendirilmiştir. Katılımcı tablosu Şekil 3.5 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
participant_id	bigint	NOT NULL	nextval('participant_participant_id_seq'::regclass)	
corporation_id	integer	NOT NULL		
unit_id	integer	NOT NULL		
county_id	integer	NOT NULL		
turkish_id_no	integer			1
first_name	character varying(32)	NOT NULL		
last_name	character varying(32)	NOT NULL		
phone1	character(10)	NOT NULL		
phone2	character(10)			
adres	text	NOT NULL		
postal_code	character(5)	NOT NULL		
e-mail	character varying(32)	NOT NULL		
web	character varying(32)			
linkedin	character varying(32)			
fax	character varying(12)			
notes	text			
deleted	smallint	NOT NULL	0	

Şekil 3. 5: Katılımcı Tablosu.

Kampanya_Katılımcı tablosu, çoka-çok ilişki kurabilen kampanya ve katılımcı tablolarının ilişkilendirilebilmesi için oluşturulmuş bir bağlantı tablosudur. Bağlantı tablosunun her iki tablo ile ayrı ayrı bire-çok ilişki kurabilmesi için kampanya ve katılımcı tablolarının birincil anahtarlarını (PK) yabancı anahtar (FK) olarak içermektedir. Tablonun birincil anahtarı ise her iki tablonun birincil anahtarlarından

oluşan kampanya_katılımcı_pkey birleşik anahtarıdır. Kampanya-Katılımcı tablosu Şekil 3.6 da gösterilmiştir.

Column	Type	Not Null	Default	Constraints
campaign_id	integer	NOT NULL		
participant_id	integer	NOT NULL		
deleted	smallint	NOT NULL	0	

Şekil 3. 6: Kampanya_Katılımcı Tablosu.

GNSS_Oturum Tablosu, kampanyalar dâhilinde yapılacak oturum bilgilerinin tutulması için tasarlanmıştır. Birincil anahtar olarak oturum_id seçilmiş, kampanya_id yabancı anahtarı ile kampanya tablosu ile ilişkilendirilmiştir. Tabloda ayrıca oturum adı, oturumun başlangıç ve bitiş tarihleri, veri aralığı, yükseklik açısı, katılımcı notları ve kayıt_çizelgesi_oid nitelikleri bulunmaktadır. Kayıt çizelgesi veritabanına dosya yükleme işlemi ile eklenecektir ve bunun için veri tipi binary formatta olan BLOB (*Binary Large Object*) olarak seçilmiştir. BLOB veriler, veritabanında ilgili tabloda tutulmak yerine ayrı bir tabloda tutularak ilgili tablo ile kayıt oid (*object id*)’ si üzerinden ilişkilendirilmektedir. GNSS-Oturum tablosu Şekil 3.7 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
settlement_id	bigint	NOT NULL	nextval('gnss_settlement_id_seq'::regclass)	
campaign_id	integer	NOT NULL		
settlement_name	character varying(50)	NOT NULL		
begin_date	timestamp without time zone	NOT NULL		
end_date	timestamp without time zone	NOT NULL		
record_chart	oid			
data_interval	character varying(32)	NOT NULL		
angle_of_elevation	character varying	NOT NULL		
notes	text			
deleted	smallint	NOT NULL	0	

Şekil 3. 7: GNSS_Oturum Tablosu.

Katılımcı_GNSS_Oturum tablosu, bir katılımcı birden fazla oturumda görev alabileceği için ve bir oturumda birden fazla katılımcı görev alabileceği için Katılımcı ve GNSS_Oturum tabloları arasında çoka-çok ilişki söz konusudur. Bu ilişkinin fiziksel olarak kurulabilmesi için katılımcı_gnss_oturum bağlantı tablosu oluşturulmuştur. Tablo katılımcı_id ve gnss_oturum_id birincil anahtarlarından oluşan birleşik anahtarı ve katılımcı_id ile gnss_oturum_id yabancı anahtarlarından oluşmaktadır. Katılımcı-GNSS tablosu Şekil 3.8 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
participant_id	integer	NOT NULL		
gnss_settlement_id	integer	NOT NULL		
deleted	smallint	NOT NULL	0	

Şekil 3. 8: Katılımcı_GNSS_Oturum Tablosu.

GNSS_Ölçü tablosu, GNSS oturumları ile yapılacak alımlara dair bilgilerin ve ölçme verilerinin tutulması için tasarlanmıştır. Birincil anahtar olarak ölçme_id seçilmiş, istasyon_id yabancı anahtarı ile istasyon tablosuyla, oturum_id yabancı anahtarı ile GNSS Oturum tablosuyla, uydu_sistemi_id yabancı anahtarı ile uydu sistemleri tablosuyla, seri_no yabancı anahtarı ile alıcı tablosuyla ilişkilendirilmiştir. Tabloda ek olarak istasyon tipi, ölçme metodu, alet yüksekliği, alet yüksekliği türü, doğruluk ve ölçmeyi yapan katılımcı notları nitelikleri bulunmaktadır. Alıcıdan RINEX formatta alınan meteoroloji, navigasyon ve ölçü dosyaları için veri tipi ölçü_oid, navigasyon_oid, meteoroloji_oid ile kaydedilmek üzere BLOB olarak seçilmiştir. GNSS-Ölçü tablosu Şekil 3.9 da gösterilmiştir.

Column	Type	Not Null	Default	Constraints
measurement_id	bigint	NOT NULL	nextval('gnss_measurement_measurement_id_seq'::regclass)	
settlement_id	integer	NOT NULL		
satellite_system_id	integer	NOT NULL		
station_id	integer	NOT NULL		
station_type	smallint	NOT NULL		
measurement_method	smallint	NOT NULL		
receiver_height	numeric	NOT NULL		
receiver_height_type	smallint	NOT NULL		
measurement_file	oid	NOT NULL		
navigation_file	oid	NOT NULL		
meteorology_file	oid	NOT NULL		
accuracy	character varying	NOT NULL		
notes	text	NOT NULL		
serial_no	character varying(16)	NOT NULL		
deleted	smallint	NOT NULL	0	

Şekil 3. 9: GNSS_Ölçü Tablosu.

Alıcı tablosu, kampanya kapsamında belirlenecek alanlarda mevcut ya da tesisi yapılacak istasyonlarda alım yapmak üzere kullanılacak ölçme aleti hakkındaki bilgilerin tutulacağı tablodur. Her bir alet için tek olan seri_numarası aday anahtarı, birincil anahtar olarak seçilmiştir. Bir aletle birden fazla ölçme yapılabileceği için GNSS Ölçme tablosu ile ilişkilendirilmiştir. Tabloda alıcı ismi, üretici firma, alet doğruluğu bilgileri de tutulmaktadır. Alıcı tablosu Şekil 3.10 da gösterilmiştir.

Column	Type	Not Null	Default	Constraints
serial_no	character varying(16)	NOT NULL		
receiver_name	character varying(16)	NOT NULL		
producer_firm	character varying(32)	NOT NULL		
accuracy	character varying(10)	NOT NULL		
notes	text			
deleted	smallint	NOT NULL	0	

Şekil 3. 10: Alıcı Tablosu.

Uydu Sistemleri tablosu, alım esnasında geriden kestirme yöntemi ile istasyonun konumunu belirlemek amacıyla gerçek konumundan yararlanılmak üzere seçilecek uydu sisteminin bilgisini tutmak ve GNSS Ölçme tablosu ile ilişkilendirilmek üzere tasarlanmış bir yardımcı tablodur. Uydu sistemi id' si birincil anahtar olarak atanmıştır. ALAGÖZ web sitesi kullanıcısının GNSS Ölçme formuna veri girişi esnasında, uydu sistemleri listesi (select list/drop down list) içinden seçim yapabilmesi için GPS, GLONASS ve GALILEO sistemleri veritabanına eklenmiş ve GPS Uydu sistemleri yaygın olarak kullanıldığı için varsayılan değer olarak atanmıştır. Udu Sistemleri tablosu Şekil 3.11 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
satellite_system_id	integer	NOT NULL	nextval('satellite_system_satellite_system_id_seq'::regclass)	
satellite_system_name	character varying(10)	NOT NULL		

Şekil 3. 11: Uydu Sistemleri Tablosu.

Kurum Tablosu, kampanyalarda yürütücü olarak görev alacak olan kurumların ya da katılımcıların bağlı olduğu kurumların bilgisini tutmak üzere hazırlanmıştır. Birincil anahtar olarak BIGSERIAL veri tipindeki kurum_id'si atanmıştır. Katılımcı ve kampanya tabloları ile ilişkilendirilmiştir. Kurum tablosunun adres bilgileri için kullanılacak ilçe listesi, ilçe tablosunun birincil anahtarı ile ilişkilendirilmiştir. Tabloda ayrıca kurum adı, telefon numarası, faks numarası, e-posta adresi, web sitesi adresi, kurum bünyesinde iletişime geçilecek kişi hakkında bilgi ve notlar nitelikleri bulunmaktadır. Kurum tablosu Şekil 3.12 de gösterilmiştir.

Birim tablosu, her bir kurumun bünyesindeki birimler için tasarlanmış bir tablodur. Birincil anahtar olarak birim_id kolonu/ sütunu atanmıştır. Kurum tablosunda olduğu gibi adres bilgileri için ilçe tablosu ile ilişkilendirilmiş ve birim adı, iletişim bilgileri için nitelikler eklenmiştir. Birim tablosu Şekil 3.13 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
corporation_id	bigint	NOT NULL	nextval('corporation_corporation_id_seq'::regclass)	
corporation_name	character varying(50)	NOT NULL		
address	character varying(50)	NOT NULL		
phone_1	character(10)	NOT NULL		
phone_2	character varying(13)	NOT NULL		
fax	character varying(13)			
e_mail	character varying(50)	NOT NULL		
web_site	character varying(50)			
contact_info	character varying(50)			
notes	character varying(50)			
deleted	smallint	NOT NULL 0		
county_id	integer	NOT NULL		

Şekil 3. 12: Kurum Tablosu.

Column	Type	Not Null	Default	Constraints
unit_id	bigint	NOT NULL	nextval('unit_unit_id_seq'::regclass)	
corporation_id	integer	NOT NULL		
unit_name	character varying(50)			
address	character varying(50)	NOT NULL		
phone_1	character varying(13)	NOT NULL		
phone_2	character varying(13)			
fax	character varying(13)			
e_mail	character varying(50)	NOT NULL		
web	character varying(50)			
contanc_info	character varying(50)	NOT NULL		
notes	character varying(50)			
deleted	smallint	NOT NULL 0		
county_id	integer	NOT NULL		

Şekil 3. 13: Birim Tablosu.

Bölge tablosu, kampanyalar zinciri şeklinde gerçekleşecek projenin her bir kampanyasında bölgeyi/sınırları belirlemek ve katılımcı veya kurumların adres bilgilerinde çoktan seçmeli listeler olarak formların içine döndürülmek amacıyla önceden hazırlanmış, Türkiye'nin bölgelerini içeren tablodur. Bölge_id birincil anahtarını içerir. Bölge tablosu Şekil 3.14 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
region_id	integer	NOT NULL	nextval('region_region_id_seq'::regclass)	
region_name	character varying(16)	NOT NULL		
deleted	smallint	NOT NULL 0		

Şekil 3. 14: Bölge Tablosu.

İl tablosu, Türkiye'nin seksen bir ilinin listesini içeren ve illerin bulunduğu bölgeyi göstermek için Bölge tablosu ile ilişkilendirilmiş bir tablodur. İl tablosu Şekil 3.15 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
province_id	integer	NOT NULL	nextval('province_province_id_seq'::regclass)	
region_id	integer	NOT NULL		
province_name	character varying(32)	NOT NULL		
deleted	smallint	NOT NULL	0	

Şekil 3. 15: İl Tablosu.

İlçe tablosu, her bir ilin ilçelerinin listelenebilmesi ve gerektiğinde veritabanından çağırılabilmesi için tasarlanmış bir tablodur. İl tablosu ile ilişkilendirilebilmesi için il tablosunun birincil anahtarını yabancı anahtar olarak almıştır. İlçe tablosu Şekil 3.16 da gösterilmiştir.

Column	Type	Not Null	Default	Constraints
county_id	integer	NOT NULL	nextval('county_county_id_seq'::regclass)	
province_id	integer	NOT NULL		
county_name	character varying(32)	NOT NULL		
deleted	smallint	NOT NULL	0	

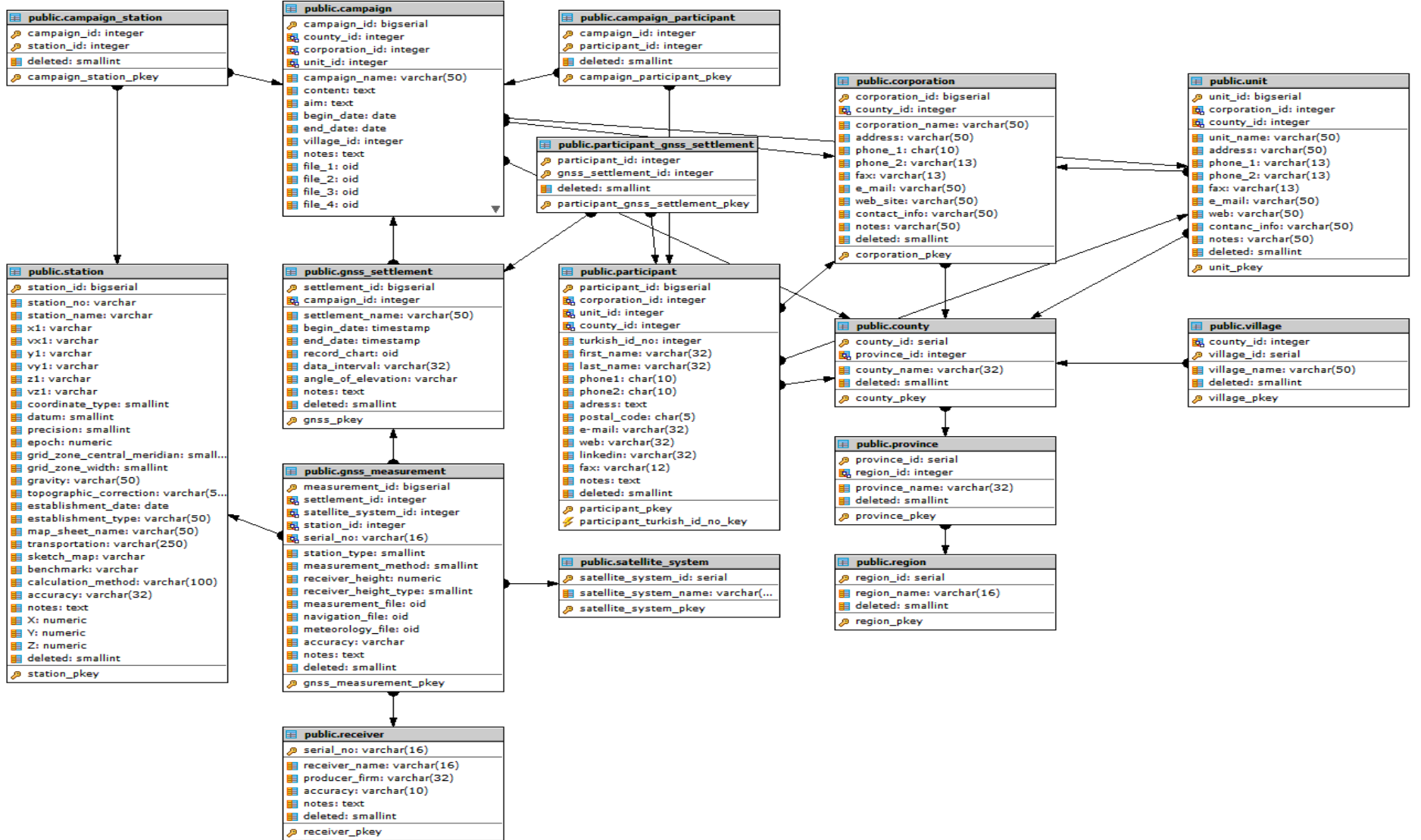
Şekil 3. 16: İlçe Tablosu.

Mahalle tablosu, ilçelerin mahalle bilgisini içeren tablodur. İlçe tablosu ile ilişkilendirilmiştir. Mahalle tablosu Şekil 3.17 de gösterilmiştir.

Column	Type	Not Null	Default	Constraints
county_id	integer	NOT NULL		
village_name	character varying(50)	NOT NULL		
deleted	smallint	NOT NULL	0	
village_id	integer	NOT NULL	nextval('village_village_id_seq'::regclass)	

Şekil 3. 17: Mahalle Tablosu.

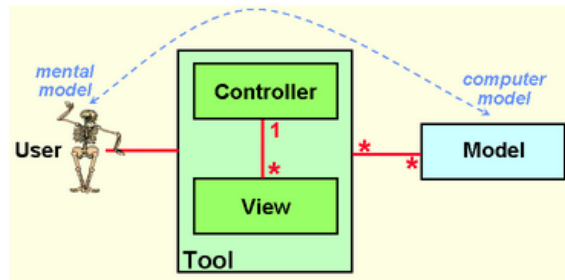
Bölge, il, ilçe ve mahalle tabloları hiyerarşik yapıda oldukları için tek bir yabancı anahtar ile tablolar arası geçiş mümkün olmaktadır. Bunun için adres bilgisinin tutulacağı tablolarda bölge, il ve ilçe tablolarının her biriyle ayrı ayrı ilişki kurulması yerine, PHP programlama dili ile ilgili veri döndürülecektir. Örneğin; bir kampanyanın kapsama alanında kalan ilçelerin veritabanında tutulabilmesi için: veritabanından çağılan bölge listesinde seçilen bölge_id ile ilişkili iller listesinden seçilecek ilin ilçeleri arasından seçilmesi yeterlidir. Yani, veritabanında kampanyanın bölge, il, ilçe bilgilerinin tutulması yerine sadece ilçe bilgileri tutulmuş olur. Bu şekilde normalizasyon kuralları gerçekleştirilerek veri tekrarının önüne geçilmiş olmaktadır. Hiyerarşinin en altında bulunmamasına rağmen her uygulama sınırı bir mahalle birimi içermeyeceğinden tablolarda ilçe verisi tutulmaktadır. İlgili tablolar bu tablo ile ayrıca ilişkilendirilmiştir. Varlık ilişki modeli aşağıdaki gibidir.



Şekil 3. 18: Varlık İlişki Modeli.

3.3 Web Yazılım

Web yazılımında, fonksiyonlardan oluştuğu için uygulamaya yönelik olan , düşük okunabilirliğe sahip, büyük oranda kod tekrarlarından oluşan dolayısı ile bilgi çıkarımı (*data mining*), hata tespiti ve geliştirilmesi zor olan Spagetti kod düzeni yerine, tekrar tekrar kullanılabilir sınıflardan (nesneler) oluşan, okunabilirliği yüksek, gerçek hayatın konu ve kavramlarının modellendiği, gelişime açık ve hata tespiti kolay olan nesne yönelimli programlama (*Object Oriented Programing, OOP*) tercih edilmiştir. Uygulama kodlarının düzenli, sistematik olması ve anlaşılabilirliğinin artırılması için sınıflar/ modüller görevlerine göre farklı katmanlarla gruplandırılmıştır. Separation of Concerns olarak anılan bu yöntemle, uygulamanın veri işleme ve yorumlama kuralları/ algoritması (*Business/Domain Logic*) ile kullanıcının etkileşeceği arayüz gibi farklı alanların birbirinden ayrılması sağlanarak uygulamanın geliştirilmesi sürecinde esnekliğin önü açılmış ve yönetimi kolaylaştırılmıştır. Bunun için, kullanıcıya yüklü miktarda verinin sunulduğu karmaşık uygulamalarda kontrol kolaylığı sağlayan ve veri ile gösterimin soyutlanması (*encapsulation*) esasına dayanan Model Görünüm Denetçiler (*Model View Controller, MVC*) mimari deseninden yararlanılmıştır. 1979 yılında, Smalltalk programcısı Trygve Reenskaug tarafından insan zihnindeki model ile bilgisayardaki digital model arasında köprü olmak üzere tanımlanan MVC daha sonra web uygulamalarında da kullanılmaya başlanmıştır. (URI-18) MVC'nin çalışma prensibi Şekil 3.19 da gösterilmiştir.



Şekil 3. 19: MVC' nin çalışma biçimi (URI-16).

Başlangıçta, thin client olarak adlandırılan ve tek başına var olamayıp başka bilgisayar ya da programlarla tam olarak çalışabilen MVC yapısında tüm model, görünüm(view) ve denetçiler (*controller*) sunucu üzerinde çalışıyordu. Bu yaklaşımda, model tamamen sunucu üzerinde yer almakta, istemcinin (*client*) istekleri

referanslar/referans-yollar (hyperlink) ya da form girdileri ile denetçiye (*controller*) iletilirken, tamamlanmış ya da güncellenmiş web sayfası görünüm (view) üzerinden görüntüleniyordu. Zaman içerisinde geliştirilen Javascript ve Backbone gibi frameworkler ile MVC bileşenlerinin istemci tarafında da çalıştırılabilmesi mümkün olmuştur (URI-18).

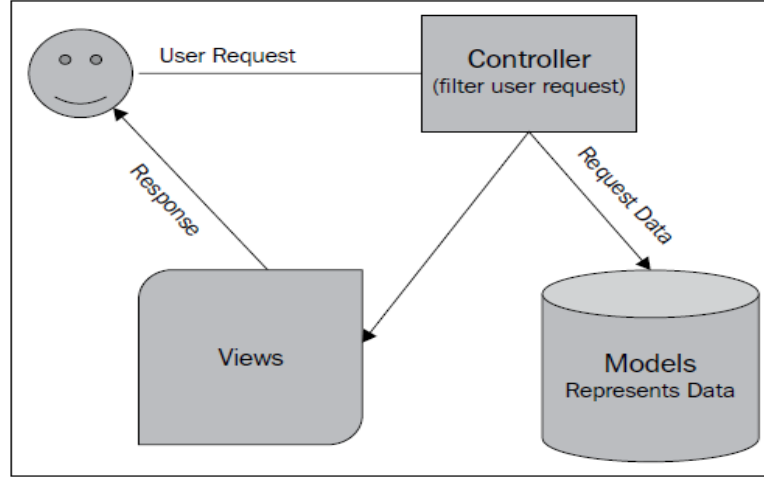
Nesne yönelimli programlama dillerinde kullanılan MVC yapısı kullanıcı arayüzü ile uygulamanın veri modellerini birbirinden ayırmak için Model View ve Controller Katmanları'ndan yararlanır. Model View Controller yapısında Model Katmanı uygulama verisini, domain logic işlemlerini ve fonksiyonlarını içerir. Domain Objects, Data Mappers ve Services olarak adlandırılan üç yapıdan oluşur. Domain Object gerçek dünyadaki varlıklara göre modellenen özellikler (*property*) toplamıdır. Depolama işlemlerinden bağımsız, veri değerlendirme işlemlerinin tanımlandığı yapıdır. Data Mappers olarak adlandırılan yapı sadece verinin depolanmasından sorumludur. Verilerin veritabanında depolanması halinde SQL'in barındığı ya da XML dosyalarında tutulması halinde (*parse*) transfer işleminin gerçekleştiği yapıdır. Domain Logic ile ilişkili Public arayüzün üretilmesini sağlarlar. Servisler, Domain Objects ile Data Mapper arasındaki etkileşimin gerçekleşmesini sağlayan yapılardır. Ayrıca Modeler ile MVC mimari dokusunun diğer bileşenleri arasındaki iletişim de yine Servisler tarafından sağlanır, (URI-19). View ve Controller arayüzlerinin aynı Model katmanına ulaşması ikisinin de aynı ServiceFactory arayüzüne ulaşması ile sağlanır.

Sunum ve Model Katmanlarından oluşan MVC yapısında verinin sunum kısmı View ve Controller Katmanları ile gerçekleştirilir. Sergilenecek her bir element için ayrı bir View-Controller çifti gerekir.

Controller Katmanı model ve görünüm arasında bir köprü görevi gören gelen isteklerin ayrıştırılarak değerlendirildiği sınıfları temsil eder.

View Katmanı ise kullanıcı arayüzündeki elementleri içeren sınıfları temsil eder. Bir başka deyişle, grafik olan herşeyle ilgilenir, modelden aldığı verinin görsel olarak kullanıcıya sunulmasını sağlar (Dangar, 2013). Bağlı olduğu modelden sunulacak veriyi sorular yoluyla alır. Ayrıca gerektiğinde, uygun mesajları göndererek modelin güncellenmesini de sağlar. Modelin bu soruları ve mesajları anlayabilmesi için view'

in modeldeki özelliklerin (*attributes*) anlamını (*semantic*) biliyor olması gerekir. Model View Controller mimari örüntüsü Şekil 3.20 de gösterilmiştir.



Şekil 3. 20: Model View Controller Mimari Örüntüsü, (Dangar, 2013).

Uygulamada MVC mimari yapısını içeren Laravel Framework’u kullanılmıştır.

3.3.1 Laravel framework

Laravel Framework’ün ilk versiyonunun beta sürümü Taylor Otwell tarafından CodeIgniter PHP Framework’ündeki mevcut açıkları kapatmak ve kullanıcılara kullanım kolaylığı sağlamak amacı ile 9 Haziran 2011 tarihinde yayınlanmıştır. Laravel 1 olarak adlandırılan ilk sürüm yerleşik kimlik doğrulama (*built in authentication*), veritabanı işlemleri için Eloquent ORM, modeller ve ilişkiler, basit rotalama (*routing*) yapıları, ön bellekleme (*caching*), oturumlar (*sessions*), görünüm (view), framework’ün modül ve kütüphaneler aracılığı ile genişletilebilir olması (*extendibility*), form ve HTML ile yardımcılar (*helpers*) gibi özellikleri içeriyordu. Ancak henüz denetçi (*controller*) özelliğine sahip olmadığı için MVC yapısına tam olarak sahip değildi. Altı aydan kısa zamanda Otwell’in framework’e geçerlilik denetimi (*validation*), sayfalama (*pagination*), unit test gibi özellikler eklemesi ve Eloquent ORM’ in özelliklerini genişletmesi ile Laravel 2 sürümü, 24 Kasım 2011 tarihinde yayınlanmıştır. Laravel 2 sürümünde denetçi (*controller*) desteğine ek olarak Blade şablon yapısının (*template*) da eklenmesi ile framework tam nitelikli bir MVC Framework’e dönüşmüştür.

22 Şubat 2012 tarihinde yayımlanan Laravel 3 sürümü ile mevcut sürümdeki ve diğer bazı framework'lerdeki rotaların (routes) denetçiler (controllers) ile ilişkilendirilmesi açığı kapatılarak, unit test entegrasyonu, Artisan komut satırı arayüzü, veritabanı migrasyonu (migration), olaylar (events), oturum sürücüler (session drivers) ve veritabanı sürücüler gibi özelliklerle framework zenginleştirilmiştir. Son gelişmeler ile Laravel, CodeIgniter ve Kohana gibi yaygın olarak kullanılan frameworkler ile yarışır hale gelmiştir. Beş ay gibi kısa bir süre sonra, Otwell tüm freamework'u Composer PHP Bağımlılık Yönetim aracı (*PHP Dependency Management Tool*) üzerinden dağıtılmak üzere bileşenler koleksiyonu ya da paketler şeklinde sıfırdan/yeniden yazmıştır. Bu sebepten dolayı Illuminate olarak da anılan Laravel 4 sürümü, 28 Mayıs 2013 tarihinde resmi olarak yayınlanmıştır. Laravel 4' e ayrıca, hiçbir Laravel sürümünde ve hatta hiçbir PHP framework'ünde daha önce olmayan veritabanı ekme (*database seeding*), mesaj kuyrukları (*message queues*), yerleşik posta (*built-in mailer*) özellikleri eklenmiş, ayrıca Eloquent ORM özellikleri daha da güçlendirilmiştir.

PHP 5.3 ile gelen yeni özellikleri kapsayan Laravel framework'u yaklaşık iki yıl kadar kısa bir süre içinde yaygınlaşmış ve Taylor Otwell ile dünya genelinde geliştiricilerden oluşan topluluğu ile büyük ilerlemeler katetmiştir (URL-20).

3.3.1.1 Framework'ün özellikleri

Framework geliştiricilerinin, 2009 yılında düzenlenen, php|tek konferansında bir araya gelerek oluşturdukları *Framework Interoperability Group (FIG)* çalışmaları kapsamında belirledikleri standartlar ile beraber çalışabilirlik ve farklı frameworkler üzerinde olsa bile kodların paylaşılabilirliği sağlanmıştır. Bu standartlar *PHP Standards Recommendation (PSR)* PSR-0, PSR-1, PSR-2, and PSR-3 olarak adlandırılmıştır. PSR-3 standartları Composer PHP Bağımlılık Yönetim Aracının çıkış noktasıdır. Bağımlılık (*Dependency*), uygulamalarda belirli görevlerde kullanılmak üzere oluşturulmuş sınıf ya da paketlerden herbiridir. Dördüncü sürümü ile Composer uyumlu hale getirilen Laravelde eposta göndermek için kullanılan PhpMailer ya da SwiftMailer sınıfları buna örnek olarak gösterilebilir (Dangar, 2013).

PHP topluluğu tarafından kullanılan ve arşivinde uygulamalarda kullanılmaya hazır, unit testleri yapılmış yedi binin üzerinde paket bulunan bir bağımlılık yönetim aracı

olan Composer'ın en önemli özelliği herhangi bir projedeki bağımlılıkları yönetebilmesidir. Örneğin; projemizde iki farklı kütüphaneye bağımlı bir kütüphane kullanıyorsak, kütüphaneleri güncellemek ya da iyileştirmek için “composer update” komutunu kullanmak yeterlidir. Bu komut ile composer.json dosyasında yer alan bağımlılıklar en son sürümleri ile güncellenir.

Laravel framework'ü diğer framework'lerden ayıran en önemli özelliklerinden birisi Composer uyumlu olduğu için ihtiyaç duyulacak bağımsız bileşenlerin doğrudan uygulamaya dâhil edilip kullanılabilir olmasıdır. Github üzerinden de ulaşılabilen Composer bileşenlerinin derlenmesi ile oluşan frameworkün, Symphony2 gibi farklı frameworklerin bileşenlerini içermesi birlikte çalışabilirlik özelliğini göstermektedir. Ayrıca kullanıcıya istediği bileşenleri seçme hakkı veren Laravel, istenilen veritabanı bileşenlerinin de tek komutla güncellenebilmesi kolaylığını ve uygulamaya özel çekirdek bileşenler geliştirilebilmesi olanağını sağlamaktadır (Dangar, 2013).

Programlama dilinden bağımsız olan ve Object Relatioanl Mapping (ORM) olarak adlandırılan teknikler/metodlar ile OOP nesneleri, veritabanı çeşidinden bağımsız olarak, veritabanındaki tablolara bağlanır ve veri alış-verişleri metod tarafından, programlama dili içinde yaratılan sanal veritabanı nesnesi üzerinden gerçekleştirilir. Laravel'de yerleşik olarak gelen Eloquent ORM ile kullanıcıya veritabanı tablo nesneleri yaratması ve bu nesneler üzerinden veritabanı işlemlerini gerçekleştirebilmesi olanağı sağlanmıştır. SQL sorgu cümleleri kullanmadan da veritabanı işlemleri gerçekleştirmenin mümkün olması yanında, tabloların nesneler üzerinden ilişkilendirilmesi ile uzun sorgu cümlelerine gerek kalmadan, nesneler aracılığıyla, birden fazla tablo üzerinde çoklu işlem yapılabilmesi de sağlanmıştır. Her veritabanı tablosu, bu tabloyla etkileşim için kullanılacak kendine has bir "Model" sahibidir (URI-21).

Migration (migrasyon) olarak adlandırılan yapılar veritabanı için bir sürüm kontrol türüdür. Veritabanındaki değişiklikler dosyalar şeklinde depolanarak farklı kurulumlarda ya da farklı ekip elemanları tarafından uygulamanın kopyalanması halinde veritabanının şemasının değiştirilmesine veya son şema durumuna göre güncel kalabilmelerine olanak sağlar. Laravel 4 buna ek olarak veri ekimi (seeding) özelliğini getirmiştir, bu sayede migrasyon ile oluşturulacak veritabanı tablosunda gerekli olacak ilk veri kayıtları (seed data) oluşturulabilir.

Bilgisayar bilimlerinde sıklıkla kullanılan bir veri yapısı olan Queue (kuyruk) yapılacak işlemlerin bir sıraya göre düzenli olarak yapılmasını sağlar. İlk Giren İlk Çıkar (*First In First Out, FIFO*) prensibi ile çalışan yapıda, sırası gelen işlem tamamlanır, sıradan çıkar ve sıra daha sonraki işleme geçer. Kuyruklar sayesinde uzun sürede tamamlanabilen işlemler sıraya alınarak sıraya göre tamamlanır. Örneğin, çoklu eposta gönderilmesi durumunda epostalar sıraya alınır, alıcıya ulaşınca kadar sırada kalır ve sonra sıradaki epostanın gönderilme işlemine geçilir (URL-22). Laravel'in Queue bileşeni farklı kuyruk servisleri için tek bir API sağlamaktadır. Kuyruklar e-mail göndermek gibi zaman harcayan görevlerin ileri bir zamana kadar ertelenmesi ile uygulamaya yapılan web isteklerinin büyük ölçüde hızlanmasını sağlar (URI-23).

Yerleşik posta özelliği olarak Laravel, SwiftMailler kitaplığının kullanımını mümkün kılmıştır. Swift Mailler gelişim sürecine, 2005 yılında Chris Chorney tarafından bir foruma yorum almak için postalanması ile başlamıştır. Başlangıçta SMTP üzerinden mail göndermek üzere tasarlanmış tek sınıflık bir proje olan SwiftMailler zaman içinde kullanıcıların da katkılarıyla hala gelişimini sürdüren bir kütüphaneye dönüşmüştür (URL-24).

3.3.2 Formlar

Web tabanlı uygulamalarda önemli rol oynayan Web formları ya da HTML formları, kullanıcılardan geri bildirim almak ya da bilgi toplamak ve işlenmesi için sunucuya göndermek üzere oluşturulan yapılardır. Kullanıcıya veritabanına veri ekleme, güncelleme, silme işlemlerini gerçekleştirebilme yetkisi veren ve ayrıca veritabanındaki verileri listeleme imkân ve kolaylığını da sağlayan bir arayüz elemanı olarak da düşünülebilir. Formlardan alınan veriler veritabanına kaydedilmeden ya da işlenmeden önce çeşitli kontrollerden geçer. Bu kontroller, forma veri girişi olup olmadığının kontrolü, girilen verilerin uygun formatta oluşu, veritabanındaki veri tipleri ile uyumluluğu, zararlı veri girişleri/ kodlarla siteye müdahalenin önlenmesi, formun gönderilmiş/onaylanmış olması amacıyla gerçekleştirilmiş filtreler ve geçerlilik denetimi (*validation*) şeklinde olabilir.

Laravel Framework'ünde formlar klasik HTML etiketleri kullanılarak ya da Laravel'in form sınıfından yararlanarak iki şekilde oluşturulabilmektedir;

Klasik HTML <form></form> etiketleriyle

<form>

Kullanıcı Adı : <input type="text" name="username">

Şifre :<input type="password" name="password">

<input type="submit" value="Giriş yap">

</form>

Laravel Form Sınıfı kullanılarak;

```
<?php echo Form::open('ilgili/route'); ?>
```

```
<?php echo Form::label('username', Kullanıcı Adı); ?>
```

```
<?php echo Form::text('username'); ?>
```

```
<?php echo Form::label('password', Şifre); ?>
```

```
<?php echo Form::password('password'); ?>
```

```
<?php echo Form::submit(Giriş yap);
```

```
<?php echo Form::close(); ?>
```

Proje kapsamında, Kampanya, İstasyon, GNSS Ölçme, GNSS Oturum, Alıcı, Kurum ve Birim formları oluşturulmuştur. Formlardan alınan bilgiler veritabanındaki ilgili tablolara işlenmektedir. Oluşturulan form örnekleri EK A'da yer almaktadır.

4. SONUÇ VE ÖNERİLER

Levha hareketlerinden biri olan levhaların uzaklaşması durumu sonucunda Atlantik okyanusunda oluşan okyanus sırtlarının yayılması ile Afrika ve Arabistan levhaları kuzey-kuzeydoğu yönünde itilmektedir. Arabistan levhasının kuzeye doğru itilmesi, Avrasya levhasının altına dalmasıyla sonuçlanmaktadır. İki kıtasal levhanın çarpışma alanında bulunan ülkemizin doğusunda kıta kabuğu kalınlaşmış ve hala kalınlaşmaya devam etmektedir. Arabistan levhasının kuzeye doğru ilerlemesi sonucu anadolu levhası batıya doğru hareket etmeye başlamıştır. Levhaların tektoniği sonucu oluşan deprensellik ve deformasyon nokta koordinatlarında zamana bağlı değişimler meydana getirmektedir. Bu durum, TUTGA koordinatlarının güncellenmesi ihtiyacını doğurmaktadır (URI-1).

TUTGA koordinatlarının güncelliğini yitirmemesi için mevcut istasyonlardan yeniden gözlem yapılması ya da gerekli yerlerde jeodezik ağın sıklaştırılması gerekmektedir. Bu ihtiyaca çözüm üretmek için tasarlanan ALAGÖZ Projesinde, Türkiye genelinde 4000’den fazla alıcı ile gerçekleştirilmesi planlanan GPS/GNSS gözlemlerinin dinamik bir web sitesi aracılığıyla bir veritabanında saklanması, bu jeodezik verilerin yönetilmesi ve açık olarak sunulması tasarlanmıştır.

Tez çalışması kapsamında, ALAGÖZ Projesinde kullanılmak üzere bir veri yönetimi sistemi tasarlanmış ve site kullanıcılarının GPS/GNSS verilerini projenin ihtiyaçları doğrultusunda tasarlanan veritabanına web formları aracılığıyla, lokal sunucuda, kaydedebilmesini sağlayan bir platform geliştirilmiştir

Bu web uygulamasının tasarımında ve yazılımında açık kaynak kodlu programlama dilleri, frameworkler ve veritabanları tercih edilmiştir. Web sitesinin yazılımı Laravel Framework’ü üzerinde PHP programlama dili ile gerçekleştirilmiş, Javascript, JQuery dillerinden, Blade Template’inden yararlanılmış, Tasarım kısmında HTML, CSS dillerinden, Bootstrap framework’ünden yararlanılmıştır. İlişkisel ve açık kaynak kodlu veritabanı olarak PostgreSQL veritabanı tercih edilmiş ve uygulamanın

geliştirilmesi sürecine zemin hazırlamak üzere açık kaynak kodlu PostGIS ile entegre edilmiştir. PHP kodlarının çalıştırılabilmesi için sunucu olarak Apache sunucusu ile PHP, MySQL/PostgreSQL entegrasyonunu ve ZendFramework, CodeIgniter, Laravel, Smyphony gibi frameworkler'e hazır erişimi sağlayan BITNAMI programı tercih edilmiştir.

Web sitesinde temel olarak, üyelik sistemi, oturum yönetimi, proje katılımcılarına dair bilgiler, proje yöneticileri tarafından sunulacak bilgiler ve yine katılımcılar tarafından sağlanacak verilerinin veritabanı ile entegrasyonunu güvenli bir şekilde sağlayacak form yapısı gerçekleştirilmiştir.

Sonraki çalışmalarda, geliştirilen platformun bir daha kapsamlı bir web uygulamasına dönüştürülerek ALAGÖZ Projesi katılımcılarının sağlayacağı gerçek verilerin ve metadatanın standartlara uygun olarak depolanması, sorgulanması, çeşitli analiz sonuçlarının haritalar üzerinde görselleştirilmesi planlanmaktadır. Devam eden süreçte uygulamanın android ortamda da çalışabilirliğinin sağlanması hedeflenmektedir.

KAYNAKLAR

- Aslan, K.,** (2013). *C# Kursu, C ve Sistem Programcıları Derneği*, İstanbul
- Aslan, K.,** (2013). *C#Uygulama-I Kursu, C ve Sistem Programcıları Derneği*, İstanbul
- Dangar, H.,** (2013). *Learning Laravel 4 Application Development*. Burmingham, United Kingdom, Packt.
- Douglas, K., Douglas S.,** (2005). *PostgreSQL The comprehensive guide to building, programming, and administering PostgreSQL databases*. Indianapolis, Indiana, SAM
- Doyle, M.,** (2010). *Beginning PHP 5.3*. Indianapolis, Indiana, Wiley.
- Gilmore, W. J., and Treat, R. H.,** (2006). *Beggining PHP and PostgreSQL 8 From Novice To Professional*. Verlag, New York, Springer.
- Lerdorf, R., Tatroe K. and MacIntyre P.,** (2006). *Programming PHP*. Sebastopol, California, O'Reilley.
- Momjian, B.,** (2001). *PostgreSQL Introduction and Concepts*. New York, New York, Addison Wesley.
- Özaslan, Ö.,** (2003). *Jeodezik Veri Tabanı Tasarımı ve Web Tabanlı Yönetimi*, (yüksek lisans tezi), İTÜ
- Riggs, S., Krosing H.,** (2010). *PostgreSQL9 Administration Cookbook Solve real-world PostgreSQL problems with over 100 simple, yet incredibly effective recipes*. Birmingham, Packt.
- Şamlı, M.,** (2011). *Her Yönüyle PHP 6*. Sultanahmet, İstanbul, Kodlab.
- Valade, J., Ballad, T. and Ballad, B.,** (2008). *PHP & MySQL Web Development*. Indianapolis, Indiana, Wiley.
- Yazıcı, V.,** (2006). *PostgreSQL ile Programlama*. GNU Özgür Belgeleme Lisansı, <http://www.students.itu.edu.tr/~yazicivo/> Redirected to: <http://vy.github.io/code/>
- URI-1**<http://tr.wikipedia.org/wiki/Levha_hareketleri>, alındığı tarih: 04.05.2014
- URI-2**< www.hgk.msb.gov.tr/uyekurulus/tujjb/tujk/alagöz.ppt>
- URI-3**<http://en.wikipedia.org/wiki/Satellite_positioning_system>
Satellite Navigation, alındığı tarih 25.06.2014
- URI-4**<<http://msdn.microsoft.com/tr-tr/library/dd490886.aspx>>, *Yazılım Sistem Mimarisi*, alındığı tarih: 25.06.2014
- URI-5** <<http://www.php.net/manual/tr/history.php.php>>, *PHP manuaels*, alındığı tarih 30.04.2013

- URI-6** <<http://tr.wikipedia.org/wiki/PHP>>, alındığı tarih: 03.05.2013
- URI-7** <<http://php.net/usage.php>>, alındığı tarih:03.05.2014
- URI-8** <<http://www.students.itu.edu.tr/~yazicivo/>>, alındığı tarih: 25.04.13
- URI-9**< <http://www.w3.org/People/Raggett/book4/ch02.html> >, *A history of HTML*
alındığı tarih:19.04.2013
- URI-10**< <http://www.w3schools.com/css/> >, alındığı tarih: 26.04.2013
- URI-11**< <http://en.wikipedia.org/wiki/Database>>, *Database*,
alındığı tarih: 21.04.2013
- URI-12**< <http://www.postgresql.org/docs/devel/static/features.html>>,
alındığı tarih:27.04.2013
- URI-13**< <http://en.wikipedia.org/wiki/PostgreSQL>> *PostgreSQL*,
alındığı tarih: 03.05.2013
- URI-14**<<http://en.wikipedia.org/wiki/PostgreSQL#Features>>, alındığı tarih: 21.04.13
- URI-15**<<http://web.itu.edu.tr/~yazicivo/doc/postgresql-ile-programlama-2>>,
alındığı tarih: 19.04.13
- URI-16** < <http://akademi.itu.edu.tr/aaydinoglu/DosyaGetir/64458/49.pdf>>, *Coğrafi Veri Yönetiminde Standart Kavramı*, alındığı tarih:25.06.2014
- URI-17** <<http://www.postgresql.org/docs/7.4/static/datatype-geometric.html>>,
alındığı tarih:03.05.2014
- URI-18** <<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>> , *Reenskaug Trygve*, alındığı tarih:03.05.2014
- URI-19** <<http://stackoverflow.com/questions/5863870/how-should-a-model-be-structured-in-mvc/5864000#5864000>>, alındığı tarih:03.05.2014
- URI-20** <<http://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging>> , alındığı tarih:03.05.2014
- URI-21**<<http://www.laravel.gen.tr/docs/eloquent>>, alındığı tarih: 03.05.2014
- URI-22**< [http://en.wikipedia.org/wiki/Queue_\(abstract_data_type\)](http://en.wikipedia.org/wiki/Queue_(abstract_data_type))>,
alındığı tarih: 03.05.2014
- URI-23**<<http://laravel.com/docs/queues>> , alındığı tarih: 03.05.2014
- URI-24**<<http://swiftmailer.org/docs/introduction.html>>, alındığı tarih: 03.05.2014

EKLER

EK A: Formlar

ALAGÖZ *AI Alıcını Gel Ölçme Zamanı*

ALAGÖZ

- Giriş Yap
- Şifremi unuttum
- Katılımcı olmak istiyorum

Lütfen giriş yapınız.

Şekil A.1: ALAGÖZ Web Uygulaması.

ALAGÖZ *AI Alıcını Gel Ölçme Zamanı*

ALAGÖZ

- Giriş Yap
- Şifremi unuttum
- Katılımcı olmak istiyorum

Email:

Password:

☐ Remember me

Şekil A.2: ALAGÖZ Web Uygulaması Giriş Sayfası

ALAGÖZ *Al Alıcını Gel Ölçme Zamanı*



Şekil A.3: ALAGÖZ Web Uygulaması Ana Sayfası.

ALAGÖZ Alıcıları Listele Alıcı ekle							
Alıcı Listesi							
Seri No	Alıcı ismi	Üretici Firma	Doğruluk	Notlar			
hc128563	Trimble Pro6H	Trimble	1cm	220 Kanal, 1000gr, Real-Time 10cm, PostProcess 1cm hassasiyet	Göster	Düzenle	Sil
tb4532892	TRIMBLE Geo7X	Trimble	1cm	220 kanal, GPS L1/L2/L2C/L5 Glonass L1/L2, Beidou(Compass) QZSS, Dahili Mesafe ölçer, Dahili Pusula	Göster	Düzenle	Sil

Şekil A.4: Alıcı Sayfası.

ALAGÖZ Alıcıları Listele Alıcı ekle	
Alıcı Ekle	
Seri No	
Alıcı ismi	
Üretici Firma	
Doğruluk	
Notlar	
Alıcı Ekle	

Şekil A.5: Alıcı Ekleme Sayfası.

Alıcı Bilgisi Göster

Alıcı Seri No: hc128563

Alıcı ismi: Trimble Pro6H

Doğruluk: 1cm

Notlar: 220 Kanal, 1000gr, Real-Time 10cm, PostProcess 1cm hassasiyet

Şekil A.6: Alıcı Görüntüleme Sayfası.

Trimble Pro6H - hc128563

Seri No

hc128563

Alıcı ismi

Trimble Pro6H

Üretici Firma

Trimble

Doğruluk

1cm

Notlar

220 Kanal, 1000gr, Real-Time 10cm, PostProcess 1cm hassasiyet

Alıcı Düzenle

Şekil A.7: Alıcı Düzenleme Sayfası.

ÖZGEÇMİŞ



Ad Soyad : Deniz BAŞAR
Doğum Yeri ve Tarihi : Konya, 22.02.1986
Adres : İTÜ Ayazağa Yerleşkesi İnşaat Fakültesi
Geomatik Mühendisliği Bölümü L206
E-posta : deniz.basar@itu.edu.tr
Lisans : 2011, İstanbul Teknik Üniversitesi, İnşaat Fakültesi,
Geomatik Mühendisliği Bölümü
Yükseklisans : 2014, İstanbul Teknik Üniversitesi, Geomatik Mühendisliği,
Geomatik Mühendisliği Programı