

T.C.
MİMAR SİNAN GÜZEL SANATLAR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**LADM STANDARTINDA MEKANSAL VERİ OLUŞTURULMASI ve WEB
ORTAMINDA ÜÇ BOYUTLU OLARAK GÖRSELLEŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Murat SÜMBÜL

Enformatik Anabilim Dalı

Bilgisayar Ortamında Sanat ve Tasarım Lisansüstü Programı

Tez Danışmanı: Doç. Dr. Ümit IŞIKDAĞ

HAZİRAN 2019

Murat SÜMBÜL tarafından hazırlanan LADM Standartında Mekansal Veri Oluşturulması ve Web Ortamında Üç Boyutlu Olarak Görselleştirilmesi adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

DOÇ. DR. ÜMİT IŞIKDAĞ

Tez Yöneticisi

Bu çalışma, jürimiz tarafından Enformatik Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan

: DOÇ. DR. ÜMİT IŞIKDAĞ

Üye

: Prof. Dr. Salih OFLUOĞLU

Üye

: Dr. Öğr. Üyesi Caner GÜNEY

Üye

: _____

Üye

: _____

Bu tez, Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

Mimar Sinan Güzel Sanatlar Üniversitesi Fen Bilimleri Enstitüsü tez yazım klavuzuna uygun olarak hazırladığım bu tez çalışmada;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel etik kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- ücret karşılığı başka kişilere yazdırmadığımı (dikte etme dışında), uygulamalarımı yaptırmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

Hayatım boyunca desteklerini ve sevgilerini bir an olsun esirgemeyen, varlıklarıyla hayatımı güzelleştiren annem Zekiye Nurdoğdu SÜMBÜL, babam Erdal SÜMBÜL ve kardeşim Münip Kutay SÜMBÜL'e, tez konusunun belirlenmesi ve ilerleyişindeki katkıları, çalışma sürecinde göstermiş olduğu ilgi, sabır ve anlayışı ile değerli hocam Doç.Dr.Ümit IŞIKDAĞ'a en içten şekilde teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖNSÖZ	iv
İÇİNDEKİLER	v
KISALTMALAR	vii
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
1.1 Problemin Tanımı	2
1.2 Araştırma Soruları	2
1.3 Araştırmanın Amacı ve Hedefleri	2
1.4 Araştırma Süreci	3
2. LİTERATÜR ARAŞTIRMASI	6
2.1 Coğrafi Bilgi Sistemleri	6
2.2 Üç Boyutlu Coğrafi Bilgi Sistemleri	8
2.3 Üç Boyutlu Kadastro	9
2.4 LADM	11
3. VERİ ve YAZILIM ALTYAPISI	16
3.1 Mekansal Veri Yapıları	17
3.1.1 İki boyutlu mekansal veriler	18
3.1.1.1 Raster veri yapısı	19
3.1.1.2 Vektör veri yapısı	19
3.1.2 İki buçuk ve üç boyutlu mekansal veriler	22
3.1.2.1 Voxel veri yapısı	23
3.1.2.2 Üç boyutlu vektör veri yapıları	24
3.1.3 Mekansal veri model ve standartları	25
3.1.3.1 Geography Markup Language (GML)	25
3.1.3.2 CityGML	26
3.1.3.3 GeoJSON	26
3.1.3.4 Arazi yönetimi temel modeli (LADM)	27
3.1.4 Mekansal veri tabanları	28
3.1.4.1 Mekansal veri tabanlarında kullanılan veri tipleri	28
3.1.4.2 Mekansal Verinin Dizinlenmesi	30
3.1.4.3 Mekansal operatörler ve SQL tabanlı mekansal sorgular	31
3.1.4.4 Mekansal uygulama rutinleri	33
3.1.4.5 Oracle veritabanı yazılımının mekansal özelliklerinin kullanım örnekleri	33
3.2 Uygulama Katmanı Teknolojileri	36
3.2.1 Java 2 Enterprise Edition	36
3.2.2 Spring Framework	38
3.2.3 Apache Maven	39
3.2.4 Apache Tomcat	40

3.2.5 JavaScript Object Notation (JSON)	41
3.3 Sunum Katmanı Teknolojileri	42
3.3.1 WebGL	42
3.3.2 JQuery	42
3.3.3 ThreeJS	43
3.3.4 CesiumJS	43
4. YÖNTEM	45
4.1 LADM Standardında Verileri Oluşturan Uygulama	47
4.1.1 Coğrafi veri setinin seçilmesi	50
4.1.2 Coğrafi verilerin oluşturulması uygulaması katmanları	51
4.1.2.1 Veri katmanı	53
4.1.2.2 Uygulama mantığı katmanı	57
4.1.2.3 Kullanıcı deneyimi katmanı	62
4.1.3 Coğrafi verilerin oluşturulması uygulaması sınıf ve sıralanım şemaları ..	63
4.1.4 Oluşturulan coğrafi verinin doğrulanması	66
4.2 Kullanıcı Etkileşim Uygulaması	66
4.2.1 Kullanıcı etkileşim uygulaması katmanları	69
4.2.1.1 Veri katmanı	71
4.2.1.2 Uygulama mantığı katmanı	71
4.2.1.3 Kullanıcı deneyimi katmanı	77
4.2.2 Kullanıcı etkileşim uygulaması sınıf ve sıralanım şemaları	80
5. SONUÇLAR VE YORUMLAR	86
5.1 Kullanıcı Testleri	86
5.2 Kullanıcı Testleri Sonuçları	89
5.2.1 Etkinlik Testleri	90
5.2.1.1 Tamamlanma oranı	90
5.2.1.2 Tamamlanan görev başına hata oranı	91
5.2.2 Verimlilik Testleri	92
5.2.2.1 Zaman temelli verimlilik	92
5.2.2.2 Genel göreceli verimlilik	92
5.2.3 Kullanıcı Memnuniyeti Testleri Sonuçları	93
5.2.3.1 Görev bazlı memnuniyet	93
5.2.3.2 Sistem bazlı memnuniyet	94
5.3 Yorumlar	95
KAYNAKLAR	98
EKLER	105

KISALTMALAR

API	: Application Programming Interface
CityGML	: City Geography Markup Language
CLI	: Call Level Interface
CSS	: Cascading Style Sheets
DI	: Dependency Injection
DOM	: Document Object Model
GeoJSON	: Geographic JavaScript Object Notation
GIS	: Geographic Information Systems
GML	: Geography Markup Language
HTML	: Hyper Text Markup Language
HTTP	: Hypertext Transfer Protocol
IoC	: Inversion of Control
ISO	: International Organization for Standardization
J2EE	: Java 2 Platform, Enterprise Edition
JMS	: Java Messaging Service
JSON	: JavaScript Object Notation
JSP	: Java Server Pages
LADM	: Land Administration Domain Model
MVC	: Model – View - Controller
OGC	: Open Geospatial Consortium
OJDBC	: Oracle Java Database Connection
OpenGL ES	: Open Graphics Library Embedded Systems
PTZ	: Pan, Tilt, Zoom
RRR	: Rights, Restrictions and Responsibilities
SQL	: Structured Query Language
UML	: Unified Modeling Language
WAR	: Web Application Archive
WebGL	: Web Graphics Library
WGS	: World Geodetic System
XLink	: XML Linking Language
XML	: Extensible Markup Language

ŞEKİL LİSTESİ

Sayfa

Şekil 3. 1 : MVC Tasarım Örüntüsü	17
Şekil 3. 2 : Raster veri yapısı	19
Şekil 3. 3 : İki boyutlu nokta, çizgi, çoklu ve kapalı çizgi vektör veri örnekleri	21
Şekil 3. 4 : İki boyutlu bir geometri ile üç boyutlu geometri oluşturulması	23
Şekil 3. 5 : Voxel ve pixel veri yapılarının karşılaştırılması	24
Şekil 3. 6 : Üç boyutlu vektör veri yapısı ..	25
Şekil 3. 7 : Temel LADM paketleri ..	28
Şekil 3. 8 : ISO 19125-2 SQL Tip İlişkileri.....	30
Şekil 3. 9 : J2EE Çok katmanlı mimarisi ..	37
Şekil 3. 10 : Spring Modülleri ...	39
Şekil 3. 11 : Apache Tomcat mimarisi	41
Şekil 3. 12 : CesiumJS tabanlı sanal küre	44
Şekil 4. 1 : UML şema tipleri	46
Şekil 4. 2 : LADM standardında verileri oluşturan uygulama kullanım durum şeması	50
Şekil 4. 3 : LADM standardında verileri oluşturan uygulama paket şeması	52
Şekil 4. 4 : LADM standardında verileri oluşturan uygulama sınıf şeması.....	64
Şekil 4. 5 : LADM standardında verileri oluşturan uygulama sıralanım şeması	65
Şekil 4. 6 : Kullanıcı etkileşim uygulaması durum şeması.....	69
Şekil 4. 7 : Kullanıcı etkileşim uygulaması uygulama paket şeması.....	70
Şekil 4. 8 : Kullanıcı etkileşim uygulaması sunucu tarafı sınıf şeması	82
Şekil 4. 9 : Kullanıcı etkileşim uygulaması istemci tarafı sınıf şeması	83
Şekil 4. 10 : Kullanıcı etkileşim uygulaması sunucu tarafı sıralanım şeması.....	84
Şekil 4. 11 : Kullanıcı etkileşim uygulaması istemci tarafı sınıf şeması	85
Şekil 5.1 : Tamamlanma Oranı.....	91
Şekil 5.2 : Tamamlanan Görev Başına Hata Oranı.....	91
Şekil 5.3 : Hedef/Saniye	92
Şekil 5.4 : Genel Göreceli Verimlilik.....	93
Şekil 5.5 : Görev Bazlı Memnuniyet	94
Şekil 5.6 : Sistem Bazlı Memnuniyet	95

TABLO LİSTESİ

Tablo 1.1 : Araştırma akış şeması	5
Tablo 4.1: Veri setinin oluşturulması kullanım durumu	48
Tablo 4.2 : Oluşturulan veri setinin doğrulanması kullanım durumu	49
Tablo 4.3 : Üç boyutlu LADM elemanlarını görselleştirme kullanım durumu.....	67
Tablo 4.4 : Üç boyutlu nesnelerin görselleştirildiği sahne üzerinde gezinme kullanım durumu	68
Tablo 5.1 Kullanıcı Testi Sonuçları	90
Tablo 5.2 Kullanıcı Memnuniyeti Testi Sonuçları	93

LADM STANDARDINDA MEKANSAL VERİ OLUŞTURULMASI VE WEB ORTAMINDA ÜÇ BOYUTLU OLARAK GÖRSELLEŞTİRİLMESİ

ÖZET

İnsanoğlu yerleşik hayata geçip, tarım ve benzeri etkinlikler ile uğraşmaya başladığından beri, mülkiyet ve dolayısıyla kadastro kavramları ortaya çıkmıştır. Kadastronun, ulusal ve uluslararası ihtiyaçlara cevap verecek şekilde gelişebilmesi adına, tarihsel süreçte pek çok çalışma yapılmıştır ve günümüzde uluslararası kabul gören modellerden birisi de LADM'dir. LADM, tüm kadastro sistemlerinde bulunan iştirakler, mekansal olgular ve hak, kısıt, sorumluluk gibi temel yönetsel olguları içermektedir. İçerdiği bu olgulara ek olarak da farklı kadastro ihtiyaçlarına cevap verebilecek özellikleri barındırmaktadır. Web teknolojilerinin gelişmesi sonucu uç kullanıcıların, veriyi sorgulama, elde etme ve görselleştirme olanakları artmıştır ve çalışmanın hedefi de bu artan olanakları kullanarak mekansal verileri uç kullanıcıların hizmetine sunmaktır.

Öncelikli olarak gerçekleştirilen literatür araştırması sonucu LADM standardı ile yapılmış çalışmalar detaylı bir şekilde incelenmiştir ve görselleştirmenin yapılabilmesi için ihtiyaç duyulan LADM standardındaki mekansal mülkiyet veri seti, sadece taban bilgisi bulunan GeoJSON biçimindeki verilerin ve kullanıcı girdilerinin alındığı bir web yazılımı aracılığıyla oluşturularak Oracle Spatial veritabanına yazılmıştır.

Çalışmanın temel hedefini gerçekleştirmek adına, oluşturulan bu veri setini kullanıcı isteklerine göre görselleştirecek bir web uygulaması geliştirilmiştir. MVC mimarisinde olan bu uygulama veri katmanı olarak veri setinin bulunduğu veritabanına, uygulama katmanı olarak kullanıcı isteklerini işleyerek istemci katmanına ileten Java uygulamasına ve istemci katmanında da mekansal mülkiyet verisinin görselleştirildiği Cesium tabanlı sanal küreye sahiptir. Kullanıcılar görselleştirme dışında, mekansal mülkiyet verisi ile ilgili bilgilere de ulaşabilmektedir.

Anahtar Kelimeler: LADM, Sanal Küre, Oracle Spatial, Cesium, Java, Spring MVC, Üç Boyutlu Kadastro

CREATING SPATIAL DATA SET FOR THE LADM STANDARD AND 3D VISUALIZATION IN WEB ENVIRONMENT

ABSTRACT

Since human beings have settled and engaged in agriculture and similar activities, the concepts of property and hence cadastre have emerged. In order to improve the cadastre to meet national and international needs, many studies have been done in the historical process and one of the internationally accepted models is LADM. The LADM includes the basic managerial facts in all cadastral systems, spatial phenomena, rights, constraints and responsibilities. In addition to these phenomena, it contains features that can meet different cadastral needs. As a result of the development of web technologies, end-users have the opportunity to query, acquire and visualize data, and the objective of the study is to present the spatial data to the end users by using these increasing possibilities.

As a result of the literature research carried out with the LADM standard, the studies of the LADM standard were examined in detail and the spatial property data set required for visualization was created by using a web software that can receive GeoJSON format data with base information and user inputs. In addition to this, created data set was written to the Oracle Spatial database.

In order to realize the main objective of the study, another web application has been developed to visualize this set of data according to user requests. This application, which is in the MVC architecture, is using created data set as a data layer. In addition to this, it has a Java application layer that processes user requests and serves the results to the client layer. Finally, the client layer has a Cesium-based virtual sphere where the spatial property data is visualized. Apart from visualization, users can also access information about spatial property data.

Keywords: LADM, Virtual Sphere, Oracle Spatial, Cesium, Java, Spring MVC, 3D
Cadastre

1. GİRİŞ

İnsanoğlu ve kadaströ sistemleri arasında devingen bir ilişkinin olduđu gözlemlenmektedir (Ting Ba, Candidate, & Williamson, 2000). Toplumların tarih boyunca değışen sosyolojik yapısı ve sahip olunan teknolojik yetenekler de bu devingen ilişkinin yönelimlerini belirlemektedir. Küreselleşen Dünya'nın getirdiđi pek çok ihtiyaca cevap verebilmek adına, günümüzde pek çok ülke, ulusal tapu ve kadaströ sistemlerini bilgi sistemlerine dönüştürme çalışmalarında bulunmaktadır. Türkiye özelinde örnek vermek gerekirse, Çevre ve Şehircilik Bakanlığına bađlı olan Tapu ve Kadastro Genel Müdürlüğü 2005 yılında (Alkan & Can, 2009) ilk sürümünü geliştirdiđi ve 2012 yılında (Tapu ve Kadastro Genel Müdürlüğü, 2005) yaygınlaşmasını ulusal düzeyde tamamladıđı “Tapu ve Kadastro Bilgi Sistemi” kamunun hizmetine sunmuştur.

Ulusal düzeyde gerçekleştirilen çalışmalara ek olarak, LADM gibi uluslararası standartlaştırma çalışmaları da gerçekleştirilmektedir. Gerçekleştirilen çalışmaların ortak hedefi, uluslararası düzeyde kullanılabilecek ve bu yetkinliđi sağlarken de devletlerin karşılaştıđı coğrafyaya özel sorunların çözümünde yol gösterebilecek bir standardın ortaya konulmasıdır. LADM standardı ile alakalı Türkiye'ye uyumlama çalışmaları da gerçekleştirilmektedir (Polat & Alkan, 2018) .

Geliştirilen coğrafi bilgi sistemleri kanunsal ve mekansal bilgilerin saklanması ve sunulması temel amaçlarına hizmet etmektedir. Tez kapsamında gerçekleştirilen çalışmaların hedefi de bu yöndedir. Öncelikli olarak mekansal mülkiyet verisinin uluslararası bir standart olan LADM çerçevesinde oluşturulması, saklanmasını ve sunulmasını sağlayan bir alt sistemin ve yazılımın geliştirilmesi gerçekleştirilmiştir. Gerçekleştirilen bu çalışmanın ışığında elde edilen mekansal mülkiyet verilerinin tez dahilinde kullanım amacı ise, çalışma kapsamında belirlenen probleme önerilecek çözüm sürecinde yapılacak görselleştirme yazılımına temel oluşturmaktır.

Mekansal verilerin görselleştirilmesi öncelikli olarak masaüstü yazılımlar aracılığıyla gerçekleştirilmekteydi ve bu sebeple genel kullanıma sunulması kısıtlı bir şekilde

olmaktaydı. Gelişen web teknolojileri, görselleştirme işlerinin uç kullanıcılara yani internet erişimi olan herkese ulaştırılmasında kolaylıklar sağlamaktadır. Özellikle web ortamında çalışabilen ve sağladığı yordamlar vasıtası ile nesnelerin devingen olarak oluşturulabilmesi sağlayan üç boyutlu sanal küre teknolojisi, mekansal verilerin görselleştirilerek kullanıcılara etkin bir şekilde sunulabilmesini sağlamıştır. Sunulan mekansal veriler ile kullanıcı tercihlerine ve girdilerine göre etkileşime geçilebilmesi de verilerin daha faydalı ve etkin bir şekilde sunulmasını sağlamaktadır.

1.1 Problemin Tanımı

LADM standardı ile tanımlanmış mekansal mülkiyet verisinin, üç boyutlu bir sanal küre üzerinde web ortamında görselleştirilmesinin zor ve sınırlı olması.

1.2 Araştırma Soruları

- LADM standardı ile ifade edilen mekansal bilgiler görselleştirilebilir mi?
- LADM standardı ile ifade edilen mekansal bilgilerin web ortamında görselleştirilmesinde nasıl bir veri saklama aracı ve yöntemi kullanılabilir?
- LADM standardı ile ifade edilen mekansal verilerin web ortamında görselleştirilmesi sürecinde nasıl bir yazılım mimarisi kullanılabilir?
- LADM standardı ile ifade edilen mekansal verilerin web ortamında görselleştirilmesi sürecinde hangi görselleştirme araç ve yöntemleri kullanılabilir?
- Görselleştirme sürecinde kullanılabilecek LADM standardına uyan örnek bir veri seti nasıl oluşturulabilir?

1.3 Araştırmanın Amacı ve Hedefleri

Bu çalışma temel olarak, mekansal verilerin saklanması ve görselleştirilmesi konularına odaklanmaktadır. Çalışma kapsamında ilk amaç, LADM standardına uygun üç boyutlu mekansal ve anlamsal olguları temsil eden veri setleri oluşturmak ve bu veri setlerini saklamaktır. Saklanan verinin web ortamından alınan kullanıcı isteklerine göre okunarak yine web ortamında bulunan bir sanal küre üzerinde

kullanıcının etkileşime girebileceği şekilde görselleştirilmesi çalışmanın ikinci amacıdır. Çalışmanın üçüncü amacı da gerçekleştirilen uygulamalar sürecinde tasarım örüntülerine ve genel kabul görmüş yazılım mimarilerine uyarak gelişime ve değişime açık yazılım sistemleri geliştirmektir. Amaçlar kapsamında geliştirilecek olan yazılım sistemlerinin, kullanılabilirlik açısından test edilmesi çalışmanın diğer amaçlarından birisidir. Geliştirilmesi amaçlanan sistemin hayata geçirilmesi ile ileride gerçekleştirilecek çalışmalara yardımcı bir yazılım sistemi oluşturmak ve LADM standardına sahip mekansal verilerin web tabanlı bir sanal küre üzerinde görselleştirilebilmesi hedeflenmektedir.

1.4 Araştırma Süreci

Araştırma başlangıcı itibariyle ilk olarak, araştırma konusunun belirlenmesi hedeflenmiştir. Konunun fikirsel olarak belirlenmesinin ardından, konunun içine dahil olduğu coğrafi bilgi sistemleri ve üç boyutlu coğrafi bilgi sistemleri ile alakalı akademik çalışmalar ulusal ve uluslararası düzeyde incelenmiştir. Gerçekleştirilen çalışmaların incelenmesi sürecinde öncelikli olarak görselleştirme ile ilgili çalışmalar hedef alınsa da farklı yönleri olan çalışmalar da inceleme sürecine dahil edilmiştir. Araştırma öncesi fikirsel olarak belirlenen araştırma konusu, gerçekleştirilen bu çalışmanın ardından daha net bir şekilde belirlenmiştir.

Üç boyutlu kadastral sistemler ve LADM standardı ile ilgili konunun belirlenmesinin ardından, bu konular ile ilgili araştırmalar gerçekleştirilmiştir. Araştırmaların amacı, belirlenen konunun iyileştirilmesi ve araştırma probleminin tanımlanmasıdır.

Tanımlanan konu ve problem ile beraber çözümün gerçekleştirilmesi için gereken teknolojik kavramlar ile alakalı akademik ve teknik çalışmalar incelenmiştir. Süreç boyunca ilk olarak benzer problemlerin çözümü için uygulanan yaklaşımlar incelenmiştir ve çözüm mimarisi seçilmiştir. Çözüm mimarisinin seçilmesi sonrasında ise mimarinin ihtiyaç duyduğu katmanlarda kullanılabilecek teknolojiler ile alakalı araştırmalar, teknolojiler hakkında sahip olunan kavramsal ve teknik bilgilerin artırılması amacı ile gerçekleştirilmiştir.

Çözüm için gerekli araştırmaların gerçekleştirilmesine müteakip çözüm için gereken yazılım sistemlerinin gerçekleştireceği işler belirlenmiştir ve mimari olarak tasarımları gerçekleştirilmiştir. Geliştirilecek yazılımın hedefleri ve mimarisinin

belirlenmesinin ardından kullanım durum şemaları ile tüm kullanım süreci belirlenmiştir ve yazılım geliştirilmiştir.

Geliştirilen yazılım ve önerilen çözümün test edilebilmesi adına ise kullanıcı testleri gerçekleştirilmiştir. Son olarak da, yürütülen kullanıcı testleri sonucu elde edilen yorumlar ve testler sürecinde yapılan gözlemler değerlendirilerek çalışmanın raporlanması amacıyla tez yazımı gerçekleştirilmiştir. Araştırmanın yöntemini ve tez içinde bulunduğu bölümleri açıklayan “Araştırma Akış Şeması” tablosu, Tablo 1.1’de görülmektedir.



Tablo 1.1 : Araştırma akış şeması

Nasıl	Ne	Neden	
Literatür Araştırması	Coğrafi bilgi sistemleri ile ilgili akademik araştırmaların incelenmesi Üç boyutlu coğrafi bilgi sistemleri ve bu sistemler aracılığıyla gerçekleştirilen görselleştirme çalışmalarının incelenmesi	Araştırma konusunun belirlenmesi amacı ile	Bölüm 2
Literatür Araştırması	Üç boyutlu kadastro sistemleri ile ilgili akademik çalışmaların incelenmesi LADM standardı kapsamında gerçekleştirilen akademik çalışmaların incelenmesi	Araştırma konusunun iyileştirilmesi amacı ile Araştırma probleminin tanımlanması amacı ile	Bölüm 2
Teknoloji Araştırması	Web ortamında kullanılan mimari yapıların incelenmesi	Benzer araştırma problemlerinin çözüm yaklaşımlarını incelemek adına	Bölüm 3
Teknoloji Araştırması	Mekansal veriler ve veri yapıları ile ilgili araştırmaların gerçekleştirilmesi Uygulama katmanı teknolojileri ile ilgili araştırmaların gerçekleştirilmesi Veri görselleştirme ile ilgili teknolojilerin ve araştırmaların incelenmesi	Çözüm önerme süreci öncesinde ilgili teknolojiler ile ilgili bilgi seviyesinin artırılması amacı ile	Bölüm 3
Yazılım Geliştirme	Veri oluşturan yazılımın geliştirilmesi Üç boyutlu olarak mekansal verileri görselleştiren yazılımın geliştirilmesi	Önerilen çözümün çalışan bir sistem olarak ortaya konulması amacı ile	Bölüm 4
Kullanıcı Testi Yapılması	Geliştirilen yazılım için kullanıcı testlerinin yapılması	Önerilen çözümün kullanılabilirlik durumunun incelenmesi	Bölüm 5
Tez Yazımı	Bulguların ve geliştirilen sistemin raporlanması		

2. LİTERATÜR ARAŞTIRMASI

Tez kapsamında yapılan çalışmalar sürecinde coğrafi bilgi sistemleri, üç boyutlu kadastro sistemleri, mekansal verilerinin modellenmesi, saklanması ve işlenmesi, LADM standardı, geometrik verilerin üç boyutlu görselleştirilmesi ve web ortamında üç boyutlu görselleştirme konularında gerçekleştirilmiş geçmiş çalışmalar ulusal ve uluslararası düzeyde incelenmiştir.

Giriş bölümünde belirtildiği gibi bu çalışma ile LADM standardında olan üç boyutlu coğrafi verilerin, web ortamında sanal bir küre üzerinde görselleştirilmesi hedeflenmiştir. Bu kapsamda dört ayrı temel konuda literatür araştırması yapılmıştır. Bunlardan ilki coğrafi bilgi sistemlerinin incelenmesi olmuştur. İkinci olarak üç boyutlu coğrafi bilgi sistemleri ve bu sistemler aracılığıyla gerçekleştirilen görselleştirme işlerinin incelenmesidir. Ardından üç boyutlu kadastro sistemleri ile ilgili çalışmalar incelenmiştir. Son olarak da LADM standardı ile alakalı gerçekleştirilen akademik çalışmalar incelenmiştir. Belirtilen incelemeler, bu bölümde dört başlık altında açıklanacaktır.

2.1 Coğrafi Bilgi Sistemleri

Coğrafi Bilgi Sistemleri (GIS), coğrafi referanslı verilerin ve semantik verilerin depolanması, sorgulanması, manipülasyonu, analizi ve gösterimi için kullanılan bilgisayar destekli sistemlerdir (Mark, Chrisman, Frank, McHaffie, & Pickles, 2018). Üç boyutlu coğrafi bilgi sistemleri ise coğrafi bilgi sistemleri çatısı altında konumlandırılan ve yükseklik bilgisine sahip coğrafi referanslı verilerin ve semantik verilerin ele alındığı bir araştırma alanıdır.

Coğrafi bilgi sistemi kavramının modern anlamda ele alındığı ilk akademik çalışmalardan biri Roger Tomlinson'un, dünyanın ilk coğrafi bilgi sistemi olan "Kanada Arazi Envanteri" sisteminin geliştirilmesi sırasında edindiği tecrübe ve bilgiyi sunduğu doktora tezidir. Coğrafi referanslı verilerin saklanması, işlenmesi, gösterilmesi ve semantik veriler ile bağlantılı bir şekilde ilişkilendirilmesi gibi

konularda bilgisayar destekli yordamlar ve öneriler tez kapsamında sunulmaktadır (Tomlinson, 1974).

Coğrafi bilgi sistemleri, bazı akademik kesimler tarafından seksenli yıllardan itibaren coğrafya bilimini yeniden canlandırma potansiyeline sahip bir disiplin olarak görülmüştür (Openshaw, 1991). Coğrafya bilimi dışında pek çok farklı bilimsel disiplin ile beraber kullanılarak verinin anlamlı bilgi olarak analizinde de rol oynamıştır. Coğrafi ve semantik verilerin aynı bağlam içerisinde incelenebildiği bu disiplin, günümüzde pek çok farklı alanda farklı teknolojik uygulamalar aracılığıyla kullanılmaktadır.

Sık olarak rastlanan kullanım alanlarından birisi şehir planlamacıları, şehir yöneticileri, kanun koyucuların ve her türlü hizmet sağlayıcılarının stratejik ve yatırımsal karar vermelerinde bilgi sağlayan ya da sunulan hizmetlerin kalitesini, sürdürülebilirliğini ve verimliliğini artıran meskun mahallerin ele alındığı coğrafi bilgi sistemi uygulamalarıdır. Vinod Kumar ve diğerleri kitaplarında, akıllı şehir kavramını ve akıllı şehirlere hizmet edebilecek bir coğrafi bilgi sistemini sürdürülebilir şehirleşme, çevreye duyarlı şehirleşme, mal taşınması, ulaşım, trafik gibi pek çok farklı açıdan Hindistan özelinde incelemişlerdir (Vinod Kumar, 2014). Coğrafi bilgi sistemlerinin, meskun bölgeler özelinde farklı konuları ele alan kapsamlı kullanımlarının dışında, bazı meskun bölgeler özelinde oluşan sorunların çözümlenmesinde de kullanıldığı görülmüştür. Örneğin Brezilya'da bulunan Goiás eyaletinde ortaya çıkan atıkların %93'ünün kanunlara aykırı bir biçimde elden çıkarılması sebebiyle oluşan sorunu çözmek adına yeni atık depolama alanlarının tespit edilmesinde yardımcı olan bir coğrafi bilgi sistemi geliştirilmiştir (Colvero, Gomes, Tarelho, Matos, & Santos, 2018). Meskun bölgelerde ve yol çevrelerinde oluşan, yayaların karıştığı trafik kazalarının incelenmesi gibi farklı tür konularda da coğrafi bilgi sistemleri kullanılabilir. 2006-2015 yılları arasında Danimarka'da yayaların karıştığı trafik kazalarının gerçekleştiği bölgelerin coğrafi olarak incelenmesi sonucunda hem alt yapısal hem de davranışsal çözüm ve kaza engelleme önerilerinde bulunulmuştur (Prato, Kaplan, Patrier, & Rasmussen, 2019).

Coğrafi bilgi sistemleri kamu sağlığı ve hastalık takibi gibi farklı alanlarda da kullanılmaktadır. Coğrafi ve anlamsal bilgileri, eş zamanlı kullanarak analizlerin yapılması ve elde edilen bilginin paylaşılmasıyla sağlık alanında da fayda sağlamaktadır. Örneğin; Akdeniz havzasında yaşayan atlar ve insanlar arasında

gözlemlenen Batı Nil hastalığının yayılmasını kontrol altına almak ve bu hastalıkla daha etkin bir şekilde uluslararası düzeyde mücadele edebilmek için web tabanlı bir coğrafi bilgi sistemi geliştirilmiştir (Savini, ve diğerleri, 2018). Sistem, hastalıkla alakalı verilerin daha etkin bir şekilde toplanabilmesini, işlenebilmesini, dağıtılabilesini sağlamaktadır ve sayede hastalıkla mücadelenin eşgüdömlü bir şekilde gerçekleştirilebilmesini yardımcı olmaktadır.

Coğrafi bilgi sistemlerinin tarım, hayvancılık ya da balıkçılık gibi ekonomik ve beslenme ile alakalı konularda da sürdürülebilirlik, karlılık, verim ve iyileştirme sağlayan uygulamaları mevcuttur. Artvin Yusufeli ilçesinde gerçekleştirilecek olan baraj inşaatları sonrasında su altında kalacak bölgeler dışında, yeni tarımsal kullanım alanlarının etkili bir şekilde belirlenebilmesi adına toprak özellikleri, eğim ve erozyon gibi özellikler incelenerek olası tarım bölgeleri verimlerine göre sınıflandırılmış ve belirlenmiştir (Akinci, Özalp, & Turgut, 2013). Hayvancılık ile alakalı olarak da; yüksek miktarda hayvan bulunan bölgelerde, çiftliklerin ve çevrenin coğrafi verileri incelenerek, hayvan çiftliklerinin riskleri belirlenmiş ve ölçölmüştür (Gallego, Calafat, Segura, & Quintanilla, 2019). Gelişmekte olan uzaktan algılama teknolojileri ve coğrafi bilgi sistemlerini eş zamanlı olarak kullanarak sürdürülebilir, karlı ve etkili bir şekilde balıkçılığın ve su ürünleri yetiştiriciliğinin yapılmasına yardımcı olan uygulamaları da bulunmaktadır (Meaden & Aguilar Manjarrez, 2013).

2.2 Üç Boyutlu Coğrafi Bilgi Sistemleri

Tez çalışmalarında gerçekleştirilen kaynak tarama sürecinde, tez kapsamında gerçekleştirilen çalışmanın içine dahil olduğı meskun bölgelerin üç boyutlu olarak ele alındığı coğrafi bilgi sistemleri incelenmiştir. Gerçekleştirilen bu araştırma kapsamında meskun bölgeler özelinde şehirler, binalar ve iç mekanların dahil olduğı konular araştırılmıştır.

Geçmişte var olmuş şehirlerin insanlara tanıtılması, nasıl korunmaları gerektiğı ile ilgili bilgi sahibi olunması ya da restorasyon süreçlerinde bir kaynak olarak kullanılmaları açısından üç boyutlu coğrafi bilgi sistemleri kullanılabilir. Eş güdümlü olarak kullanılan lazer tarama ve fotogrametrik teknikleriyle, geometrik doğruluk payı yüksek ve görsel olarak yeterli üç boyutlu şehir modelleri oluşturulabilir ve oluşturulan bu modeller web ortamında da sunularak turizm, şehir

planlama ya da kültürel mirasın ileri nesillere aktarılması gibi amaçlar için kullanılabilmektedir (Balsa-Barreiro & Fritsch, 2018) (Chenaux, ve diğerleri, 2019). Kadastro haritaları ve lazer tarama teknikleri beraber kullanılarak elde edilen 2 boyutlu eski bina temelleri bilgisi işlenerek üç boyutlu şehir modelleri de elde edilebilmektedir (Ingensand, Nappez, Produit, & Chassin, 2018). Sondaj sonucu elde edilen toprak katmanlarından elde edilen veriler işlenerek şehirlerin geçmişteki paleotopografik modellenmeleri de yapılmaktadır (Garcia-León, ve diğerleri, 2017). Şehirlerin geçmişteki haritaları ve bu haritalarda bulunup günümüze kadar gelen binaların fotoğrafları kullanılarak da şehir modellemeleri yapılabilmektedir ve bu modellemeler sanal gerçeklik uygulamalarına dönüştürülerek, modellenen eski şehrin içinde gezinme deneyimi sağlayabilmektedir (Ramsey, 2017).

Tarihsel şehirlerin yanı sıra modern şehirlerin de modellenmesi ve görselleştirilmesi birçok fayda sağlamaktadır. Modern şehirler de tarihsel şehirlerin modellendiği yöntemlerle modellenemediği daha farklı şekillerde de modellenmektedir. Günümüzde şehirlerin içinde bulunan binaların kadastral olarak taban bilgileri sık bir şekilde karşımıza çıksa da yükseklik bilgilerini bulabilmek daha zordur ve bu gibi durumlarda, kadastral taban bilgileri ve makine öğrenme kullanılarak, üçüncü boyut bilgisi bulunmayan şehirlerin üç boyutlu olarak modellenmesi mümkündür (Biljecki, Ledoux, & Stoter, 2017). Bina tabanlarının vektör bilgileri ve uzaktan algılama sonucunda elde edilen yükseklik bilgileri makine öğrenme teknolojisiyle beraber kullanılarak da şehirlerin üç boyutlu modellemeleri gerçekleştirilebilmektedir (Park & Guldmann, 2019). Üç boyutlu şehir modelleme sürecinde kullanılan bir diğer yöntem ise karasal lazer tarayıcıları ve karasal fotogrametri yöntemiyle elde edilen mekansal verilerin kullanılması ile gerçekleştirilmektedir (Sahin, ve diğerleri, 2012). Havadan çekilen eğik fotogrametri görüntüleriyle aracılığıyla da üç boyutlu şehir modellemeleri gerçekleştirilebilmektedir (Lemmen & van Oosterom, 2003).

2.3 Üç Boyutlu Kadastro

Geleneksel kadastral yaklaşımda araziler, binalar ve bağımsız bölümler ile ilgili bilgiler genellikle iki boyutlu kadastro haritaları ile ifade edilmektedir. Smart ve Priebbenow'a göre geçtiğimiz on yılda, kadastro sisteminin algısı ve kullanım tipleri önemli ölçüde gelişmiştir (Smart & Priebbenow, 2018). Gelişen Dünya'nın gereksinimlerini tam anlamıyla karşılayamayan iki boyutlu kadastral yaklaşım

günümüzde üçüncü boyuta da ihtiyaç duymaktadır. Lemmen ve Oosterem'a göre üçüncü boyutun gerekli olduğu bazı uygulamalar, binaların hacmine dayalı vergilendirmenin gerçekleştirildiği, peyzajdaki değişikliklerin görsel analizlerinin yapıldığı, cep telefonu şebekeleri için vericilerin konumlarının belirlendiği, doğal kaynakların ve yeraltının modellenmesinin yapıldığı uygulamaları da kapsayan pek çok tip uygulamalardır (Lemmen & van Oosterom, 2003).

Üç boyutlu kadastral çalışmalarının ve uygulamalarının eşgüdümlü olarak çalıştığı ve hizmet ettiği en önemli olgular, çalışmanın gerçekleştirildiği bölge özelinde uygulanan kadastral kanunlar, gereklilikler ve kısıtlardır. Polonya özelinde tasarlanmak istenen üç boyutlu kadastral sistem için öncelikli olarak kullanılan ve teknik olarak incelenen araştırma materyali, emlak kadastro ile ilgili kanunsal dökümanlardır (Mika & Jurkiewicz, 2018). Kanunlar genel olarak beraberinde kurallar ve kısıtlamalar getirmektedir. Gelen bu kurallar ve kısıtlamalar da üç boyutlu kadastral uygulamaların çerçevesinin, gelişebileceği alanların ve gelişim sürecinde karşılaştığı zorlukların belirlenmesinde yol gösterici olarak kullanılmaktadır. Üç boyutlu kadastro sistemlerin karşılaştığı hukuki sorunların akademik olarak incelendiği ve bu sorunların sistemlere olan etkisinin araştırıldığı çalışmalar gerçekleştirilmektedir (Ho, Rajabifard, Stoter, & Kalantari, 2013). Sorunların incelendiği çalışmaların yanı sıra, uluslararası hukuk ölçeğinde uyumun ve ortak terimlerin sağlanması alanlarında da çalışmalar yapılmaktadır (Dimopoulou & Elia, 2013). Üç boyutlu kadastral kanunlara yol gösterecek çalışmalara da akademik olarak gerçekleştirilmektedir (Kitsakis & Dimopoulou, 2014).

Üç boyutlu kadastral sistemlerin oluşturulmasında, hali hazırda bulunan 2 boyutlu sistemlerin güncellenmesi ve geliştirilmesi kullanılan yaklaşımlardan birisidir. Atina Teknik Üniversitesi' nin örnek olarak kullanıldığı bir çalışmada, üniversite kampüsünün kadastral ve mimari planları kullanılarak, kampüs üç boyutlu ve kadastral bilgileri içeren bir şekilde modellenmiştir (Tsiliakou, Labropoulos, & Dimopoulou, 2013). Gereksinimler sebebiyle iki boyutlu kadastral sistemden, bina bilgilerinin de temsil edilebildiği bir kadastral sisteme geçiş yapan Slovenya'da, olası tüm üç boyutlu kadastral nesnelerin temsil edilemediği hukuki olarak görülmüştür ve bu sistemde bulunan veriler kullanılarak daha yetkin bir kadastral sistemin geliştirilmesi için çalışmalar yapılmıştır (Drobež, Fras, Ferlan, & Lisec, 2017). 150

yılı aşkın bir süredir hukuki olarak üç boyutlu kadastral sistem kullanan Yeni Zelanda’da, üç boyutlu kadastral olgular fotoğraflar, kesit planları ve yükselti planları aracılığıyla temsil edilmektedirken, üç boyutlu sayısal bir kadastral sisteme geçmeye yönelik çalışmalar yapılmaktadır (Gulliver, Haanen, & Goodin, 2017).

Hali hazırda bulunan 2 boyutlu kadastral sistemlerin verileri analiz edilerek, yeni bir sistem geliştirilmesi yaklaşımına rastlamak da mümkündür. Çin’in Shenzhen bölgesinde kullanılmakta olan kadastral sistemin incelenmesi sonucu elde edilen bulgular ışığında sahip olan verileri kullanarak çalışabilecek ayrı bir üç boyutlu kadastral sistem geliştirilmiştir (Guo, ve diğerleri, 2013). Üç boyutlu kadastral sistemlerin geliştirilmesi sürecinde teknik tasarımların yapıldığı ve karşılaşılan teknik sorunların analizinin gerçekleştirildiği çalışmalar mevcuttur (Shojaei D. , Olfat, Briffa, & Rajabifard, 2017). Üç boyutlu kadastral sistemlerin prototip olarak sadece birkaç yapı için geliştirildiği çalışmalar da mevcutta gerçekleştirilmektedir (Shojaei D. , Olfat, Rajabifard, & Briffa, 2018).

2.4 Land Administration Domain Model (LADM)

Tez kapsamında temel alınan standart LADM ile alakalı okumalar, Paulsson ve Paasch’ın gerçekleştirdiği LADM konusundaki literatür anketinde belirlenen teknik ve emlak kayıt konuları ve yasal ve örgütsel konular başlıkları altında bulunan teknik ve emlak kayıt konularını kapsayacak şekilde gerçekleştirilmiştir (Paulsson & Paasch, 2015). Yasal ve örgütsel düzeyde LADM’e uyum sürecinde gerçekleştirilmesi önerilen süreçleri vaka çalışmaları ile örnekleyen bazı çalışmalar teknik süreçlerden farklı bir bakış açısına sahip olduğu için tez kapsamında gerçekleştirilen araştırmalar kısmında değinilmemiştir (Kalantari, Dinsmore, Urban-Karr, & Rajabifard, 2015).

LADM bir standart olduğu için teknolojik uygulamaların gerçekleştirildiği teknik çerçeveyi ve sınırlarını belirlemek için kullanılmaktadır. Gerçekleştirilen uygulamanın nasıl ve hangi teknik araçlar vasıtasıyla yapılacağı ile ilgili uygulamayı gerçekleştiren kurumların ya da şahısların tercihinin kalmıştır. Tercih sürecinde ya da sonrasında gerçekleştirilecek olan olası hataları ve eksiklikleri giderebilmek adına ve daha doğru teknik uygulamalar geliştirilmesi için akademik çalışmalar mevcuttur. Üç boyutlu kadastral sistemlerin LADM standardında uygulanmasında, İsviçre’de geliştirilen nesne ilişkisel modelleme dili olan INTERLIS ile eşgüdümlü çalışmasının

incelendiği ve örneklendiği vaka çalışmaları ile karşılaşmak mümkündür (Kalogianni, Dimopoulou, Quak, & Van Oosterom, 2017). LADM standardına uygun, katmansal mekansal nesneler ile çalışması hedeflenen kavramsal, mantıksal ve fiziksel tasarım süreçlerinin konu edildiği bir veri tabanı yönetim sisteminin geliştirilmesini gerçekleştiren ve bu veri tabanı yönetim sistemi ile de Malezya özelinde bir vaka çalışmasının gerçekleştirildiği çalışmalar da bulunmaktadır (Nasorudin, Hassan, Zulkifli, & Rahman, 2016). Mevcut üç boyutlu kadastral veri modellerini inceleyen ve bu veri modellerinde bulunan eksiklikleri ya da iyileştirilmesi gereken konuları inceleyerek daha yetkin ve ihtiyaçlara hizmet eden bir veri modelinin geliştirilmesi hedefleyen çalışmalar ile karşılaşmak da mümkündür (Aien, Kalantari, Rajabifard, Williamson, & Wallace, 2013). Belediyeler, vergi daireleri, bankalar ve devlet daireleri gibi pek çok kurumun ihtiyaç duyduğu üç boyutlu kadastral verilerin web ortamında paylaşılmasını sağlayan web hizmetlerinin incelendiği ve bu web hizmetlerin sebep olduğu zorlukların üstesinden gelmek amacıyla LADM standardına uyumlu bir web hizmetinin geliştirildiği ve hizmet sonucunun üç boyutlu olarak görselleştirildiği çalışmalar gerçekleştirilmektedir (Zardiny & Hakimpour, 2015).

Bina, tarla, arsa ya da ortak kullanım alanları gibi olguların mekansal bilgilerine doğrudan ya da dolaylı olarak etki eden topolojik ögeler üç boyutlu kadastral sistemlerin geliştirilmesinde ve etkin kullanılmasında önem arz etmektedir. Üç boyutlu kadastral sistemlerde topolojik bilgilerin temsili açısından da çalışmalar gerçekleştirilmiştir. Zulkifli ve diğerlerinin çalışmasında, farklı üç boyutlu topolojik modellerin özellikleri incelenmiştir ve gerçekleştirilecek uygulamanın kullanım amacı özelinde nasıl bir topolojik modelin kullanılması gerektiği ve gerçekleştirilecek uygulamanın gereksinimlerine göre bir topolojik modelin kullanılması gerekliliği gibi öneriler sunulmuştur (Zulkifli, Abdul Rahman, & Van Oosterom, 2015). Var olan topolojik veri modellerinin incelenmesinin yanı sıra LADM standardına uygun yeni bir topolojik modelin geliştirilmesinin gerçekleştirildiği çalışmalarda mevcutta gerçekleştirilmektedir (Zulkifli, Rahman, & Hassan, 2016).

LADM standardı çatısı altında gerçekleştirilen çalışmalar arasında kat mülkiyeti ya da yer altı kullanım alanlarının incelendiği farklı tip çalışmalar da mevcuttur. Çin kadastral kanunları ve LADM standardı incelenerek, ülke özelinde kat mülkiyeti

sahiplik yapısının modellendiği ve vaka çalışmaları ile bu modellerin üç boyutlu olarak görselleştirildiği çalışmalar gerçekleştirilmektedir (Li, Wu, Zhu, Duan, & Luo, 2016). Kat mülkiyeti ile alakalı ülkeler ya da bölgeler düzeyinde karşılaştırılmaların gerçekleştirilmesi açısından da LADM standardı kullanılabilir. Kanada’da bulunan Quebec ve Fransa’da bulunan Alsace Moselle bölgelerinde bulunan kat mülkiyetine dayalı kadastral olgular LADM standardına uygun bir şekilde modellenmiştir ve iki bölgenin kadastral sistemleri arasında gerçekleştirilen kıyas sonucu, standardın kıyas yapmak için kullanılabilirliği sorgulanmıştır (Pouliot, Vasseur, & Boubehrezh, 2013). Güney Kore’de artan şehirleşme sonucu yer altında bulunan kullanım alanların artmasıyla oluşan kadastral sorunları çözmeye adına LADM standardına uygun ve üç boyutlu bir veri modeli önerisi gerçekleştirilmiştir (Kim & Heo, 2017). Yer altında bulunan kadastral öğelerin LADM standardı kapsamında temsil edilmesi ve bu temsilin üç boyutlu olarak görselleştirilmesi adına yeni teknolojilerin geliştirilmesi ve denenmesi ile alakalı çalışmalar da mevcuttur (Janecka & Bobíková, 2018).

LADM standardının ülkeler özelinde uygulandığı ve üç boyutlu kadastral sistemlerin üretildiği pek çok akademik kaynak bulunmaktadır. Kaynakların ortak özelliği, hali hazırda bulunan kadastral sistemlerin incelenerek LADM standardına uyacak şekilde yeniden düzenlenmesi süreçlerini açıklamaları, süreçler boyunca karşılaşılan sorunların tanımlanması ve çözümleri için yaklaşımlar sunmalarıdır. Hukuki açıdan mekansal bilgilerin kullanıldığı tapu kurumu ve fiziki açıdan mekansal bilgilerin kullanıldığı kadastro kurumu gibi iki farklı kuruma sahip olan Polonya’da, bu iki kurumun eş zamanlı çalışabileceği ortak bir bilgi platformunun kurulması ve daha anlaşılır bir kadastral modelin gerçekleştirilmesini hedeflenerek ülke çapında kullanılması önerilen bir üç boyutlu mekansal sistemin LADM standardında tasarlandığı bir çalışma mevcuttur (Bydłosz, 2015). Polonya benzeri iki farklı tapu ve kadastro kurumuna sahip olan Malezya’da da LADM standardına uygun tekil bir coğrafi bilgi sistemi geliştirilmesi ihtiyacı duyulmaktadır ve ihtiyacı gidermek adına önerilere sahip akademik çalışmalar gerçekleştirilmektedir (Zulkifli, ve diğerleri, 2015). Çalışma özelinde incelenen ülkenin kadastral sistemini geliştirmeyi hedefleyen ve bu süreç boyunca da uluslararası standartlara uygun ve kadastral veri alışverişini destekleyen LADM standardında bir veri modelinin geliştirildiği çalışmalar da incelenmiştir (Polat & Alkan, 2018). 2012 yılında LADM standardına

uygun iki boyutlu bir kadastral sistem tasarlanan Kore’de mevcut sistemin üç boyutlu bir sisteme evrilmesi için gerekli incelemelerin ve geliştirilmelerin yapıldığı, elde edilen bulgular ışığında bu sürecin nasıl yürütülmesi gerektiği ile ilgili önerilerin sunulduğu çalışmalar gerçekleştirilmiştir. (Lee, Kim, Kwak, Lee, & Choi, 2015). 2014 yılında yürürlüğe giren kanunlar ışığında ulusal bir sayısal kadastral sistem geliştirilmesi gerekliliğini belirleyen Çekya hükümetine bu süreçte yardımcı olunması açısından LADM standardına uygun, mekansal, hukuksal ve anlamsal verilerin dahil olduğu bir sistemin geliştirildiği çalışmalar gerçekleştirilmiştir (Janečka & Souček, 2017). Hırvatistan için de benzer bir çalışma gerçekleştirilmiştir, binaların ve binaların içinde bulunan daire ve ofis gibi birimlerin de üç boyut olarak örnekler temelinde görselleştirildiği, LADM standardına uygun çalışmalar gerçekleştirilmiştir (Vučić, Mađer, Roić, & Vranić, 2017). Tarla, arsa, bina bina gibi genel geçer kadastral öğeler dışında deniz kıyıları ve orman gibi ekonomik olarak sürdürülebilir değer katan kullanım alanları için de farklı yönetmelikleri ve kanunları olan Çin Halk Cumhuriyeti özelinde LADM standardına uygun iki ve üç boyutlu bir kadastral sistemin gerçekleştirilmesi için çalışmalar gerçekleştirilmektedir (Yu, Li, He, Zhao, & Li, 2017). LADM standardı geliştirilebilir ve bölgesel bazı kısıtlara uyumlandırılabilir bir yapıya sahip olduğu için, standardın uygulandığı modern kadastral sistemler dışında kalan yöresel sorunları çözmek için kullanılabilir. Günümüzde standardın bir parçası olmasa da Brezilya’da yaşayan kabilelerin kadastral haklarının belirlenmesi ve korunması adına gerçekleştirilen çalışmalar vardır (Paixao, ve diğerleri, 2015).

LADM standardı içerisinde olan ve kadastral öğelere dahil iç mekanlarla alakalı çalışmalar da gerçekleştirilmektedir. İç mekanlarda erişilebilirlik ve erişim izni konularını, LADM standardı çerçevesinde bulunan sahiplik ve kullanım hakkı bilgileriyle beraber inceleyen araştırmalar gerçekleştirilmiştir (Alattas, ve diğerleri, 2017). İç mekanlarda ki erişilebilirlik ve erişim izni konularında yapılan bir diğer araştırmada ise, acil durumlarda gerçekleşecek olan bina tahliyelerinde LADM modelinin kullanılmasıyla alakalı üç boyutlu bir vaka analizi çalışması gerçekleştirilmiştir (Alattas, Van Oosterom, Zlatanova, Hoeneveld, & Verbree, 2018). OGC standardı olan ve iç mekanlar özelinde verilerin temsili ile alakalı bilgiler içeren IndoorGML’in, erişilebilirlik yönünden geliştirilmesini hedefleyen bu geliştirme sürecinde LADM standardında bulunan erişilebilirlik ile alakalı kuralların

uygulanması gerektiğini öneren çalışmalar da gerçekleştirilmiştir (Zlatanova, Van Oosterom, Lee, Lic, & Lemmen, 2016).

Geleneksel kadastral yaklaşımdan üç boyutlu kadastro sistemlerine geçiş sürecinde, iç mekanlarla alakalı bilgilerin toplanması sürecinde standard olmamış ve farklı tip kullanımlara sahip yöntemler yerine daha hızlı veri elde edilmesini sağlayan yöntemlerin önerildiği çalışmalar da gerçekleştirilmektedir (Jamali, Boguslawski, Duncan, Gold, & Abdul Rahman, 2013).

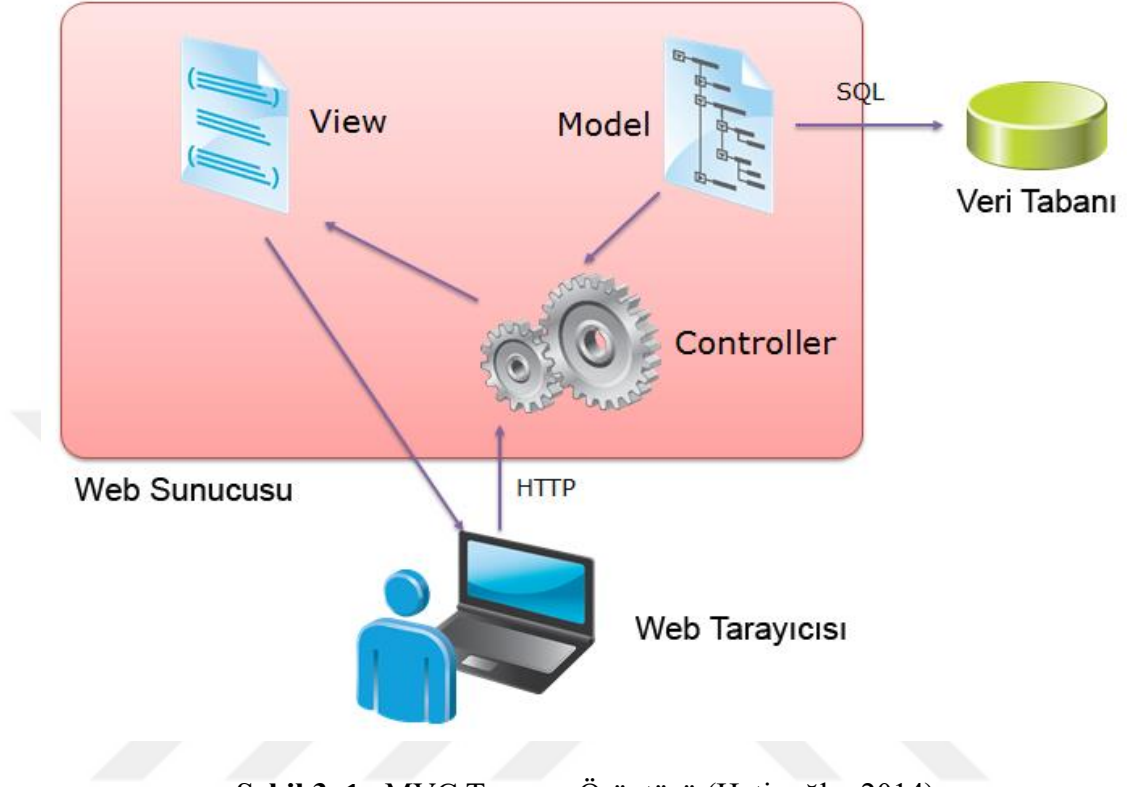
LADM standardına dahil edilmesi önerilen, değerlendirilen ve standart ile ilişkili geliştirilen çalışmalara da rastlamak mümkündür. Çalışmalar genellikle, standardın sağladığı çatı altında farklı konuların incelenmesi ya da değerlendirilmesi amacıyla gerçekleştirilmektedirler. Athanasiou ve diğerleri çalışmalarında, denizde ve kıta sahanlığında bulunan alanların yönetimi ve temsili ile alakalı LADM'e eklenebilecek bir ek model önermişlerdir ve bu modelin Yunanistan ve Trinidad Tobago Cumhuriyeti özelinde uygulandığı bir vaka çalışmasını sunmuşlardır (Athanasiou, ve diğerleri, 2017). LADM standardının sağladığı çatının kullanılmasıyla taşınmazların ve vergilendirilecek kişilerin tanımlanmasını, taşınmazların değerlendirilmesini ve bu süreçte elde edilen bilgilerle vergilendirmenin gerçekleştirilmesini sağlayan bir modelin önerildiği çalışmalar da gerçekleştirilmektedir (Çağdaş, ve diğerleri, 2016). Günümüzde gelişen teknoloji ve şehirleşmenin sonucu olarak ortaya çıkan yüksek katlı binaların beraberinde getirdiği sorunlardan birisi olan gün ışığı kullanım hakkı ile ilgili LADM standardına uyumlu bir model önerisi çalışması gerçekleştirilmiştir ve bu öneri ile alakalı Çin Halk Cumhuriyeti kanunları rehber alınarak bir vaka çalışması yapılmıştır (Li, ve diğerleri, 2019).

3. VERİ ve YAZILIM ALTYAPISI

Tez kapsamında geliştirilen iki ayrı yazılım modülü bulunmaktadır. Temel olarak tanımlanmaları gerekirse modüllerden birisi mekansal veri oluşturulması işlemini gerçekleştirmektedir. İkinci modül de oluşturulan mekansal verinin web ortamına aktarılması, görselleştirilmesi ve kullanıcı ile etkileşim işlemlerinin yerine getirilmesini sağlamaktadır. Geliştirilen iki yazılım modülü de veri tabanı kullanan, uygulama katmanında yapılması istenen iş ile ilgili mantıksal işlemler gerçekleştiren ve web tabanlı bir arayüze sahip olan uygulamalardır. Üç farklı katmanda işlemleri gerçekleştiren bu yazılım modüllerinde tasarım aşamasında ulaşılmak istenen hedef, katmanların birbirine olan bağımlılıklarının en alt düzeyde tutulmasıdır. Katmanlar arasındaki bağ azaldıkça, katmanların farklı bir projede kullanılabilme, katmanlarda kullanılan teknolojilerin güncellenebilme ve muadilleri ile değiştirilebilme imkanları artmaktadır.

Tasarım aşamasında hedeflere ulaşabilmek adına modüllerin mantıksal modellemelerinde mimari ve tasarım örüntülerden faydalanılmıştır. Tasarım örüntüleri yazılım ortamında oluşan problemleri ve bu problemleri çözmek amacıyla tekrar tekrar kullanılacak çözüm yöntemlerini tanımlamak için kullanılmaktadır (Alexander, Ishikawa, & Silverstein, 1977). Faydalanılan mimari örüntü de Model-View-Controller (MVC) olarak isimlendirilmektedir. MVC mimari örüntüsünde uygulama bileşenleri üç mantıksal kavram olarak ayrılmaktadır. Bunlar; uygulamada kullanılan veri modeli, bu veri modelinin kullanıcıya sunum şekli ve kullanıcının bu model ile etkileşime geçme şeklidir (Krasner & Pope, 1988). MVC mimari örüntüsü sahip olduğu katmanlar arasında iletişimde ve genel çalışma prensibinde tasarım örüntülerinden faydalanmaktadır (Gamma, Helm, Johnson, & Vlissides, 1994). Nesne yönelimli yazılım ve kullanıcı arayüz tabanlı web uygulamaları geliştirme sürecinde tecrübe edilen sorunlar ve uygun çözümler ile ilgili yol gösterici olan bu kavramları temel alarak tez kapsamında geliştirilen iki modül de MVC mimari örüntüsü doğrultusunda katmanlara ve paketlere ayrılmıştır. Gerçekleştirilen bu ayrıştırma işleminin yazılım bileşeni görevleri bazında gösterimi Şekil 3.1’de

sunulur. Katmanlar arasında gerçekleşen iletişim arayüzler aracılığıyla gerçekleşmektedir ve bu arayüzlerin kullanılması için gereken kurallara uyulduktan sonra modüller kolayca değiştirilebilmektedir.



Şekil 3. 1 : MVC Tasarım Örüntüsü (Hatipoğlu, 2014).

Bu bölümde tez kapsamında kullanılan MVC mimari örüntüsünün model, uygulama ve sunum katmanlarında kullanılan teknolojiler ile ilgili alt yapı bilgileri verilmiştir. Ayrıca bahsi geçen teknolojilerin kullanım nedenleri ve örnekleri de bulunmaktadır.

3.1 Mekansal Veri Yapıları

Bilgisayar biliminde kullanıldığı bağlamda, veri yapıları; verilerin verimli bir şekilde erişilebilmesi ve değiştirilebilmesi için, verileri bilgisayarda organize etmenin ve saklamanın bir yoludur (Cormen, Thomas H. and Leiserson, Charles E. and Rivest, Ronald L. and Stein, 2009). Daha genel bir ifade ile de “Bir veri yapısı, veriler üzerinde bir dizi işlemi gerçekleştirmek için kullanılan bir yöntemdir” (Brass, 2008). Veri yapılarının tasarımları, bu veri yapısını kullanacak verilerin toplanma, saklanma, erişilme, işlenme, sunulma ve iletilme yöntemlerine göre şekillenmektedir. İlgili işlemleri gerçekleştirirken karşılaşılan kısıtlar, eksiklikler ya da kazanımlar,

veri yapısının kullanım alanlarının seçimlerinde göz önünde bulundurulması gereken birer etkidir.

Mekansal veri; nokta, çizgi, dörtgen, bölge, yüzey ve hacimlerden oluşur (Samet, 1994). Mekansal veri, uzayda bulunan bir nesnenin konum, şekil, coğrafi bilgi ve bu özellikleri arasında bulunan ilişkilerin bilgisi olarak da genişletilebilir. Gelişen veri edinme ve saklama teknolojileri ile mekansal veri ile ilgili yapılan tanımlar da gelişmektedir.

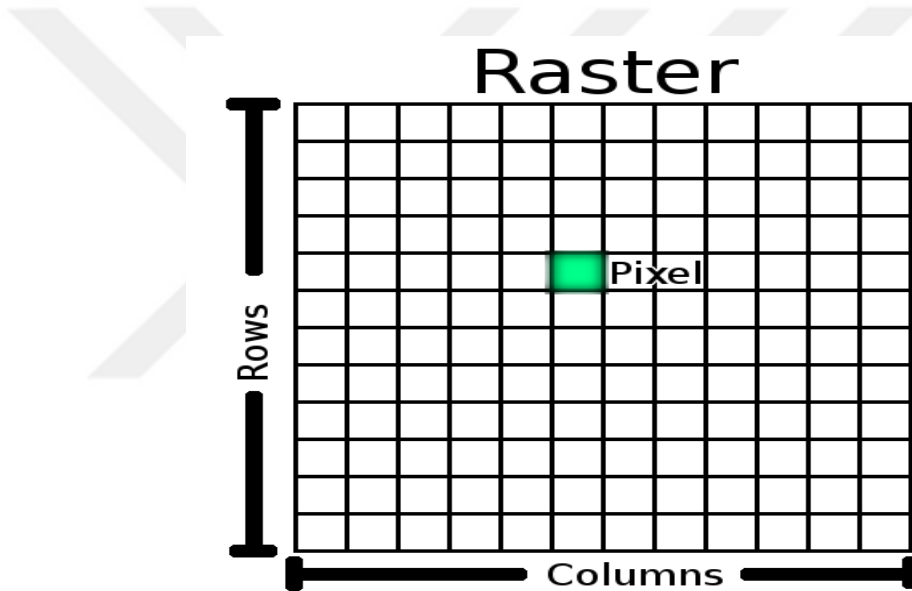
Mekansal veriler için de toplanma, saklanma, erişilme, işleme, sunulma ve iletilme işlemlerinde kullanılabilmek için veri yapılarından faydalanılmaktadır. Mekansal veriler, belirli bir alanda özelleştirilmiş veriler olmaları ve üzerinde gerçekleştirilecek işlemlerin çeşitlilik göstermesi sebebiyle kendileri için tasarlanmış veri yapılarına ihtiyaç duymaktadırlar. Mekansal veriler belirttikleri uzay boyutlarına göre iki ve üç boyutlu mekansal veriler biçiminde tanımlanabilirler, ayrıca iki boyutlu mekansal verilere yükseklik bilgisi üst veri olarak eklenerek 2.5 boyutlu mekansal veriler de tanımlanabilir (Abdul-Rahman & Pilouk, 2008).

3.1.1 İki boyutlu mekansal veriler

İki boyutlu mekansal veriler, yükseklik bilgisine sahip olmayan ya da bu bilgidan arındırılmış verilerdir. Yükseklik bilgisine sahip olmadıkları için genellikle Dünya üzerinde ya da seçilen taban üzerinde bulunan nesnelerin konum ve şekil özelliklerini belirtmektedirler. İki boyutlu mekansal veriler kullanılarak gerçekleştirilen görselleştirmelere haritalar ya da krokileri örnek verilebilir. Coğrafi bir olgu, mekansal veri yapıları ile temsil edilmek istendiğinde; raster ve vektör olarak iki şekilde sınıflandırılabilir. Belirtilen iki mekansal veri tipi teorik olarak alan ve ayrık nesneleri coğrafi olarak temsil etmek için kullanılabilir; ancak raster ile alanlar ve vektör ile de ayrık nesneler arasında mantıksal olarak daha kuvvetli bağlar vardır (Longley, Goodchild, Maguire, & Rhind, 2005). Bir tabanlık üzerine ayrık nesnelerin yerleştirildiği modeller gibi çok çeşitli mantıksal ilişkilerin kurulması gereken uygulamalarda beraber kullanımları sık sık görülmektedir. Örneğin; gerçek dünyanın üzerine yol gibi bir iki boyutlu ayrık nesne yerleştirilmek istendiğinde Dünya tabanı olarak raster veri yapısı, ayrık nesnelerin temsili için de vektör veri yapısı kullanılmaktadır.

3.1.1.1 Raster veri yapısı

Raster veri yapısında, bilgisayar ortamında temsil edilmek istenen uzay satırlar ve sütunlar halinde hücrelere bölünerek ve işlem sonucunda oluşan her hücreye hem kendisine özel ve çevresinde olan diğer hücreler ile paylaştığı bilgiler eklenerek saklanılır. Bölme işleminde genellikle aynı büyüklükte hücreler kullanılır, yaygın olarak da kare tercih edilir. Bölünen bir alanın satırları, sütunları ve örnek bir pixel Şekil 3.2’de gösterilmiştir. Raster veri yapısında belirlenen hücrelerin boyutları ile görselleştirme sonucundaki çıktının çözünürlüğü arasında ters bir orantı vardır. Kullanılan hücrelerin boyutları büyüdükçe çıktının çözünürlüğünün kalitesi azalmaktadır. Yapılan bu işlemlere ek olarak bazı gerçeklemlerde her hücre için üst veri de eklenebilir.



Şekil 3. 2 : Raster veri yapısı (QGIS çevrim içi dokümanı, 2018).

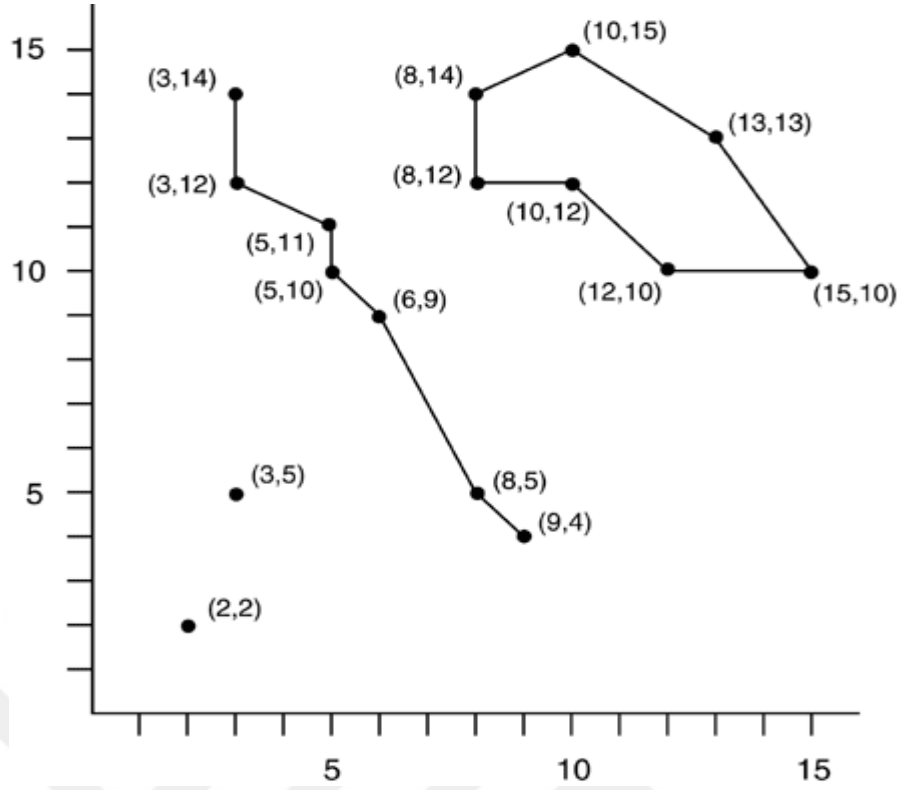
Raster veri yapıları genel olarak performans iyiliği ve mantıksal bağın kolay kurulması sebebi ile birbirine bağlı ve devamlılık sağlayan alanların bütün olarak temsil edilmek istendiği durumlarda kullanılır. Raster veri genellikle uydu görüntülemeleri ve havadan fotoğraflama yöntemleri ile elde edilir.

3.1.1.2 Vektör veri yapısı

Vektör veri yapısı, çizgisel ve yön bilgisine sahip olunan durumlarda coğrafi özellikleri bilgisayar ortamında temsil etmek için kullanılır (Kennedy, 2001). İki boyutlu uzayda coğrafi özellikler x ve y koordinatlarından oluşurlar. En temel x ve y koordinat çifti ile noktalar elde edilir ve bu noktaların sıralı şekilde listelenmesi ile

izgiler meydana gelmiř olur. Benzer bir řekilde izgiler de listeli bir řekilde ifade edildiğinde drtgen ve oklu-izgileri oluřturur; oluřturulan bu mekansal veri yapıları ile de yzeyler ifade edilebilir. Benzer yaklařım ile  boyutlu coęrafi zellikler de temsil edilebilmektedir.

Vektr veri yapıları genel olarak ayrıık nesnelerin bilgisayar ortamında temsil edilmek istendięi durumlarda kullanılırlar. Ayrıık nesneler, alanlardan daha kk olmalarına karřı genellikle daha fazla detaya sahip oldukları iin modellenmek istendiklerinde bu detayları tmyle ya da ana hatlarıyla belirtmek gerekmektedir. Vektr veri yapısında, temsil edilmek istenen ayrıık nesneler matematiksel bir dil ile ifade edilir. Vektr veri yapılarının kullanılmasının dięer bir sebebi de bu tip mekansal verilerin katman olarak st ste kullanılabilmesinin bilgisayar ortamında daha kolay olmasıdır. Mekansal verileri mantıksal olarak farklı řekillerde ayrıştırıp katman olarak kullanmak hem mekansal veri setlerinin farklı uygulamalarda kullanılabilmesini saęlar hem de grselleřtirme srecinde kullanıcı deneyimi artırır. Kullanıcılar grntlemek istedikleri coęrafi zellikleri seerek grntledikleri katmanları deęiřtirebilirler ve istedikleri grsel sonuca daha sade bir řekilde ulařabilirler. rneęin; yolların, konutların ve sosyal binaların grselleřtirildięi bir coęrafi bilgi sisteminde ayrıık yapılar tiplerine gre ayrıştırılır ve katmanlar olarak temsil edilirse, kullanıcı istedięi katman ya da katmanları seerek daha kiřiselleřtirilmiř bir deneyim yařayabilir. řekil 3.3’de iki boyutlu bir dzlem zerinde nokta, izgi, oklu ve kapalı izgi geometrileri gsterilmiřtir.



Şekil 3. 3 : İki boyutlu nokta, çizgi, çoklu ve kapalı çizgi vektör veri örnekleri (Gruver, 2018).

Nokta geometrisi:

İki boyutlu mekansal veri yapıları bağlamında nokta geometrisi x ve y koordinatlarından oluşan en temel mekansal veri temsidir. Nokta geometrisi konum ya da göreceli konum bilgisi dışında herhangi bir fiziksel özellik belirtmez. Sadece konum bilgisi içeren bir mekansal veri tipi olduğu için diğer mekansal veri yapılarının temellerini de oluşturmaktadır. Örneğin iki nokta ve bu noktaları birleştiren yol ya da yön bilgisine sahip olduğumuzda, çizgi geometrisini elde edebiliriz. Bir diğer örnek olarak da sahip olunan nokta ve yarı çap bilgileri kullanılarak matematiksel denklemler yardımıyla çember ve çemberin kapsadığı alan bilgileri temsil edilebilir.

Çizgi geometrisi:

Çizgi geometrisi en az iki nokta ve bu noktaları birleştiren yol ya da yol yönü ile ifade edilebilir. İki nokta ile çizilen doğruların başlangıç ve bitiş noktaları arasında 180 derece açı olmak zorundadır. İki den fazla nokta ile çizilen doğrularda ise böyle bir düzenin olması gerekmez. Örneğin beş nokta ile ifade edilmiş bir çizgisel geometri, eğri bir coğrafi özelliği temsil etmek için kullanılabilir. Coğrafi olan özellikleri çizgisel geometri ile daha isabetli belirtebilmek için, geometriyi oluşturan

noktaların sayıları artırılabilir. Bir önceki örnekte belirtilen beş nokta ile zikzak benzeri bir şekil çizilebilirken, noktalar arasına daha fazla yeni nokta eklenerek köşeler yumuşatılır ve eğri yapılar ortaya çıkarılabilir.

Çizgi geometrisi bilgisayar ortamında saklanırken kendisini oluşturan noktaların dizisi olarak saklanır. Saklanan dizide bulunan nokta sayısı arttıkça saklama alanı da doğrusal olarak artmaktadır. Saklanan dizide bulunan nokta sayısının artması ayrıca belirtmek istenen coğrafi özelliğin temsil kalitesini de yükselteceği için saklama alanı ve temsil kalitesi arasında da doğru orantılı bir ilişki söz konusudur. Bir nokta ve eğri bir çizgi geometrisi elemanı kullanılarak, çember (kapalı alan) oluşturulabilir. İki adet çizgi geometrisi elemanı arasında kalan alan ile de coğrafi bir alan temsil edilebilir. Üç ve daha fazla çizgi geometrisi ile de kapalı alanlar ve yüzeyler tanımlanabilir.

Çoklu ve kapalı çizgi geometrisi:

Çizgi geometrisi bileşenleri, ardışık olarak bir dizi şeklinde ifade edildikleri zaman çoklu ve kapalı çizgi geometrisi oluşmaktadır (Boissonnat & Teillaud, 2010). İfade edilen bu dizide her çizgisel eleman kendisinden sonra gelen eleman ile ortak bir noktada kesişmektedir. Bu şekilde çoklu çizgi geometrisi oluşmaktadır. Eğer başlangıç ve bitiş noktaları da kesişiyorsa ortaya çıkan çoklu çizgi geometrisi kapalı çizgi geometrisidir ve kapalı bir alanı ifade etmektedir. Çoklu çizgi geometrisinin kullanım alanlarının başında eğrilik derecesi yüksek ve detay belirtmenin gerekli olduğu coğrafi özelliklerin temsil edildiği durumlar gelmektedir.

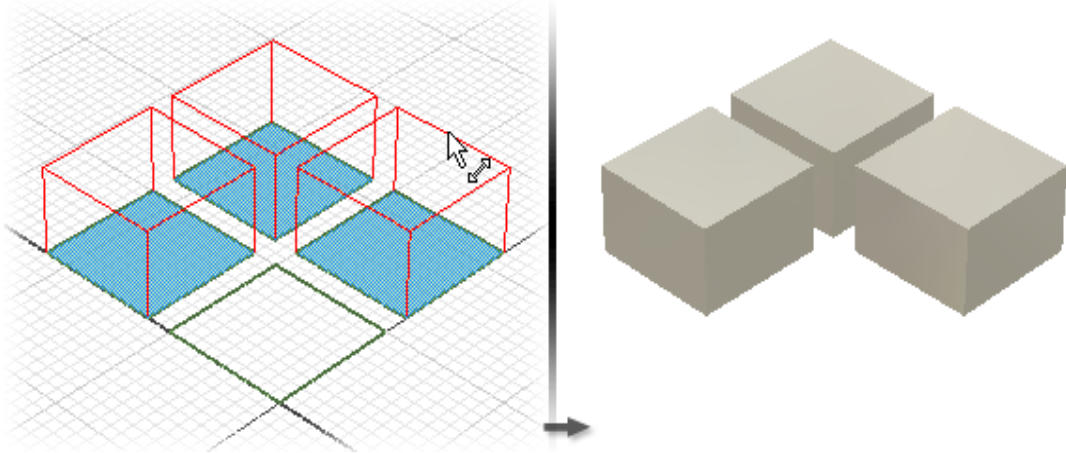
Çoklu çizgi geometrisinin bilgisayar ortamında saklanması gerektiğinde iki farklı yöntem izlenebilir. Birinci yöntem, çoklu çizgi geometrisi oluşturan tüm noktaların bir nokta dizisi halinde saklanması. Diğer yöntem ise geometriyi oluşturan çizgi geometrisi elemanlarının bir dizi şeklinde tutulması. İki yöntem de birbirine benzemekle beraber verinin üzerinde gerçekleştirilecek işlemlerin özelliklerine göre birbirlerine karşı avantaj yaratabilir. Örneğin bir nesne yönelimli programlama uygulamasında nesne sayısı azaltılması gerektiği zaman bileşenleri çizgi geometrisi şeklinde tutmak sayıyı azaltmaya yardımcı olabilir.

3.1.2 İki buçuk ve üç boyutlu mekansal veriler

İki boyutlu mekansal veriler ile tüm coğrafi özellikler temsil edilmek istendiğinde yaşadığımız Dünya ile ilgili tüm mantıksal bağlar kurulamaz. Üç boyutlu Dünyada

yükseklik bilgisinin veriden elde edilecek anlamı değiştireceği durumlarda yükseklik bilgisinin de mekansal veri yapılarına eklenmesi elde edilen anlamı artırır ve gerçek Dünya ile arasında kurulan mantıksal bağları güçlendirir. Bilgisayar ortamında üç boyutlu sayısal modelleme teknikleri geliştikçe, Dünya ile ilgili mekansal verilerin kullanım alanları ve önemi artmıştır (Abdul-Rahman & Pilouk, 2008).

İki buçuk ve üç boyutlu mekansal verinin farkı yükseklik bilgisinin veriye ekleniş biçimi ile değişmektedir. İki buçuk boyutlu mekansal veri, yükseklik bilgisinin tabanda bulunan noktalar temel alınarak eklenmesiyle oluşur. Örneğin bir çatı ile ilgili mekansal veri oluşturulmak istenirse, çatının tüm kesişim noktaları iki boyutlu düzlemde nokta olarak tanımlanır ve bu nokta dizisine bileşenlere özel ya da genel yükseklik bilgisi eklenir. Üç boyutlu mekansal veriler ise yükseklik bilgisini kendilerini oluşturan nokta, çizgi, çoklu çizgi vb. gibi bileşenler ile birlikte içerir. Örneğin x, y ve z koordinatlarından oluşan 8 nokta ya da 12 çizgi (Her biri 2 nokta ile ifade edilen) ile bir küp tanımlanabilir. Şekil 3.4’de sol tarafta bulunan iki boyutlu düzlemlerin (mavi bölgeler) yükseltilecek, sağ tarafta bulunan üç boyutlu geometrilere dönüştürüldüğü gösterilmektedir.

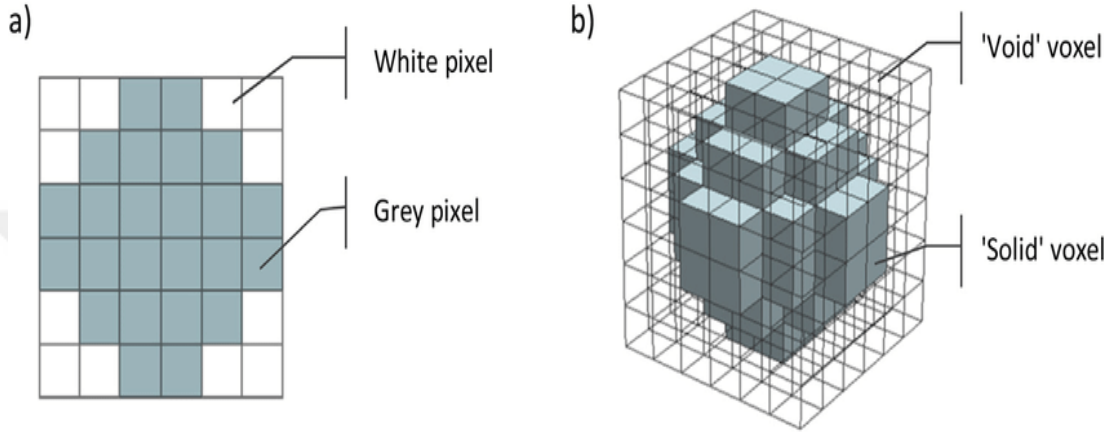


Şekil 3. 4 : İki boyutlu bir geometri ile üç boyutlu gometri oluşturulması (Autodesk çevrim içi dokümanı, 2014).

3.1.2.1 Voxel veri yapısı

Voxel veri yapısı, raster veri yapısı ile benzerlikler göstermektedir. Voxel veri yapısında temsil edilmek istenen 3 boyutlu uzay eşit hacimlere bölünür ve bu bölümler veri yapısının temel elemanlarını oluştururlar. Elemanlar tek başlarına konum bilgisine sahip değildir; sadece temsil ettikleri hacim, boyutları ve temsil ettikleri hacim içerisindeki konumları bilinmektedir. Modellenmek istenen hacim,

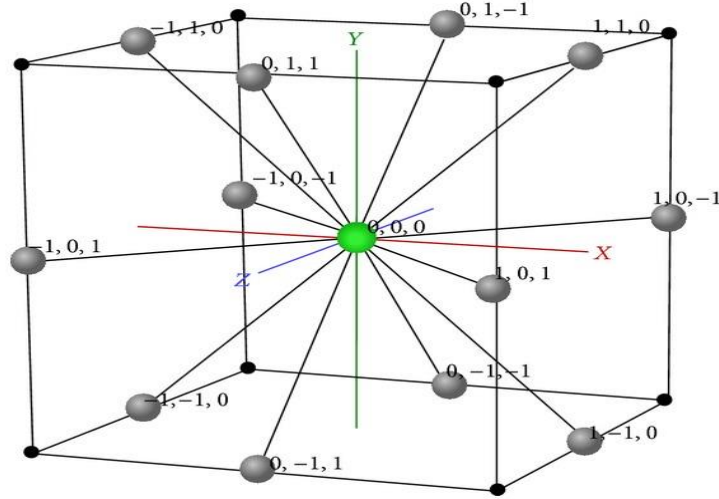
voxellerin birbirlerine göreceli olarak konumlandırılmasıyla oluşturulur. Genellikle boşluk içermeyen hacimlerin modellenmesinde kullanılır. Voxel veri yapısı 1 metreküp ya da 1 milimetre küp hacmi temsil etmek için kullanılabilir. Veri yapısının hacmi küçüldükçe temsil edilmek istenen geometri ile ilgili detaylar daha belirgin olmaktadır ancak görselleştirme performansı ve saklama maliyeti artmaktadır. (Hugues, J. F., Van Dam, A., McGuire, M., Skylar, D. F., Foley, J. D., Feiner, S. K., & Akeley, 2014). Şekil 3.5’de pixel ve voxelin aralarındaki fark gösterilmektedir.



Şekil 3.5 : Voxel ve pixel veri yapılarının karşılaştırılması (Aremu, ve diğerleri, 2017).

3.1.2.2 Üç boyutlu vektör veri yapıları

Üç boyutlu vektör veri yapıları, iki boyutlu vektör veri yapılarının z koordinatı ile zenginleştirilmiş halidir. Temsil edilmek istenen hacim üç boyutlu vektörler yardımıyla bilgisayar ortamında saklanabilir. Örneğin bir küp temsil edilmek istendiğinde, üstünde ve altında bulunan dörder kenar ve yan yüzeylerinde bulunan 4 kenar ile toplam 12 vektör ile temsil edilebilmektedir. Vektör veri yapılarında küp örneği için farklı yaklaşımlar da kullanılmaktadır; aynı noktadan x, y ve z boyutlarına eşit şekilde uzanan üç vektör ile de küp temsil edilebilmektedir. Üç boyutlu vektörlerin temsil yöntemleri bulundukları platforma, teknik kısıtlara ya da modellenmek istenen hacimlere göre farklılık gösterebilmektedir. Üç boyutlu vektör veri yapılarının sunduğu artılardan biri temsil edilmek istenen hacmin dinamik olarak matematiksel işlemlerle ölçeklendirilebilmesidir. Küp örneğine dönülecek olursa, 1 metrelik vektörlerle modellenmiş bir hacim vektör boyutu arttırılarak kolayca ölçeklendirilebilir. Şekil 3.6’da 1 birimlik ölçü ile modellenmiş üç boyutlu vektör veri yapısı gösterilmektedir.



Şekil 3. 6 : Üç boyutlu vektör veri yapısı (Rashid, Newton, Hoque, & Sattar, 2013).

3.1.3 Mekansal veri model ve standartları

Mekansal verilerin üzerinde gerçekleştirilen saklama, iletilme, görselleştirme ya da dönüştürme gibi işlemlerin çeşitliliğinin ve kullanım alanlarının artması standartlaşma ihtiyacını da beraberinde getirmiştir. Standartlaşma yöntemleri; yapılan iş, modellenmek istenen mekansal olgunun tipi, teknolojiyi geliştiren firma ya da bir grup tarafından yönetilebilmektedir. Mekansal veri modelleri, temsil ettikleri mekansal verinin kullanım amacına göre mekansal veri yapılarını kullanırlar ya da standart dışında tutarlar. Örneğin üç boyutlu Dünya üzerinde binalar modellenmek istendiğinde, Dünya tabanı olarak iki boyutlu raster veri yapısı ve modellenen binalar için de üç boyutlu vektör veri yapısını standart haline getirmiş bir veri modeli, voxel veri yapısını standart dışında tutabilir. Mekansal veri modelleri kullanım amaçlarına göre anlamsal verinin içerilmesine izin verdiği gibi sadece coğrafi olguların temsil edilmesine de izin verebilirler.

3.1.3.1 Geography Markup Language (GML)

Geography Markup Language (GML) coğrafi ve anlamsal verilerin standart bir şekilde temsil edilmesi için geliştirilmiş, XML tabanlı bir uygulamadır (Burggraf, 2006). GML, Open Geospatial Consortium (OGC) tarafından geliştirilmiş ve güncellenmekte olan bir uygulamadır. GML coğrafi verilerin modellenmesinde kullanıldığı gibi XML tabanlı olduğu için modellenen verilerin iletiminde ve saklanması da kullanılabilir. XML tabanlı bir uygulama olduğu için

kullanım amacına göre geliştirmeye açık bir uygulamadır, örneğin coğrafi veriler dışında anlamsal verilerin de modellenmesini destekler.

GML ile mekansal veri modellenmek istendiğinde, konsept olarak tasarlanmış temel şekil, ölçü, koordinat sistemi gibi sınıfları içeren temel GML şemaları kullanılmaktadır. Temel şemalar özellik, geometri ve XLink şemaları olarak üçe ayrılmışlardır. Özellik şemasının tasarlanma amacı sayısal olarak modellenmek istenen coğrafi olguların ve anlamsal kavramların isim, tip ve değer bilgilerine sahip özellikler cinsinden ifadesini sağlamaktır. Geometri şemasında temsil edilmek istenen olguların şekilleri ile ilgili temel nokta, çizgi gibi sınıflar tanımlanmıştır. XLink şemasında ise XML tabanlı GML şemalarında elemanlar arasındaki doküman düzeyindeki ilişkiler tanımlanmıştır. Temel şemalarda bulunan konsept sınıflar kullanılarak ve genişletilerek modellenmek istenen coğrafi olgular özelinde uygulama şeması geliştirilmektedir (The Open Geospatial Consortium (OGC), 2007).

3.1.3.2 CityGML

CityGML temelde bir GML uygulama şemasıdır. Üç boyutlu şehir modellerinin ve arazinin saklanması ve iletimi için kullanılan bu GML uygulama şeması Open Geospatial Consortium (OGC) tarafından geliştirilir ve güncellenir. Uygulama şemasının geliştirilme amacı üç boyutlu şehir modellerinde ve arazilerde kullanım sıklıkları yüksek olan nesne, özellik ve ilişkilerin temel olarak tanımlanması ve bu olguların yeniden kullanımını sağlayarak yüksek performanslı şehir modellerinin gerçekleştirilmesidir.

3.1.3.3 GeoJSON

GeoJSON mekansal ve anlamsal verilerin saklanması ve iletimi için geliştirilmiş JSON tabanlı bir metinsel biçimdir (Butler, Gillies, & Schaub, 2016). GeoJSON ile geniş bir alan, tek bir nesne, özellikler ve özellik dizileri temsil edilebilmektedir. GeoJSON nokta, çizgi dizisi, çoklu çizgi, çoklu nokta, çoklu çizgi dizisi ve geometri koleksiyonu gibi mekansal veri modellerini kullanmaktadır. Anlamsal özelliklerin ifadesi için de özellik ve özellik koleksiyonları kullanır. GeoJSON biçimi temelde bir JSON nesnesidir. JSON nesnelerinin GeoJSON olarak özelleştirilmesi ise JSON nesnelerinde zorunlu bir alan olan tip değerinin GeoJSON standardında belirtilen tip sınıfların birisi olarak tanımlanması ile gerçekleştirilebilmektedir.

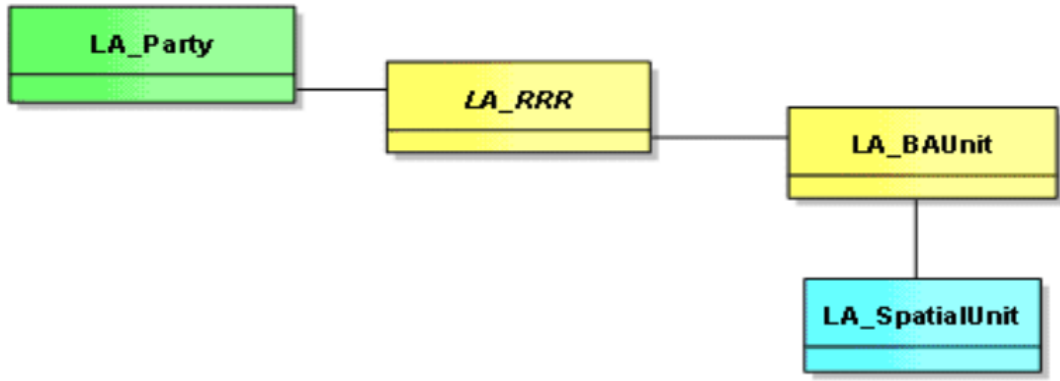
Günümüzde kullanımı artan JavaScript tabanlı teknolojiler tarafından işlenmesi ve iletilmesi XML tabanlı teknolojilere göre daha yüksek performansa sahip olduğu için web ortamına coğrafi veri aktarılmak istendiğinde ilk tercih edilecek biçimlerden birisidir.

3.1.3.4 Arazi yönetimi temel modeli (LADM)

LADM, Uluslararası Standartlar Teşkilatı (ISO) tarafından ISO 19152 kod numarasıyla tescillenmiş bir arazi yönetim modelidir. LADM uluslararası düzeyde beraber çalışabilen ya da iletişime geçebilen sayısal hizmetlerin geliştirilmesinde yol gösterebilmek için geliştirilmiş mantıksal ve soyut şemalar içeren bir modeldir. LADM'nin geliştirilmesi ardında yatan sebep coğrafi verilerin saklanması ve iletilmesi gibi işlemler için standartlar olmasına karşın coğrafi bilgi sistemlerinde kullanılabilecek uluslararası standartlaştırılmış bir arazi yönetim modelinin eksikliğidir (Lemmen C. , 2012). LADM bağlamında arazi, yer altı ya da üstünde bulunan kara ve su dahil tüm mekansal olguları kapsamaktadır.

Önceki bölümlerde bahsedilen mekansal veri modellerinden ayrıldığı en temel noktalar belli bir alan üzerinde özelleştirilmiş olması (özelleştirilmesine rağmen genişletilmesi önünde bir engel bulunmamaktadır) ve soyut bir model olarak tanımlanmasıdır. Soyut bir model olduğu için standartları çerçevesinde diğer modelleri içerek de genişletilebilir. LADM üç temel temel pakette bulunan dört sınıftan oluşmaktadır (Lemmen C. , 2012). Şekil 3.7'de de bu üç temel paketler ve sınıflar bulundukları pakete göre renklendirilerek gösterilmişlerdir. LADM standardı kapsamında bahsi geçen temel paketler ve sınıflar aşağıda özetlenmiştir

- **İştirakler (Party):** Modelleme sürecinde işlemleri gerçekleştiren şahıs, grup ve tarafları temsil eder.
- **Temel yönetim birimleri (Basic administration unit):** Haklar, kısıtlar ve sorumlulukları da içerir (RRR).
- **Mekansal birimler (Spatial unit):** Parsel, bina ve kamusal alanlar gibi mekansal olguların temsili.



Şekil 3. 7 : Temel LADM paketleri (Kalogianni & Dimopoulou, 2014).

3.1.4 Mekansal veri tabanları

Mekansal verilerin saklanması ve sorgulanması konularında özelleştirilmiş ya da ek işlevler ile zenginleştirilmiş veritabanı yönetim sistemleridir. Mekansal veritabanı yönetim sistemleri aynı zamanda veri bütünlüğünün korunmasını ve içerdiği veri üzerinde mekansal analiz yapılabilmesini sağlar (Bertino, ve diğerleri, 1997). Mekansal veriler için zenginleştirilmiş özelliklerin eklenmesinin temel sebebi sorgu ve analiz işlemlerinin etkili bir şekilde gerçekleştirebilme ihtiyacıdır. Basit bir yaklaşım ile mekansal veri geleneksel veri tabanı yönetim sistemlerinde bulunan metin biçimlerinde saklanmak istenirse metin boyutlarının çok büyümesi sebebiyle sorgular, aramalar ve analiz işlemleri uzun zaman harcamalarına rağmen istenen sonuçları veremeyecektir. Mekansal veri tabanlarına ait işlev ve özellikleri sınıflandırılmak istendiğinde dört alt başlık oluşturulabilir (Yeung & Hall, 2007).

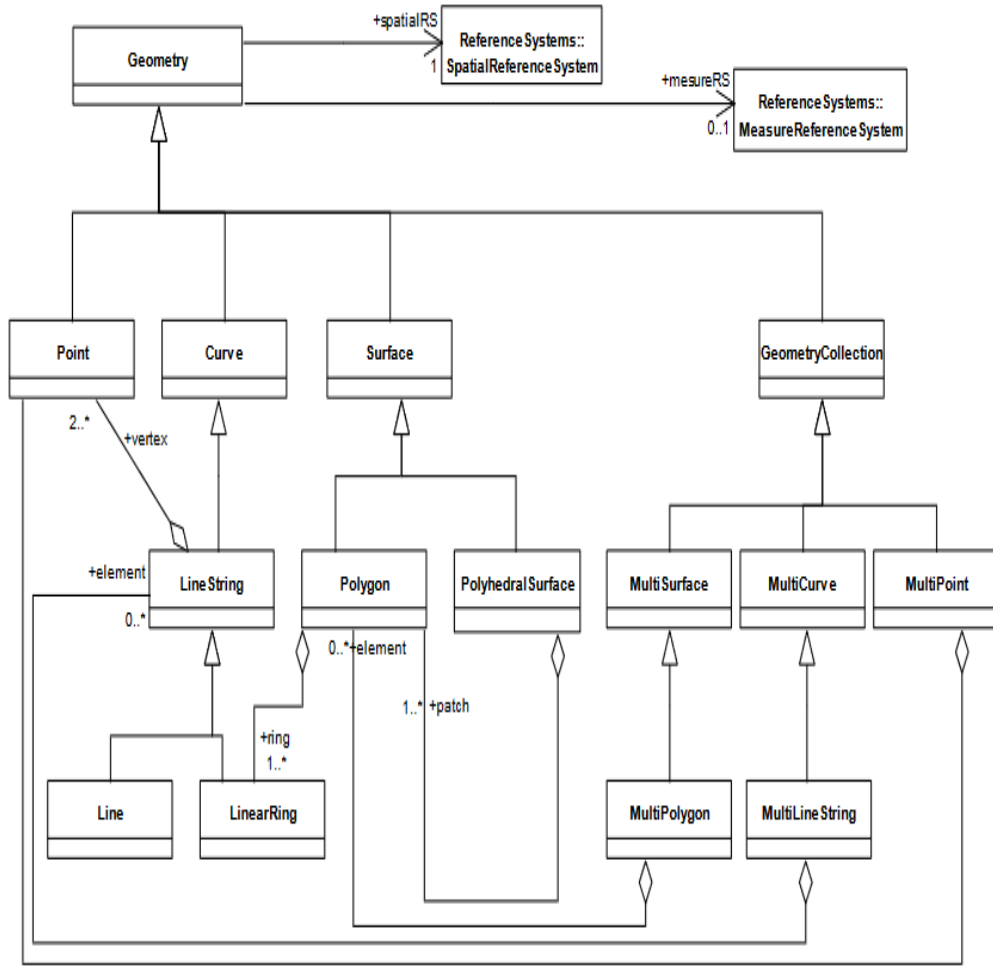
- Mekansal Veri Tabanlarında Kullanılan Veri Tipleri
- Mekansal Verinin Dizinlenmesi
- Mekansal Operatörler ve Sorgular
- Mekansal Uygulama Rutinleri

3.1.4.1 Mekansal veri tabanlarında kullanılan veri tipleri

Mekansal veri tabanlarında kullanılan veri tipleri, veri tabanı yönetim yazılımını üreten firmaların geliştirdikleri ürünlerin kullanım amaçlarına göre çeşitlenmektedir. Bu çeşitlilik veri tabanı yönetim yazılımının kullandığı teknolojilere göre de artmaktadır. Çeşitliliğin artması ile mekansal veri tabanlarında kullanılan veri tiplerinde standartlaşma ihtiyacı oluşmuştur. Standartlaşma ihtiyacını doğuran durumların başında ise mekansal veri ile saklandığı veri tipi arasında bulunan

mantıksal bağı güçsüzleştirme gereksinimi bulunmaktadır. Bu bağ ne kadar kuvvetli ise verinin başka bir veri tabanına aktarımı, başka bir uygulamaya iletilmesi ve üzerinde analizlerin gerçekleştirilmesi o denli zorlaşmaktadır. Genel kabul gören standartlaştırma yaklaşımlarından birisi Open Geospatial Consortium (OGC) tarafından geliştirilmekte olan “Implementation Standard for Geographic Information - Simple feature access“ standardıdır. Bu standart Uluslararası Standartlar Teşkilatı (ISO) tarafından da ISO 19125 kod numarası ile kabul görmektedir. Standart, başka bir ISO standardı olan, SQL Çağrı Seviyesi Arabirimi (SQL / CLI) (ISO - International Organization For Standardization, ISO/IEC 9075-3:2003, 2003) üzerine temellendirilmiştir. ISO 19125, SQL Çağrı Seviyesi Ara birimini kullanarak mekansal özellik koleksiyonlarının depolanmasını, alınması, sorgulanması ve güncellenmesini destekleyen standart bir SQL şeması tanımlamaktır.

Özellik koleksiyonları, mekansal ve özniteliksel (mekansal olmayan, anlamsal) verilerden oluşabilmektedirler. Özniteliksel veriler veri tabanı yönetim sistemi kapsamında bulunan veri tipleri ile kolonlar olarak temsil edilebildikleri için tanımlanan şemayı kullanan sistemin kontrolünde gerçekleştirilebilirler. Örneğin bir tarlada mülkiyet hakkına sahip olan kişi sayısı bu tip bir veri olarak düşünülebilir ve veri tabanı yönetim sisteminin sahip olduğu veri tipleri ile temsil edilebilir. Mekansal verinin saklanması için de geometri tipleri kullanılmaktadır. Standart ile belirlenen geometri tiplerini saklayabilecek SQL veri tipleri üreticiler arasında isim ve gerçekleştirme noktalarında farklılık gösterse de standart şemasına uyduğu sürece benzer özelliklere sahiptirler. Mekansal veriler de veri tabanlarında kolon olarak saklanır ve özniteliksel verilere sahip kolonlar ile beraber kullanılarak tabloların oluşturulmasında kullanılırlar. Tablo ile gerçekleştirilen bir veri modeli hem mekansal hem de özniteliksel veriye sahip olacağı için mekansal verilerin tüm yönleriyle saklanmasında kullanılır. ISO 19125-2 standardı kapsamında bulunan veri tipleri arasında gerçekleşen ilişkilerin gösterimi Şekil 3.8’de gerçekleştirilmiştir. Temel olan “Geometry” veri tipinden türetilen daha kapsamlı veri tiplerinin gösterimi bu şekil kapsamındadır.



Şekil 3. 8 : ISO 19125-2 SQL Tip İlişkileri

3.1.4.2 Mekansal Verinin Dizinlenmesi

Mekansal veri tabanları sadece verinin saklanması işlemini gerçekleştirmezler. Saklanan veriye erişim işleminin de gerçekleştirilmesini sağlarlar. Veri seti büyüdükçe ve erişilmek istenen veri ile ilgili istekler karmaşılaştıkça, istekleri doğru bir şekilde cevaplamanın yanıt süresini ve işlem maliyeti artmaktadır. Artan maliyetlerin azaltılması için mekansal veri tabanlarında bulunan veriler dizinlenebilir. Bu işlem zorunlu olmasa da maliyetleri azaltmak tüm sistemler için yararlı bir durum olduğu için büyük hacimde veriye sahip uygulamalarda bir zorunluluğa dönüşmektedir (Bertino, ve diğerleri, 1997).

Geleneksel veri tabanlarında kullanılan dizinleme yöntemleri mekansal veriler üzerinde uygulandığında verinin hacminin yüksek olması ya da mantıksal yapısının karmaşık olmasından dolayı istenen performans artırımlarını gerçekleştiremeyebilir.

Bu durumu düzeltebilmek adına mekansal verilerin dizinlenmesi için yeni yöntemler ya da var olan yöntemlerin üzerine ek özellikler geliştirilmiştir.

3.1.4.3 Mekansal operatörler ve SQL tabanlı mekansal sorgular

Mekansal veri sorguları geleneksel veri tabanı veri tipleri üzerinde gerçekleştirilen sorgulara göre daha karmaşık sorgulardır; bu durumun nedeni ise iki ya da üç boyutlu veri tipleri üzerinde sorguların gerçekleştirilmesi ve sonuçların görsel olarak sunulmak istenmesidir (Yeung & Hall, 2007). Gerçekleştirilecek olan mekansal sorgularda standartlaşma ihtiyacı oluşmuştur ve bu ihtiyaca da Open Geospatial Consortium (OGC) mekansal operatörleri standartlaştırarak cevap vermiştir. OGC mekansal operatörleri 3 ana başlık altında sınıflandırmıştır (Open Geospatial Consortium, 2010). Bunlar;

❖ Temel Operatörler

- Spatial Reference: Geometrinin referans sistemini belirtir.
- Envelope: Geometri etrafında çizilebilecek en küçük sınırlayıcı dörtgeni belirtir.
- Export: Geometriyi farklı bir biçimde dışa aktarır.
- IsEmpty: Geometrinin boşluk durumunu kontrol eder.
- IsSimple: Geometrinin tipinin basit geometri sınıflarına dahil olma durumunu kontrol eder.
- Boundry: Geometrinin sınırları belirtir.

❖ Topolojik Operatörler

- Equal: Mekansal veri elemanları arasında eşitliği test eder.
- Disjoint: Mekansal veri elemanları arasında ayrıklığı test eder.
- Intersect: Mekansal veri elemanları arasında iki boyutlu düzlemde kesişimi test eder.
- Touch: Mekansal veri elemanlarının birbirine dokunma durumunu test eder.
- Cross: Mekansal veri elemanları arasında üç boyutlu düzlemde kesişimi test eder.
- Within: Mekansal veri elemanlarının birbiri içinde olma durumunu test eder.

- Contain: Mekansal veri elemanlarının birbirini tümüyle kapsama durumunu test eder.
- Overlap: Mekansal veri elemanlarının üst üste olma durumunu test eder.
- Relate: Mekansal veri elemanları arasında ilişki durumunu test eder.

❖ Mekansal Analiz Operatörler

- Distance: Geometri noktaları arasında bulunan en kısa mesafeyi belirler.
- Buffer: Girdi olarak kullanılan geometri ve uzaklık değerlerine istinaden, girdi geometrisinden belirtilen uzaklık kadar olan bölgedeki tüm noktaları içeren bir geometri bilgisini belirtir.
- ConvexHull: Girdi olarak kullanılan geometri seti içindeki tüm elemanları kapsayacak minimum boyuttaki konveks geometriyi belirtir.
- Intersection: İki geometrinin kesişim kümesini ifade eden geometriyi döndürür.
- Union: İki geometrinin birleşim kümesini ifade eden geometriyi döndürür.
- Difference: İki geometrinin fark kümesini ifade eden geometriyi döndürür.
- SymDifference: İki geometrinin simetrik fark kümesini ifade eden geometriyi döndürür.

Mekansal operatörlerin, SQL ile beraber kullanılması ile mekansal veri üzerinde mekansal ve özniteliksel sorguların sağlıklı bir şekilde gerçekleştirilebilmesi sağlanmıştır. Mekansal operatörler standart SQL dilinde bulunan operatörlerin gerçekleştirdiği işlemlerden farklı işlemleri SQL operatörleriyle beraber kullanılarak gerçekleştirebilmektedir. Operatörlerin beraber ve geleneksel veri tabanı yönetim sistemleri üzerinde çalışabilmesi ile üç farklı sorgu biçimi gerçekleştirilebilmektedir; bunlar mekansal, özniteliksel ve karışık sorgulardır (Egenhofer, 1994). Mekansal sorgular sadece mekansal veri tipinde olan veri tabanı kolonları üzerinde, özniteliksel sorgular geleneksel veri tabanı veri tiplerinde olan kolonlar üzerinde ve karışık sorgular da hem mekansal hem de özniteliksel veriye sahip kolonlar üzerinde aynı anda gerçekleştirilen sorgulardır. Gerçekleştirilen sorgular standart SQL söz

diziminde bulunan “SELECT - FROM – WHERE” ifade kalıbına uygun bir şekilde gerçekleştirilebilir. “SELECT “ifadesi ardından seçilecek kolon ya da kolonların ismi gelmektedir. “FROM” ifadesinden sonra üstünde seçim yapılan tablo ya da tabloların ismi gelmektedir. “WHERE” ifadesinden sonra ise sorgu kısıtları ya da yeni bir sorgu başlangıcı gelebilir.

3.1.4.4 Mekansal uygulama rutinleri

Mekansal uygulama rutinleri için geliştirilmiş bir standart bulunamamıştır. Veri tabanı yönetim yazılımını geliştiren firma tarafından eklenen bu özellikler mekansal verinin hayat döngüsünde kritik bir işleve sahip değildir. Bu uygulama rutinlerinin amacı uygulama geliştiricilerinin iş yükünü azaltmak, sık yapılabilecek işlemler için hazır özellikler hazırlamak ve veri tabanı yönetim sistemine piyasa ile rekabet avantajı sağlamak olabilir. Mekansal veriyi toplu olarak farklı biçimlerde dışarı aktaran, farklı biçimlerde saklanan veriyi içeri aktaran, mekansal veri tabanları arasında veri dönüşümü yapan, mekansal veriyi ön bellekte saklayarak performans artıran, mekansal veriyi programlanabilir hale getiren, yedekleme ve acil durum geri dönüş noktaları oluşturan rutinler örnek olarak gösterilebilir.

3.1.4.5 Oracle veritabanı yazılımının mekansal özelliklerinin kullanım örnekleri

Tez kapsamında geliştirilen uygulamalarda Oracle tarafından geliştirilen 11g versiyon numaralı veri tabanı yönetim yazılımı kullanılmıştır. Bu yazılımın kullanılmasının iki temel sebebi vardır. Birinci olarak uygulama katmanında kullanılacak olan Java dili ile aynı firma tarafından geliştirilmiş olmalarıdır. Aynı firma tarafından geliştirilen bu iki teknoloji arasında gerçekleştirilmek istenen iletişimin yine bu firma tarafından geliştirilmiş standartlar çerçevesinde problemsiz bir şekilde gerçekleştirilebileceği öngörülmüştür. İkinci seçim kısıtı ise bahsi geçen teknoloji ve standartları kullanan uygulama geliştirici topluluğunun ve dolayısıyla bilgi havuzunun geniş olmasıdır.

Önceki bölümlerde açıklanan mekansal veri tabanlarında bulunan işlev ve özelliklerin uygulamalarını da belirterek, Oracle veritabanı özelinde kullanımı ile ilgili örnekler aşağıdaki gibidir.

- Oracle veri tabanı yönetim sistemi mekansal verilerin tablo kolonlarında saklanması için veri tipi desteği sunmaktadır ve bu veri tipinin (“SDO_GEOMETRY”) halihazırda kullanımda olan SQL dili ile beraber

kullanılmasına izin vermektedir. Id, isim, su bulunma durumu ve mekansal verinin beraber tutulduğu arazileri belirtmek için kullanılan “BASIC_LAND” isimli bir tablo yaratılmak istenirse aşağıdaki SQL kodu ile yaratılabilir.

```
CREATE TABLE BASIC_LAND (  
    LANDID" NUMBER PRIMARY KEY,  
    NAME VARCHAR2(32),  
    HASWATER NUMBER(1),  
    SHAPE SDO_GEOMETRY);
```

- Yaratılan tabloya iki boyutlu ve beş noktadan oluşan "oracle_first_land" isimli bir çokgen eklenmek istenirse aşağıdaki kod kullanılabilir.

```
INSERT INTO BASIC_LAND VALUES(  
    1, 'oracle_first_land', 1,  
    SDO_GEOMETRY(  
        2003,--İki boyutlu çokgeni belirten sabit  
        NULL, NULL,  
        SDO_ELEM_INFO_ARRAY(1,1003,1),--Çokgen ile ilgili tanımsabiti  
        SDO_ORDINATE_ARRAY(5,1, 8,1, 8,6, 5,7, 5,1)--Çokgenin noktaları ));
```

- Yaratılan tablonun mekansal veri içeren kolonu Oracle tarafından sunulan temel algoritma ile dizinlenmek istenirse aşağıdaki uygulama rutini ile gerçekleştirilebilir.

```
CREATE INDEX index_basic_land ON BASIC_LAND (SHAPE) INDEXTYPE IS  
MDSYS.SPATIAL_INDEX;
```

- Tabloda bulunan mekansal ve mekansal olmayan kolonlar üzerinde sorgu işlemleri aşağıdaki şekilde yapılabilir. Gerçekleştirilen bu sorgular birden fazla alan için de yapılabilir.

```
SELECT BASIC_LAND.SHAPE FROM BASIC_LAND WHERE BASIC_LAND.LANDID=1;  
SELECT BASIC_LAND.HASWATER FROM BASIC_LAND WHERE BASIC_LAND.LANDID=1;
```

- Arazilerin tutulduğu tablo ile ilişkili arazi sahiplerinin tutulduğu mekansal olmayan veri içeren “LAND_OWNER” tablosu ve ilişki aşağıdaki gibi tanımlanabilir. Bu tabloda id, isim, soy isim, Türkiye Cumhuriyeti Kimlik Numarası, yaş ve arazi tablosu ile ilişkiyi belirten alan bilgileri bulunmaktadır.

```
CREATE TABLE LAND_OWNER (  
    OWNERID NUMBER PRIMARY KEY,  
    BASICLANDREF NUMBER(27) NOT NULL ,  
    NAME VARCHAR2(32),  
    SURNAME VARCHAR2(32),  
    TCKN VARCHAR2(11),  
    AGE NUMBER(3));
```

```
ALTER TABLE LAND_OWNER
```

```
ADD CONSTRAINT owner_land_FK FOREIGN KEY (BASICLANDREF) REFERENCES  
BASIC_LAND(LANDID);
```

- Arazi sahibi tablosuna, önceden eklenen arazi ile ilişkili bir kayıt aşağıdaki gibi eklenebilir.

```
INSERT INTO LAND_OWNER VALUES(1, 1, 'Murat', 'Sümbül', '43489345678', 28);
```

- Arazi sahibinin bilgileri ile arazi ile ilgili mekansal, mekansal olmayan ve karışık sorgular gerçekleştirilebilir.

```
SELECT SHAPE FROM BASIC_LAND WHERE LANDID IN (SELECT BASICLANDREF FROM  
LAND_OWNER WHERE NAME LIKE 'Murat');
```

```
SELECT HASWATER FROM BASIC_LAND WHERE LANDID IN (SELECT BASICLANDREF FROM  
LAND_OWNER WHERE TCKN LIKE '43489345678');
```

```
SELECT SHAPE, HASWATER FROM BASIC_LAND WHERE LANDID IN (SELECT BASICLANDREF  
FROM LAND_OWNER WHERE NAME LIKE 'Murat' AND TCKN LIKE '43489345678');
```

- Topolojik ve mekansal analiz operatörleri aracılığıyla mekansal veri üzerinde işlemler gerçekleştirilebilir. Bu işlemler kabul ettikleri parametrelere göre bir ya da iki mekansal veri üzerinde gerçekleştirilebilir.

```
-- Yeni bir şekil ekle
```

```
INSERT INTO BASIC_LAND VALUES(
```

```
2, 'oracle_second_land', 1,
```

```
SDO_GEOMETRY(
```

```
2003,--İki boyutlu çokgeni belirten sabit
```

```
NULL, NULL,
```

```
SDO_ELEM_INFO_ARRAY(1,1003,1),--Çokgen ile ilgili tanım sabitleri
```

```
SDO_ORDINATE_ARRAY(3,3, 6,3, 6,5, 4,5, 3,3)--Çokgenin noktaları ));
```

```
-- INTERSECT (Mekansal Analiz Operatörü)
```

```
SELECT SDO_GEOM.SDO_INTERSECTION(land1.SHAPE, land2.SHAPE, 0.005)
```

```
FROM BASIC_LAND land1, BASIC_LAND land2
```

```
WHERE land1.NAME='oracle_first_land' AND land2.NAME='oracle_second_land';
```

3.2 Uygulama Katmanı Teknolojileri

Sunucu tarafında gerçekleştirilen tüm işlemleri içeren mantıksal katmandır. Veri tabanı ile iletişim kurmak, veri tabanı tablolarını Java sınıfları olarak modellemek, veri tabanından veriyi Java sınıfları olarak elde etmek, elde edilen veriyi işlemek, işlenen veriyi dışarıya aktarmak ve istemci tarafından iletilen isteklerin cevaplanması işlemlerini gerçekleştirir. Uygulama katmanı kendi içinde de yapılan işlerin çeşidine göre mantıksal ve fiziksel olarak katmanlanabilir (Brown, Niswonger, & Hester, 2001) (Fowler, 2002).

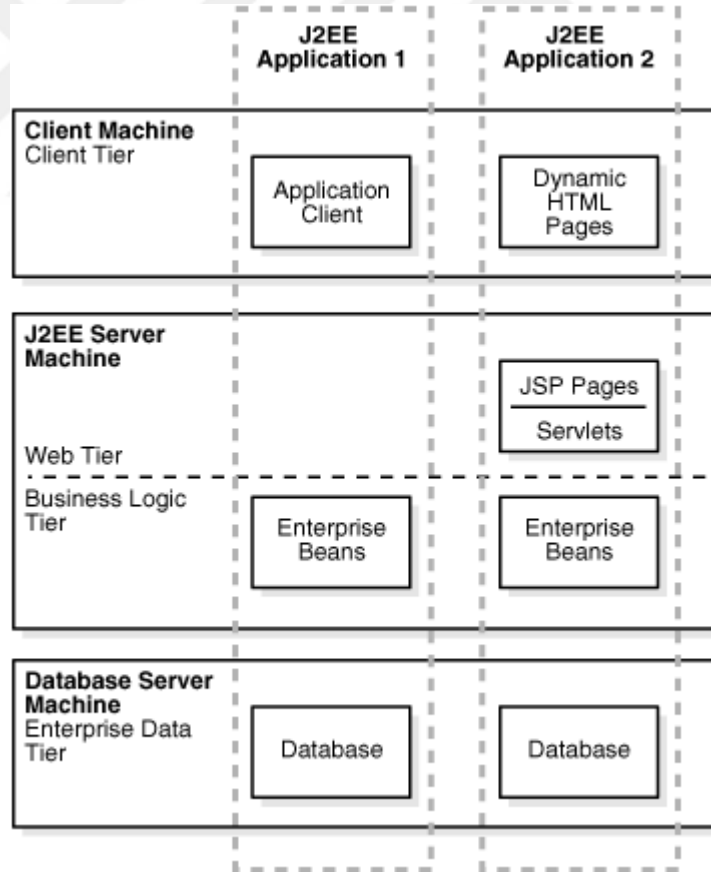
3.2.1 Java 2 Enterprise Edition

Java 2 Enterprise Edition ya da J2EE birden fazla katmana sahip kurumsal uygulamaların geliştirilmesi için tasarlanmış Java programlama dili tabanlı bir spesifikasyonlar bütünüdür (McGovern, Adatia, Fain, Gordon, & Henry, 2003). Standart Java programlama dilinin sunduğu yetenekler ile basit ve bir kaç işi yerine getirebilen uygulamalar kodlanabilirken; web siteleri, web servisleri ya da sunucu bileşenleri gibi daha karmaşık işleri yerine getirebilecek uygulamaların geliştirilebilmesi spesifikasyonlara ihtiyaç duyulmuştur.

Belirlenen spesifikasyonların temel amacı, kurumsal firmaların ürün geliştirme süreçlerinde ihtiyaç duydukları temel iş birimlerinin genel kabul görmüş tasarım örüntülerine uygun bir biçimde standartlaştırılmasıdır. Amaçlanan diğer bir fayda ise gerçekleştirilen iş birimlerini modüler bir hale getirerek, yazılım bileşenleri arasında bulunan bağımlılığı azaltmaktır. Modüller arasındaki iletişim uygulama programlama arayüzleri aracılığıyla gerçekleştirilir ve bu durum bağımlılığın azalmasında büyük rol oynar. Örneğin; spesifikasyonlar belirlenmeden önce veri tabanı ile beraber çalışması gereken uygulamalarda; bağlantı ve iletişim, uygulamayı geliştiren ekip tarafından programlanmaktaydı. Bu durum mimari, tasarımsal ve programatik hataların daha sık oluşmasına sebebiyet vermekteydi. Spesifikasyonlar belirlendikten sonra ise uygulamayı geliştiren ekipler belirlenen uygulama programlama arayüzlerini kullanarak hem daha az emek ve zaman harcayarak hem de standartlara uyarak istenen işleri yerine getirebilmektedirler.

Geliştirilen spesifikasyonlar güncel yazılım trendlerine, tasarım örüntülerine, mimarilerine ve ihtiyaçlarına göre sürekli güncellenmektedirler. Spesifikasyonları belirleyen grup dışında, üçüncü kişiler tarafından geliştirilen teknolojilerin de

spesifikasyonlara dahil edilmesi bu güncellemeleri daha güçlü kılmaktadır. Bu güncellemeler geçmiş özellikleri kapsayacak ya da yok saymayacak şekilde tasarlandıkları için spesifikasyonlara uyan yazılımlara, gelişimlere açıklık ve sürdürülebilirlik gibi özellikler katmaktadır. Şekil 3.9’da iki farklı J2EE uygulamasının mimarisi belirtilmiştir. Belirtilen bu mimarilerde yazılım bileşenlerinin bulunduğu fiziksel konumlar (sunucu makine, istemci makine, veri tabanı sunucu makinesi vs.) ve katmanlar (istemci, web, uygulama mantığı vs.) belirtilmişlerdir. Tez kapsamında geliştirilen uygulamalarda kullanılan mimari, şekilde gösterilen ikinci uygulamanın mimarisi ile benzer özelliklere sahiptir. Tez kapsamında böyle bir mimarinin kullanılmasının altında yatan motivasyon ise iş birimlerini olabildiğince bölerek sadece tek işi gerçekleştirebilen yazılım bileşenlerini, gerçekleştirdikleri iş ile ilgili yerlerde bulunan paketlerde konumlamaktır. Böyle bir yaklaşım ile yazılım bileşenleri arasında bulunan bağımlılık olabildiğince az bir seviyede tutulmuş olacaktır.



Şekil 3. 9 : J2EE Çok katmanlı mimarisi (Oracle çevrim içi dokümanı. J2EE, Web Services, and Internet Applications, 2013).

3.2.2 Spring Framework

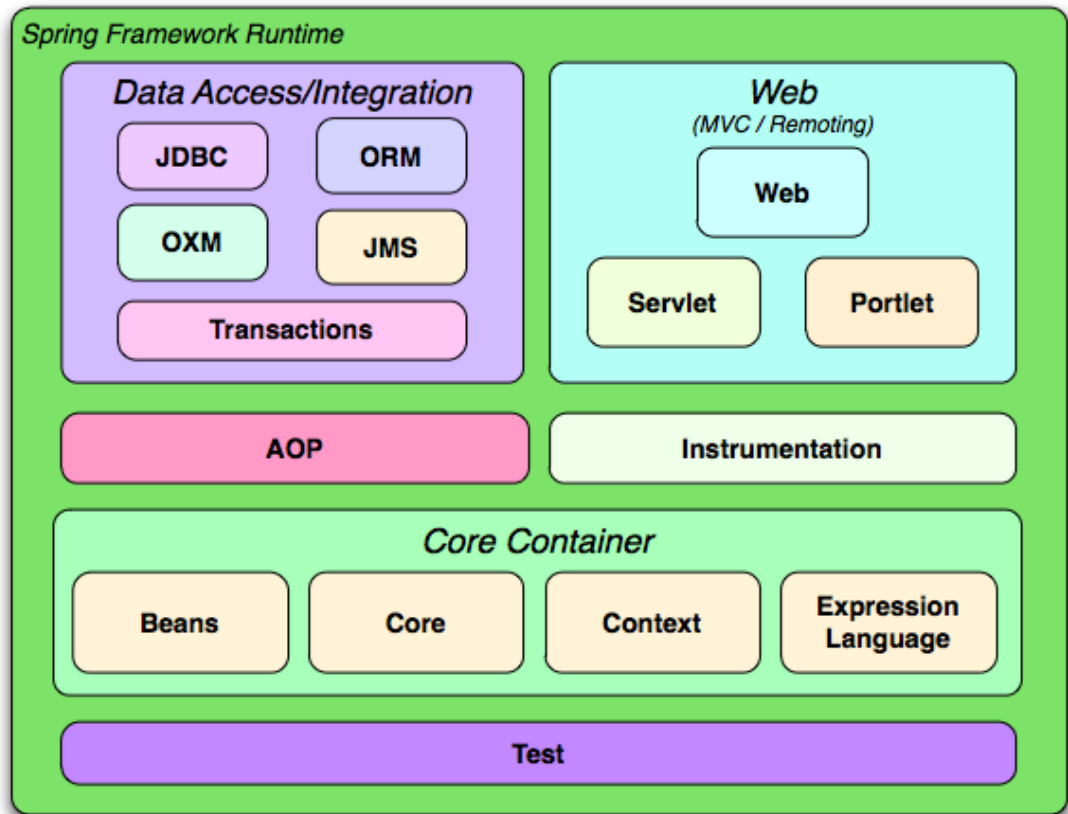
Spring Framework Java platformu için açık kaynak olarak geliştirilmiş bir uygulama çatısıdır. Spring Framework Java uygulamalarının geliştirilmesi için geniş kapsamlı yazılımsal ve tasarımsal alt yapı desteği sağlamaktadır. Sağlanan bu alt yapı desteğinin amacı uygulama geliştiricilerinin alt yapıdan ziyade geliştirilen uygulamaya yoğunlaşmalarını ve daha az efor ile istenen işleri gerçekleştirmelerini sağlamaktır. Spring Framework çok katmanlı mimariye sahip uygulamaların her katmanı için kısmen ya da tamamen alt yapı ve tasarımsal destek verdiği için yaygın kullanıma sahiptir. Verilen alt yapı desteğinin konfigürasyonu XML tabanlı dosyalar ve Java açıklamaları ile proje bazında gerçekleştirilebilir. İki farklı konfigürasyon tipi beraber de kullanılabilir.

Spring Framework'ün sağladığı en önemli özelliklerden biri “Kontrolün Ters Çevrilmesi” (Inversion of Control - IoC) tasarım prensibinin uygulanmasını sağlayan bir konteynere sahip olmasıdır. Prensibin temel amacı beraber çalışması beklenen yazılım bileşenlerinin arasındaki bağımlılığı en aza indirmek, bileşenlerin yaşam döngülerini yönetmek ve bileşenlerin tek başlarına genişletilmelerinin önünü açmaktır. Spring Framework içerdiği yazılım bileşenlerini sakladığı konteyner aracılığıyla, bileşeni kullanmak isteyen diğer bileşenin yaratılma ya da tetiklenme anında bileşenler arasında ilişki kurarak bu tasarım prensibini gerçekleştirir.

Spring Framework modüler bir yapıya sahiptir ve içerdiği modüller tek ya da toplu olarak uygulama geliştirmek için kullanılabilir. Şekil 3.10'da Spring Framework kapsamında bulunan modüller gösterilmiştir. Spring Framework'ün içerdiği yaklaşık 20 modül de kendi içlerinde verdikleri hizmetlere göre gruplanmıştır. Bu gruplar;

- Çekirdek Konteyner: IoC Konteyneri, Bağımlılık Enjeksiyonu (Dependency Injection, DI), JavaBean Fabrikası, bileşenlere erişim için kullanılan bağlam ve çalışma zamanında bileşenlerin erişilmesi ve değiştirilmesini sağlayan modülleri içerir.
- Veri Erişimi ve Entegrasyonu: Veri tabanı ile iletişim kurulması, veri tabanı iletişim süreçlerinin yönetilmesi, nesneler ve veri tabanı tabloları arasında ilişki kurulması, nesneler ve XML şemaları arasında ilişki kurulması, Java Messaging Service (JMS) tabanlı mesajların oluşturulması ve işlenmesi gibi işlemleri gerçekleştiren modülleri içerir

- Web: JavaServlet, JavaPortlet, web uygulama bağlamı, yaşam döngüsü yönetimi, genel kabul görmüş web tabanlı tasarım örüntülerinin uygulamaları gibi modülleri içerir.
- Test: Bileşenlerin beraber ya da ayrı olarak test edilmesini sağlayan modülleri içerir.
- Cehpe Yönelimli Programlama: Yazılmış kodların yapısı değiştirilmeden yeni özelliklerin eklenmesi ve özellikleri genelleştirilmesini kolaylaştıran modülleri içerir.
- Enstrümantasyon: Bileşenlerin kullanıldıkları ortamlara göre yüklenmesini ve yönetilmesini sağlayan modülleri içerirler.



Şekil 3. 10 : Spring Modülleri (Spring çevrim içi dokümanı, 2004).

3.2.3 Apache Maven

Apache Maven, kaynak kodlardan sunucuya yüklenebilen paket oluşturmak, uygulamada kullanılan üçüncü parti özelliklere olan bağımlılıkları yönetmek gibi farklı işleri yapabilen bir proje yönetim otomasyonudur (O'Brien, ve diğerleri, 2008). Üçüncü parti bağımlılıkların isimlerini, versiyon numaralarını, kaynak konumlarını ve nasıl kullanılacaklarını tanımlamaya yarayan XML tabanlı bir konfigürasyon

dosyasına sahip olan Apache Maven; HTTP üzerinden bu bağımlılıkları kod geliştirme ortamına tanımlar. Kod geliştirme ortamına aktarılan üçüncü parti bağımlılıklar kaynak kodların derlenmesinde kullanılır. Ayrıca kaynak kodlardan sunucuya yüklenecek paket oluşturulmak istendiğinde, üçüncü parti bağımlılıkların pakete dahil edilmesi işlemini de gerçekleştirmektedir. Oluşturulan bu pakette üçüncü parti bağımlılıkların kaynak kodlar tarafından nasıl kullanılacağı bilgisi de bulunmaktadır.

Üçüncü parti bağımlılıklarının her dosya bazında ayrı ayrı yönetilmesi hataya açık bir süreç olduğu için Apache Maven tek bir konumdan yönetilen XML tabanlı konfigürasyon dosyası ile bu süreçte hata yapılma şansını azaltmaktadır. Ayrıca bağımlılıklar için versiyon güncellemelerinde dinamik bir şekilde süreci yönettiği için kullanıcılarına kolaylık sağlamaktadır. Grafik tabanlı kod geliştirme arayüzlerine entegre olan eklentileri ile de gerçekleştirilmek istenen konfigürasyonlar kolay bir şekilde uygulanabilmektedir.

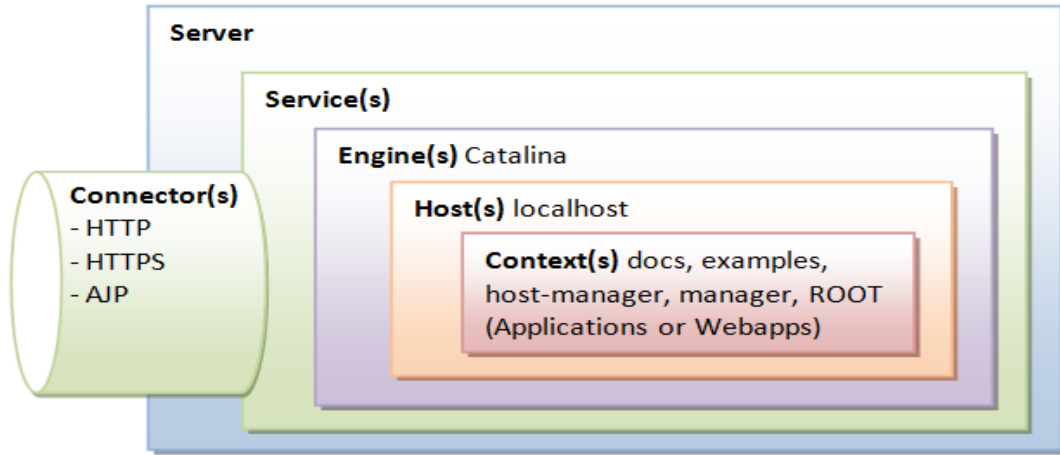
Tez kapsamında geliştirilen modüllerde Apache Maven, üçüncü parti bağımlılıkların yönetimi ve sunuculara yüklenecek kaynak kodların bağımlılıklar ile beraber paketlenmesi için kullanılmıştır.

Spring Framework, Oracle Java Database Connection (OJDBC) gibi bağımlılıklar XML tabanlı bir dosya ile yönetilmiştir. Ayrıca kaynak kodları; Web Archive dosyası olarak .war uzantılı bir şekilde sunucuya yüklenebilecek dosya biçimine getirilmiştir.

3.2.4 Apache Tomcat

Apache Tomcat; açık kaynak kodlu, Java tabanlı ve Java Servlet, Java Server Pages(JSP) ve Java WebSocket gibi J2EE spesifikasyonlarını implemente eden bir web uygulama konteyneridir (Vukotic & Goodwill, 2011). J2EE spesifikasyonlarını temel alan bir uygulama konteyneri olduğu ve muadillerine göre daha az dosya boyutuna sahip olduğu için kullanımı ile sık karşılaşılr. Gerçekleştirdiği temel işlem JSP ile kodlanmış web sayfalarını Java Servlet spesifikasyonlarına bağlı kalarak web ortamında istemcilerin erişimine sunabilmektir. Tez kapsamında geliştirilen web tabanlı yazılımlar .war dosya biçimine getirildikten sonra Apache Tomcat yardımı ile web üzerinden ulaşımına sunulmuştur. Geliştirilen iki farklı uygulama için de aynı özelliklere sahip farklı konumlarda bulunan web uygulama konteynerleri

kullanılmıştır. Şekil 3.11’de tez kapsamında kullanılan Apache Tomcat web uygulama konteynerinin mimarisi belirtilmiştir.



Şekil 3. 11 : Apache Tomcat mimarisi (Chuan, 2011).

3.2.5 JavaScript Object Notation (JSON)

Sunucu – istemci mimarisine sahip web uygulamalarında; sunucuya iletilen isteklere göre, sunucu işlemleri gerçekleştirir ve istemciye cevap olarak işlem sonucunu gönderir. Sunucuların cevaplarının yapısı uygulama mimarisine, teknik kısıtlara ya da performans ihtiyaçlarına göre farklılık gösterebilirler. Günümüzde istemci olarak genellikle web tarayıcıları kullanılmaktadır. Web tarayıcıları üzerinde işlem yapabilme kabiliyeti teknik olarak arttıkça, istemciye cevap olarak gönderilen bilgilerin içeriği de değişmektedir. Web teknolojilerinin ilk zamanlarında doğrudan tüm sayfanın bilgisi gönderilirken, günümüzde sayfanın bir bölümü ya da sayfada gösterilecek verinin içeriği gönderilmektedir. Gönderilen cevabın tipine göre tarayıcı üzerinde işlemler gerçekleştirilerek cevabın gösterimi sağlanmaktadır.

JavaScript Object Notation (JSON) yapısal verinin iletimi için kullanılan küçük boyutlu, insan tarafından okunabilen, programlama dillerinden bağımsız bir metin biçimidir (Crockford, 2006). JSON metin biçimi JavaScript standartlarını temel olarak oluşturulduğu için XML gibi benzer amaç için kullanılan teknolojiler ile işleme hızı açısından kıyaslandığında avantaj sağlamaktadır (Nurseitov, Paulson, Reynolds, & Izurieta, 2009). Tez kapsamında geliştirilen uygulamanın sunum katmanında kullanılan teknolojiler de JavaScript temelli oldukları için veri iletiminde JSON metin biçimi tercih edilmiştir. Diğer bir tercih sebebi de gerçek zamanlı olan uygulama sunum katmanında, sunucu tarafından iletilen cevabın hızlı bir şekilde işlenip sunulabilmesini sağlayabilmektir.

3.3 Sunum Katmanı Teknolojileri

İstemci olarak kullanılan web tarayıcısı üzerinde veri işleme, veri görselleştirme ve kullanıcı ile etkileşimi yönetme gibi işlemleri gerçekleştiren teknolojilerdir. Tez çalışması kapsamında gerçekleştirilen uygulamanın sunum katmanında JavaScript temelli kütüphaneler kullanılmıştır. JavaScript temelli kütüphanelerin kullanılmasının başlıca sebepleri tüm web tarayıcıları tarafından desteklenmeleri ve geniş bir geliştirici ağına sahip olmalarıdır.

Geliştirilen uygulama kapsamında sunum katmanı, kullanıcı ile gerçek zamanlı etkileşime geçtiği için görselleştirme performansının zaman boyutunda artırılması hedeflenmiştir. Hedefe ulaşmak amacıyla sunucu tarafından dönen cevap salt veri formatındadır ve istemci tarafında işlenerek görselleştirilir.

3.3.1 WebGL

WebGL, kâr amacı gütmeyen teknoloji şirketler birliği Khronos Group tarafından yönetilen, web ortamında 3 boyutlu grafik çizimi sağlanmasını amaçlayan bir web uygulama programlama arayüzüdür. OpenGL ES 2.0 üzerine temellendirilmiştir, otomatik hafıza yönetimli JavaScript gibi dillere uyum sağlaması için küçük değişikliklere uğramış olsa da WebGL OpenGL ES'ten çok da farklı değildir (Jackson & Gilbert, 2019). Uyumlu web tarayıcısında herhangi bir eklenti kurulmasını gerektirmeden, HTML5'in "Canvas" elementi üzerinde 3 boyutlu grafik çizimi yapılmasını mümkün kılmaktadır.

Microsoft (Internet Explorer), Mozilla (Firefox), Apple (Safari), Google (Chrome) ve Opera (Opera) gibi web tarayıcısı geliştiren şirketler Khronos WebGL çalışma grubunun üyeleridirler. Genel olarak masaüstü bilgisayarlarda çalışan tarayıcı versiyonları tarafından destekleniyor olsa da mobil platformlarda çalışan tarayıcılar tarafından desteği de artmaktadır.

3.3.2 JQuery

JQuery tanımlandığı web sayfası üzerinde bulunan elemanları programatik olarak değiştirmek, web sayfalarının yaşam döngülerine müdahale etmek, web üzerinde animasyonlar gerçekleştirmek, kullanıcı girdilerini dinlemek, asenkron istek üretmek ve tarayıcı tarafından oluşturulan Domain Object Model (DOM) ile uygulama

programlama arayüzü aracılığıyla iletişime geçmek için kullanılan bir JavaScript kütüphanesidir (jQuery Foundation, 2019).

Web tarayıcıları sunucudan gelen HTML cevapları doğrudan tarayarak görselleştirmezler. Gelen cevaplar ağaç benzeri bir veri modelinde saklanır ve ardından görselleştirme işlemi gerçekleştirilir. Ağaç benzeri bu veri modeli Domain Object Model olarak isimlendirilmiştir. JQuery'nin genel olarak Kabul görmesinin ve kullanılmasının temel sebeplerinden birisi DOM üzerinde gerçekleştirilmek istenen işlemlerin kolaylıkla ve tarayıcıdan bağımsız bir şekilde gerçekleştirilebilmesini sağlamasıdır. JQuery'nin JavaScript yazımına benzer bir yazım biçimi vardır ancak söz dizimi tamamen aynı değildir.

3.3.3 ThreeJS

ThreeJS, WebGL uygulama programlama arayüzü teknolojisini kullanan bir JavaScript kütüphanesidir. Geliştirilmesindeki temel amaç WebGL teknolojisinin tarayıcı bağımsız bir şekilde kullanılmasını kolaylaştırmaktır. Diğer bir katkısı da WebGL teknolojisini doğrudan kullanarak gerçekleştirilmek istendiğinde çok fazla yazılımsal efor gerektiren işlemlerin standartlaştırarak programcı için kolay hale getirilmesidir (Dirksen, 2014). ThreeJS kullanılarak WebGL teknolojisinin gerçekleştirebildiği işlemlerin neredeyse tamamı, matematiksel işlemlere çok fazla ihtiyaç duymadan nesneler ve davranışları aracılığıyla kodlanabilmektedir. Getirdiği faydaların yanı sıra küçük dosya boyutu da kullanımının yaygın olmasındaki nedenlerden birisidir.

3.3.4 CesiumJS

CesiumJS, WebGL teknolojisini kullanan ve Dünyanın web ortamında üç boyutlu şekilde görselleştirmesini sağlayan açık kaynak kodlu bir JavaScript kütüphanesidir. Görselleştirilen Dünya sanal küre olarak da isimlendirilebilmektedir. CesiumJS tarafından görselleştirilen Dünya küresi üzerinde temel geometrik şekillerin çizilmesini sağlayan metotlara da sahiptir. Bu metotlar ile Dünya küresi üzerine yapışık ya da yükseklik özelliğine sahip şekiller görselleştirilebilmektedir. Görselleştirme işleminde paralel programlama kullanımına izin verdiği için de gerçekleştirilen işlemin zaman olarak performansını arttırmaktadır. Görselleştirme için kullandığı yaklaşım nesne tabanlıdır. Dünya üzerine çizilen her olgu bir nesnedir ve programatik olarak erişilerek üzerinde istenilen işlemler

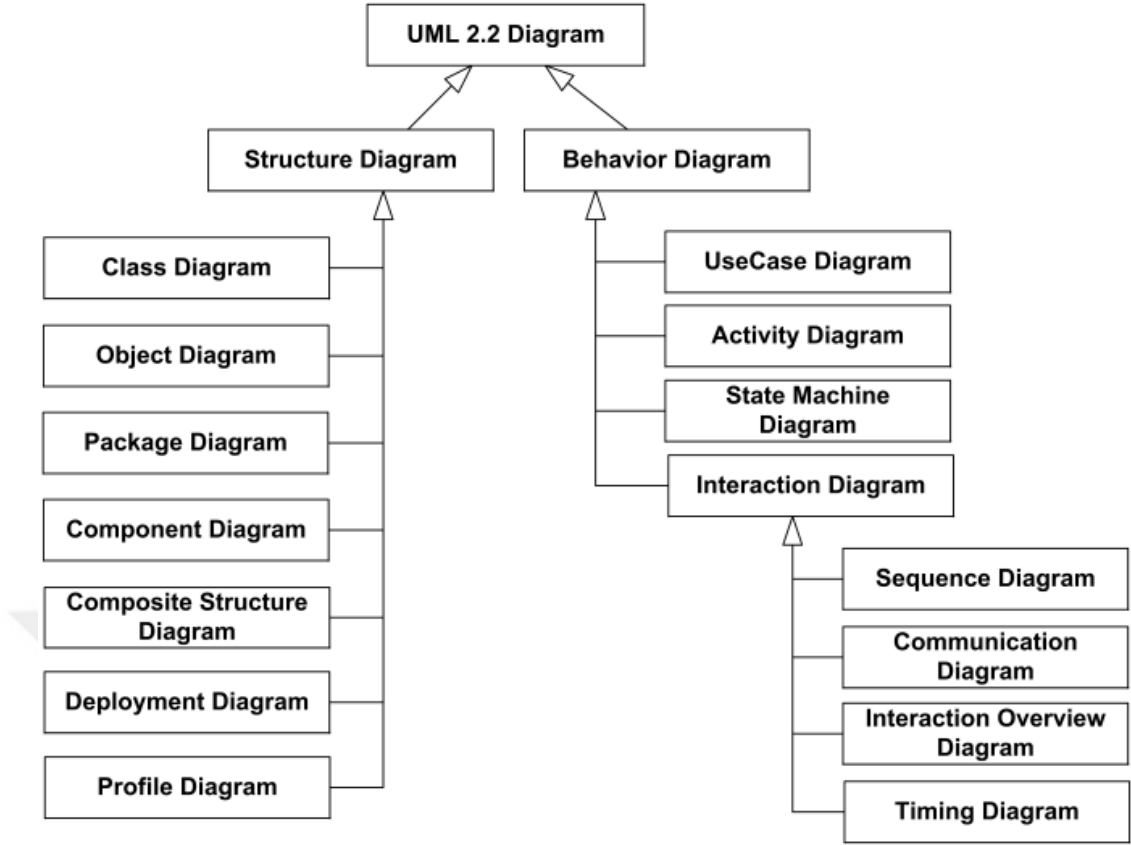
gerçekleştirilebilmektedir. Şekil 3.12’de web ortamında bulunan CesiumJS tabanlı örnek bir sanal küre gösterilmektedir.



Şekil 3. 12 : CesiumJS tabanlı sanal küre

4. YÖNTEM

LADM standardının web ortamında üç boyutlu olarak görselleştirilebilmesi için geliştirilen sistem kapsamında iki temel yazılım modülü tasarlanmıştır. Tasarlanan bu modüller kullanım amaçlarına göre açıklanacak olursa; temel alınacak olan LADM standartlarına uygun veriyi oluşturup veri tabanına ya da dosyaya aktaran modül ve LADM standardındaki veriyi web tabanlı kullanıcı arayüzüne kullanıcı sorgularına göre servis ederek web arayüzünde veriyi işleyip görselleştiren modül olarak tanımlanabilirler. Yazılım modüllerinin geliştirilmesi üç ana süreç kapsamında tamamlanmıştır. Bu süreçler işlev çözümlenmesi, mimari tasarım ve fiziksel tasarım süreçleri olarak ayrılabilir. Süreçler boyunca Unified Modeling Language (UML) standartlarına uygun şema modellemeleri gerçekleştirilmiştir ve nesne yönelimli yazılım mühendisliği yaklaşımlarına bağlı kalmıştır. Gerçekleştirilen süreçlerde faydalanan UML şemalarının sınıflandırılması Şekil 4.1’de gösterilmiştir. Tez kapsamında gerçekleştirilen çalışmalarda yazılım bileşenlerini ve süreçlerini modelleyebilmek amacıyla hem yapısal hem de davranışsal UML şemalarından faydalanılmıştır



Şekil 4. 1 : UML şema tipleri (Fakhroutdinov, 2018).

İşlev çözümlenmesi sürecinde kullanım durumlarından faydalanılmıştır. Kullanım durumları sistem dışı bir kullanıcının tasarlanmak istenen sistem ile gerçekleştirdiği etkileşimlerin ve sistem çıktılarının belirtildiği bir yazılım mühendisliği tekniğidir. Kullanım durumlarından faydalanılmasındaki amaç sistemin gerçekleştirmeyi hedeflediği davranışı bütün olarak, iç işleyişini açığa vurmadan tanımlamaktır (Rumbaugh, Jacobson, & Booch, 2005). Kullanım durumları ışığında davranışsal UML şemalarından olan kullanım durum şemaları da oluşturulmuştur.

İşlev çözümlemesi sürecinde belirlenen sistem davranışı mimari tasarım sürecinde alt işlevlere bölünmüştür. Sistem kapsamında gerçekleştirilen işlevler tiplerine göre paketlere ayrılmışlardır ve yapısal UML şemalarından olan paket şemaları ile temsil edilmişlerdir. Paket şemaları, sistemin mantıksal olarak paketlere ayrılması ve bu paketler arasında olan bağımlılıkların modellenmesi sağlamaktadır. Paketler MVC tasarım örüntüsünde bulundukları katman göz önüne alınarak belirlenmişlerdir ve şemalarda konumlandırılmışlardır.

Fiziksel tasarım sürecinde önceden belirlenen işlevleri gerçekleştiren sınıflar, servisler ve yordamlar kodlanmıştır. Kodlama işlemiyle beraber bu bileşenler

arasında entegrasyon çalışmaları da gerçekleştirilmiştir. Bileşenler arasında olan bağıın azaltılması amacıyla, etkileşimler arayüzler aracılığıyla gerçekleştirilmiştir. Yazılım bileşenleri gerçekleştirdikleri işler temelinde gruplanarak paket şemaları oluşturulmuştur. Sistem bileşenlerinin sahip oldukları özellikler ve davranışların, bileşenler arasında ilişki ile beraber gösterildiği yapısal UML şemalarından olan sınıf şemaları modellenmiştir. Sınıflar temel alınarak yaratılan nesnelerin arasında gerçekleştirilen iletişimi ve bu nesnelerin yaşam süreçlerini modellemek için de UML etkileşim şemalarından olan sıralanım şemalarından faydalanılmıştır.

4.1 LADM Standardında Verileri Oluşturan Uygulama

Uygulama geliştirme sürecinde öncelikli olarak, kullanım durumlarının tasarlandığı planlama etkinliği gerçekleştirilmiştir. Kullanım durumlarının oluşturulmasındaki temel amaç, hedef kullanıcı ve sistem arasında gerçekleşmesi beklenen etkileşimin özelliklerini, amacını, sınırlarını ve çıktısını belirlemektir. Etkileşimin normal ya da olağan dışı akışlar olarak incelenmesi ve olası hataların öngörülmesi işlemleri de gerçekleştirilmektedir. Uygulama kapsamında kullanıcı ve sistem arasında gerçekleşmesi beklenen iki kullanım durumu bulunmaktadır. Bunlar veri setinin oluşturulması ve doğrulanması kullanım durumlarıdır. Her iki kullanım durumunda da birincil aktör olarak kullanıcı bulunmaktadır. Tablo 4.1’de veri setinin oluşturulması kullanım durumu açıklanmıştır. Bahsedilen kullanım durumunun normal akışında devam edebilmesi için gerekli veri biçimi ihtiyaçları ve olası hata durumları da belirtilmiştir. Tablo 4.2’de de sistem tarafından oluşturulan veri setinin doğrulanması kullanım durumu açıklanmıştır. İkinci kullanım durumu, ilk kullanım durumunda bahsedilen süreç gerçekleştikten sonra oluşabileceği için aralarında içerme ilişkisi kurulmuştur. Tasarlanan bu kullanım durumları ile de Şekil 4.2’de bulunan kullanım durum şeması modellenmiştir. Modellenen bu şema da kullanıcının gerçekleştirebileceği etkileşimleri görselleştirmektedir.

Tablo 4.1: Veri setinin oluşturulması kullanım durumu

Kullanım Durumu İsmi : Veri Setinin Oluşturulması	ID:1.1	Tip:Detaylı
Birincil Aktör: Kullanıcı		
Paydaş(lar): Kullanıcı		
Kısa Tanım: Kullanıcı sistemin çalışabilmesi için gerekli bilgi girişini sistem kısıtlamaları uyarınca gerçekleştirir. Sistem girdileri işleyerek oluşturduğu LADM standardına uygun veri setini kullanıcının belirlediği konuma aktarır.		
Tetikleyici: Kullanıcı, temel alınacak GeoJSON dosya, kat sayısı, kat yüksekliği ve aktarım tipi ile ilgili bilgi girişini tamamlar ve sunucu tarafına iletir.		
İlişki: Kullanıcı İçerir: Hiç Genişletir: Hiç		
Sürecin Normal Akışı 1. Kullanıcı JSP tabanlı web uygulamasına tarayıcı aracılığıyla ulaşır. 2. Kullanıcı geçerli biçimde olan GeoJSON dosyayı, kat sayısını ve yüksekliğini sisteme girer. 3. Kullanıcı veri tabanı ya da dosya tipinde olan aktarım yöntemlerinden birinin seçimini yapar ve ilgili aktarım yöntemi için gerekli hedef veri tabanı adresi, kullanıcı adı ve şifresini ya da dosya adresi bilgilerini sisteme girer. 4. Kullanıcı bir girdi elemanı aracılığıyla sistemi tetikler. 5. Uygulama girdilerin doğrulunu kontrol eder ve sistemin akışa devam etmesine karar verir. 6. GeoJSON biçiminde bulunan dosyada bulunan elemanlar ilgili Java sınıflarına dönüştürülür. 7. Dönüştürülen Java sınıfları kat sayısı ve yüksekliği bilgileri kullanılarak z boyutu bilgisi ile zenginleştirilir. 8. Zenginleştirilmiş Java sınıfları temel alınarak Oracle 11g veri tabanına eklenebilecek SQL cümleleri oluşturulur. 9. Oluşturulan cümleler kullanıcı seçimine göre belirtilen veri tabanı ya da dosyaya aktarılır.		
Alternatif /Olağan Dışı Akışlar 1a. Hizmet adresine bağlantı sırasında iletişim hatası oluşur. 2a. Yüklenen dosya biçimi uygun olmadığı için teknik hata oluşur.		

- 2b. Kat sayısı bilgisi eksik ya da tam sayı olmadığı için teknik hata oluşur.
- 2c. Kat yüksekliği bilgisi eksik ya da geçerli bir sayı olmadığı için teknik hata oluşur.
- 3a. Aktarım tipi seçimi eksik olduğu için teknik hata oluşur.
- 3b. Aktarım tipi ile alakalı diğer girdiler eksik olduğu için teknik hata oluşur.
- 5a. İstemci tarafından gönderilen bilgiler tekrar kontrol edilir ve sunucuya iletim sırasında bilgiler değiştirilmiş ise teknik hatalar oluşur.
- 6a. GeoJSON elemanları standart dışında oluşturulmuş ise teknik hata oluşur.
- 9a. Kullanıcı tarafından sağlanan bilgiler ile veri tabanına bağlanılamaz ise iletişim hatası oluşur.
- 9b. SQL cümleleri veri tabanına aktarılamaz ise teknik hata oluşur.
- 9c. Kullanıcı tarafından sağlanan dosya konumda bulunamaz ya da erişime açık değil ise iletişim hatası oluşur.

Son Durum

Kullanıcı tarafından girişi gerçekleştirilen GeoJSON biçiminde dosya temel alınarak, LADM standardında ve Oracle 11g veri tabanı yönetim sistemi ile uyumlu veri seti oluşturulur.

Tablo 4.2 : Oluşturulan veri setinin doğrulanması kullanım durumu

Kullanım Durumu İsmi : Veri Setinin Doğrulanması	ID:1.2	Tip:Detaylı
Birincil Aktör: Kullanıcı		
Paydaş(lar): Kullanıcı		
Kısa Tanım: Veri setinin oluşturulması kullanım sürecinin normal akış ile sonlanması sonucunda işlenen GeoJson elemanları hem işlenme sürecinde hem de işlendikten sonra aktarım sürecinde doğrulanır.		
Tetikleyici: Veri setinin oluşturulması kullanım sürecinin normal akış içinde sonlanması.		
İlişki: Kullanıcı		
İçerir: Veri Setinin Oluşturulması		
Genişletir: Hiç		
Sürecin Normal Akışı		
1. Yüklenen GeoJSON dosyasında bulunan elemanlar işlenirken, sahip oldukları mekansal özellikler analiz edilerek işlemin devam etme durumuna karar verilir.		

Devam etmeyen işlemler sistem tarafından saklanır.

2. İşleme sonucunda oluşan SQL cümleleri aktarım sürecinde hata ile karşılaşılırsa, ilgili SQL cümlesi sistem tarafından saklanır.

3. GeoJSON dosyasında bulunan elemanların işlenme ve aktarılma durumları hesaplanır.

4. Hesaplamaların sonucu web sayfası üzerinden kullanıcıya sunulur.

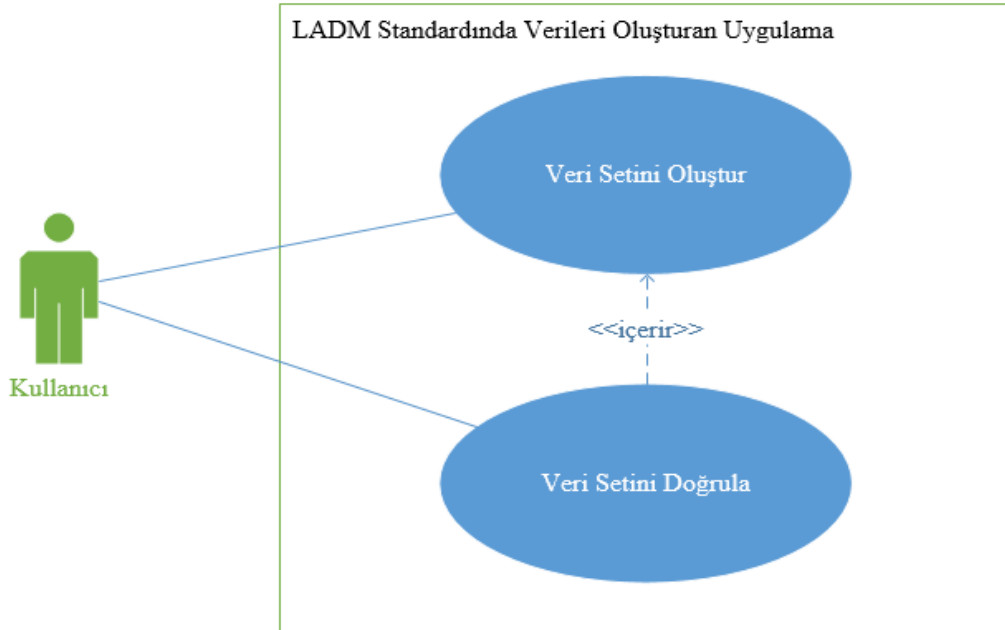
Alternatif /Olağan Dışı Akışlar

1a. İşlenen eleman GeoJSON standardı dışında ise teknik hata oluşur ve hata durumu saklanır.

2a. Aktarılmak istenen SQL cümlesinin çalıştırılması ya da yazılması sırasında teknik ya da iletişim hatası oluşur ve hata durumu saklanır.

Son Durum

Kullanıcı tarafından girişi gerçekleştirilen GeoJSON biçiminde dosya temel alınarak oluşturulan LADM standardında verinin doğrulanması gerçekleştirilir.



Şekil 4. 2 : LADM standardında verileri oluşturan uygulama kullanım durum şeması

4.1.1 Coğrafi veri setinin seçilmesi

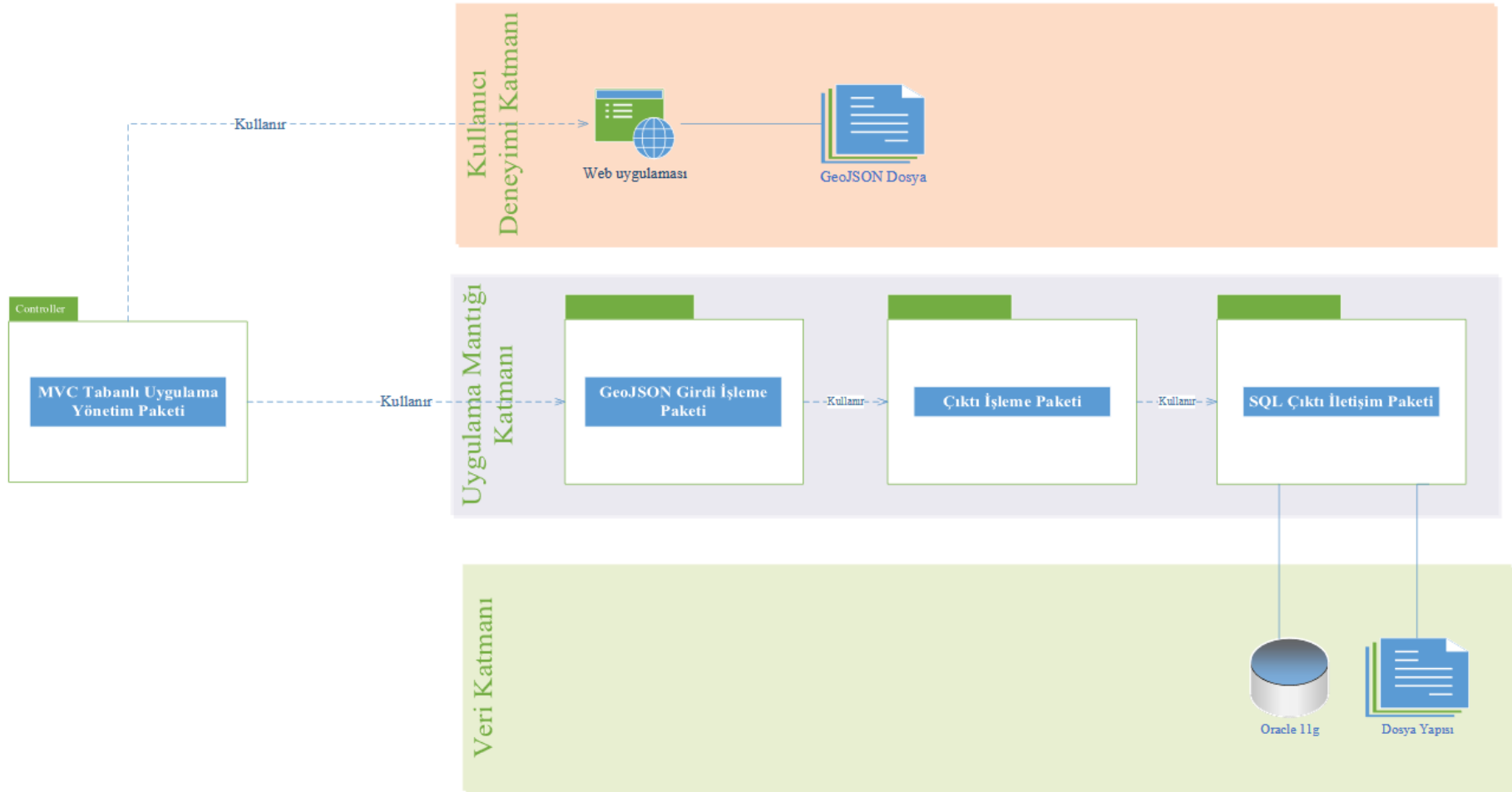
LADM standardına uygun verilerin oluşturulabilmesi amacıyla ilk olarak koordinat bilgisine sahip olan bir veri seti elde edilmesi gerekmektedir. Seçilen veri setinde olması gereken özellik iki adet nokta ile tanımlanmış noktalardan oluşan kapalı çizgi

geometrisinde iki boyutlu alan bilgileridir. Yazılım modülünün kullanım alanlarını genişletmek adına, seçilen veri seti genel kabul görmüş ve web ortamında kullanımı uygun olan mekansal veri modellerinden birisi ile biçimlenmiş setlerden seçilmek istenmiştir. Seçimin bu şekilde yapılması sonucunda, karar kılınan mekansal veri modeline uygun tüm veri setleri LADM standardına uygun bir şekilde dönüştürülebilecektir. Alan temsil eden veri setinin birbirine yakın ya da komşu olması da gözetilmiştir. Bu durumun sebebi de görselleştirme yapıldığı zaman nesnelerin birbiri arasındaki kesişim, uzaklık gibi ilişkilerin kullanıcıya etkili bir şekilde sunulmasını sağlamaktır.

LADM standardında verilerin oluşturulabilmesi amacıyla GeoJSON veri modelin karar kılınmıştır. Bu kararın alınmasındaki başlıca sebep GeoJSON veri modelinde, görselleştirilmek istenen iki nokta ile temsil edilen kapalı çizgi geometrileri temsilinin standartlaştırılmış anahtar kelimeler aracılığıyla yapılmasıdır. Örnek verilecek olursa; “geometry” anahtar kelimesiyle verinin mekansal veri olarak tanımlanması, “type”, “Polygon” ve “coordinates” anahtar kelimeleriyle de sırasıyla mekansal veri tipi, cinsi ve koordinat değerleri tanımlanması standartlaştırılmıştır. Anahtar kelimelerin kullanılması sonucunda seçilen veri setinin içeriği ne olursa olsun yazılım modülü ile işlenebilecektir. Ayrıca ileride bu anahtar kelimeler değişir ya da standartlara yeni anahtar kelimeler eklenirse yazılımın güncellenmesi, anahtar kelimeler özellik dosyalarından yönetileceği için daha az maliyetli olacaktır.

4.1.2 Coğrafi verilerin oluşturulması uygulaması katmanları

Uygulama kapsamında geliştirilen ve kullanılan yazılım bileşenleri gerçekleştirdikleri işler ve çok katmanlı mimaride bulundukları konum göz önüne alınarak gruplanmışlardır. Gruplama işlemi gerçekleştirilirken yapısal UML şemalarından olan paket şemalarından faydalanılmıştır. Uygulama içerisinde yazılım paketlerini içeren katmanlar ve bu katmanlar arasında iletişimi kuran bir yönetim paketi bulunmaktadır. İletişimi kuran yönetim paketinin görevi istemci tarafında gösterilmek istenen web formunun kullanıcıya sunulması, formdan girişi alınan bilgilerin istemciye iletilmesi ve sunucunun cevabının istemci tarafına iletilmesi işlemlerinin bulunduğu süreçlerin yaşam döngülerinin kontrolünü sağlamaktır. Şekil 4.3’de gösterildiği üzere uygulama içerisinde üç katman bulunmaktadır, bunlar; veri, uygulama mantığı ve kullanıcı deneyimi katmanlarıdır.



Şekil 4. 3 : LADM standardında verileri oluşturan uygulama paket şeması

4.1.2.1 Veri katmanı

Oluşturulan verilerin saklanması için LADM standardında bulunan sınıfların saklanmasına uygun bir tablo yapısı tasarlanmıştır. Tablo yapısı tasarlanırken standartta belirlenen ilişkilerin bağımlılığının en az şekilde olması ve veri tekrarının olmaması için tablolar arası ilişkiler de yeni tablolar ile temsil edilmişlerdir. LADM standardında bulunan tüm sınıfların arasından tez kapsamında kullanılacak sınıfların seçiminin yapılması sürecinde tezin 3.1.3.4 bölümünde tanımlanan LADM temel sınıflarının kullanılmasında karar kılınmıştır. Kararın verilmesindeki sebepler ise bu sınıflar aracılığıyla LADM standardına tamamen bağlı kalınacak olunması ve standardın temel paketlerinin hepsinin içinde bulunan en az bir sınıfın tez kapsamında kullanarak daha kapsayıcı bir sonuca ulaşılacak olunmasıdır. LADM standardında bulunan tüm sınıfların tez kapsamında kullanılmamasındaki sebep ise standartta bulunan çoğu sınıfın anlamsal bilgileri saklıyor olması ve tezin amacı olan görselleştirme ile ilgilerinin olmamasıdır. Ayrıca mekansal bilgilerin oluşturulacağı dosya kullanıcı tarafından giriliyorken ve anlamsal bilgiler sistem tarafından rastgele oluşturuluyorken anlamsal bilgilerin görselleştirme sürecine dahil edilmesi ek bir kazanım sağlamayacaktır. Tez kapsamında kullanılan sınıflarda görselleştirme ile ilgisi olmayan alanlar boş bırakılmıştır ancak tez kapsamı gelecek çalışmalarda genişletilmek istenirse bu alanlar da genişletme amacına kullanılabileceği için oluşturulmuşlardır.

Temel sınıfların tanımlanması sürecinde, tanımlanmak istenen sınıf başka bir sınıfın referansına alan olarak sahip ise referans gösterilen sınıf da tez kapsamına dahil edilmektedir. Referans gösterilen sınıflar, alan olarak sahip olan sınıfın özelliklerini tanımladıkları için önceden tanımlanmış değerler ile oluşturulacaktır. Bu sınıfların veri tabanına aktarılma sürecinde öncelikle tabloları yaratılacak ardından veriler uygun olan şekilde bu tablolara eklenecektir. Belirtilen bu süreç tüm uygulamalar çalışmadan önce gerçekleştirilecektir.

İştirakler paketi

İştirakler paketi kapsamında bulunan üç sınıf tez kapsamına dahil edilmiştir. Dahil edilen bu sınıflardan LADM temel sınıflarından olan “LA_PARTY” sınıfıdır. Bu sınıf şahıs, grup ya da tarafları temsil etmek için kullanılmaktadır. Sadece anlamsal bilgileri içeren bu sınıf id, dış id, tip ve rol tipi bilgilerine sahiptir. Tip ve rol tipi bilgileri de sırasıyla “LA_PARTYTYPE” ve “LA_PARTYROLETYPE” sınıflarında

tutulmaktadır ve referans olarak gösterilmektedir. “LA_PARTYTYPE” sınıfında tutulan tip bilgisi şahıs, tüzel kişilik, vakıf, grup ya da temel yönetim birimi (BAUnit) gibi iştirak tipi bilgileri içerir. “LA_PARTYROLETYPE” sınıfında tutulan rol tipi bilgisi ise gözlemci, noter, banka, finans sağlayıcısı, vergi mükellefi, çiftçi, vatandaş gibi iştirakin rolünü belirtir. İlgili tabloların tanımlanması işlemini gerçekleştiren SQL cümleleri aşağıda belirtilmişlerdir.

```
CREATE TABLE "LADM"."LA_PARTYTYPE" (  
  "CID" NUMBER(27) NOT NULL,  
  "NAME" VARCHAR2(30 BYTE) NOT NULL,  
  "DESCRIPTION" VARCHAR2(100 BYTE) NULL,  
  "CODE" NUMBER(3) NOT NULL);
```

```
CREATE TABLE "LADM"."LA_PARTYROLETYPE" (  
  "CID" NUMBER(27) NOT NULL ,  
  "NAME" VARCHAR2(30 BYTE) NOT NULL ,  
  "DESCRIPTION" VARCHAR2(100 BYTE) NULL ,  
  "CODE" NUMBER(3) NOT NULL);
```

```
CREATE TABLE "LADM"."LA_PARTY" (  
  "EXTPID" NUMBER(27) ,  
  "NAME" VARCHAR2(100 BYTE) NOT NULL,  
  "PID" NUMBER(27) NOT NULL,  
  "LA_PARTYROLETYPEPEREF" NUMBER(27),  
  "LA_PARTYTYPEPEREF" NUMBER(27) NOT NULL);
```

Temel yönetim birimleri paketi

Temel yönetim birimleri paketi kapsamında bulunan iki sınıf tez kapsamına dahil edilmiştir. Dahil edilen bu sınıflardan LADM temel sınıflarından olan “LA_BAUNIT” sınıfıdır. Bu sınıf ev, garaj, tarla gibi temel yönetsel birimleri temsil etmektedir. Sadece anlamsal bilgileri içeren bu sınıf id, isim, tip bilgilerine sahiptir. Tip bilgisi “LA_BAUNITTYPE” sınıfında tutulmaktadır ve referans olarak gösterilmektedir. “LA_BAUNITTYPE” sınıfında tutulan tip bilgisi temel mülkiyet, kiralanmış ya da kullanım hakkı gibi anlamsal bilgileri içermektedir. İlgili tabloların tanımlanması işlemini gerçekleştiren SQL cümleleri aşağıda belirtilmişlerdir.

```
CREATE TABLE "LADM"."LA_BAUNITTYPE" (  
  "RID" NUMBER(27) NOT NULL,  
  "NAME" VARCHAR2(30 BYTE) NOT NULL,  
  "DESCRIPTION" VARCHAR2(100 BYTE) NULL,  
  "CODE" NUMBER(3) NOT NULL);
```

```
CREATE TABLE "LADM"."LA_BAUNIT" (  
  "UID" NUMBER(27) NOT NULL,  
  "NAME" VARCHAR2(100 BYTE),  
  "LA_BAUNITTYPEPEREF" NUMBER(27) NOT NULL);
```

Mekansal birimler paketi

Mekansal birimler paketi kapsamında bulunan üç sınıf tez kapsamına dahil edilmiştir. Dahil edilen bu sınıflardan LADM temel sınıflarından olan “LA_SPATIALUNIT” sınıfıdır. Bu sınıf mekansal bir olguyu temsil etmektedir. Örneğin bir parsel, su ya da hacim bu sınıf kullanılarak temsil edilir. Temel yönetsel birimleri yapılandırmak için bu sınıftan faydalanılır. Sınıfı id, etiket, dış adres referansı, referans nokta, boyut tipi, alan değeri, hacim değeri ve yüzey ilişkisi bilgilerine sahiptir. Sınıfın oluşturulması için zorunluluğa sahip olan bilgi id bilgisidir. Diğer bilgiler modelleme işleminin gerçekleştirileceği bağlamın gerekliliğine göre kullanılabilir. Tez kapsamında sadece boyut tipi bilgisi kullanılacaktır. Sınıf içerisinde bulunan diğer alanların kullanılmamasının nedeni ise görselleştirme ile ilgili alakalarının bulunmamasıdır. Boyut tipi bilgisi “LA_DIMENSIONTYPE” sınıfında tutulmaktadır ve referans olarak gösterilmektedir. “LA_DIMENSIONTYPE” sınıfında tutulan tip bilgisi sıfır, bir, iki ya da üç boyutlu gibi mekansal boyut bilgilerini içermektedir.

Tez kapsamında oluşturulan mekansal verilerin tutulduğu LADM sınıfı ise “LA_BOUNDARYFACE” sınıfıdır. Diğer sınıflar anlamsal bilgileri ve mekansal olgu bütünü temsil etmek için kullanılırken bu sınıf doğrudan mekansal bilgiyi saklamaktadır. Sınıfı id, mekansal bilgi, kaynak referansı ve mekansal bilginin metin olarak adres değeri bilgilerini saklamaktadır. Tez kapsamında mekansal bilgi Oracle veri tiplerinden olan “SDO_GEOMETRY” veri tipinde saklanmaktadır. Kaynak ve adres metni bilgisi ise boş bırakılmıştır. İlgili tabloların tanımlanması işlemini gerçekleştiren SQL cümleleri aşağıda belirtilmiştir.

```
CREATE TABLE "LADM"."LA_DIMENSIONTYPE" (  
  "SUID" NUMBER(27) NOT NULL,  
  "NAME" VARCHAR2(30 BYTE) NOT NULL, -- 0D, 1D, 2D, 3D, liminal, etc.  
  "DESCRIPTION" VARCHAR2(100 BYTE) NULL,  
  "CODE" NUMBER(3) NOT NULL);
```

```
CREATE TABLE "LADM"."LA_SPATIALUNIT" (  
  "SUID" NUMBER(27) NOT NULL ,  
  "LABEL" VARCHAR2(100 BYTE),  
  "EXTADDRESSID" VARCHAR2(100 BYTE),  
  "LA_REFERENCEPOINT" NUMBER(27),  
  "LA_DIMENSIONTYPEPEREF" NUMBER(27),  
  "LA_AREAVAUEREF" NUMBER(27),  
  "LA_VOLUMEVAUEREF" NUMBER(27),  
  "LA_SURFACERELATIONREF" NUMBER(27));
```

```
CREATE TABLE "LADM"."LA_BOUNDARYFACE" (  
  "BFID" NUMBER(27) NOT NULL,
```

```
"GEOMETRY" MDSYS.SDO_GEOMETRY,  
"LA_SOURCEREF" NUMBER(27),  
"LOCATIONBYTEXT" VARCHAR2(250 BYTE));
```

GeoJSON özellik dizisi tabloları

Tez kapsamında oluşturulacak olan veri seti GeoJSON tabanlı olduğu için veri kaybını önleyebilmek adına, LADM ve GeoJSON standartları arasında bir ilişkinin kurulması gerekmektedir. Mekansal bilgiler iki modelde de tutulduğu için bu ilişkinin ek bir tablo ile kurulması gerekmemektedir. LADM standardında "LA_BOUNDARYFACE" sınıfında tutulan mekansal bilginin bir benzeri GeoJSON standardında "geometry" dizisi içerisinde bulunan "type" ve "coordinates" alanlarında tutulmaktadır. GeoJSON standardında dizi halinde tutulan "properties" alanının LADM sınıfı ile ilişkisini sağlayabilmek adına "GEOJSON_PROPERTY" tablosu oluşturulmuştur. Tabloda id, özellik ismi ve özelliğin değerinin tanımlanabileceği üç alan bulunmaktadır. Tabloya eklenen her kayıt, "properties" dizisi içerisinde bulunan isim/değer çiftine karşılık gelecektir. Özellik dizisinde bulunan elemanların hangi "geometry" ile ilişkili olduğu bilgisini belirtmek için de "LA_BOUNDARYFACE" sınıfından elemanlar ile temsil edilen "LA_SPATIALUNIT" sınıfı ile "GEOJSON_PROPERTY" tabloları arasındaki ilişkinin tutulduğu "SPATIALUNIT_GEOJSON_PROPERTY" isimli tablo oluşturulmuştur. Özellik dizisinden elde edilen verilerin her "LA_BOUNDARYFACE" elemanı için kaydedilmesinin önüne geçmek için böyle bir yöntem izlenmiştir. GeoJSON özellik dizisinin tutulmasını sağlayan tabloları oluşturan SQL cümleleri aşağıdaki gibidir.

```
CREATE TABLE "LADM"."GEOJSON_PROPERTY" (  
    "ID" NUMBER(27) NOT NULL,  
    "PROPERTY_NAME" VARCHAR2(1000 BYTE),  
    "PROPERTY_VALUE" VARCHAR2(1000 BYTE));  
  
CREATE TABLE "LADM"."SPATIALUNIT_GEOJSON_PROPERTY" (  
    "ID" NUMBER(27) NOT NULL,  
    "LA_SPATIALUNITREF" NUMBER(27) NOT NULL,  
    "GEOJSON_PROPERTYREF" NUMBER(27) NOT NULL);
```

LADM sınıfları arası ilişki tabloları

Veri tekrarının azalması ve SQL sorgularının performansının artırılması amacıyla LADM standardında önerilen sınıflar arası ilişkiler, ilişki tabloları aracılığıyla gerçekleştirilmiştir. Tez kapsamında oluşturulan veriler sınıf tablolarında tutulurken bu veriler arasında gerçekleşen ilişkiler id ve referans id bilgileri içeren bu tip tablolarda tutulmuşlardır.

Tez kapsamında LADM standardına göre tasarlanan ilk ilişki tablosu, mekansal verilen tutulduğu “LA_BOUNDARYFACE” ve mekansal olguların tutulduğu “LA_SPATIALUNIT” arasındaki ilişkiyi temsilen oluşturulan “LA_SPATIALUNIT_BOUNDARYFACE” tablosudur. İlişkiye gerçek uygulamadan örnek verilecek olunursa; dört duvar, bir taban ve de bir çatıdan oluşan bir oda veri tabanına eklenmek istendiğinde duvarlar, çatı ve taban için altı “LA_BOUNDARYFACE” kaydının tekil anahtarları ve odanın kendisi için oluşturulan bir “LA_SPATIALUNIT” kaydının tekil anahtarını içeren altı kayıt bahsedilen tabloya eklenecektir. İşlem sonucunda tabloya eklenen tüm veriler bir odayı oluşturmak için gereken yüzeylerin bilgisini tekil anahtarlar aracılığıyla ilgili tablolara ulaşarak kullanıcıya verecektir. İlişki tablosunu oluşturan SQL cümlesi aşağıdaki gibidir.

```
CREATE TABLE "LADM"."LA_SPATIALUNIT_BOUNDARYFACE" (1
  "ID" NUMBER(27) NOT NULL,
  "LA_SPATIALUNITREF" NUMBER(27) NOT NULL,
  "LA_BOUNDARYFACEREF" NUMBER(27) NOT NULL);
```

LADM standardına göre tasarlanan ikinci ilişki tablosu mekansal olguların tutulduğu “LA_SPATIALUNIT” sınıfı ile temel yönetim birimlerinin tutulduğu “LA_BAUNIT” sınıfının ilişkilendirildiği “LA_BAUNIT_SPATIALUNIT” tablosudur. Tablo ile açıklanan ilişki, mekansal olguların yönetim birimi olarak karşılığını belirtmektedir. İlişki tablosunu aşağıdaki gibi tanımlanabilir.

```
CREATE TABLE "LADM"."LA_BAUNIT_SPATIALUNIT" (
  "ID" NUMBER(27) NOT NULL,
  "LA_BAUNITREF" NUMBER(27) NOT NULL,
  "LA_SPATIALUNITREF" NUMBER(27) NOT NULL);
```

Tasarlanan üçüncü tablo ise iştiraklerin tutulduğu “LA_PARTY” sınıfı ile yönetim birimlerinin tutulduğu “LA_BAUNIT” sınıfı arasındaki ilişkinin kurulduğu “LA_PARTY_BAUNIT” tablosudur. İlişki, iştirakler ile yönetim birimleri arasında anlamsal bağ kurmak için kullanılmaktadır. Aşağıdaki SQL cümlesi ile ilişki tablosu tanımlanabilir.

```
CREATE TABLE "LADM"."LA_PARTY_BAUNIT" (
  "ID" NUMBER(27) NOT NULL,
  "LA_PARTYREF" NUMBER(27) NOT NULL,
  "LA_BAUNITREF" NUMBER(27) NOT NULL);
```

4.1.2.2 Uygulama mantığı katmanı

Uygulama katmanında kullanıcı deneyimi katmanından alınan bilgiler işlenerek LADM standardında verilerin oluşturulması ve veri tabanına ya da dosyaya

kaydedilmesi işlemi gerçekleştirilmektedir. İşlemleri gerçekleştiren yazılım bileşenleri sorumluluklarına göre üç paket olarak gruplanmışlardır. Bu paketler; GeoJSON girdi işleme, çıktı işleme ve SQL çıktı iletişim paketi olarak isimlendirilmişlerdir.

GeoJSON işleme paketi

GeoJSON işleme paketi kullanıcı deneyimi katmanında fiziksel konumu iletilen dosyayı konumlandırarak işlemlerine başlamaktadır, eğer dosya konumlandırılmaz ise ön yüze hata ile ilgili mesaj dönerek işlemi sonlandırmaktadır. Dosya konumlandırıldıktan sonra dosya okuma işlemi gerçekleştirilmektedir, eğer dosya yazılım tarafından açılmaz ya da .json uzantısı ile biçimlendirilmemiş ise ön yüze ilgili hata mesajı ileterek işlemlerini durdurmaktadır. Dosya okuma süreci başarılı bir şekilde sonlandırıldıktan sonra dosya içerisinde bulunan elemanlar GeoJSON standartlarında bulunan anahtar kelimelerine göre kontrol edilirler. Gerçekleştirilen kontrollerin amacı okunan elemanların GeoJSON standartlarına tamamen uygun olmasını sağlamaktır. İşleme sürecinde aşağıdaki isim/değer anahtar kelimeleri kontrol edilmektedir.

- Elemanın “type” isminde değer “Feature” anahtar kelimesini içermesi durumu kontrol edilir.
- Elemanın “geometry” isminde değer olarak da bir isim/değer dizisini içermesi durumu kontrol edilir.
- İsim/değer dizisinde “type” isminde ve “Polygon” değerinde anahtar kelimeler içermesi durumu kontrol edilir.
- İsim/değer dizisinde “coordinates” isminde ve iki sayıdan oluşan noktaları temsil eden verileri içermesi durumu kontrol edilir. Listede bulunan ilk ve son nokta çiftinin aynı olması durumu kontrol edilmektedir; bu kontrolün amacı geometrinin kapalı çizgi geometrisi olma durumunu gelecek işlemlerden önce garantiye almaktır.
- Elemanın “properties” isminde değer olarak da bir isim/değer dizisini içermesi durumu kontrol edilir ancak bu isim/değer çiftinin bulunması zorunlu değildir. GeoJSON standartları tarafından ek özellikleri temsil etmek için opsiyonel olarak tanımlanmıştır.

Kontrol sürecinden geçen elemanlar Java nesnelere dönüştürülmek amacıyla tekrar işlenirler. İşlemin amacı geometri ve özellikleri içeren sınıfları oluşturmak ve bu

sınıfları kullanarak GeoJSON elemanını tamamen temsil eden “GeoJSONElement” isimli Java sınıfını oluşturmaktır. “GeoJSONElement” sınıfı “Geometry” sınıfına, “Property” sınıfı listesine ve “type” isimli “String” tipinde sınıf değişkenlerine sahiptir. Dönüşüm sürecinde GeoJSON elemanının değerleri, oluşturulacak nesnenin alanlarına taşınır. Taşıma işlemi sonucunda oluşan nesneler liste şeklinde çıktı işleme paketinde bulunan sınıflara iletilir.

Çıktı işleme paketi

Çıktı işleme paketinde bulunan sınıfların hedefi kendisine iletilen nesne listesinde bulunan elemanları, kullanıcı tarafından sağlanan kat sayısı ve yüksekliği bilgileri ile zenginleştirmektir. İşlenecek olan “GeoJSONElement” nesnelerinin bu safhada temsil ettiği mekansal olgu ikili koordinat bilgileri ile kodlanmış noktaları içeren, yeryüzüne paralel ve bitişik çoklu çizgi geometrileridir. Geometri temsilleri taban kabul edilerek yeryüzüne dik duvarlar ve yer yüzüne paralel tavan nesnelerinin “GeoJSONElement” olarak yaratılması işlemi listede bulunan her eleman için gerçekleştirilmektedir. İşlem sırasında duvarların kenar yükseklikleri ve tavanın yerden yüksekliği kullanıcı tarafından belirlenen değer ile hesaplanır. Hesaplanan yükseklikler oluşturulacak olan nesnenin sahip olduğu “Geometry” sınıfında bulunan “coordinates” dizisinde noktayı temsil eden iki değer yanına eklenerek tutulmaktadır; böylece sınıfta bulunan alan artık noktaları x, y ve z koordinatları şeklinde temsil etmektedir (Taban olan nesne için 0 yükseklik değeri kullanılır). Kat sayısı değeri ile de bu işlemin o taban için kaç kere tekrarlanacağına karar verilir. Aynı taban için gerçekleştirilen her işlem sonunda kalınan yükseklik değeri bir sonraki işleme aktararak yer yüzünden yukarı doğru tavan, taban ve duvarları temsil eden “GeoJSONElement” nesneleri oluşturulmaktadır. Taban olan geometrinin işleme dahil olabilmesi için kaç kenarı olduğunun bir önemi yoktur sadece kapalı olması yeterlidir, işleme sürecinde kenar sayısından bağımsız bir şekilde duvar oluşturulacak şekilde yordam geliştirilmiştir. Tabanı temsil eden elemanın z boyut değeri sıfır olarak kabul edildikten sonra tabanı oluşturan ardışık iki nokta oluşturulacak duvarın alt başlangıç noktaları kabul edilir. Alt başlangıç noktaları yükseklik yani z boyutu ile zenginleştirilerek duvarın üst noktaları yaratılır. Alt ve üst kısımda elde edilen toplam dört nokta ile bir duvar temsil edilmektedir. Tabanı oluşturan noktalar üzerinde bu işlem nokta çiftleri aynı olana kadar saat yönünde gerçekleştirilir ve tüm duvarlar oluşturulmuş olur. Oluşturulan bu duvarların üst

noktaları ile tavan da oluşturulabilir. Tüm bu yüzeyler bütünü ile kapalı, üç boyutlu bir mekansal hacim temsil edilmiş olur. Her bir taban ile ayrı ayrı gerçekleşen bu işlemlerin sonucunda apartmanları temsil eden ve LADM standartlarına uyumlu Java nesneleri oluşturulmaktadır.

SQL çıktı iletişim paketi

SQL çıktı iletişim paketinin amacı kendisine iletilen “GeoJSONElement” nesnelerini dönüştürerek, Oracle 11g veri tabanı yönetim sistem üzerinde tutulan LADM standardında tasarlanmış tablolara eklenebilecek SQL cümleleri üretmektir. Üretilen SQL cümleleri kullanıcı tarafından girişi yapılan veri tabanı tablo ve alan ismine uygun olarak oluşturulmaktadır. Oluşturulan cümlelerin isimlendirilmesi kullanıcı tarafından gerçekleştirilse de tablo yapısı ve verinin saklanma biçimi LADM standartlarına uygun olarak yapılmaktadır. Saklama işlemi gerçekleştirilirken mekansal olguların temsili LADM standartlarında tanımlanmış iki tablo ve birbiriyle ilişkileri aracılığıyla sağlanır; bu tablolar “LA_BOUNDARYFACE” ve “LA_SPATIALUNIT” tablolarıdır. “LA_BOUNDARYFACE” tablosunda bulunan kayıtlar iki ya da üç boyutta bir yüzeyi temsil eden temel geometriler olarak standartlaştırılmışlardır. “LA_SPATIALUNIT” tablosunda bulunan kayıtlar da adres, alan, boyut tipi, hacim gibi öz niteliklere ve bu öz niteliklerle ilgili hesaplama yordamlarına sahip olan mekansal birimlerin temsili için standartlaştırılmış. Bahsi geçen iki tablonun da tekil anahtarları vardır ve bu anahtarlar iki tablo arasında ilişki tablosu yaratmak için kullanılabilmektedir. Tez kapsamında daha önce bahsedilen “LA_SPATIALUNIT_BOUNDARYFACE” isimlendirilmesi ile bir ilişki tablosu tasarlanmıştır. Tabloda üç alan bulunmaktadır bunlar; kayıt tekil anahtarı ile “LA_SPATIALUNIT” ve “LA_BOUNDARYFACE” kayıtlarına referans olan tekil anahtarlar. Tabloda tutulan referans anahtarlar veri bütünlüğünü sağlayabilmek adına çift olarak da tekillik sağlama zorunluluğuna sahiplerdir. İlişki tablosunun da tutulan kayıtların belirttiği temel bilgi ise referansı olan bir “LA_SPATIALUNIT” kaydını mekansal olarak oluşturmak için gereken “LA_BOUNDARYFACE” referans kayıtları bütünüdür.

“GeoJSONElement” nesneleri işlenirken yer üzerinde bulunan bir yüzeyden referans alınarak taban, duvarlar, tavan olarak yaratılan ve x, y, z boyutlarını temsil eden kapalı geometrilerinden “LA_BOUNDARYFACE” tablosuna eklenebilecek şekilde SQL cümleleri oluşturulmaktadır. İşlenen her taban için de ayrıca bir

“LA_SPATIALUNIT” SQL kayıt cümlesi oluşturulmaktadır. Her referans yüzey için gerçekleştirilen işlemler sonrası oluşturulan bu SQL kayıt cümlelerinin oluşturulmasından sonra “LA_SPATIALUNIT” ve “LA_BOUNDARYFACE” kayıtları arasında ilişkilerin tutulduğu tabloya kayıt eklemek için de “LA_BOUNDARYFACE” sayısı kadar “LA_SPATIALUNIT” ve “LA_BOUNDARYFACE” tekil anahtarlarına sahip ilişki kayıtlarının temsil edildiği SQL cümleleri oluşturulur. Bir taban için işlem bittiği zaman; tabanı, tavanı ve tüm duvarlarını temsil eden “LA_BOUNDARYFACE” kayıtları, bir adet “LA_SPATIALUNIT” kaydı ve “LA_SPATIALUNIT” ile ilişkili “LA_BOUNDARYFACE” kayıtlarını tanımlayabilecek SQL cümleleri oluşturulmaktadır.

“GeoJSONElement” nesnesi ile önceki paragrafta bahsedilen SQL cümleleri oluşturulduktan sonra, nesnede bulunan “Property” sınıfı listesinde bulunan her bir özellik elemanı için “GEOJSON_PROPERTY” tablosuna isim/değer çifti kullanılarak kayıt ekleyebilecek SQL cümleleri oluşturulmaktadır. Gerçekleştirilen her bir özellik ekleme cümlesi oluşturma işlemi sonrası da “SPATIALUNIT_GEOJSON_PROPERTY” tablosu için, son eklenen özellik ve “LA_SPATIALUNIT” arasındaki ilişkiyi temsil eden bir SQL ekleme cümlesi oluşturulmaktadır. İlişki, iki tabloda bulunan tekil anahtarlar aracılığıyla kurulmaktadır.

Oluşturulan her “LA_SPATIALUNIT” kayıt ekleme cümlesi için mekansal olgunun temel yönetim birimi olarak karşılığı olan “LA_BAUNIT” tablosuna kayıt ekleme ihtiyacı doğmuştur. İhtiyacın sebebi LADM standardında bulunan temel sınıflara tam uyum sağlayabilmektir. Kayıt ekleyebilmek amacıyla oluşturulan SQL cümlesinde isim alanı veri oluşturma sürecinde rastgele olarak doldurulmaktadır. Tip değeri de önceden tanımlanan değerler arasından rastgele seçilmektedir. Kayıt ekleme cümlesi oluşturulduktan sonra “LA_BAUNIT” ve “LA_SPATIALUNIT” arasındaki ilişkiyi temsilen “LA_BAUNIT_SPATIALUNIT” tablosuna kayıt ekleyecek SQL cümlesi de oluşturulmaktadır. Gerçekleştirilen bu işlem sonrası mekansal olgu, bir temel yönetim birimi ile temsil edilmiştir.

Mekansal olgu ile temel yönetim birimi arasındaki ilişkinin kurulması sağlandıktan sonra bahsi geçen temel yönetim birimi ile iştirakler arasında da bir ilişkinin kurulması gerekmektedir. İlişkinin kurulabilmesi için öncelikle rastgele değerler ile

bir iştirak kaydı ekleyecek SQL cümlesi oluşturulmuştur. Oluşturulan bu cümlede isim alanı rastgele harfler, tip ve rol tipi alanı da önceden girilmiş değerler arasından rastgele seçilen bir referans ile doldurulmuştur. İşlemin ardından iştirak ve temel yönetim biriminin arasındaki ilişkinin tutulduğu “LA_PARTY_BAUNIT” tablosuna kayıt ekleyen SQL cümlesi tekil anahtarlar aracılığıyla oluşturulmuştur.

Her bir “GeoJSONElement” sınıfı için gerçekleştirilen tüm bu SQL cümlesi oluşturma işlemi süreci sonrasında elde edilen SQL cümleleri ile LADM standardına uygun ve GeoJSON standardında bulunan tüm verileri temsil edebilecek bir veri seti oluşturulmuştur. Oluşturulan bu veri setinde mekansal bilgilerin tamamı ve anlamsal bilgilerin bir bölümü GeoJSON kaynak dosyasından gelmekteyken, LADM sınıf tabloları rastgele bilgiler ile doldurulmuştur.

4.1.2.3 Kullanıcı deneyimi katmanı

Kullanıcı deneyimi katmanı kapsamında sade bir bilgi giriş formu tasarlanmıştır. Bilgi giriş formunun amacı kullanıcı tarafından sağlanan veri setinin ve bilgilerin uygulama katmanına ulaştırılması ve uygulama katmanı tarafından gerçekleştirilen işlemlerin sonuçlarının kullanıcıya gösterilmesidir. Geliştirilen kullanıcı bilgi giriş form uygulaması istemci tarafında JSP, HTML, CSS, JavaScript ve JQuery teknolojileri kullanmıştır. JSP ve HTML teknolojisi ile form üzerinde bulunan girdi alanları ve diğer yapısal alanlar kodlanmıştır. CSS teknolojisi ile form üzerinde bulunan elemanların görsel özellikleri güçlendirilmiştir. JavaScript ve JQuery teknolojileriyle de form üzerinde bulunan elemanların doğrulanmaları, kısıtlanmaları ve birbirlerine bağlı davranışları kodlanmıştır. İstemci tarafından girişi alınan veriler Spring MVC teknolojisi aracılığıyla Java sınıfı olarak sunucu tarafından işleneceği bir şekilde uygulama katmanına iletilir. Uygulama katmanında işlenen bilgiler ile ilgili bir sorun ortaya çıkarsa ya da işlem başarılı bir şekilde sonuçlanırsa, uygulama katmanında isteğe cevap veren Java sınıfı tarafından, istemci tarafına sonuç mesajı döner ve bu mesaj Spring teknolojisinden yararlanılarak yakalanıp kullanıcıya sunulur. Kullanıcı tarafından sağlanması beklenen on bilgi vardır, bunlar;

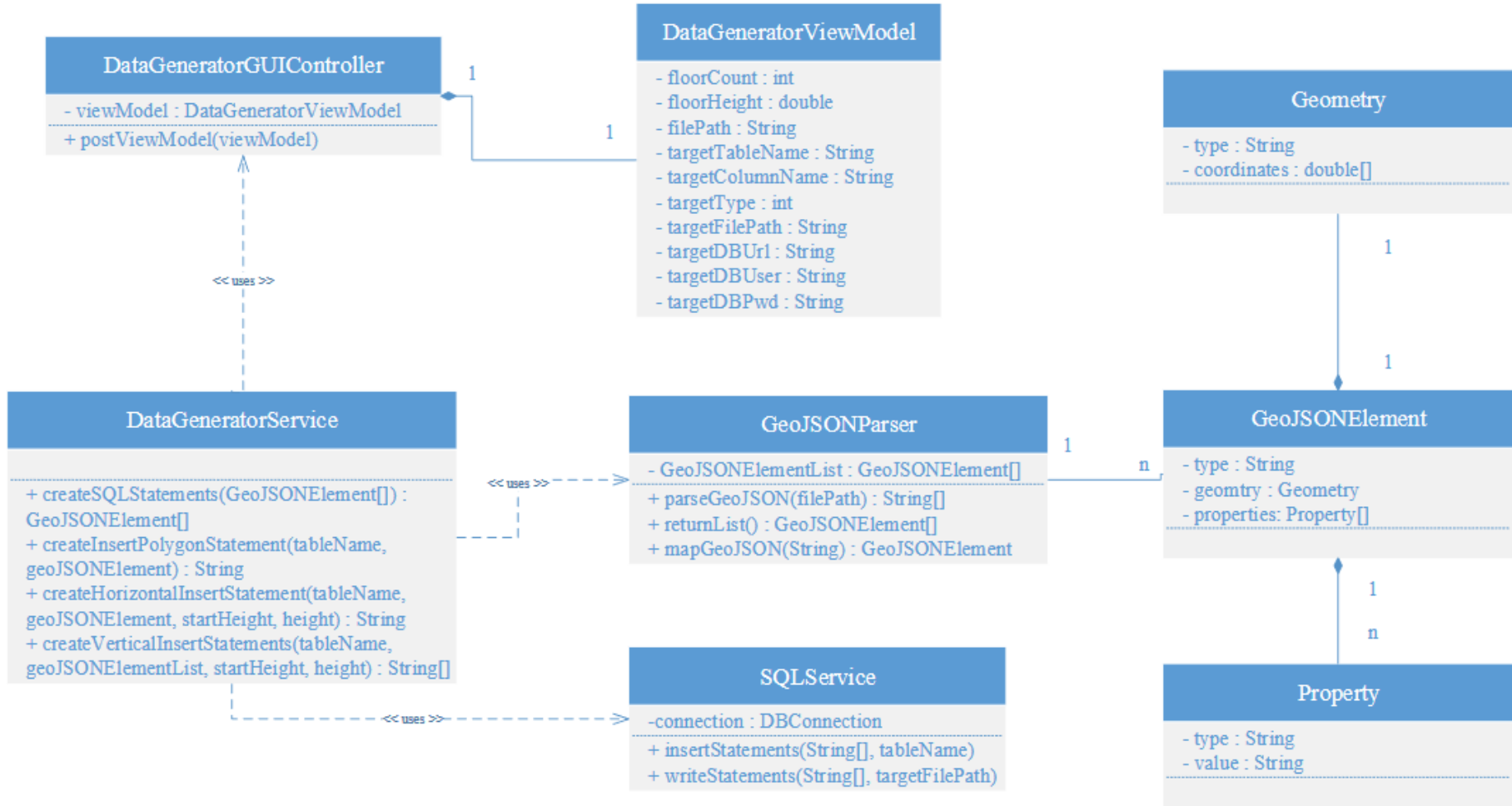
- Oluşturulmak istenen apartmanların azami kat sayısı bilgisi. Bu bilgi tam sayı ve dolu olmalıdır.
- Oluşturulmak istenen apartmanların kat yüksekliği bilgisi. Bu değer kesirli ve dolu olmalıdır.

- Veri setinin bulunduğu dizinin fiziksel adres bilgisi. Bu bilgi karakter dizisi şeklinde, dolu ve .json uzantısı ile biçimlendirilmiş olmalıdır.
- Oluşturulacak olan LADM standardında verinin kayıt edileceği hedef SQL tablosunun isim bilgisi. Bu bilgi karakter dizisi şeklinde ve dolu olmalıdır.
- Oluşturulacak olan LADM standardında verinin kayıt edileceği hedef SQL tablosunda mekansal verilerin temsil edileceği alanın isim bilgisi. Bu bilgi karakter dizisi şeklinde ve dolu olmalıdır.
- Veri setinin işlendikten sonra kayıt edileceği ortamı belirlemek adına “Veri Tabanı” ve “Dosya” değerlerine sahip bir seçim kutusu. Seçim yapılması zorunlu kılınmıştır.
- “Dosya” değeri seçilmiş ise sonuçların yazılması istenen dosyanın fiziksel adres bilgisi. Bu bilgi karakter dizisi şeklinde ve dolu olmalıdır.
- “Veri Tabanı” değeri seçilmiş ise üç farklı karakter dizisi alanının doldurulması zorunlu kılınmaktadır. Bu alanlar hedef veri tabanı adresi, veri tabanı kullanıcı ismi ve şifresidir. Kullanıcı tarafından girişi gerçekleştirilen bu alanlar ile veri tabanı bağlantısı kurmak amaçlanmıştır.

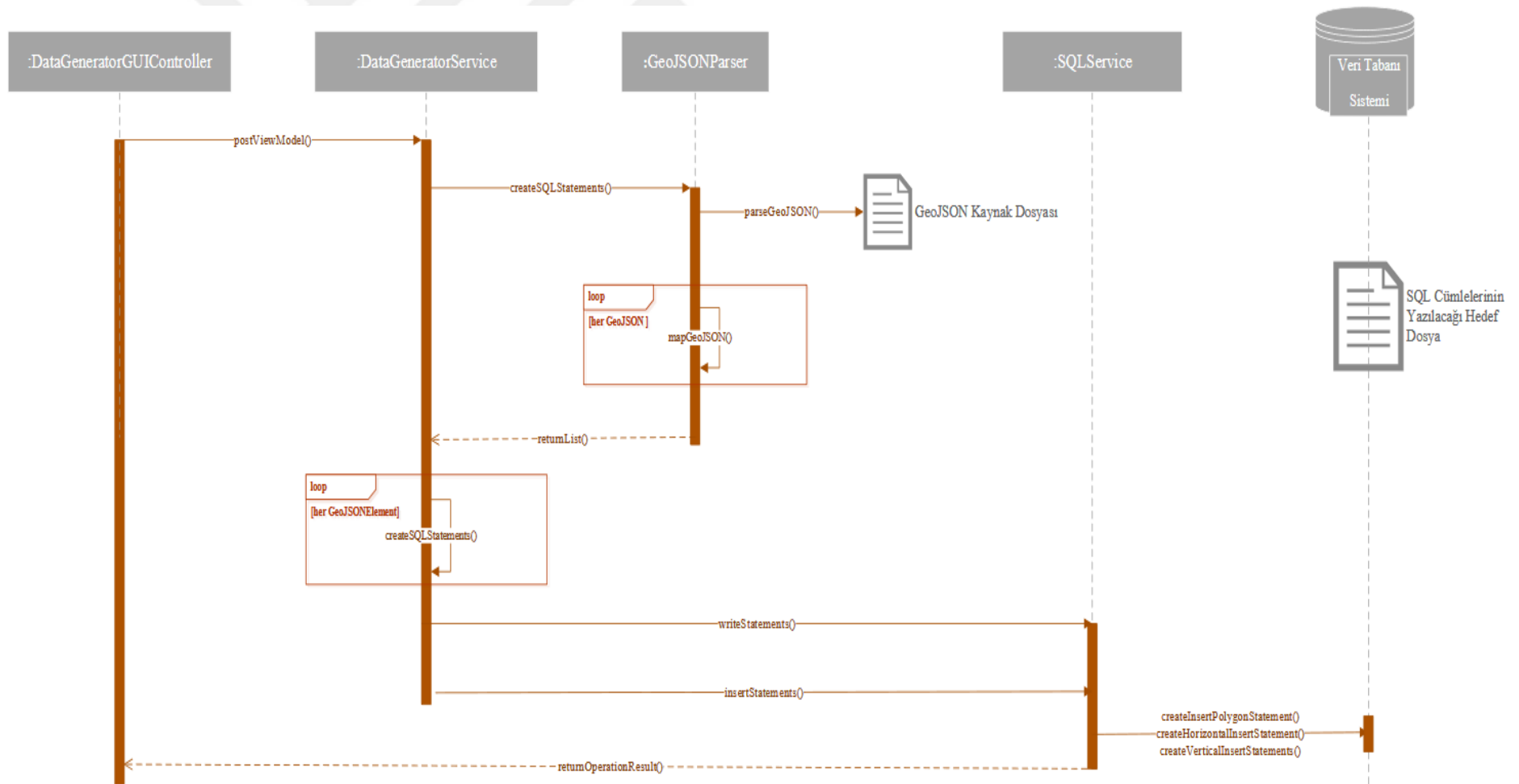
Kullanıcı tarafından girilen tüm bilgilerin doluluk ve gereken kısıtlara uyma durumu istemci tarafında da kontrol edilmektedir. Geliştirilen uygulamanın veritabanı ve dosya hedefli iki kullanımı için de geliştirilen arayüzleri, Ekler bölümünde bulunan Ek A başlığı altında gösterilmiştir.

4.1.3 Coğrafi verilerin oluşturulması uygulaması sınıf ve sıralanım şemaları

Şekil 4.4’de geliştirilen uygulamada kullanılan sınıfların belirtildiği sınıf şeması gösterilmektedir. Şema kapsamında sınıflarda bulunan alanlar, sınıfların sahip olduğu yordamlar ve sınıfların birbirleri arasındaki iletişim belirtilmiştir. Şekil 4.5’de de kullanıcı ve yazılım bileşenleri arasında gerçekleşen işlemlerin sıralanım şeması gösterilmektedir. Sıralanım şemasında gerçekleştirilen işlemler bir zaman çizelgesi halinde sol taraftan, işlemin sonlandığı sağ tarafa doğru bir sıra halinde belirtilmektedir. Modellenen iki şemada da tez kapsamında görselleştirme üzerine çalışıldığı için sadece mekansal bilgilerin tutulduğu sınıflar gösterilmişlerdir.



Şekil 4. 4 : LADM standardında verileri oluşturan uygulama sınıf şeması



Şekil 4. 5 : LADM standardında verileri oluşturan uygulama sıralanım şeması

4.1.4 Oluşturulan coğrafi verinin doğrulanması

Oluşturulan coğrafi verinin tutarlı, anlam ifade eden ve bütünsel olabilmesi ayrıca kullanıcının bilgilendirilmesi amacıyla coğrafi bilgi işlem sürecinde yazılımsal olarak doğrulanmaktadır. Tez kapsamında gerçekleştirilen doğrulama işlemlerinde genel mantık olarak işlenen veri ile sonuç veri arasındaki sayı ilişkisi kontrol edilmektedir. Doğrulama işlemi iki seviyede gerçekleştirilmektedir bunlar; GeoJSON elemanları işlenirken ve kayıtlar veri tabanı ya da dosyaya aktarılırken. GeoJSON veriler işlenirken ilk olarak kaç noktadan oluştuklarına, elemanın “geometry” isminde olan değerinde bulunan “coordinates” isminde ve iki sayıdan oluşan noktaları temsil eden bilgiler aracılığıyla bakılır. Belirtilen alanda bulunan nokta çifti sayısı kadar duvar oluşturulmaz ise işlem yok sayılarak sonraki taban için devam eder ve hatalı kayıt adedi kullanıcıya iletilir. Tüm işlemler sonuçlanınca oluşan kayıtların sayısı ve veri tabanına ya da dosyaya yazılan kayıtların sayısı karşılaştırılarak da ikinci seviye doğrulama işlemi gerçekleştirilir. Doğrulama işlemleri sonucunda oluşabilecek farklılıklar ile ilgili bilgiler de kullanıcıya iki seviye bazında sunulur.

4.2 Kullanıcı Etkileşim Uygulaması

Kullanıcı etkileşim uygulamasının görevi LADM standardında oluşturulup Oracle 11g veri tabanına kaydedilen LADM standardında verilerin, kullanıcı tarafından web ortamında koordinat bazında sorgulanmasını sağlamak ve sorgu sonuçlarını web ortamında sanal bir küre üzerinde görselleştirerek sanal küre üzerinde gezinme imkanı sağlamaktır. Tablo 4.3’de üç boyutlu LADM elemanlarını görselleştirilmesi kullanım durumu açıklanmıştır. Bahsedilen kullanım durumunun normal akışında devam edebilmesi için gerekli veri biçimi ihtiyaçları ve olası hata durumları da belirtilmiştir. Tablo 4.4’de üç boyutlu nesnelerin görselleştirildiği sahne üzerinde gezinme kullanım durumu açıklanmıştır. İkinci kullanım durumu, ilk kullanım durumunda bahsedilen süreç gerçekleştikten sonra oluşabileceği için aralarında içerme ilişkisi kurulmuştur. Tasarlanan bu kullanım durumları ile de Şekil 4.6’da bulunan kullanım durum şeması modellenmiştir. Modellenen bu şema da kullanıcının gerçekleştirebileceği etkileşimleri görselleştirmektedir.

Tablo 4.3 : Üç boyutlu LADM elemanlarını görselleştirme kullanım durumu

Kullanım Durumu İsmi: Üç Boyutlu LADM Elemanlarını Görselleştir	ID:2.1	Tip: Detaylı
Birincil Aktör: Kullanıcı		
Paydaş(lar): Kullanıcı		
Kısa Tanım: Kullanıcı kapsayan kutu sorgusunu sunucu tarafına gönderir. Sorgu sonucu dönen LADM standardında olan mekansal ve anlamsal veri seti web ortamında bulunan CesiumJS tabanlı bir sanal küre üzerinde görselleştirilir.		
Tetikleyici: Kullanıcı üç boyutlu noktalar ile belirtilen kapsayan kutu (bounding box) sorgusunu CesiumJS tabanlı web arayüzü aracılığıyla başlatır.		
İlişki: Kullanıcı İçerir: Hiç Genişletir: Hiç		
Sürecin Normal Akışı 10. Kullanıcı JSP ve CesiumJS tabanlı web uygulamasına tarayıcı aracılığıyla ulaşır. 11. Kullanıcı nokta giriş elemanları aracılığıyla üç boyutlu kapsayan kutu sorgusunu gerçekleştirir. 12. Sunucu kapsayan kutu sorgusunu tablo yapısına uygun SQL cümlesine dönüştürür. 13. Sunucu JDBC aracılığıyla sorguyu veri tabanı üzerinde çalıştırır. 14. Sunucu sorgu sonucunu Java nesnelere dönüştürür. 15. Java nesneleri z koordinatı değeri ile zenginleştirilir. 16. Java nesneleri GeoJSON elemanlarına dönüştürülür ve istemciye gönderilecek cevaba eklenir ve istemci tarafına gönderilir. 17. CesiumJS tabanlı arayüze üzerinde çalışan JQuery cevap içinden GeoJSON elemanlarını ayırıştırır. 18. Ayırıştırılan elemanlar CesiumJS tarafından sanal küre üzerinde görselleştirilir.		
Alternatif /Olağan Dışı Akışlar 1a. Hizmet adresine bağlantı sırasında iletişim hatası oluşur. 2a. İstek gönderilirken iletişim hatası oluşur. 3a. Sorgu oluşturulurken teknik hata oluşur.		

- 4a. Sorgu çalıştırılırken iletişim hatası oluşur.
- 4b. Sorgu çalıştırılırken teknik hata oluşur.
- 5a. Dönüşüm işlemi sürecinde teknik hata oluşur.
- 6a. Zenginleştirme sürecinde teknik hata oluşur
- 7a. Cevap gönderilirken iletişim hatası oluşur.
- 7b. GeoJSON' a dönüştürülme sürecinde teknik hata oluşur.
- 8a. Ayırıştırılma sürecinde teknik hata oluşur.
- 9a. Görselleştirme sürecinde teknik hata oluşur.

Son Durum

Sorgu sonucunda elde edilen veri seti web ortamında bulunan sanal küre üzerinde görselleştirilir.

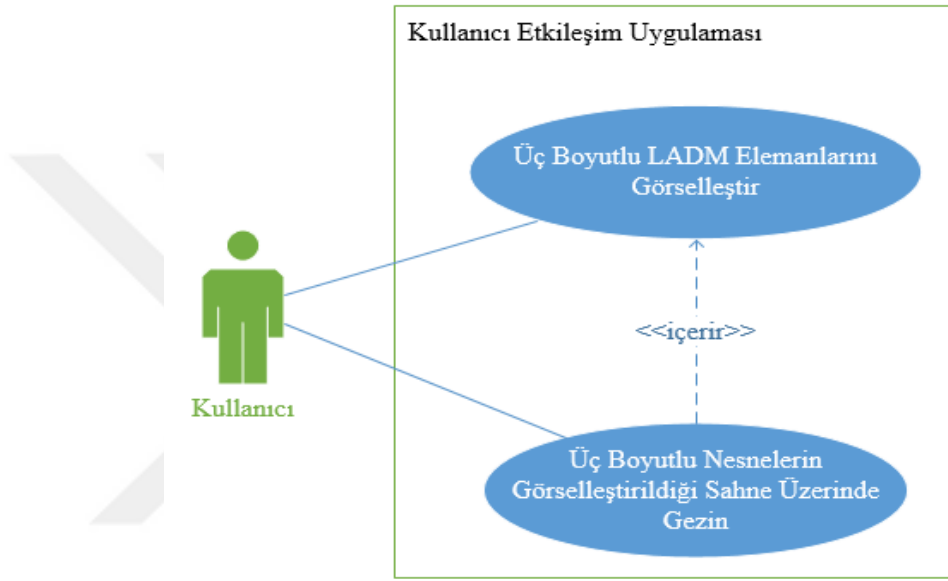
Tablo 4.4 : Üç boyutlu nesnelerin görselleştirildiği sahne üzerinde gezinme kullanım durumu

Kullanım Durumu İsmi : Üç Boyutlu Nesnelerin Görselleştirildiği Sahne Üzerinde Gezin	ID:2.2	Tip: Detaylı
Birincil Aktör: Kullanıcı		
Paydaş(lar): Kullanıcı		
Kısa Tanım: Kullanıcı, kapsayan kutu sorgusu sonucu görselleştirilmesi gerçekleştirilen LADM standardındaki veri setinin bulunduğu sanal küre üzerinde gezinme ve seçim işlemlerini gerçekleştirir.		
Tetikleyici: Kullanıcı görselleştirme sürecini tamamlar.		
İlişki: Kullanıcı		
İçerir: Kullanım Durumu 2.1		
Genişletir: Hiç		
Sürecin Normal Akışı 1. LADM standardında bulunan üç boyutlu veri setinin sanal küre üzerinde görselleştirilmesi tamamlanır. 2. Kullanıcı, kamera açısını PTZ (Pan, Tilt, Zoom) komutları ile değiştirir. 3. Kullanıcı, sanal küre üzerinde gezinti yapar. 4. Kullanıcı, görselleştirilen elemanları seçerek anlamsal bilgilere ulaşır.		
Alternatif /Olağan Dışı Akışlar		

- 1a. Görselleştirme işlemi sırasında teknik hata oluşur.
- 2a. Komutların işlenmesi sırasında teknik hata oluşur.
- 3a. Gezinti sırasında teknik hata oluşur.
- 4a. Anlamsal bilginin elde edilmesi sırasında teknik hata oluşur.

Son Durum

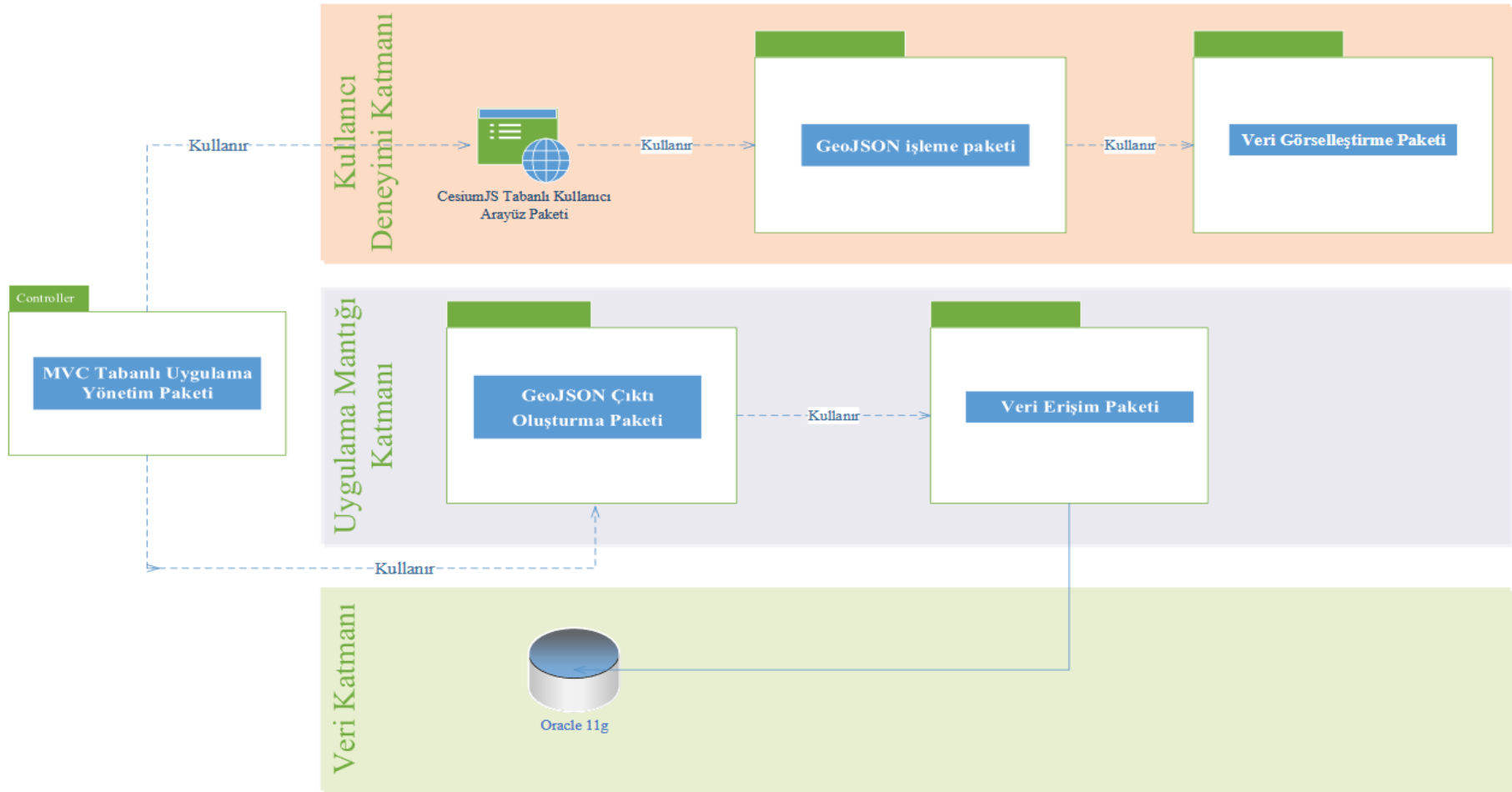
Kullanıcı sanal küre üzerinde gezinir ve seçim yapar.



Şekil 4. 6 : Kullanıcı etkileşim uygulaması durum şeması

4.2.1 Kullanıcı etkileşim uygulaması katmanları

Tez kapsamında geliştirilen kullanıcı etkileşim uygulaması içerisinde yazılım bileşenlerinin bulunduğu paketleri içeren üç adet katman ve bu katmanlar arasında iletişimi kuran bir paket bulunmaktadır. İletişimi kuran paketin görevi istemci tarafında gösterilmek istenen sanal küre içeren web formunun ve verilerin kullanıcıya sunulması, kullanıcı ara yüzünden girişi alınan bilgilerin istemciye iletilmesi ve sunucunun cevabının istemci tarafına iletilmesi işlemlerinin bulunduğu süreçlerin yaşam döngülerinin kontrolünü sağlamaktır. Uygulama kapsamında geliştirilen yazılım bileşenleri bir önceki uygulamada olduğu gibi yapsal UML şemalarından olan paket şemaları ile Şekil 4.7’de gösterildiği üzere modellenmiştir. Uygulama içerisinde üç katman bulunmaktadır, bunlar; veri, uygulama mantığı ve kullanıcı deneyimi katmanlarıdır.



Şekil 4. 7 : Kullanıcı etkileşim uygulaması uygulama paket şeması

4.2.1.1 Veri katmanı

Kullanıcı etkileşim uygulaması veri tabanı olarak bir önceki uygulama kapsamında oluşturulan veri tabanını kullanmaktadır. Veri tabanında bulunan tüm tabloların Java sınıfı olarak karşılığı da kullanıcı etkileşim uygulamasının veri katmanını oluşturmaktadır. LADM ve GeoJSON standartları kapsamında oluşturulan veri tabanı tablolarından sorgular sonucunda elde edilen verilerin programatik olarak işlenebilmesi amacıyla bu yaklaşım uygulanmıştır.

Sınıflar oluşturulurken tablolarda “NUMBER” veri tipinde bulunan alanların Java sınıflarında karşılıkları sayısal alanların tiplerine göre “int” ya da “BigDecimal” veri tiplerinde tutulmuşlardır. “VARCHAR2” olarak tutulan tablo alanları ise “String” veri tipinde tutulmuşlardır. Tablolarda referans olarak tutulan alanlar Java sınıflarına alan olarak eklenirken de Java nesneleri olarak eklenmişlerdir.

4.2.1.2 Uygulama mantığı katmanı

Uygulama katmanının temel görevleri görselleştirilmenin yapıldığı Model-View-Controller mimarisine sahip web uygulamasını enstrümente etmek, kullanıcı isteklerini işlemek, veri tabanı işlemleri gerçekleştirmek ve istemci tarafına işlenen verinin anlamlı bir şekilde iletilmesini sağlamaktır. İşlemleri gerçekleştiren yazılım bileşenleri sorumluluklarına göre üç paket olarak gruplanmışlardır. Bu paketler; MVC uygulama enstrümantasyon, veri erişim ve GeoJSON çıktı oluşturma paketleri olarak isimlendirilmişlerdir.

MVC Uygulama enstrümantasyon paketi

MVC Uygulama enstrümantasyon paketi uygulamanın web ortamında çalışabilmesini, istemci tarafından iletilen verilerin doğrulanmasını ve işlenmesini, veri tabanı sorgularının yapılabilmesi, yapılan sorgular sonucu gerçekleştirilen veri işleme sürecinin başlatılması ve bu süreç sonuçta ulaşılan verilerin tekrardan istemci tarafına iletilmesini sağlamaktadır. Paket içerisinde karşılama sayfasına yönlendirme, istemci tarafından veri kabul etme ve istemci tarafına veri iletme isteklerini cevaplayan üç yordamı içeren bir “SpringController” tipinde Java sınıfı bulunmaktadır.

Sınıfın sahip olduğu yordamlardan ilki uygulamaya erişmek istenen adrese gelen istekleri işleyen yordamdır. Yordam, istekleri uygulamanın karşılama sayfasına yönlendirerek görevini sonlandırmaktadır. İstemci tarafından veri kabul edilmesi

işlemini gerçekleştiren ikinci yordam da karşılama sayfasında bulunan veri giriş alanlarına kullanıcı tarafından doldurulan değerlerin sunucu tarafına iletilmesini ve doğrulanmasını sağlamaktadır. Verinin iletilmesi işlemi bir Spring özelliği olan veri bağlama mekanizması ile gerçekleştirilir. Mekanizma istemci tarafında bulunan tüm alanları bir Java sınıfının alanları ile eşleştirir ve kullanıcı tarafından tetiklenen istek ile veriler sunucuya bir Java sınıfı olarak iletilir. Verinin doğrulanması işlemi için oluşturulan ve görselleştirilmek istenen verinin uyduğu standartlar kısıt olarak kabul edilmişlerdir. Kullanıcıdan girmesi istenen noktaların Dünya Geodezi Sistemi 1984 (WGS 84) standardı çerçevesinde (National Geospatial - Intelligence Agency, 2014) olması için kontroller kullanıcı deneyimi katmanında gerçekleştirildiği gibi sunucu tarafında da gerçekleştirilmektedir. Bu kontroller;

- Boylam ifade eden değer -180 ile 180 arasında olması.
- Enlem ifade eden değer -90 ile 90 arasında olması.
- Yükseklik ifade eden değer -1000 ile 1000 arasında olmasıdır.

İstemci tarafından iletilen veriler standartlara uymuyor ise üçüncü yordam tarafından kullanıcıya hata mesajı gönderilmektedir. Veriler doğru ise üçüncü yordam sahip olduğu değerler ile GeoJSON çıktı oluşturma paketinde bulunan sınıfları tetikler ve bu paketin çıktısını istemci tarafına iletir. İşlem sonucu hatalı ise istemci tarafına hata mesajı gönderilir, işlem sonucu doğru ise de GeoJSON çıktı oluşturma paketinden iletilen GeoJSON biçiminde bulunan verileri istemciye iletilir.

Veri Erişim Paketi

Veri erişim paketinde bulunan sınıflar GeoJSON çıktı oluşturma paketinin gerçekleştirdiği işlemler sürecinde kullanıcıdan alınan bilgiler ile tetiklenerek çalışırlar. Çalışma mantığının bu şekilde kurulmasının sebebi bağımlılık enjeksiyonu pratiğini kullanarak paketler arasında bulunan ilişkinin olabilecek en alt seviyede tutulabilmesini sağlamaktır. İki paket arasında bulunan ilişki bir arayüz ile sağlandığı için de farklı bir veri tabanı yönetim sistemi ya da veri kaynağı ile çalışılacağı zaman sadece veri erişim paketinde değişiklikler yapılarak sistemin olağan çalışma akışına devam etmesi sağlanacaktır. Örneğin PostGIS veri tabanı yönetim sistemine taşınma yapılması gerektiği durumda sorgunun oluşturulması dahil veri tabanı ile alakalı işlemler bu paket kapsamında yapıldığı için arayüz kısıtları sağlandığı her durumda gerçekleştirilen değişiklikler diğer yazılım bileşenleri etkilemeyecektir.

Paket içerisinde bulunan sınıfların yordamlarından ilki, kullanıcı tarafından girilen WGS 84 standartlarında olan noktaları ve LADM standartlarında oluşturulup doldurulan tabloları sorgulamak için kullanılacak SQL cümlelerini parametrik olarak oluşturur. LADM standardında mekansal bilginin tutulduğu en temel tablo “BoundaryFace” olduğu için sorguların mekansal olan kısımlarının bu tablo ve mekansal veri tutan alanları üzerinde gerçekleştirilmesi hedeflenmiştir. Sorgu SQL cümlesi oluşturulurken tablo ve alan ismi parametrik olarak değiştirilebilmektedir. Sorgu cümlesi oluşturulurken Oracle tarafından sağlanan “SDO_ANYINTERACT” isimli mekansal analiz operatöründen faydalanılmaktadır. Operatör mekansal veriler arasında içinde bulunma, eşit olma gibi ilişkileri test eder ve herhangi bir ilişkiye sahip olan kayıtların sorgulanabilmesini sağlar. Belirtilen mekansal analiz operatörünün SQL cümlesi içerisinde bulunma durumu da parametrik bir şekilde gerçekleştirilmektedir. Parametre değeri değiştirilerek farklı mekansal analiz operatörleri de kullanılabilir.

Sorgulama cümlesi oluşturulurken kullanıcı tarafından iletilen üç boyutlu noktalar ile oluşturulan kapsayan kutu hacmi (“Bounding box”) içerisinde bulunan yüzeylerin sorgulanması hedeflenmiştir. “SDO_ANYINTERACT” mekansal analiz operatörü parametre olarak Oracle Veri tabanının mekansal veri tipi olan “SDO_GEOMETRY” tipinde değerleri kabul ettiği için öncelikle istemci tarafından iletilen noktaların bu biçime çevrilmesi işlemi gerçekleştirilmiştir. “SDO_GEOMETRY” tipi beş temel parametre ile oluşturulmaktadır. Bunlar;

- Mekansal verinin boyutunu ve tipini belirten sabit ve dört basamaklı sayı değeri. Dört basamaklı sayı değerinin ilk basamağı boyutu belirtirken, kalan üç değer de mekansal verinin tipini belirtir. Örneğin; ilk basamakta bulunan 2, 3 ya da 4 değeri boyutu belirtirken, 001 değeri noktayı, 002 değeri çizgiyi, 003 değeri de çoklu yüzleri belirtmektedir. Tez kapsamında sorgulanacak üç boyutlu yüzeyleri temsil etmek adına 3 ve 003 değer çiftleri birleştirilerek mekansal analiz operatörüne iletilecek dört haneli ve üç boyutlu çok yüzleri belirten değer oluşturulmaktadır.
- Mekansal verinin tanımlandığı mekansal kayıt tekil anahtar değeri. Tez kapsamında gerçekleştirilecek sorgular WGS 84 standardına sahip noktalar temel alınarak gerçekleştirileceği için bu standardın Oracle veri tabanında karşılığı olan 4985 değeri kullanılmaktadır.

- Mekansal verinin bulunduğu referans nokta değeri. Mekansal veriler, oluşturulan yüzeyler aracılığıyla temsil edileceği için referans nokta değeri boş (“NULL”) bırakılmıştır.
- Mekansal verinin tanımlandığı bilgilerin bulunduğu Oracle tarafından parametreleri tanımlanan dizi değeri “SDO_ELEM_INFO_ARRAY” dizisi üç değer ile tanımlanır bunlar; başlangıç ofseti, eleman tipi, yorumlama tipi değerleridir. Tez kapsamında oluşturulan sorguda bulunan offset değeri 1, eleman tipi değeri olarak düz çizgilerden oluşturulan yüzeyleri temsil eden 1003 değeri ve yorumlama tipi olarak da 1 değeri kullanılmıştır. Örnek değer olarak “SDO_ELEM_INFO_ARRAY (1, 1003, 1)” gösterilebilir.
- Mekansal verileri temsil eden nokta dizisi değeri. Oracle tarafından parametreleri tanımlanan dizi değeri “SDO_ORDINATE_ARRAY” en az bir nokta koordinat değeri ile tanımlanmaktadır. Tez kapsamında noktaların değerleri, kullanıcıların ilettiği noktalar ile belirtilmektedir. Kullanıcının girdiği iki adet üç boyutlu nokta çifti ile diziyi “SDO_ORDINATE_ARRAY (30, 40, 0, 180, 90, 1000)” şeklinde oluşturabiliriz. Dizide bulunan ve virgül ile ayrılan ilk üç nokta ilk noktayı, diğer üç nokta da ikinci noktayı belirtmektedir.

“SDO_GEOMETRY” veri tipinde bir değer oluşturulduktan sonra tablo ismi ve mekansal analiz operatörü ile tüm haliyle SQL sorgu cümlesi oluşturulabilir. Oluşturulan sorgu cümlesi “SELECT * FROM tablo_adı WHERE mekansal_operatör (SDO_GEOMETRY, alan_adı)” şeklinde parametrik olarak gösterilebilir. Örnek bir sql sorgu cümlesi aşağıdaki gibidir.

```
SELECT * FROM TR_BOUNDARYFACE bf WHERE SDO_FILTER(bf.geometry,  
SDO_GEOMETRY(3003, 4985, NULL, SDO_ELEM_INFO_ARRAY(1,1003, 3 ),  
SDO_ORDINATE_ARRAY(30, 40, 0, 180, 90, 1000 )) ) = 'TRUE';
```

Sorgu cümlesi oluşturulduktan sonra sorgulama işlemini gerçekleştirecek yordama parametre olarak iletilerek tetikleme işlemi gerçekleştirilir. Sorgu işleminin gerçekleştirilebilmesi için üç parametre gerekmektedir. İlk parametre tetikleme anında iletilen sorgu cümlesidir. İkinci parametre programatik olarak veri tabanı bağlantısının gerçekleştirilmesini sağlayan veri tabanı sürücüsüdür. Veri tabanı sürücüsü olarak Oracle tarafından geliştirilmiş ve kullanılan Oracle 11g veri tabanı ile uyumlu Oracle Java Database sürücüsü (OJDBC) kullanılmaktadır. Sürücü

bağımlılığını projeye dahil edilirken Apache Maven proje yönetim otomasyonu kullanılmıştır. Sürücü olarak kullanılan Java arşivinin “SDO_API” programlama arayüzüne sahip olması gerekmektedir. Belirtilen arayüz veri tabanında bulunan “SDO_GEOMETRY” tipinde alanların programatik olarak işlenebilmesini sağlayan sınıflara sahiptir. Ayrıca Oracle veri tabanı yönetim sisteminin mekansal olarak sunduğu tüm işlevlerin de gerçekleştirilebilmesini sağlayan sınıflara sahiptir. Son parametre olarak da bağlanılacak olan veri tabanının fiziksel adres bilgisi, kullanıcı adı ve şifresi ile oluşturulan ve Oracle tarafından geliştirilen “Connection” sınıfıdır.

Tüm parametreler tanımlandıktan sonra sorgulama işlemini gerçekleştiren yordam, belirtilen fiziksel adreste bulunan veri tabanına kullanıcı adı ve şifresi ile sürücünün yardımı ile bağlanarak sorgu cümlesini hedef veri tabanında çalıştırır. Belirtilen adreste veri tabanı bulunamazsa, kullanıcı adı ya da şifresi yanlışsa, sürücü teknik bir hatayla karşılaşır ya da bağlantı gerçekleştirilemezse istemci tarafına gerçekleşen hatanın tipini göre hata mesajları iletilmektedir. Sorgu çalıştırma işlemi başarılı bir şekilde çalışır ancak kayıt dönmez ise de istemci tarafına belirtilen noktalar arasında kayıt dönmediğini belirten bilgilendirme mesajı iletilir. Sorgunun çalıştığı ve kayıt döndüğü durumlarda ise Oracle tarafından geliştirilen ve veri tabanı işlem sonuçlarının tutulduğu “ResultSet” sınıfı, sorgu sonucunda dönen mekansal ve anlamsal değerlere sahip alanlar ile doldurulur. Belirtilen sınıf kullanılan veri tabanı yönetim sisteminden bağımsız olarak sorgular sonucu doldurulabilecek bir sınıf olduğu için, veri erişim paketinden GeoJSON paketine iletilir ve gerçekleştirilecek diğer işlemlerin bu paket içerisinde bulunan sınıflar tarafından yapılması sağlanır. Tasarımın bu şekilde yapılmasının sebebi sistem farklı bir veri tabanı yöneti sistemi üzerinde çalıştırılmak istendiğinde sadece sorgu cümlesi, sürücüyü ve bağlantı parametrelerini değiştirerek aynı “ResultSet” sınıf değerini elde edebilmektir.

GeoJSON çıktı oluşturma paketi

Veri erişim paketinden dönen kayıt seti GeoJSON çıktı oluşturma paketi tarafından işleme alınmaktadır. Gerçekleştirilen işlemler, dönen “ResultSet” sınıfı içerisinde bulunan her kayıt satırı için ayrı ayrı gerçekleştirilmektedir ve sonuç olarak da tez kapsamında geliştirilmiş “SpatialViewElement” isimli sınıfın nesnelerinden üretmeyi hedeflemektedir. Nesne üretim sürecinde “ResultSet” nesnesi içinde bulunan kayıt satırlarındaki anlamsal ve mekansal veriler birbirinden farklı şekilde işlenmektedir. İşlenme sürecinde anlamsal veriler tip kontrolünden geçirilip Java dilinde bulunan

hazır veri tipleri ile eşleştirilmektedir. Örneğin işlenen kayıta bulunan anlamsal veri, metin biçiminde ve veri tabanında “VARCHAR” tipinde saklanıyor ise Java sınıfı oluşturulurken eşleşebileceği “String” veri tipiyle temsil edilir ve oluşturulacak olan nesnenin bir alan değeri olarak geçici şekilde kaydedilir. Aynı şekilde sayısal biçimde bulunan anlamsal alanlar da tip kontrolünden geçirilir ve Java dilinde karşılıkları olan veri tipleri ile eşleştirilerek oluşturulan nesnenin bir alanı olarak geçici şekilde kaydedilir. Geçici şekilde kaydedilmelerini sebebi GeoJSON üretim sürecinde tüm bu verilerin metinleştirilerek bir diziye eklenecek olmasıdır.

Mekansal veriler ile ilgili nesneleştirme süreci de tip kontrolü ile başlamaktadır. Kayıt satırlarında bulunan mekansal veri tip kontrolünden geçirildikten sonra Oracle tarafından geliştirilmiş ve veri tabanı işlemlerinde ilgili alandan gelen veriyi salt bit değeri olarak Java ortamında saklayabilen “Struct” sınıfına dönüştürülür. Mekansal veri salt bit biçimindeyken üzerinde işlem yapmak ya da bilgilerini okuyabilmek programatik olarak riskli olacağı için ileri bir dönüşüm daha gerçekleştirilmektedir. Gerçekleştirilen dönüşüm sırasında Oracle tarafından geliştirilen ve Oracle Java DataBase sürücüsüne (OJDBC) dahil edilen “JGeometry” sınıfından faydalanılmaktadır. “JGeometry” sınıfı Oracle veri tabanı yönetim sisteminde bulunan “SDO_GEOMETRY” veri tipinin, Java programlama dilinde kullanılacak karşılığı olması için geliştirilmiş bir sınıftır. Ayrıca “Struct” gibi salt bit içeren Java sınıflarını, eğer doğru biçimlendirilmişlerse, kendi tipine dönüştürebilecek yordamlara da sahiptir. Dönüşüm sürecinde de “JGeometry” sınıfının bu işlemleri gerçekleştiren yordamları kullanılmıştır. Kayıt satırında bulunan mekansal veriler “Struct” sınıfına dönüştürüldükten sonra “JGeometry” sınıfında bulunan yordamlar aracılığıyla “JGeometry” sınıfının nesneleri olacak şekilde dönüştürülmüşlerdir. Dönüşüm sonucunda elde edilen nesnelerin sahip olduğu özellikleri elde etmek ya da nesnelerin üzerinde işlemler gerçekleştirmek programatik olarak daha az riske sahip olmuştur. Örneğin bir “Struct” nesnesinin tanımlandığı mekansal kayıt tekil anahtarını elde etmek için bit düzeyinde işlem yapmak gerekirken, “JGeometry” sınıfına dönüştürülmüş bir nesnenin bu özelliği sınıf tarafından sunulan yordam aracılığıyla gerçekleştirilebilir.

“ResultSet” nesnesi içinde bulunan kayıt satırlarındaki mekansal veriler “JGeometry” nesnelerine, anlamsal olan metin ve sayı tabanlı veriler de ilgili Java veri tiplerine dönüştürüldükten sonra hedeflenen “SpatialViewElement” nesnesi

dönüşüm işlemi gerçekleştirilmektedir. Belirtilen nesne üç veri alanına sahiptir. Bu alanlar; tip, geometri ve özellik çiftlerinden oluşan bir veri dizisidir. Her nesne için tip alanına GeoJSON standartlarında bulunan ve mekansal bir özelliğin belirtildiğini açıklayan “Feature” anahtar kelimesi metinsel olarak eklenir. Geometri alanına ise dönüştürülen “JGeometry” nesnesi yazılır. Gerçekleştirilen sorguya göre okunan ve önceden geçici olarak kaydedilen metin ve sayı tabanlı anlamsal veriler de metinleştirilerek özellik çiftleri olarak veri dizisi alanına taşınır. Tüm işlemlerin sıralı olarak gerçekleştirilmesi sonucunda “SpatialViewElement” nesneleri her kayıt için oluşturulmuş olur.

“SpatialViewElement” sınıfı GeoJSON standartlarına uygun şekilde dönüştürülebilecek bir Java sınıfı olarak tasarlanmıştır. GeoJSON veri formatında oluşturulan dosyalarda temel olarak üç alan bulunmaktadır. Bu alanlar; isim belirten “name”, veri tipi belirten “type” ve mekansal verilerin kendilerini belirten “features” alanlarıdır. Oluşturulan GeoJSON biçimli dosyada, isim belirten “name” alanına okunan tablo ve okuma tarihi alanı birleştirilerek taşınır. “Type” alanına GeoJSON anahtar kelimelerinde olan ve bir dizi ile mekansal verilerin temsil edileceğini belirten “FeatureCollection” değeri yazılır. “Features” alanına ise tip, geometri ve özellik dizisi halinde üç alan yazılır ve bu işlem oluşturulan tüm “SpatialViewElement” nesneleri için tekrarlanır. Tip değeri olarak “Feature” anahtar kelimesi yazılır. Geometri alanında mekansal verinin tipi ve koordinat dizisi alanı yazılır. Tez kapsamında kapalı çoklu çizgiler ile çalışıldığı için tip değerine “Polygon” anahtar kelimesi taşınmıştır. Koordinat alanına da mekansal veri tabanından okunan ve “JGeometry” sınıfına dönüştürülen nesnelerin “SDO_ORDINATE_ARRAY” değerleri taşınmıştır. Özellik dizisine de “SpatialViewElement” nesnelerinde bulunan özellik dizisi taşınır. Gerçekleştirilen dönüşüm işlemlerinin ardından GeoJSON standardında bir cevap dosyası oluşturulmuş olur ve paketi tetikleyen MVC Uygulama enstrümantasyon paketine iletilir.

4.2.1.3 Kullanıcı deneyimi katmanı

Kullanıcı deneyimi katmanının temel görevleri kullanıcının bilgi girişi sağlayabileceği bir veri giriş formunun sunulmasını sağlamak, sunucu ile bilgi alışverişini kullanıcı komutlarına göre yönetmek, sunucudan gelen GeoJSON standardında olan cevabı işlemek, işlenen cevabı üç boyutlu bir küre üzerinde

görselleştirmek ve kullanıcıdan gelen girdilere göre bu küre üzerinde gezinme deneyimini sunmaktır. Katmanın gerçekleştirdiği işlemler sürecinde kullanılan, tez kapsamında geliştirilen ve dış kaynaklardan edinilen yazılım bileşenleri üç paket olarak ayrılmışlardır. Bu paketler kullanıcı arayüz paketi, GeoJSON işleme paketi ve veri görselleştirme paketleridir.

Kullanıcı arayüz paketi

Kullanıcı arayüz paketi içerisinde web ortamında sunulan ve kullanıcının etkileşime geçebileceği yazılım bileşenlerini bulunmaktadır. Yazılım bileşenleri geliştirilirken Java Server Pages (JSP), Hyper Text Markup Language (HTML), Cascading Style Sheet (CSS), JQuery, CesiumJS, ThreeJS teknolojileri kullanılmıştır. İstemci tarafından yorumlanacak şekilde kodları geliştirilen kullanıcı arayüzü iki bölümden oluşmaktadır.

Arayüzün ilk bölümünde üzerinde verilerin henüz görselleştirilmediği CesiumJS tabanlı sanal küre bulunmaktadır. Sanal küre, üzerinde fare ve klavye aracılığıyla gezinme, yakınlığa uzaklaşma, sağladığı programla arayüzü aracılığıyla programatik olarak nesne eklenebilme ve eklenen nesneleri fare ile seçerek veri gösteren alanların sayfaya eklenmesini sağlayan özelliklere sahiptir.

Arayüzün ikinci bölümünde de kullanıcıdan bilgi girişinin ve girilen bilginin kullanıcı tarafından tetiklenerek sunucu tarafına iletilmesini sağlayan bileşenler bulunmaktadır. Bölümde ilk olarak, sorguların gerçekleştirileceği üç boyutlu noktaların kullanıcı tarafından belirlenebilmesi amacıyla konumlandırılan veri giriş alanları ve bu alanların kullanımlarını yönlendiren metin alanlar bulunmaktadır. Kullanıcıdan beklenen ve üç koordinat değeri ile ifade edilen her nokta için x, y ve z değerlerini kabul eden sayı giriş alanları bulunmaktadır. Bu alanlardan kabul edilen verilerin Dünya Geodezi Sistemi 1984 (WGS 84) standardı çerçevesinde olması adına JavaScript dili ile kontroller gerçekleştirilmektedir ve kullanıcının, sistem tarafından kabul edilebilecek veri biçimi ile ilgili hata mesajlarıyla bilgilendirilmesi sağlanmaktadır. Gerçekleştirilen kontroller sonucunda aşağıdaki kısıtlara tamamen uyan değerler sistem tarafından kabul edilir.

- X değerinin -180 ile 180 arasında olması.
- Y değerinin -90 ile 90 arasında olması.
- Z değerinin de -1000 ile 1000 arasında olması.

Kullanıcının, sistem tarafından kabul edilen biçimlere uygun veri girişi yapması sağlandıktan sonra bölüme konumlandırılan bir buton aracılığıyla formda bulunan tüm bilgiler sunucu tarafına iletilir. Sunucu tarafında bir hata alınırsa bu hata kullanıcıya iletilmektedir. Eğer bir hata ile karşılaşılmaz ise sunucudan dönen GeoJSON biçimindeki cevap geldiği anda JQuery teknolojisinin sağladığı yordamlar aracılığıyla GeoJSON işleme paketi bileşenlerinin kullanıldığı süreç başlar.

GeoJSON işleme paketi

GeoJSON işleme paketinin temel görevleri sunucudan gelen cevapları dinleyerek araya girmek, gelen cevabı işlemek ve görselleştirme sürecini işlenen cevap ile tetiklemektir. Kullanıcının girdiği bilgilere istinaden sunucunun verdiği cevap istemci tarafına iletildiği anda cevap içerisinde bulunan GeoJSON dosyası paket içerisinde bulunan yordamların aracılığıyla ayrıştırılır. Ayrıştırılan GeoJSON biçimindeki dosya içeriğinden, JQuery teknolojisinden faydalanılarak, JavaScript nesneleri üretilmesi işlemi gerçekleştirilir. Üretilen bu JavaScript nesneleri, veri tabanında sorgulanan mekansal verilerin tüm bilgilerine sahip bir şekilde oluşturulmaktadır. GeoJSON dosyası içerisinde bulunan tüm veriler ile bir JavaScript nesnesi üretildikten sonra bu nesneler bir dizi veri yapısına gönderilir ve oluşan bu dizi veri yapısıyla da üç boyutlu sanal küre üzerinde görselleştirme işlemlerini gerçekleştirecek veri görselleştirme paketi bileşenleri tetiklenir.

Veri görselleştirme paketi

Veri görselleştirme paketinin görevi kendisine iletilen JavaScript nesnelerini kullanarak kullanıcı arayüzünde bulunan üç boyutlu sanal küre üzerinde nesneleri görselleştirmektir. Ayrıca kullanıcının gezinme esnasında seçimlerini dinleyerek gerçekleştirilmek istenen işleme göre sahip olduğu yordamları tetikler. Üç boyutlu sanal küre Microsoft'un sunduğu Bing teknolojisi ile birlikte WebGL teknolojileri kullanılarak Cesium firması tarafından geliştirilmiştir. Cesium firması geliştirdiği bu web ortamında çalışan sanal küre üzerine görsel nesnelerin eklenebileceği bir programlama arayüzü de kullanıma sunmuştur. Veri görselleştirme paketi kapsamında gerçekleştirilen işlemler gerçekleştirilme zamanlarına göre ikiye ayrılmaktadırlar. Bu işlemler JavaScript nesnelerinin görselleştirilmesi ve kullanıcı etkileşimi işlemleri olarak ayrılmaktadırlar.

JavaScript nesnelerinin görselleştirilmesi sürecinde paket bileşenlerine iletilen nesne dizisinde bulunan elemanlar kullanılarak üç boyutlu sanal küre üzerine CesiumJS'in

sunduğu programlama arayüzü aracılığıyla eklenir. Ekleme işleminden önce kullanıcı arayüzüne eklenen sanal küre programatik olarak seçilir ve CesiumJS’de bulunan “Scene” isimli JavaScript sınıfı olarak tutulur. İşlemin sebebi eklenecek olan nesnelerin kullanıcıya gösterilmekte olan küreye eklenmesi gerekliliğidir. Ekleme işlemi gerçekleştirebilmek adına CesiumJS’de bulunan “primitive” şekil tipi kullanılmaktadır. Bahsedilen şekil tipini oluşturmak için isim, id, özellik dizisi ve “Polygon” bilgileri gerekmektedir. İsim ve “id” bilgileri olarak GeoJSON elemanlarında bulunan özellik dizisi içerisinde bulunan alanlardan ilgili bilgiler kullanılmaktadır. “Polygon” olarak da yine GeoJSON elemanlarının içerisinde bulunan üç boyutlu noktalardan oluşan koordinat bilgileri kullanılmaktadır. Tüm bilgiler “primitive” şekil tipini oluşturan yordama aktarıldıktan sonra şekil tipi “Scene” nesnesine eklenir ve görselleştirme işlemi kayıt için tamamlanmış olur. Belirtilen işlem nesne dizisinde bulunan tüm elemanlar için gerçekleştirilene kadar da kullanıcı ile etkileşim sistem tarafından engellenir.

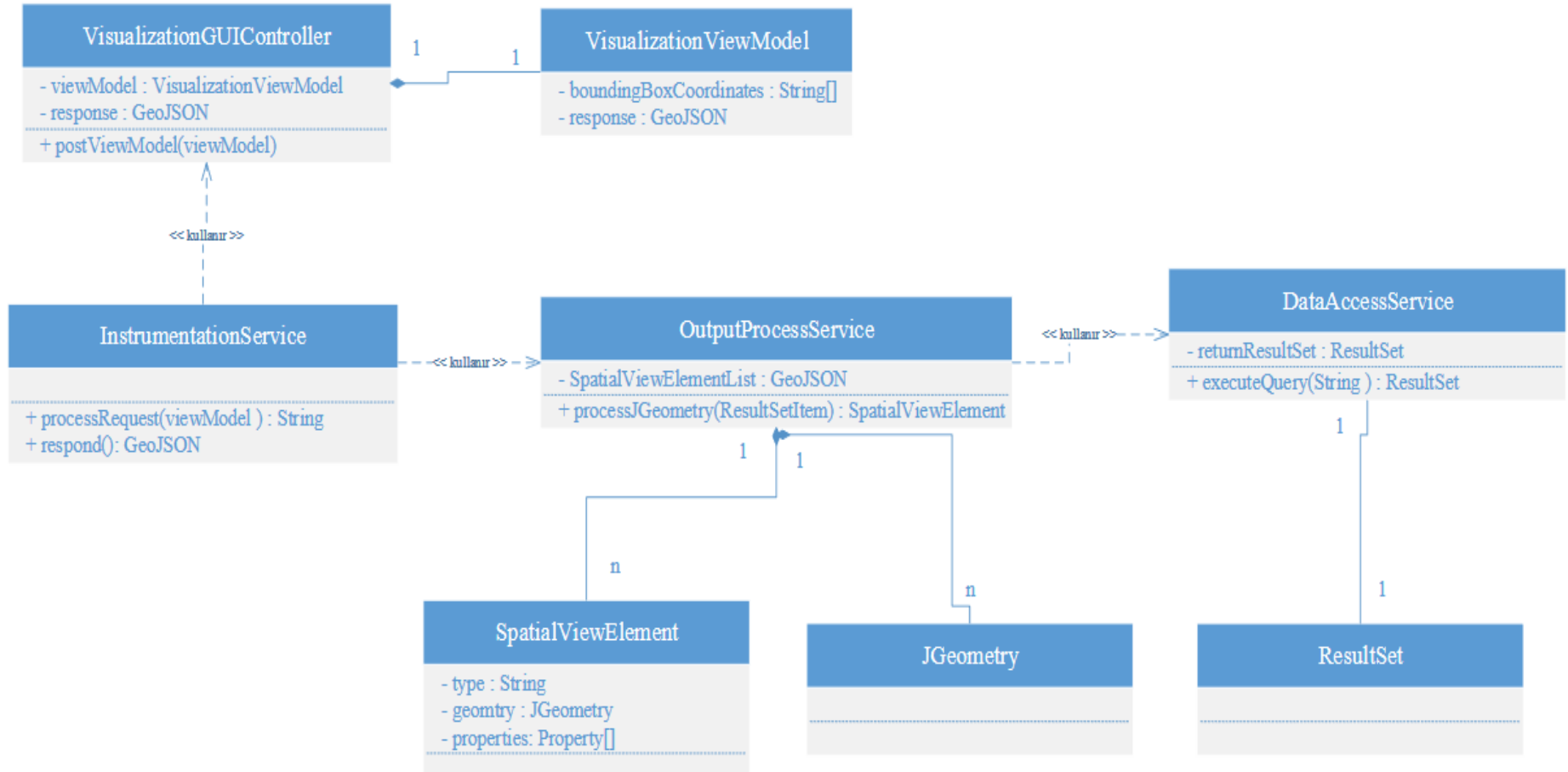
Görselleştirme sistem tarafından gerçekleştirildikten sonra kullanıcıların sistem ile etkileşime geçme kısıtı kaldırılır ve kullanıcı etkileşimin işlemleri süreci başlar. Kullanıcı arayüzünde bulunan sanal kürenin görünüş açısı klavye ve fare aracılığıyla değiştirilerek kullanıcıya gezinme deneyimi sunulur. Kullanıcı aynı cihazlarla girdiler sağlayarak görüş açısını uzaklaştırır ya da yakınlaştırır. Bahsedilen özelliklere ilave olarak kullanıcı görselleştirilmiş nesnelerden birini seçtiği zaman, o nesne ile ilgili bilgilendirme elemanı ekranın sağ üst köşesinden belirir. Bilgilendirme elemanı içerisinde seçilen nesne ile ilgili anlamsal bilgiler bulunmaktadır. Anlamsal veriler iletilen GeoJSON dosyasında bulunan kayıtlar içerisinden, seçilen nesnenin sahip olduğu “id” değeri ile ayrıştırılarak gösterilmektedir. Kullanıcı etkileşime geçmek istediği verilerin bulunduğu alanı değiştirmek istediğinde yeniden sorgulama işlemi yapabilir ve tüm süreç tekrarlanır.

4.2.2 Kullanıcı etkileşim uygulaması sınıf ve sıralanım şemaları

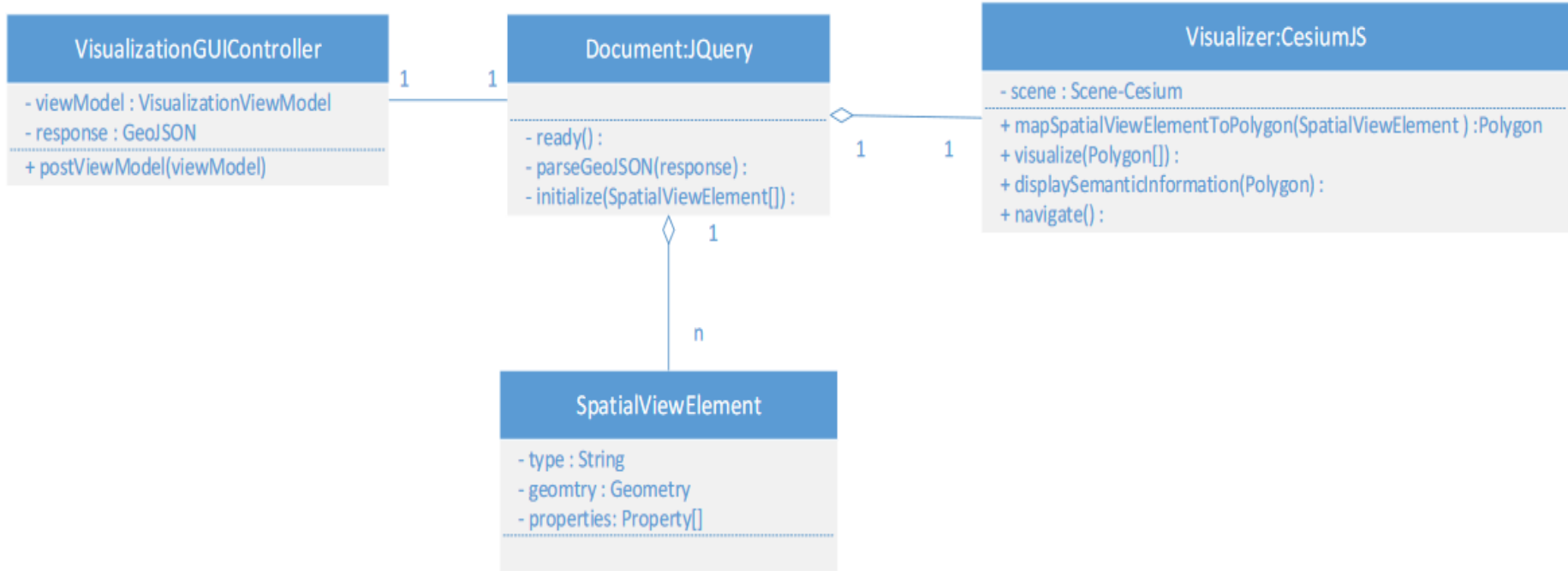
Kullanıcı etkileşim uygulaması kapsamında geliştirilen yazılım bileşenlerinin sınıf ve sıralanım şemaları modellenirken, istemci ve sunucu tarafında bulunan sınıflar ve yazılım bileşenleri ayrı olarak ele alınmıştır. Gerçekleştirilen bu ayrıştırma işleminin ana sebebi ise sınıfların ve sıralanımların beraber çalışmalarına karşın bulundukları süreçlerin ve katmanların farklılaşmasıdır.

Şekil 4.8’de geliştirilen uygulamada kullanılan ve sunucu tarafında bulunan sınıfların belirtildiği sınıf şeması gösterilmektedir. Şekil 4.9’da da uygulamanın istemci tarafında bulunan sınıfların belirtildiği sınıf şeması gösterilmektedir. Belirtilen iki şekilde ortak olarak bulunan tek sınıf “VisualizationGUIController” sınıfıdır. Bahsi geçen sınıf MVC tabanlı uygulama yönetim paketinde bulunmaktadır ve istemci ile sunucu arasında gerçekleştirilen iletişimi sağladığı için iki şemada da bulunmaktadır.

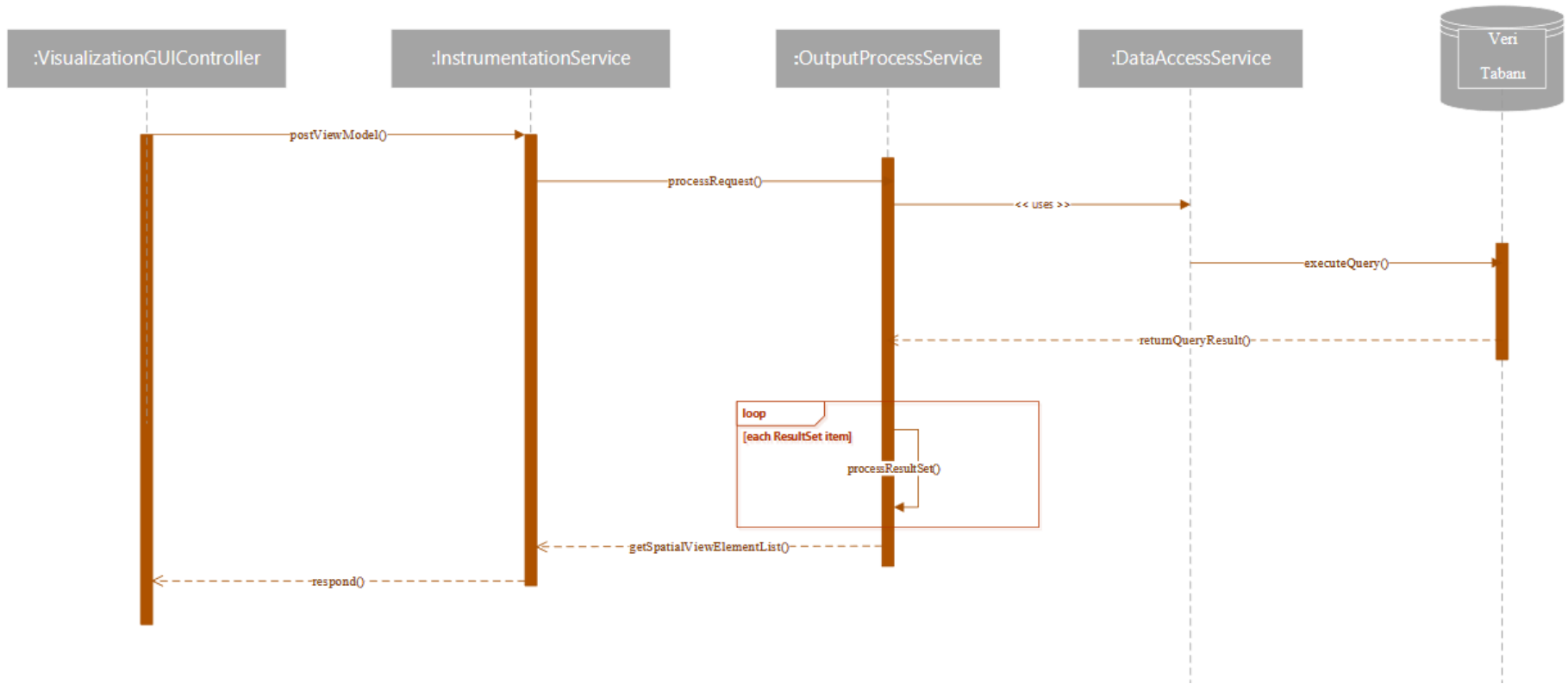
Şekil 4.10’da da sunucu tarafında bulunan yazılım bileşenleri arasında gerçekleşen işlemlerin sıralanım şeması gösterilmektedir. İstemci tarafından iletilen koordinat bilgileriyle sunucu tarafında bir sorgunun gerçekleştirilmesi ve sorgu sonucunun işlenerek cevap olarak tekrardan istemciye iletilmesi sürecinin işlem sıralanımı gösterilmektedir. Şekil 4.11’de de kullanıcı ve istemci tarafında bulunan yazılım bileşenleri arasında gerçekleşen işlemlerin sıralanım şeması gösterilmektedir. Sunucudan cevap olarak iletilen bilgilerin işlenmesi ve görselleştirilmesi işlemlerinin sıralanımı şema kapsamında belirtilmektedir. Görselleştirme süreci tamamlandıktan sonra kullanıcının mouse aracılığıyla gerçekleştirdiği seçme ve gezinme işlemleri de bu şema kapsamında gösterilmiştir. Uygulama kapsamında modellenen tüm şemalarda tez kapsamında görselleştirme üzerine çalışıldığı için sadece mekansal bilgilerin tutulduğu sınıflar gösterilmişlerdir. Anlamsal bilgilerin saklandığı LADM ve GeoJSON standardında bulunan tablolarda tutulan veriler modellenmeseler de bu süreçler içerisinde bulunmaktadır.



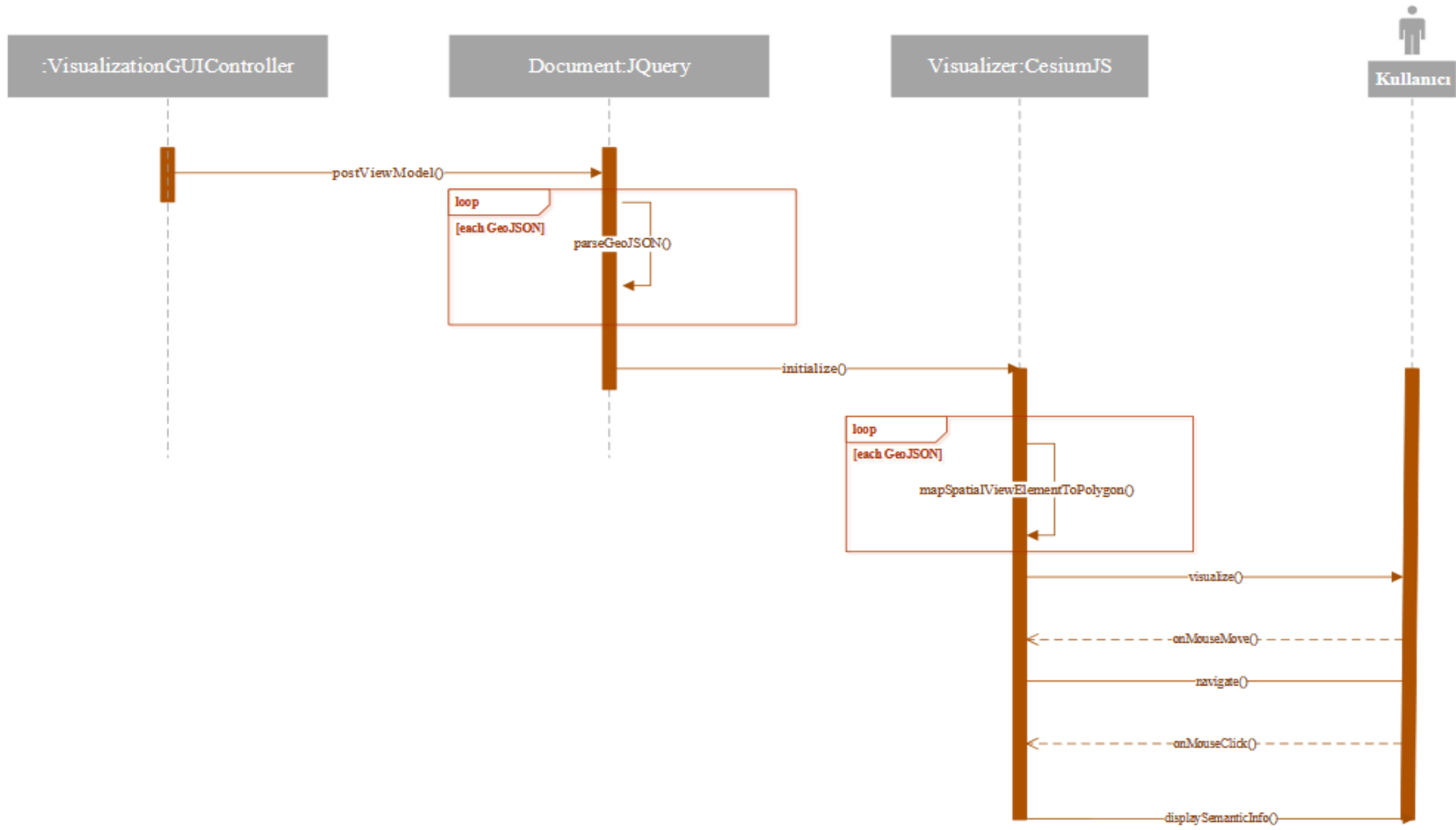
Şekil 4. 8 : Kullanıcı etkileşim uygulaması sunucu tarafı sınıf şeması



Şekil 4. 9 : Kullanıcı etkileşim uygulaması istemci tarafı sınıf şeması



Şekil 4. 10 : Kullanıcı etkileşim uygulaması sunucu tarafı sıralanım şeması



Şekil 4. 11 : Kullanıcı etkileşim uygulaması istemci tarafı sınıf şeması

5. SONUÇLAR VE YORUMLAR

5.1 Kullanıcı Testleri

Tez kapsamında geliştirilen yazılımların kullanılabilirlik kalitelerini ölçmek adına uluslararası standartlardan faydalanılmıştır. ISO tarafından önerilen bir standard olan ISO/IEC 9126 bünyesinde tanımlı olan ve yazılım kullanımlarındaki kalite ölçümlerini gerçekleştirmek için kullanılan ISO/IEC 9126-4 standardı, kullanıcı testlerinin gerçekleştirilmesinde kullanılmıştır.

Standard kapsamında üç ana kalite ölçütü belirlenmiştir bunlar; etkinlik, verimlilik ve kullanıcı memnuniyetidir (ISO - International Organization For Standardization, Software engineering -- Product quality -- Part 4: Quality in use metrics (ISO/IEC 9126-4)), 2004). Etkinlik ölçümünün yapılması için öncelikle, kullanıcılara belirli işleri gerçekleştiren görevler verilmektedir ve bu süreç izlenmektedir. Kullanıcılar; tamamlayabildiği her görev için bir, tamamlayamadığı görevler için de sıfır puan almaktadır. Kullanıcılar, kendilerine verilen tüm görevleri tamamladıklarında ise aldıkları puan değeri verilen görev sayısına bölünür ve sonuç yüz ile çarpılarak yüzdelik bir oran belirlenir. Test kapsamında bulunan tüm kullanıcıların ortalaması hesaplanarak da etkinlik ölçümünün bir kısmı gerçekleştirilmiş olur. Etkinlik ölçümünün diğer kısmı da, test sırasında gözlemlenen hataların toplam sayısı ile gerçekleştirilen görevlerin toplam sayısının bölünmesi ile elde edilmektedir.

$$Etkinlik = \frac{Başarı ile tamamlanan görev sayısı}{Toplam görev sayısı} \times \% 100 \quad (5.1)$$

Verimlilik ölçümünün yapılabilmesi adına iki temel ölçüt vardır bunlar; zaman temelli ve genel göreceli verimlilik ölçütleridir. Zaman temelli verimliliğin ölçülebilmesi için öncelikle görevlerin gerçekleştirilme zamanlarına uygun bir zaman ölçütü belirlenmelidir. Görece basit işlerde saniye ve dakika zaman birimleri kullanılırken zor ve tamamlaması uzun görevlerde dakika ve saat zaman birimleri kullanılmaktadır. Zaman birimi belirlendikten sonra kullanıcı testleri sırasında,

verilen görevler süresince geçen zaman bilgisi kaydedilir. Başarı ile sonuçlanan görevler için, çıkan sonuç bir ile bölünür, başarısız görevler için ise sonuç sıfır olarak hesaplanır. Tüm görevler ve kullanıcılar için elde edilen bu değerler toplanarak, kullanıcı ve görev sayısının çarpımı ile bölünerek zaman temelli verimlilik hesaplanır. Belirtilen hesaplama için gerekli formül aşağıdaki gibidir.

$$\text{Zaman Temelli Verimlilik} = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{N * R} \quad (5.2)$$

n_{ij} = j kullanıcısı tarafından gerçekleştirilen i görevinin sonucu

t_{ij} = j kullanıcısı tarafından gerçekleştirilen i görevinin süresi

N = Görev sayısı

R = Kullanıcı sayısı

Genel göreceli verimliliğin ölçülebilmesi adına da bir formülasyon bulunmaktadır ve ölçülen zaman değerleri kullanılmaktadır. Başarı ile sonuçlanan görevler için, çıkan sonuç ve süre çarpılır, başarısız görevler için ise sonuç sıfır olarak hesaplanır. Tüm görevler ve kullanıcılar için elde edilen bu değerler toplanarak, tüm kullanıcılar için toplam geçen süre ile bölünür ve çıkan sonuç yüz ile çarpılarak zaman temelli verimlilik oranı hesaplanır. Belirtilen hesaplama için gerekli formül aşağıdaki gibidir.

$$\text{Genel Göreceli Verimlilik} = \frac{\sum_{j=1}^R \sum_{i=1}^N n_{ij} t_{ij}}{\sum_{j=1}^R \sum_{i=1}^N t_{ij}} \times \% 100 \quad (5.3)$$

n_{ij} = j kullanıcısı tarafından gerçekleştirilen i görevinin sonucu

t_{ij} = j kullanıcısı tarafından gerçekleştirilen i görevinin süresi

N = Görev sayısı

R = Kullanıcı sayısı

Kullanıcı memnuniyetini ölçebilmek adına görev ve sistem kavramının anket yoluyla değerlendirildiği iki farklı soru seti kullanılmaktadır. Görev bazında memnuniyeti ölçebilmek adına, kullanıcılara verilen her görevi bitirir bitirmez Lewis'in

alışmalarında önerdiği anket soruları sorulmaktadır (Lewis, 1991). Sorular, 1 – “Kesinlikle katılıyorum” ile 7 – “Kesinlikle katılmıyorum” cevapları öleğinde cevaplandırılabilir ya da “Uygulanamaz” diye cevaplandırılabilir. Anket kapsamında sorulan sorular aşığıdaki gibidir.

- Görevi tamamlama kolaylığından memnunum
- Görevi tamamlamak için geçen süreden memnunum
- Görevi tamamlarken gösterilen destek bilgilerinden (çevrimiçi yardım, mesajlar vb.) memnunum

Sistemin genel olarak sağladığı kullanıcı memnuniyetini ölçebilmek adına Brooke’un alışmaları temel alınmıştır (Brooke, 1995). Tüm görevler bittikten sonra kullanıcılara on adet soru sorulur ve sorular, 5 – “Kesinlikle katılıyorum” ile 1 – “Kesinlikle katılmıyorum” cevapları öleğinde cevaplandırılabilir. Anket kapsamında sorulan sorular aşığıdaki gibidir.

- Sistemi sık sık kullanmak istediğimi düşünüyorum
- Sistemi gereksiz yere karmaşık buldum
- Sistemin kullanımının kolay olduğunu düşündüm
- Sistemin kullanabilmek için teknik bir kişinin desteğine ihtiyacım olacağını düşünüyorum
- Sistemdeki çeşitli fonksiyonların iyi entegre olduğunu gördüm
- Sistemde çok fazla tutarsızlık olduğunu düşündüm
- Çoğu insanın bu sistemi çok hızlı bir şekilde kullanmayı öğreneceğini düşünüyorum
- Sistemi kullanmak için çok hantal buldum
- Sistemi kullanırken kendimi çok güvende hissettim
- Sisteme girmeden önce bir çok şey öğrenmem gerekiyordu

Kullanıcı testlerinin gerçekleştirileceği grup seçilirken, kullanıcıların bilgisayar okur yazarlığı ve genel teknik bilgilerinin seviyesi gözetilmiştir. Bilgi seviyesinin gözetilmesinin sebebi ise veri tabanı ve JSON gibi teknolojiler ile ilgili teknik bilgilere sahip olmayan kullanıcıların, sistemi kullanmakta zorluk çekeceğinin öngörülmesidir. Gözetilen bilgi seviyesinin sağlanabilmesi açısından testler, Garanti Teknoloji firmasında çalışan on bilgisayar mühendisi ile yürütülmüştür. Testi gerçekleştirecek kişilere, test öncesinde, alışma kapsamında bulunan konular ile

ilgili genel bilgilendirme yapılmıştır. Bilgilendirmenin sebebi, bilinirliği az olan çalışma konusu ile ilgili temel bilgilerin kullanıcılara sunulmasıdır.

Testler sırasında kullanıcılara, yöntem bölümünde bahsedilen dört kullanım durumunu, veri oluşturma ve doğrulama kullanım durumu birleştirilerek, görev olarak gerçekleştirmeleri istenmiştir. Kullanım durumları ile ortaya çıkarılmış görevler aşağıdaki gibidir.

- Veri setinin oluşturulması ve oluşturulan veri setinin doğrulanması
- Üç boyutlu LADM elemanlarının görselleştirilmesi
- Üç boyutlu LADM elemanlarının görselleştirildiği sahne üzerinde gezinme

Testler sırasında kullanıcıların kullanabilmesi için, doğruluğu önceden kontrol edilmiş GeoJSON dosya ve veritabanı bağlantı bilgileri temin edilmiştir. Verilen bu bilgilere ek olarak kullanıcı eğer birinci ya da ikinci görevi tamamlamaz ise o görevin tamamlandığı bir senaryodan devam etmektedir. Testlerin bu şekilde yapılmasının sebebi tüm görevlerin puanlanabilmesi içindir.

5.2 Kullanıcı Testleri Sonuçları

Kullanıcıların, tüm görevleri gerçekleştirirken aldığı puan, görev sürecinde geçen süre ve yaptıkları hata sayılarını gösteren tablo Tablo 5.1'deki gibidir.

Tablo 5.1 Kullanıcı Testi Sonuçları

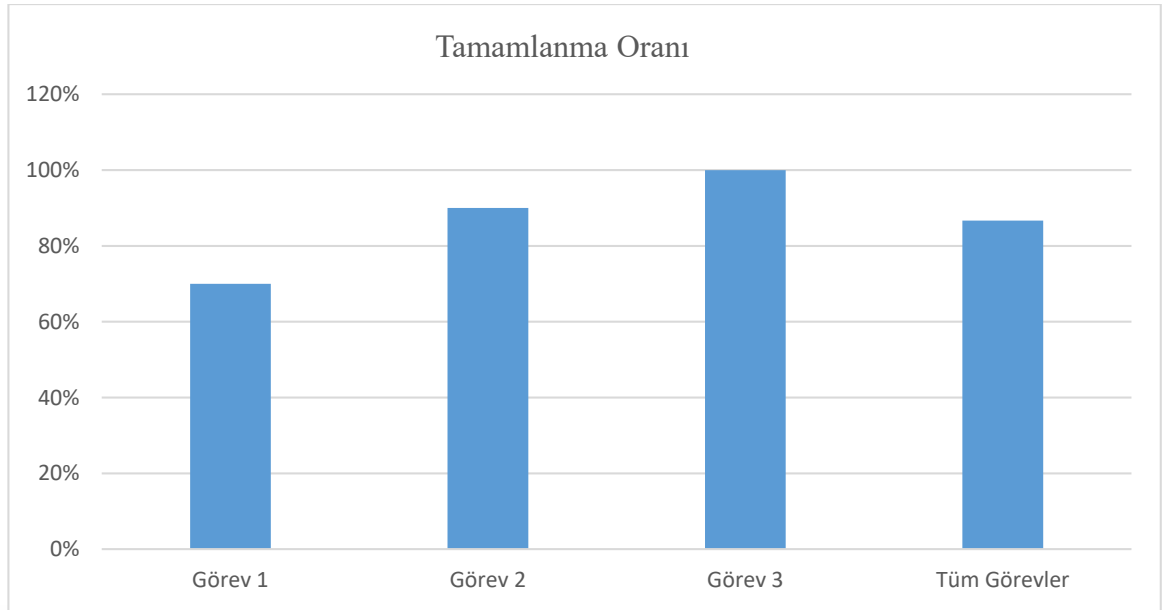
		Görev 1	Görev 2	Görev 3
Kullanıcı 1	Puan	0	1	1
	Süre	65 sn	20 sn	15 sn
	Hata	2	0	0
Kullanıcı 2	Puan	1	1	1
	Süre	54 sn	26 sn	13 sn
	Hata	2	1	0
Kullanıcı 3	Puan	1	1	1
	Süre	55 sn	29 sn	17 sn
	Hata	3	1	0
Kullanıcı 4	Puan	0	1	1
	Süre	56 sn	30 sn	16 sn
	Hata	1	1	0
Kullanıcı 5	Puan	1	1	1
	Süre	45 sn	25 sn	20 sn
	Hata	2	0	0
Kullanıcı 6	Puan	1	1	1
	Süre	48 sn	32 sn	11 sn
	Hata	0	0	0
Kullanıcı 7	Puan	0	0	1
	Süre	76 sn	26 sn	25 sn
	Hata	5	3	2
Kullanıcı 8	Puan	1	1	1
	Süre	67 sn	33 sn	24 sn
	Hata	2	1	0
Kullanıcı 9	Puan	1	1	1
	Süre	54 sn	19 sn	20 sn
	Hata	1	2	0
Kullanıcı 10	Puan	1	1	1
	Süre	46 sn	16 sn	19 sn
	Hata	2	0	0

5.2.1 Etkinlik Testleri

Test sürecinde elde edilen verilen ışığında, geliştirilen yazılımların etkinlik testleri sonuçları aşağıdaki gibidir.

5.2.1.1 Tamamlanma oranı

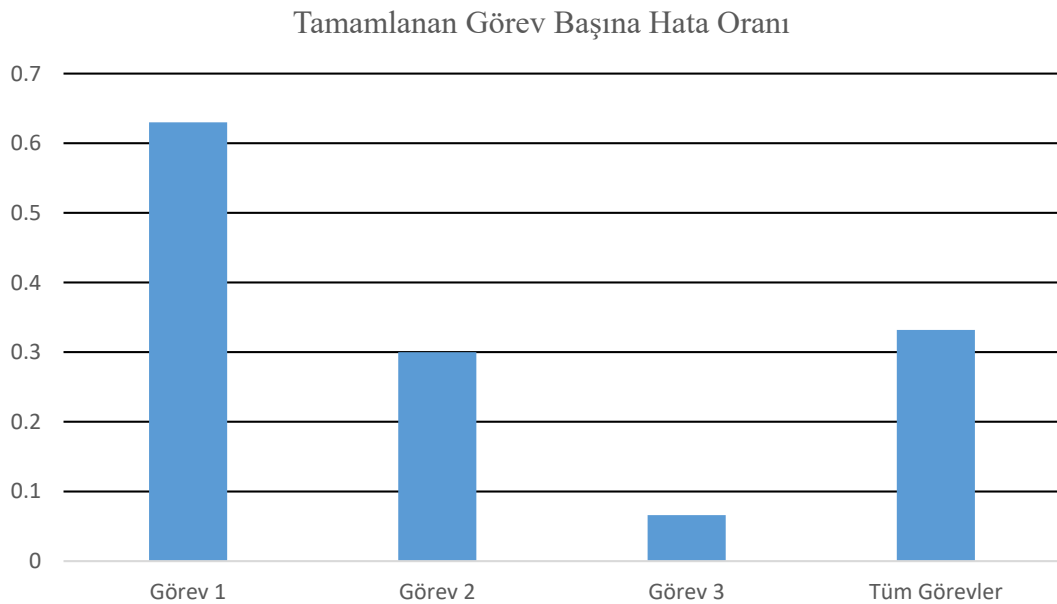
Kullanıcıların gerçekleştirdiği görevlerin tekil ve ortalama tamamlanma oranı aşağıdaki gibidir. En düşük tamamlanma oranına %70 ile birinci görev sahiptir, en yüksek tamamlanma oranına ise %100 ile üçüncü görev sahiptir. Görevlerin ortalama tamamlanma oranı ise %87'dir. Oran ile ilgili bilgiler Şekil 5.1'de gösterilmiştir.



Şekil 5.1 : Tamamlanma Oranı

5.2.1.2 Tamamlanan görev başına hata oranı

Kullanıcıların görevler sürecinde yaptıkları toplam hata sayısının, toplam görev sayısına bölünmesi ile hesaplanmıştır. En yüksek hata oranı her görev için 0.63 hata ile birinci görevde gözlemlenmiştir. En düşük hata oranı ise görev başı 0.066 ile üçüncü görevde gözlemlenmiştir. Tüm görevlerde karşılaşılan ortalama hata oranı ise görev başına 0.332'dir. Görev başına hata oranı ile ilgili bilgiler Şekil 5.2'de gösterilmiştir.



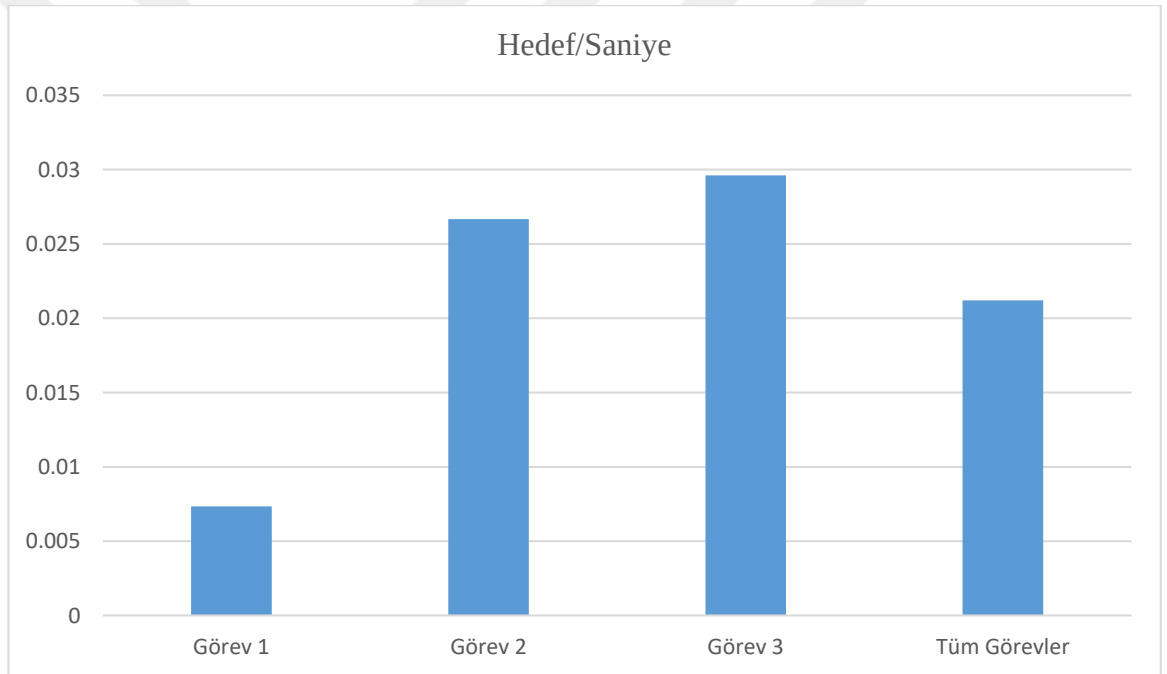
Şekil 5.2 : Tamamlanan Görev Başına Hata Oranı

5.2.2 Verimlilik Testleri

Testler sürecinde alınan puanlar ve geçen zaman ile ilişkili olan verimlilik testlerinin sonuçları aşağıdaki gibidir.

5.2.2.1 Zaman temelli verimlilik

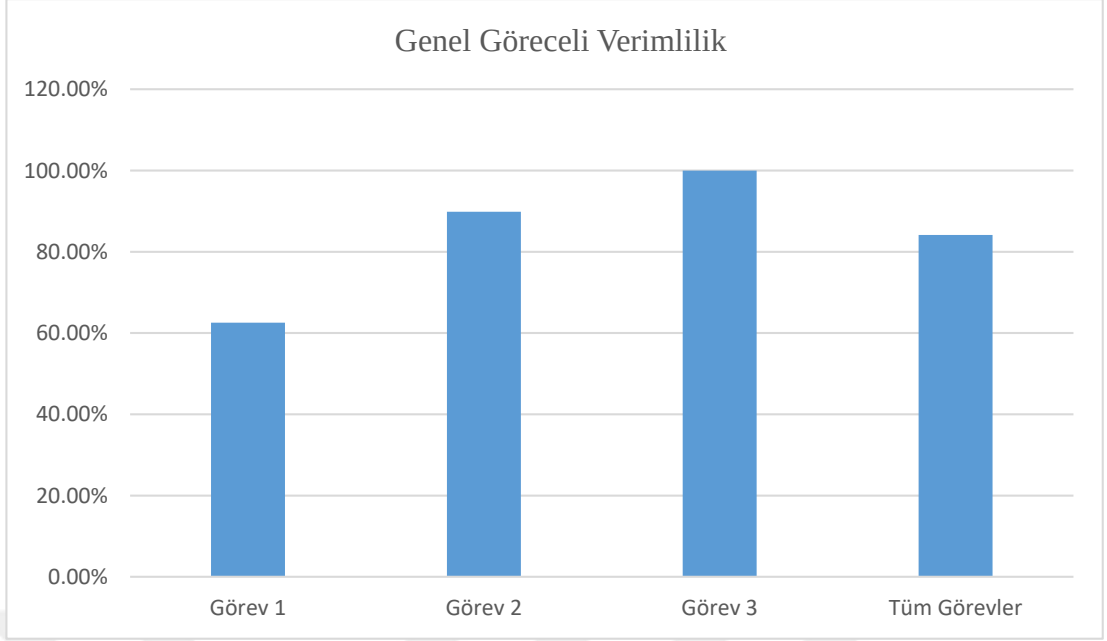
Görevleri gerçekleştirme sürecinde her geçen saniyede ulaşılan hedef sayısının belirlendi ölçümlerde 0.0073365 değerine sahip olan birinci görev en düşük zaman temelli verimliliğe sahiptir. 0.029609 değerine sahip üçüncü görev ise en yüksek zaman temelli verimliliğe sahip görev olmuştur. Tüm görevlerin ortalama zaman temelli verimliliği ise 0.0212046666666667 değerine sahiptir. Zaman temelli verimlilik ile ilgili test sonuçları Şekil 5.3’de gösterilmiştir.



Şekil 5.3 : Hedef/Saniye

5.2.2.2 Genel göreceli verimlilik

Başarılı geçen görevlerin süreleri ve geçen toplam süre bilgileri aracılığıyla hesaplanan genel göreceli verimlilik bakımından en düşük değere %62.55 ile birinci görev sahiptir. Üçüncü görev, %100 genel göreceli verimlilik ile en yüksek değere sahiptir. Tüm görevlerin ortalaması ise %84.13 değerine sahiptir. Genel göreceli verimlilik ile ilgili test sonuçları Şekil 5.4’de gösterilmiştir.



Şekil 5.4 : Genel Göreceli Verimlilik

5.2.3 Kullanıcı Memnuniyeti Testleri Sonuçları

Kullanıcı memnuniyetini görev ve sistem bazlı ölçebilmek adına gerçekleştirilen anketlerin sonuçları Tablo 5.2’ de gösterilmektedir. Kullanıcı bazında her görev ve sistemin tümü için verilen puanlar tabloda gösterilmektedir. Edinilen bilgilerin ışığında görev ve sistem bazlı kullanıcı memnuniyeti değerleri bulunacaktır. Kullanıcı memnuniyeti testleri sonuçları Tablo 5.2’ de gösterilmiştir.

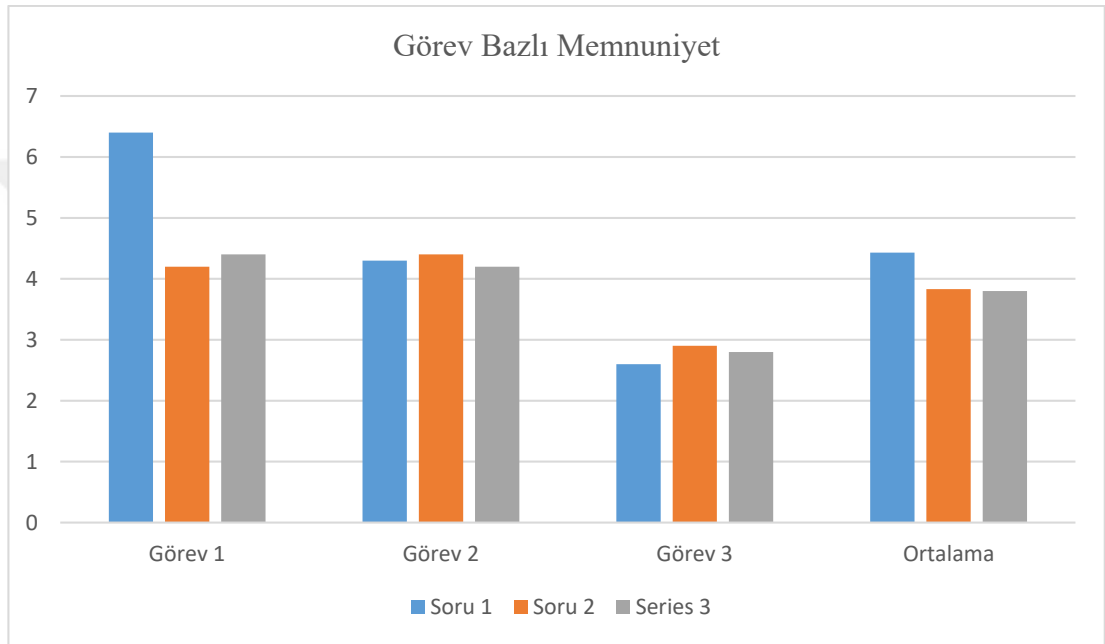
Tablo 5.2 Kullanıcı Memnuniyeti Testi Sonuçları

	Görev 1	Görev 2	Görev 3	Sistem
Kullanıcı 1	7,5,5	6,5,4	3,3,4	1,4,3,3,4,2,1,2,2,5
Kullanıcı 2	7,4,4	5,5,6	2,3,3	2,4,3,4,4,2,2,3,4,4
Kullanıcı 3	6,4,6	4,5,6	3,3,3	3,5,3,4,5,2,3,4,4,5
Kullanıcı 4	6,4,6	4,4,3	4,3,3	3,4,4,5,2,3,3,4,5,5
Kullanıcı 5	7,4,4	4,3,3	2,3,3	2,3,4,4,2,2,2,3,4,4
Kullanıcı 6	7,5,3	5,4,4	2,4,4	1,2,3,3,3,2,2,2,4,4
Kullanıcı 7	6,4,3	4,5,4	2,2,2	1,2,3,3,3,4,4,2,3,3
Kullanıcı 8	6,4,4	3,4,4	4,5,3	2,3,3,2,3,4,5,4,3,2
Kullanıcı 9	5,3,4	4,5,4	2,1,1	2,4,3,2,3,5,4,3,4,5
Kullanıcı 10	7,5,5	4,4,4	2,2,3	1,3,2,3,2,3,4,2,5,5

5.2.3.1 Görev bazlı memnuniyet

Görev bazlı memnuniyeti ölçebilmek adına her görev sonrası kullanıcılara sorulan sorulardan görev bazlı memnuniyeti ölçmek istenmiştir. Görev bazlı kolaylık sorusu

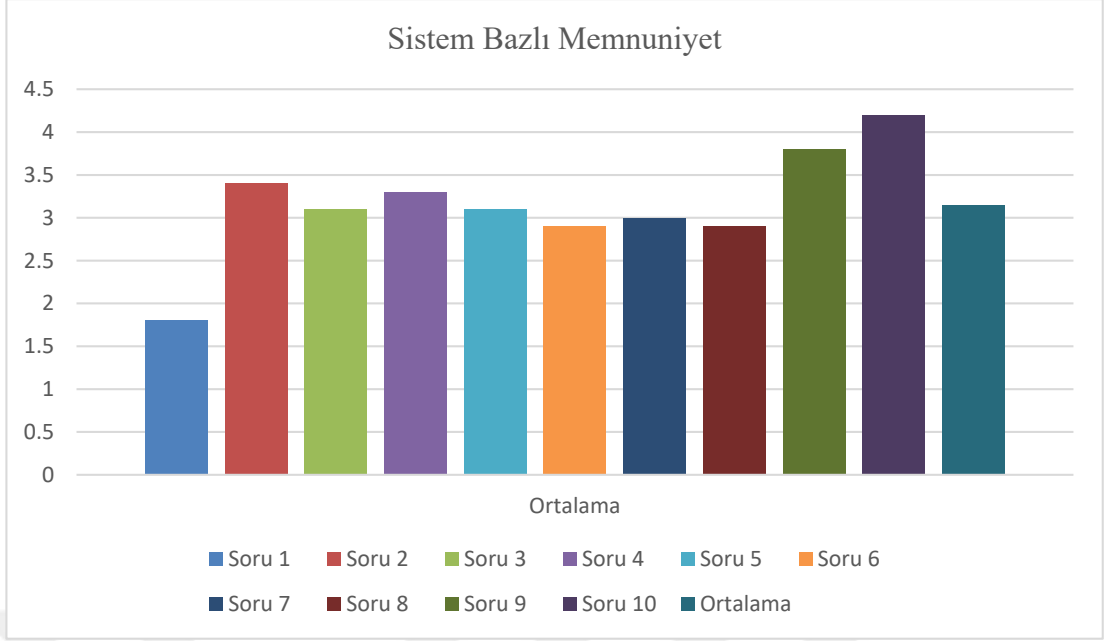
için verilen cevaplardan en düşük puanı birinci görev 6.4 ile en yüksek puanı ise üçüncü görev 2.6 ile almıştır. Ortalama olarak da 4.43 değeri ortaya çıkmıştır. Görev bazlı geçen süre ile alakalı soru için en düşük puanı ikinci görev 4.4 ile en yüksek puanı ise 2.9 ile üçüncü görev almıştır. Ortalama olarak da 3.83 değeri ortaya çıkmıştır. Destek bilgileri ile alakalı soruda ise birinci soruya benzer bir sonuç ortaya çıkmıştır. Verilen cevaplardan en düşük puanı birinci görev 4.4 ile en yüksek puanı ise üçüncü görev 2.8 ile almıştır. Ortalama olarak da 3.8 değeri ortaya çıkmıştır. Görev bazlı memnuniyet testi sonuçları Şekil 5.5’ de gösterilmektedir.



Şekil 5.5 : Görev Bazlı Memnuniyet

5.2.3.2 Sistem bazlı memnuniyet

Sistem bazlı memnuniyeti ölçmek için test sonunda kullanıcılara on adet soru sorulmuştur ve bir ile beş arasında bir puan vermeleri istenmiştir. Bir puanı kesinlikle katılmıyorum, beş puanı ise kesinlikle katılıyorum cevabına denk gelmektedir. Ortalama olarak da tüm sisteme 3.15 puan verilmiştir. Sistem bazlı memnuniyet testi sonuçları Şekil 5.6’ de gösterilmektedir.



Şekil 5.6 : Sistem Bazlı Memnuniyet

5.3 Yorumlar

Gelişen teknoloji ve çeşitlenen bilgiye ulaşma alışkanlıkları, sahip olunan her tip verinin işlenme ve sunulma şekillerini değiştirmektedir. Artan internet kullanıcı sayısı ve bilgi ihtiyacı ile mekansal mülkiyet verilerinin de modellenme, saklanma, işlenme ve sunulma şekilleri değişmektedir. Bu tez çalışmasında, LADM standardı ile tanımlanmış mekansal mülkiyet verisinin, web ortamında bulunan üç boyutlu bir sanal küre üzerinde görselleştirilmesi sağlanmıştır.

Görselleştirme sürecinde karşılaşılan ilk zorluk, görselleştirilecek bir veri setinin bulunmaması olmuştur. Mekansal mülkiyet verileri, ülkelerin tapu ve kadastro ile ilgili kurumları tarafından saklandığı ve mülkiyet hakkı gibi gizli kalması istenebilecek bilgileri içerdiği için elde edilememiştir. Görselleştirmenin buna rağmen gerçekleştirilebilmesi adına veri seti, tez kapsamında geliştirilen web tabanlı bir yazılım aracılığı ile oluşturulmuştur. Veri setini oluşturan yazılımın kullanım alanlarının çeşitlendirilmesi adına da yazılımın çalışması için gerekli olan girdiler ve yazılımın ürettiği çıktılar uluslararası standartlara uygun olarak belirlenmiştir. Karşılaşılan bir diğer zorluk ise, veri setinin saklanması sırasında kullanılması hedeflenen veritabanı yönetim sistemlerinin temel aldıkları ISO standardına uymalarına rağmen benzer bir sözdizimine sahip olmamalarıdır. Sözdiziminin farklılaşması ile hedef veritabanında yaratılacak tablolar ve eklenecek kayıtlar için gerekli komutlar farklı veritabanı yönetim sistemlerinde çalışmamaktadır. Sorunun

giderilebilmesi adına hedef veritabanı yönetim sistemi olarak sadece Oracle 11g kullanılmıştır. İleriki çalışmalarda hedef veritabanına göre komut farklılaştırılması yapılarak, oluşturulan veri seti farklı veritabanı yönetim sistemlerinde çalıştırılabilir.

Veri setinin oluşturulması ile görselleştirme işlemini gerçekleştirecek web yazılımına, kullanıcı isteklerine göre veri iletiminin sağlanması ile alakalı teknik bir sorun ile karşılaşmıştır. LADM standardı çerçevesinde veritabanı yönetim sisteminde saklanan veri, web ortamına iletilirken ve görselleştirirken web teknolojilerinin sağladığı genel programlama arayüzlerine ve standartlarına uymamaktadır. Uyumun gerçekleştirilebilmesi için LADM standardı çerçevesinde saklanan veri, web programcılığında sıkça kullanılan ve web tarayıcıları tarafından etkili bir biçimde işlenebilen GeoJSON biçimine dönüştürülerek istemcilere sunulmuştur. İstemci tarafında gerçekleştirilen görselleştirme işlemlerinde ise karşılaşılan en önemli sorun, performans sorunudur. LADM standardı kapsamında ayrı birer olgu olarak saklanan bina yüzeylerinin sayısı arttıkça, istemci tarafından işlenmeleri ve görselleştirilmeleri, kullanımı zorlaştıracak seviyelerde yavaşlamaktadır.

Tez çalışmasında geliştirilen web uygulamaları için gerçekleştirilen kullanıcı testleri sonucunda etkinlik, verimlilik ve kullanıcı memnuniyeti konularında diğer görevlere nazaran en düşük sonucu birinci görev olan veri seti oluşturma görevi almıştır. Görevin en düşük sonucu almasının nedenlerinden birisi görevin kullanıcıdan ziyade sistem yöneticisi tarafından gerçekleştirilmesi gereken bir görev olması olabilir. Diğer bir sebep ise görevin diğer görevlere göre daha fazla teknik terim bilgisi, girdi ve süre gerektirmesi olabilir. Kullanıcıların testler esnasında en olumlu puanları verdikleri görev ise görselleştirme sahnesi üzerinde etkileşimin gerçekleştirildiği görev olmuştur. Görev sonucunun diğer görevlere göre daha olumlu olmasının en büyük sebebi, süreç boyunca kullanıcıların sadece fare ile işlem yapıp, sonuç olarak gösterilen bilgileri görüntülemeleridir. Görev sürecinde yapılması gereken işlemlerin, genel olarak internet ortamında bulunan harita uygulamalarına olan benzerliği ve kullanıcıların benzer görevlere olan alışkanlığı da görevin olumlu puan almasında etkili olmuştur.

Çalışmanın devamı olarak teknik ve işlev anlamda pek çok yeni çalışma yapılabilmektedir. Teknolojik olarak geliştirilebilecek yönler; veritabanı çeşitliliği getirmek ya da ilişkisel nesne haritalaması yapmak, NOSQL veritabanı ile çalışan bir

sistem geliřtirmek ve bulut üzerinde mikroservis mimarisine d n řt rmek  rnek olarak verilebilir.  řlev olarak da sistem geliřtirilebilmektedir.  rneęin; yeni LADM sınıfları eklenerek farklı iřlemleri destekleyebilecek bir sistem oluřturulabilir. Ayrıca kullanıcılar g sterilen veriler  eřitlendirilebilir.

 alıřma sayesinde LADM, ISO ve modern yazılım standartlarına uyumlu, hedeflenen iřlemleri ger ekleřtirebilen prototip bir yazılım sistemi geliřtirilmiřtir. Geliřtirilen bu yazılım sistemi ile LADM standardının temel alındıęı uygulamalarda kullanılabilecek ve d n řt r lebilecek bir sistem ve tecr be elde edilmiřtir. Sahip olunan bu olgular yardımıyla gelecekte LADM ile ilgili yeni  alıřmaların ger ekleřtirilebilmesi olasıdır.



KAYNAKLAR

- Abdul-Rahman, A., & Pilouk, M. (2008). *Spatial data modelling for 3D GIS*. Berlin: Springer.
- Aien, A., Kalantari, M., Rajabifard, A., Williamson, I., & Wallace, J. (2013). Towards integration of 3D legal and physical objects in cadastral data models. *Land Use Policy*, 35, 140-154.
- Akinci, H., Özalp, A., & Turgut, B. (2013). Agricultural land use suitability analysis using GIS and AHP technique. H. Akinci, A. Özalp, & B. Turgut içinde, *Computers and Electronics in Agriculture* (Cilt 97, s. 71-82).
- Alattas, A., Van Oosterom, P., Zlatanova, S., Hoeneveld, D., & Verbree, E. (2018). Using the combined ladm-indoorgml model to support building evacuation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4), 11-23.
- Alattas, A., Zlatanova, S., Van Oosterom, P., Chatzinikolaou, E., Lemmen, C., & Li, K.-J. (2017). Supporting Indoor Navigation Using Access Rights to Spaces Based on Combined Use of IndoorGML and LADM Models. *ISPRS International Journal of Geo-Information*, 6(12), 384.
- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A Pattern Language: Towns, Buildings, Construction*. New York: Oxford University Press.
- Alkan, M., & Can, E. (2009). Tapu ve Kadastro Bilgi Sisteminin (TAKBİS) Geçmiş ve Gelişim Sürecinin Dünya Perspektifi Bazında İrdelenmesi. *TMMOB Harita ve Kadastro Mühendisleri Odası 12. Türkiye Harita Bilimsel ve Teknik Kurultayı*. Ankara.
- Aremu, A., Brennan-Craddock, J., Panesar, A., Ashcroft, I., Hague, R., Wildman, R., & Tuck, C. (2017, 1 1). A voxel-based method of constructing and skinning conformal and functionally graded lattice structures suitable for additive manufacturing. *Additive Manufacturing*, 13, 1-13.
- Athanasiou, K., Sutherland, M., Kastrisios, C., Tsoulos, L., Griffith-Charles, C., Davis, D., & Dimopoulou, E. (2017). Toward the Development of a Marine Administration System Based on International Standards. *ISPRS International Journal of Geo-Information*, 6(7), 194.
- Autodesk çevrim içi dokümanı. (2014). *Extrude dialog box*. <https://knowledge.autodesk.com/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2014/ENU/Inventor/files/GUID-4BBB5048-E351-49EC-9041-B82A562CD9BA-htm.html> adresinden alındı
- Balsa-Barreiro, J., & Fritsch, D. (2018). Generation of visually aesthetic and detailed 3D models of historical cities by using laser scanning and digital photogrammetry. J. Balsa-Barreiro, & D. Fritsch içinde, *Digital Applications in Archaeology and Cultural Heritage* (Cilt 8, s. 57-64).
- Bertino, E., Ooi, B., Sacks-Davis, R., Tan, K.-L., Zobel, J., Shidlovsky, B., & Catania, B. (1997). *Indexing Techniques for Advanced Database Systems*. Norwell, MA, USA: Kluwer Academic Publishers.
- Biljecki, F., Ledoux, H., & Stoter, J. (2017). Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, 64, 1-18.
- Boissonnat, J., & Teillaud, M. (2010). *Effective computational geometry for curves and surfaces: With 120 figures and 1 table*. Berlin: Springer.
- Brass, P. (2008). *Advanced Data Structures*. New York: Cambridge University Press.

- Brooke, J. (1995, 5). SUS: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- Brown, K., Niswonger, J., & Hester, G. (2001). *Enterprise Java Development with IBM Websphere*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Burggraf, D. (2006). Geography Markup Language. *Data Science Journal*, 5, 178-204.
- Butler, H., Gillies, S., & Schaub, T. (2016). *The GeoJSON Format*. <https://tools.ietf.org/html/rfc7946> adresinden alındı
- Bydłosz, J. (2015). The application of the Land Administration Domain Model in building a country profile for the Polish cadastre. *Land Use Policy*, 49, 598-605.
- Çağdaş, V., Kara, A., Van Oosterom, P., Lemmen, C., Isikdag, Kathmann, R., & Stubkjær, E. (2016). AN INITIAL DESIGN of ISO 19152:2012 LADM BASED VALUATION and TAXATION DATA MODEL. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2W1), 145-154.
- Chenau, A., Murphy, M., Pavia, S., Fai, S., Molnar, T., Cahill, J., . . . Corns, A. (2019). a Review of 3D Gis for Use in Creating Virtual Historic Dublin. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9(9), 249-254.
- Chuan, C. (2011). *Apache Tomcat 7*. https://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_More.html l adresinden alındı
- Colvero, D., Gomes, A., Tarelho, L., Matos, M., & Santos, K. (2018). Use of a geographic information system to find areas for locating of municipal solid waste management facilities. *Waste Management*, 77, 500-515.
- Cormen, Thomas H. and Leiserson, Charles E. and Rivest, Ronald L. and Stein, C. (2009). *Introduction to Algorithms, Third Edition* (3 b.). Cambridge (İngiltere): Mit Press.
- Crockford, D. (2006). *The application/json Media Type for JavaScript Object Notation (JSON)*. <https://tools.ietf.org/html/rfc4627> adresinden alındı
- Dimopoulou, E., & Elia, E. (2013). Towards common basis for 3D Cadastres from legal perspective. *Survey Review*, 45(333), 410-418.
- Dirksen, J. (2014). *Three.js Essentials* (1 b.). Birmingham: Packt Publishing Limited.
- Drobež, P., Fras, M., Ferlan, M., & Lisec, A. (2017). Transition from 2D to 3D real property cadastre: The case of the Slovenian cadastre. *Computers, Environment and Urban Systems*, 62, 125-135.
- Egenhofer, M. (1994). Spatial SQL : A query and presentation language Spatial SQL : A Query and Presentation Language. *Knowledge and Data Engineering, IEEE Transactions on*, 6(July), 86-95.
- Fakhroutdinov, K. (2018). *The Unified Modeling Language*. <https://www.uml-diagrams.org/uml-22-diagrams.html> adresinden alındı
- Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Gallego, A., Calafat, C., Segura, M., & Quintanilla, I. (2019). Land planning and risk assessment for livestock production based on an outranking approach and GIS. *Land Use Policy*, 83, 606-621.

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-oriented Software*. Reading, MA: Addison-Wesley Longman Publishing Co., Inc.
- Garcia-León, J., Ros-Sala, M., García Martín, A., Torres Picazo, M., Cerezo Andreo, F., & Ramallo Asensio, S. (2017). Reconstrucción virtual paleotopográfica de la ciudad histórica de Cartagena (España). *Virtual Archaeology Review*, 8(16), 61.
- Gruver, A. (2018). Gruver, A. Part VI: Objects vs. Fields (and Vector vs. Raster). https://www.e-education.psu.edu/geog486/11_p8.html adresinden alındı
- Gulliver, T., Haanen, A., & Goodin, M. (2017). A 3D Digital Cadastre for New Zealand and the International Opportunity. *ISPRS International Journal of Geo-Information*, 6(11), 375.
- Guo, R., Li, L., Ying, S., Luo, P., He, B., & Jiang, R. (2013). Developing a 3D cadastre for the administration of urban land use: A case study of Shenzhen, China. *Computers, Environment and Urban Systems*, 40, 46-55.
- Hatipoğlu, A. (2014). MVC Nedir? Model-View-Controller Mimarisi. hatipoglu.blogspot.com.tr/2014/03/mvc-nedir-model-view-controller-mimarisi.html adresinden alındı
- Ho, S., Rajabifard, A., Stoter, J., & Kalantari, M. (2013). Legal barriers to 3D cadastre implementation: What is the issue? *Land Use Policy*, 35, 379-387.
- Hugues, J. F., Van Dam, A., McGuire, M., Skylar, D. F., Foley, J. D., Feiner, S. K., & Akeley, K. (2014). *Computer Graphics: Principles and Practice* (2 b.). Upper Saddle River: Addison-Wesley.
- Ingensand, J., Nappes, M., Produit, T., & Chassin, T. (2018). Automated reconstruction of 3d buildings in historic city centers from lidar data and 2d building footprints. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4/W6), 19-23.
- ISO - International Organization For Standardization. (2003). *ISO/IEC 9075-3:2003*. <https://www.iso.org/standard/34134.html> adresinden alındı
- ISO - International Organization For Standardization. (2004). Software engineering - - Product quality -- Part 4: Quality in use metrics (ISO/IEC 9126-4)).
- Jackson, D., & Gilbert, J. (2019). *WebGL 2.0 Specification*. <https://www.khronos.org/registry/webgl/specs/latest/2.0/> adresinden alındı
- Jamali, A., Boguslawski, P., Duncan, E., Gold, C., & Abdul Rahman, A. (2013). Rapid indoor data acquisition for LADM-based 3D cadastre model. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(2W1), 153-156.
- Janecka, K., & Bobíková. (2018). Registering the underground objects in the 3D cadastre: A case study of wine cellar located in the vineyard area Tokaj. *Acta Montanistica Slovaca*, 23, 260-270.
- Janečka, K., & Souček, P. (2017). A Country Profile of the Czech Republic Based on an LADM for the Development of a 3D Cadastre. *ISPRS International Journal of Geo-Information*, 6(5), 143.
- JQuery Foundation. (2019). *JQuery API*. <http://api.jquery.com/> adresinden alındı
- Kalantari, M., Dinsmore, K., Urban-Karr, J., & Rajabifard, A. (2015). A roadmap to adopt the Land Administration Domain Model in cadastral information systems. *Land Use Policy*, 49, 552-564.
- Kalogianni, E., & Dimopoulou, E. (2014). Design of a Model for Managing Public Property in Greece.

- Kalogianni, E., Dimopoulou, E., Quak, W., & Van Oosterom, P. (2017). LADM and Interlis as a perfect match for 3D cadastre. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4W7), 23-26.
- Kennedy, H. (2001). *The ESRI Press Dictionary of GIS Terminology* (1 b.). Redlands, CA: ESRI Press.
- Kim, S., & Heo, J. (2017). Development of 3D underground cadastral data model in Korea: Based on land administration domain model. *Land Use Policy*, 60, 123-138.
- Kitsakis, D., & Dimopoulou, E. (2014). 3D cadastres: legal approaches and necessary reforms. *Survey Review*, 46(338), 322-332.
- Krasner, G., & Pope, S. (1988). A Cookbook for Using the Model-view Controller User Interface Paradigm in Smalltalk-80. *J. Object Oriented Program.*, 1, 26-49.
- Lee, B., Kim, T., Kwak, B., Lee, Y., & Choi, J. (2015). Improvement of the Korean LADM country profile to build a 3D cadastre model. *Land Use Policy*, 49, 660-667.
- Lemmen, C. (2012). *A Domain Model for Land Administration* (Cilt 78). Delft.
- Lemmen, C., & van Oosterom, P. (2003). 3D Cadastres. *Computers, Environment and Urban Systems*, 27(4), 337-343.
- Lewis, J. (1991, 5). Psychometric evaluation of an after-scenario questionnaire for computer usability studies: The ASQ. *SIGCHI Bull.*, 23, 78-81.
- Li, L., Lei, Y., Tang, L., Yan, F., Luo, F., & Zhu, H. (2019). A 3D spatial data model of the solar rights associated with individual residential properties. *Computers, Environment and Urban Systems*, 74, 88-99.
- Li, L., Wu, J., Zhu, H., Duan, X., & Luo, F. (2016). 3D modeling of the ownership structure of condominium units. *Computers, Environment and Urban Systems*, 59, 50-63.
- Longley, P., Goodchild, M., Maguire, D., & Rhind, D. (2005). *Geographic Information Systems and Science* (2 b.). John Wiley & Sons.
- Mark, D., Chrisman, N., Frank, A., McHaffie, P., & Pickles, J. (2018). *The GIS History Project*. http://www.ncgia.buffalo.edu/ncgia/gishist/bar_harbor.html adresinden alındı
- McGovern, J., Adatia, R., Fain, Y., Gordon, J., & Henry, E. (2003). *J2EE 1.4 Bible* (1 b.). Indianapolis, IN: Wiley Pub.
- Meaden, G., & Aguilar Manjarrez, J. (2013). Advances in geographic information systems and remote sensing for fisheries and aquaculture. *FAO. Fisheries Technical Paper*, 111.
- Mika, M., & Jurkiewicz, M. (2018). Legal and technological obstacles on the road to creating the 3D cadastre in Poland. *Acta Scientiarum Polonorum Formatio Circumiectionis*, 2, 135-143.
- Nasorudin, N., Hassan, M., Zulkifli, N., & Rahman, A. (2016). Geospatial database for strata objects based on land administration domain model (LADM). *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4W1), 329-334.
- National Geospatial - Intelligence Agency. (2014). *NGA Geomatics - WGS 84*. National Geospatial - Intelligence Agency: <http://earth-info.nga.mil/GandG/update/index.php?dir=wgs84&action=wgs84> adresinden alındı

- Nurseitov, N., Paulson, M., Reynolds, R., & Izurieta, C. (2009). Comparison of JSON and XML Data Interchange Formats: A Case Study. N. Nurseitov, M. Paulson, R. Reynolds, & C. Izurieta içinde, *Computer Applications in Industry and Engineering (Caine)* (s. 157-162).
- O'Brien, T., Casey, J., Fox, B., Snyder, B., Van Zyl, J., & Redmond, E. (2008). *Maven: The Definitive Guide: The Definitive Guide* (Cilt 4). Sebastopol, CA: O'Reilly.
- Open Geospatial Consortium, I. (2010). *Simple feature access - Part 2: SQL Option*. http://portal.opengeospatial.org/files/?artifact_id=25354%5Cnpapers2://publication/uuid/5D0E556A-2227-49B6-A1EF-35AEFBDE9550 adresinden alındı
- Openshaw, S. (1991). A View on the GIS Crisis in Geography, or, Using GIS to Put Humpty-Dumpty Back Together Again. *Environment and Planning A: Economy and Space*, 23(5), 621-628.
- Oracle çevrim içi dokümanı. *J2EE, Web Services, and Internet Applications*. (2013). https://docs.oracle.com/cd/B14099_19/core.1012/b13994/webstapp.htm adresinden alındı
- Paixao, S., Hespanha, J., Ghawana, T., Carneiro, A., Zevenbergen, J., & Frederico, L. (2015). Modeling indigenous tribes' land rights with ISO 19152 LADM: A case from Brazil. *Land Use Policy*, 49, 587-597.
- Park, Y., & Guldman, J. (2019). Creating 3D city models with building footprints and LIDAR point cloud classification: A machine learning approach. *Computers, Environment and Urban Systems*, 75, 76-89.
- Paulsson, J., & Paasch, J. (2015). The Land Administration Domain Model - A literature survey. *Land Use Policy*, 49, 546-551.
- Polat, Z., & Alkan, M. (2018). Research Article DESIGN SPATIAL AND NON-SPATIAL DATA MODELLING BASED ON THE LADM STRUCTURE FOR LAND REGISTRY AND CADASTRAL SYSTEM IN TURKEY. *Sigma Journal of Engineering and natural Sciences*, 36(4), 1081-1095.
- Pouliot, J., Vasseur, M., & Boubehrezh, A. (2013). How the ISO 19152 Land Administration Domain Model performs in the comparison of cadastral systems: A case study of condominium/co-ownership in Quebec (Canada) and Alsace Moselle (France). *Computers, Environment and Urban Systems*, 40, 68-78.
- Prato, C., Kaplan, S., Patrier, A., & Rasmussen, T. (2019). Integrating police reports with geographic information system resources for uncovering patterns of pedestrian crashes in Denmark. *Journal of Transport Geography*, 74, 10-23.
- QGIS çevrim içi dokümanı. (2018). *Raster Data*. https://docs.qgis.org/2.8/en/docs/gentle_gis_introduction/raster_data.html adresinden alındı
- Ramsey, E. (2017). Virtual Wolverhampton: Recreating the Historic City in Virtual Reality. *International Journal of Architectural Research: ArchNet-IJAR*, 11(3), 42.
- Rashid, M., Newton, M., Hoque, M., & Sattar, A. (2013). Mixing Energy Models in Genetic Algorithms for On-Lattice Protein Structure Prediction. *BioMed Research International*, 2013(924137), 1-15.
- Rumbaugh, J., Jacobson, I., & Booch, G. (2005). *The Unified Modeling Language Reference Manual*. Boston: Addison-Wesley.
- Sahin, C., Alkis, A., Ergun, B., Kulur, S., Batuk, F., & Kilic, A. (2012). Producing 3D city model with the combined photogrammetric and laser scanner data in the example of Taksim Cumhuriyet square. C. Sahin, A. Alkis, B. Ergun, S.

- Kulur, F. Batuk, & A. Kilic içinde, *Optics and Lasers in Engineering* (Cilt 50, s. 1844-1853).
- Samet, H. (1994). *The Design and Analysis of Spatial Data Structures*. Reading, Mass: Addison-Wesley.
- Savini, L., Tora, S., Di Lorenzo, A., Cioci, D., Monaco, F., Polci, A., . . . Conte, A. (2018). A web geographic information system to share data and explorative analysis tools: The application to West Nile disease in the Mediterranean basin. *PLoS ONE*, 13(6).
- Shojaei, D., Olfat, H., Briffa, M., & Rajabifard, A. (2017). 3D DIGITAL CADASTRE JOURNEY in Victoria, Australia. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(4W5), 117-123.
- Shojaei, D., Olfat, H., Rajabifard, A., & Briffa, M. (2018). Design and Development of a 3D Digital Cadastre Visualization Prototype. *ISPRS International Journal of Geo-Information*, 7(10), 384.
- Smart, M., & Priebbenow, R. (2018). Designing a 3D Cadastral System Demonstrator : A Case Study Designing a 3D Cadastral System Demonstrator : A Case Study. M. Smart, & R. Priebbenow içinde, *6th International FIG 3D Cadastre Workshop* (s. 1-16). Delft: Hollanda.
- Spring çevrim içi dokümanı. (2004). *Introduction to Spring Framework*. <https://docs.spring.io/spring/docs/3.0.0.M4/reference/html/ch01s02.html> adresinden alındı
- Tapu ve Kadastro Genel Müdürlüğü. (2005). *Tapu ve Kadastro Bilgi Sistemi (TAKBİS)*. <https://www.tkgm.gov.tr/tr/sayfa/tapu-ve-kadastro-bilgi-sistemi-takbis> adresinden alındı
- The Open Geospatial Consortium (OGC). (2007). *Geography Markup Language çevrim içi dokümanı*. <http://www.opengeospatial.org/standards/gml> adresinden alındı
- Ting Ba, L., Candidate, P., & Williamson, I. (2000, 5). Land Administration and Cadastral Trends: The Impact of the Changing Humankind-Land Relationship and Major Global Drivers.
- Tomlinson, R. (1974). Geographical Information Systems, Spatial Data Analysis and Decision Making in Government. R. Tomlinson içinde, *Research Tools in Natural Resource and Environmental Economics* (s. 377-401).
- Tsiliakou, E., Labropoulos, T., & Dimopoulou, E. (2013). Transforming 2D cadastral data into a dynamic Smart 3D model. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, XL-2/W2(2), 105-111.
- Vinod Kumar, T. (2014). *Geographic Information System for Smart Cities*. Sahibabad: Ghaziabad: Opal Publishing Group.
- Vučić, N., Mađer, M., Roić, M., & Vranić, S. (2017). Towards a Croatian 3D cadastre based on the LADM. *4th International Workshop on Geoinformation Science: GeoAdvances 2017*, 4(4), 399-409.
- Vukotic, A., & Goodwill, J. (2011). *Apache Tomcat 7*. Berkeley, CA: Apress.
- Yeung, A., & Hall, G. (2007). *Spatial database implementation and project management*. Dordrecht: Springer.
- Yu, C., Li, L., He, B., Zhao, Z., & Li, X. (2017). LADM-based modeling of the unified registration of immovable property in China. *Land Use Policy*, 64, 292-306.

- Zardiny, A., & Hakimpour, F. (2015). 3D Web Services for Visualization and Data Sharing in 3D Cadastre. *International Journal of 3-D Information Modeling*, 4(4), 1-15.
- Zlatanova, S., Van Oosterom, P., Lee, J., Lic, K., & Lemmen, C. (2016). LADM and INDOORGML for SUPPORT of INDOOR SPACE IDENTIFICATION. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4(2W1), 257-263.
- Zulkifli, N., Abdul Rahman, A., & Van Oosterom, P. (2015). An overview of 3D topology for LADM-based objects. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 40(2W4), 71-73.
- Zulkifli, N., Abdul Rahman, A., van Oosterom, P., Tan, L., Jamil, H., Teng, C., . . . Chan, K. (2015). The importance of Malaysian Land Administration Domain Model country profile in land policy. *Land Use Policy*, 49, 649-659.
- Zulkifli, N., Rahman, A., & Hassan, M. (2016). Design of 3D topological data structure for 3d cadastre objects. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 42(4W1), 325-327.

EKLER

EK A: Veri Oluřturma Uygulaması Önyüzü

Veri Oluřturma Formu

Kat Yükseklięi

Kat Sayısı

0

Kaynak GeoJSON Dosya Konumu

Browse...

No file selected.

Aktarım Tipi

☒ Veri Tabanı

☐ Dosya

Hedef Veritabanı Adresi

Hedef Veritabanı Kullanıcı Adı

Hedef Veritabanı Kullanıcı Şifresi

Veri Oluřtur

Şekil A.1 : Veri Tabanı Bağlantısı Hedefine Veri Oluřturma Önyüzü

Veri Oluřturma Formu

Kat Yükseklięi

Kat Sayısı

0

Kaynak GeoJSON Dosya Konumu

Browse...

No file selected.

Aktarım Tipi

☐ Veri Tabanı

☒ Dosya

Hedef Dosya Konumu

Veri Oluřtur

Şekil A.2 : Hedef Dosyaya Veri Oluřturma Önyüzü