



# Web Tabanlı Coğrafi Bilgi Sistemi

Yazılım Mühendisliği Ana Bilim Dalı  
Tezsiz Yüksek Lisans Bitirme Projesi Raporu

Hasan İleri  
Y210240003

Proje Danışmanı: Doç. Dr. Aytuğ Onan

Ocak 2023

İzmir Kâtip Çelebi Üniversitesi Fen Bilimleri Enstitüsü öğrencisi **Hasan İleri** tarafından hazırlanan **Web Tabanlı Coğrafi Bilgi Sistemi** başlıklı bu çalışma tarafımızca okunmuş olup, yapılan savunma sınavı sonucunda kapsam ve nitelik açısından başarılı bulunarak jürimiz tarafından YÜKSEK LİSANS BİTİRME PROJESİ olarak kabul edilmiştir.

**ONAYLAYANLAR:**

**Proje Danışmanı:** **Doç. Dr. Aytuğ Onan**  
İzmir Kâtip Çelebi Üniversitesi

**Proje Eş-danışmanı:**

**Jüri Üyeleri:**

**Savunma Tarihi:**

# Yazarlık Beyanı

Ben Hasan İleri, başlığı Web Tabanlı Coğrafi Bilgi Sistemi olan bu projemin ve proje raporunda sunulan bilgilerin şahsıma ait olduğunu beyan ederim. Ayrıca:

- Bu çalışmanın bütünü veya esası bu üniversitede Yüksek Lisans derecesi elde etmek üzere çalıştığım süre içinde gerçekleştirilmiştir.
- Daha önce bu projenin herhangi bir kısmı başka bir derece veya yeterlik almak üzere bu üniversiteye veya başka bir kuruma sunulduysa bu açık biçimde ifade edilmiştir.
- Başkalarının yayımlanmış çalışmalarına başvurduğum durumlarda bu çalışmalara açık biçimde atıfta bulundum.
- Başkalarının çalışmalarından alıntıladığımda kaynağı her zaman belirttim. Proje raporunun bu alıntılar dışında kalan kısmı tümüyle benim kendi çalışmamdır.
- Kayda değer yardım aldığım bütün kaynaklara teşekkür ettim.
- Projede başkalarıyla birlikte gerçekleştirilen çalışmalar varsa onların katkısını ve kendi yaptıklarımı tam olarak açıkladım.

Tarih: 05.01.2023

---

# Web Tabanlı Coğrafi Bilgi Sistemi

## ÖZ

Projede coğrafi bilgi sistemleri ve web teknolojileri üzerine araştırma yapılarak, coğrafi bilgi sistemi yazılımlarının kullanım alanları ve yazılım teknolojilerine dair bilgiler raporda sunulmuştur. Uygulama projesi olarak web tabanlı bir coğrafi bilgi sistemi gerçekleştirilmiştir. Uygulama projesinde Bing harita sağlayıcısı tarafından sağlanan uydu haritası altlık olarak kullanılmıştır. Uygulama projesinde kullanıcıya harita üzerinde çizim yapma ve veri tabanına kaydetme imkânı sunulmuştur. Bunun yanı sıra kullanıcıya veri tabanında bulunan coğrafi verilerin harita üzerinde doğru konumda gösterilmesi sağlanmıştır. Uygulama üzerinde kullanıcının eklediği verilerin birbiriyle olan konumsal ilişkilerine dair analizler sunulmuştur. Uygulama senaryosunda mahalle ve kapı çizimi yapan kullanıcı mahalle sınırları içine çizilen kapının öznitelik bilgilerinde mahalle yapılan coğrafi analiz sonucu gelmektedir. Ayrıca kullanıcıya harita üzerinde uzunluk ve alan ölçme imkânı da sağlanmıştır. Kullanıcının harita üzerinden herhangi bir konumsal veriyi seçerek konumsal verinin öznitelik bilgilerine erişme imkânı sağlanmış bunun yanı sıra harita bulunan görünüm yüksek çözünürlük içeren pdf çıktı alma ve geojson formatındaki coğrafi verilerin harita üzerinde gösterilmesi imkânı sağlanmıştır. Uygulama projesi asp.net teknolojileri kullanılarak C#, Javascript, Html gibi diller kullanılarak geliştirilmiştir. Ayrıca projede jquery, openlayers, turfjs, pdfjs kütüphaneleri kullanılmıştır.

**Anahtar Sözcükler:** asp.net, c#, jquery, turfjs, openlayers, pdfjs

# Web Based Geographic Information System Software

## Abstract

In the project, research was conducted on geographic information systems and web technologies, and information on the usage areas of geographic information system software and software technologies were presented in the report. A web-based geographic information system was realized as an application project. In the application project, the satellite map provided by Bing map provider was used as a base. In the application project, the user has the opportunity to draw on the map and save it to the database. In addition, the user is provided with the ability to display the geographical data in the database in the correct position on the map. Analyzes of the spatial relationships between the data added by the user on the application are presented. In the application scenario, the user who draws a neighborhood and a door comes as a result of the geographical analysis of the neighborhood in the attribute information of the door drawn within the neighborhood boundaries. The user is also provided with the opportunity to measure length and area on the map. The user has the opportunity to access the attribute information of the spatial data by selecting any spatial data on the map, as well as the ability to print out the view with the map in high resolution pdf and display geographic data in geojson format on the map. The application project was developed using asp.net technologies using languages such as C#, Javascript, Html. In addition, jquery, openlayers, turfjs, pdfjs libraries were used in the project.

**Keywords:** asp.net, c#, jquery, turfjs, openlayers, pdfjs

# Teşekkür

Bitirme projesi çalışmalarında bana zaman yaratan değerli eşim Gülşah İleri ve tecrübeleri ile yolumu aydınlatan takım arkadaşlarım Enerjisa Spatial Force ve GISM Ekiplerine teşekkür ederim.

# İçindekiler

Yazarlık Beyanı .....	iii
Öz .....	iv
Abstract .....	vi
Teşekkür .....	vi
Şekiller Listesi.....	ix
Tablolar Listesi.....	xi
Kısaltmalar Listesi .....	xii
Semboller Listesi.....	xiii
<b>1 Giriş .....</b>	<b>1</b>
<b>2 Konumsal Veri Tipleri, Saklanması ve Analizi .....</b>	<b>4</b>
2.1 Konumsal Veri Tipleri .....	4
2.2 Konumsal Verinin Saklanması .....	6
2.3 Konumsal Veri Analizi .....	7
<b>3 Web tabanlı Coğrafi Bilgi Sistemlerinin Temel Çalışma Prensipleri .....</b>	<b>9</b>
<b>4 Uygulama projesi Altyapısı ve Kullanımı .....</b>	<b>11</b>
4.1 Uygulama Projesi Altyapısı .....	11
4.2 Uygulama Kullanımı.....	16
4.2.1 Uygulama Kullanım Diyagramları.....	17
4.2.1.1 Sisteme Giriş UC-1 .....	17
4.2.1.2 Harita Yakınlaşma, Uzaklaşma UC-2 .....	18
4.2.1.3 Mahalle Çizimi UC-3.....	19
4.2.1.4 Kapı Çizimi UC-4 .....	20
4.2.1.5 Adres Listeleme UC-5 .....	21
4.2.1.6 Adres Listesinden Konuma gitme UC-6 .....	22
4.2.1.7 Harita Üzerinden Bilgi Alma UC-7 .....	23
4.2.1.8 Kapı Bilgilerini Güncelleme UC-8 .....	24
4.2.1.9 Mahalle Bilgilerini Güncelleme UC-9 .....	25
4.2.1.10 Kapı Silme UC-10 .....	26
4.2.1.11 Mahalle Silme UC-11 .....	27
4.2.1.12 Harita Temizleme UC-12 .....	28
4.2.1.13 Pdf Çıktı Alma UC-13 .....	29
4.2.1.14 Uzunluk Ölçme UC-14 .....	30
4.2.1.15 Alan Ölçme UC-15 .....	31

4.2.1.16 İlk görünümüne gitme UC-16 .....	32
4.2.1.17 Eller serbest modu UC-17.....	32
4.2.1.18 Geojson Formatında Dosya Yükleme UC-18 .....	33
4.2.1.19 Koordinat Bilgisi görme UC-19 .....	34
4.2.1.20 Sistemden Çıkış UC-20 .....	34
<b>Kaynaklar .....</b>	<b>35</b>
<b>Özgeçmiş .....</b>	<b>36</b>



# Şekiller Listesi

Şekil 1.1	CBS Tarihsel Süreci .....	2
Şekil 2.1.1	LineString .....	4
Şekil 2.1.2	CircularString.....	4
Şekil 2.1.3	CompoundCurve .....	5
Şekil 2.1.4	Poligon .....	5
Şekil 2.1.5	CurvePolygon.....	5
Şekil 2.1.6	MultiLineString.....	6
Şekil 2.1.7	MultiPolygon.....	6
Şekil 3.1	Açılır Pencere .....	9
Şekil 3.2	Geoserver .....	10
Şekil 3.3	Web Cbs Çalışma Şeması .....	10
Şekil 4.1.1	MVC.....	11
Şekil 4.1.2	Örnek Model.....	12
Şekil 4.1.3	Openlayers Kütüphanesi kod örneği .....	13
Şekil 4.1.4	Turfjs kod örneği .....	13
Şekil 4.1.5	Veri tabanı şeması.....	13
Şekil 4.1.6	Ado.net örnek kodu .....	14
Şekil 4.1.7	Giriş Ekranı .....	14
Şekil 4.1.8	Ana ekran .....	15
Şekil 4.1.9	Çalışma Şeması .....	15
Şekil 4.2.1.1.1	Sisteme Giriş.....	17
Şekil 4.2.1.1.2	Harita Yakınlaşma, Uzaklaşma.....	18
Şekil 4.2.1.1.3	Mahalle çizimi .....	19
Şekil 4.2.1.1.4	Kapı çizim.....	20
Şekil 4.2.1.1.5	Adres Listeleme .....	21
Şekil 4.2.1.1.6	Adres Listesinden konuma gitme.....	22
Şekil 4.2.1.1.7	Harita Üzerinden Bilgi Alma .....	23
Şekil 4.2.1.1.8	Kapı Bilgilerini Güncelleme .....	24

Şekil 4.2.1.1.9	Mahalle Bilgilerini Güncelleme.....	25
Şekil 4.2.1.1.10	Kapı Silme .....	26
Şekil 4.2.1.1.11	Mahalle Silme .....	27
Şekil 4.2.1.1.12	Harita Temizleme.....	28
Şekil 4.2.1.1.13	Pdf Çıktı Alma .....	29
Şekil 4.2.1.1.14	Uzunluk Ölçme .....	30
Şekil 4.2.1.1.15	Alan Ölçme .....	31
Şekil 4.2.1.1.16	İlk görünümüne gitme.....	32
Şekil 4.2.1.1.17	Eller Serbest .....	32
Şekil 4.2.1.1.18	Geojson Dosya yükleme .....	33
Şekil 4.2.1.1.19	Koordinat Görme .....	34
Şekil 4.2.1.1.20	Sistemden Çıkış .....	34

# Tablolar Listesi

Tablo 4.2.1 Kullanım Durum Diyagramları .....	16
---	----

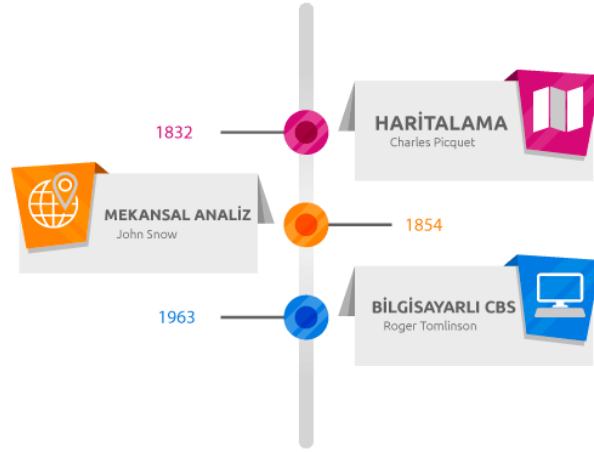
# Kısaltmalar Listesi

m	Metre
km	Kilometre
cbs	Coğrafi bilgi sistemleri
mvc	Model view controller
wms	Web Map Service
wfs	Web Feature Service

# Bölüm 1

## Giriş

Coğrafi Bilgi Sistemleri (CBS), yeryüzüne ait bilgileri belirli bir amaca yönelik olarak toplama, bilgisayar ortamında depolama, güncelleştirme, kontrol etme, analiz etme ve görüntüleme gibi işlemlere olanak sağlayan bir karar destek sistemidir. CBS alanında yapılmış uygulamaların ve CBS başlangıcının ilk olarak, 1832 yılında Charles Picquet'in Paris'in 48 ilçesinde kolera salgınını haritalaması, ardından 1854 yılında İngiliz Doktor John Snow'un Londra'da kolera nedeniyle artan ölümleri göstermek için aynı prensipte harita oluşturarak mekânsal olarak salgının kaynağını tespit edip sorunun önüne geçmesi şeklinde yer almaktadır. CBS kavramı ilk kez 1960'lı yılların başında tanıtılıp ilerleyen yıllarda yeni bir disiplin olarak araştırıldı ve geliştirildi. Roger Tomlinson doğal kaynakların yönetilebilir envanterinin oluşturulması için Kanada hükümeti tarafından görevlendirdi. Kanada'nın tüm doğal kaynak verilerini birleştirebilmek için bilgisayar kullanmayı düşünüyordu ve Kanada'nın ulusal arazi kullanım yönetim programına başlamasını sağlayan, büyük boyutlardaki veriyi depolayıp işleyebilmek için otomatik bilgi işlem sistemini yarattı. Bu sistemi CBS olarak adlandırdı. Roger Tomlinson'ın Kanada Coğrafi Bilgi Sistemleri'ni başlatmak, planlamak ve geliştirmek için yaptığı çalışmalar 1963'te ilk bilgisayarlı CBS ile sonuçlandı. Bu süreci 80'lerin başından itibaren piyasaya sürülen ticari masaüstü yazılımları izledi. Günümüzde ise özellikle bilgisayar teknolojisinde yaşanan gelişmelere bağlı olarak Coğrafi Bilgi Sistemleri ve teknoloji'lerinde web ve mobil platformlarda da büyük gelişmeler olmuştur. CBS Türkiye pazarında 1990'lı yıllarda özellikle Windows 95 sonrası masaüstü CBS programlarıyla yaygınlaşmıştır. 2000 yılı sonrasında kurumsal CBS sistemleri yayılmaya başlanmıştır. (Başarsoft 2022)



Şekil 1.1 CBS Tarihsel Süreci (Başarsoft 2022).

Web tabanlı coğrafi bilgi sistemleri gelişen internet teknolojileri sayesinde masaüstü yazılımların yerini hızla almaktadır. Web tabanlı çözümler zaman ve maliyet olarak kullanıcılara avantajlar sağlamıştır. Öncelikli olarak sadece bilgi gösterimi ve analiz için kullanılan web tabanlı coğrafi bilgi sistemleri günümüzde coğrafi veri ekleme, silme, düzenleme işlevlerini de gerçekleştirebilecek düzeye gelmiştir. Enerji sektörü envanter takibi ve analizi, askeri simülasyon ve tematik, kent bilgi sistemleri, arazi bilgi sistemleri, araç takip, tapu ve kadastro sistemleri, Trafik bilgi sistemi vb. birçok alanda web tabanlı coğrafi bilgi sistemlerinin kullanıldığını görebiliriz. Web tabanlı coğrafi bilgi sistemleri temel olarak konumsal verinin kullanıcıya web tarayıcısında gösterilmesi ve düzenlenmesi görevini görür. Yapısal olarak genellikle veri tabanı, uygulama ve kullanıcı katılımcılarından oluşur. Bu proje türüne göre farklılar gösterebilir. Bilgi teknolojilerinin gelişimine paralel olarak coğrafi bilgi sistemleri içinde birçok üçüncü parti yazılımlar ve kütüphaneler geliştirilmiş ve günümüzde geliştirilen coğrafi bilgi sistemi yazılımları bu yazılımlardan ve kütüphanelerden istifade edilerek geliştirilmektedir.

Klasik CBS’de elde edilen verilerin tek bir donanım (bilgisayarda) ortamında tutulması ve bunun için de güçlü bir yazılım ve donanıma ihtiyaç duyulması, en önemli dezavantaj olarak kullanıcıların karşısına çıkmaktadır. Klasik CBS’deki bu dezavantajlı durum, WEB CBS’de feature ve harita servislerin internet tabanıyla kullanıcılara kolay bir şekilde ulaştırılmasına imkân sağlamaktadır. WEB CBS’nin

temelinde bulut bilişim teknolojileri yer almaktadır. Özellikle klasik CBS'den WEB CBS'ye geçiş, bulut bilişimde meydana gelen gelişimlerle daha hızlı gerçekleşmektedir. (İrcan MR, Duman N 2020)

Web tabanlı coğrafi bilgi sisteminin oluşturulması için öncelikle bir CBS tasarım ve uygulama aşamalarından geçilmesi gerekmektedir. İlk olarak mevcut durum belirlenmelidir. Kullanılacak yazılım ve donanım özellikleri saptanmalıdır. Daha sonra hangi konuda CBS hazırlanacaksa o konuda araştırma yapılmalıdır. Gereksinimlerin belirlenmesi ve bu gereksinimleri karşılayacak şekilde sistem tasarlanır. Bu tasarımda veri tabanı tasarımı ve gerekli katman/katmanlar belirlenir. Katmanın hangi tipte (nokta, çizgi, alan) olması gerektiği hazırlanacak CBS çalışmasının ölçeğine uygun olmalıdır. Katmanda hangi özniteliklerin olması gerektiği tasarlanır. Sistemde sözel verilere gereksinim olması durumunda da gerekli öznitelikler de saptanır. Burada önemli olan sisteme veri girişinde veri tekrarlarının olmamasıdır. Veri tekrarının önlenmesi için veriler arasında bire-bir ya da bire-çok ilişkiye göre bağlantıların sağlanması gerçekleştirilir. Bunun için de katman ya da tablolar arasında bağlantı için ortak özniteliklerin oluşturulması ve bu duruma göre uygun şekilde veri girişinin yapılması gereklidir. (Gümüşay MÜ 2017)

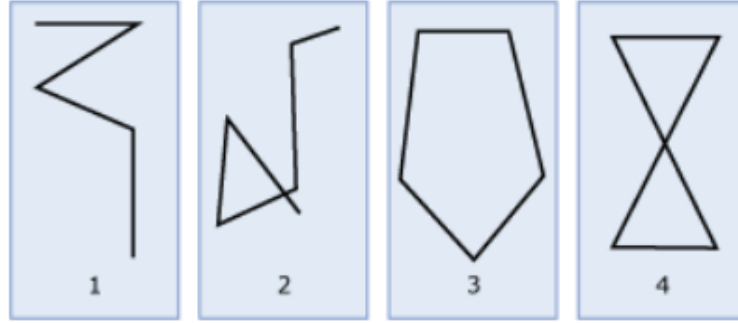
## Bölüm 2

# Konumsal Veri Tipleri, Saklanması ve Analizi

### 2.1 Konumsal Veri Tipleri

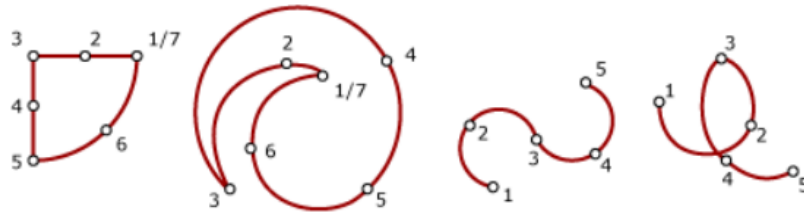
Konumsal veri tipleri ilgili verinin konum bilgisinin niteliğini temsil ederler. On adet temel veri tipi bulunmaktadır:

- Point (Nokta): Konumsal veri tipi için Point tipi, tek bir konumu temsil eder; bu tip için tek bir X, Y koordinat verisini barındırır. (Microsoft 2022)
- LineString: LineString noktalar dizisini ve bunları birleştiren çizgi parçalarını temsil eden tek boyutlu bir konumsal tiptir. (Microsoft 2022)



Şekil 2.1.1 LineString(Microsoft 2022)

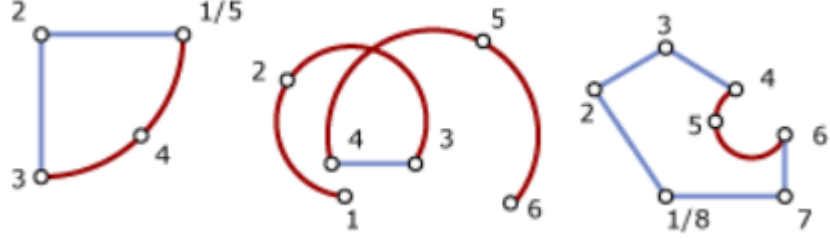
- CircularString: Dairesel yay parçası koleksiyonudur. Bir daireSEL yay parçası, iki boyutlu bir düzlemde üç nokta ile tanımlanan kavisli bir parçadır; birinci nokta üçüncü nokta ile aynı olamaz. (Microsoft 2022)



Şekil 2.1.2 CircularString (Microsoft 2022)

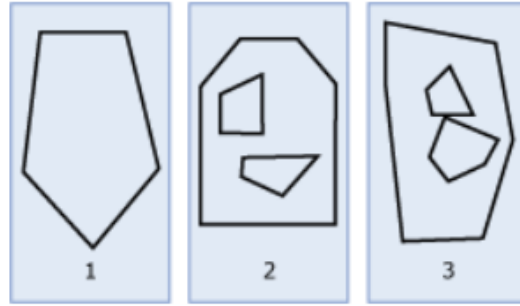


- CompoundCurve: CurcularString veya LineString koleksiyonudur. En az bir CircularString veya LineString örneği içerir. CircularString veya LineString örneklerinin sırası sürekli olmalıdır. (Microsoft 2022)



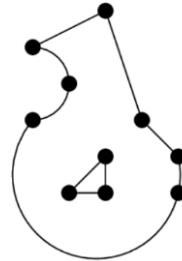
Şekil 2.1.3 CompoundCurve(Microsoft 2022)

- Poligon: En az üç farklı noktası bulunan bir çokgen örneğidir. Çokgen içinde boşluk oluşabilir. Bir Çokgenin dış ve herhangi bir iç halkası, sınırını tanımlar. Halkaların içindeki boşluk, Poligonun içini tanımlar. (Microsoft 2022)



Şekil 2.1.4 Poligon (Microsoft 2022)

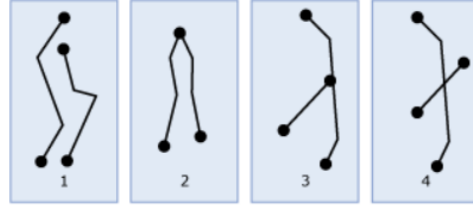
- CurvePolygon: Dairesel yay parçalarını içermesi açısından poligon örneğinden farklıdır. (Microsoft 2022)



Şekil 2.1.5 CurvePolygon (Microsoft 2022)

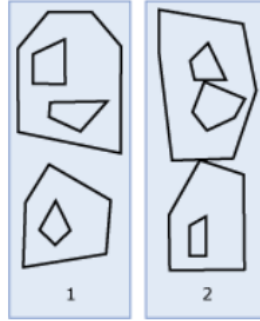
- MultiPoint: Sıfır veya daha fazla noktadan oluşan bir koleksiyondur. (Microsoft 2022)

- MultiLineString: Sıfır veya daha fazla geometri ya da geometricLineString örneğinden oluşan bir koleksiyondur. (Microsoft 2022)



Şekil 2.1.6 MultiLineString (Microsoft 2022)

- MultiPolygon: Sıfır veya daha fazla Poligon örneğini içeren bir koleksiyondur.



Şekil 2.1.7 MultiPolygon (Microsoft 2022)

- GeometryCollection: Sıfır veya daha fazla geometri tipi örneğinden oluşan bir koleksiyondur. Bir GeometryCollection boş olabilir. (Microsoft 2022)

## 2.2 Konumsal Verinin Saklanması

Bir noktanın konumunun tanımlanabilmesi için koordinatlar kullanılır. Bu koordinatlar tanımlı bir yer biçimi temel alınarak tanımlanmışlardır. Bu tanımlı yüzey üzerindeki koordinatların düzlem (harita) koordinatlara dönüştürülmesi için projeksiyonlar kullanılır. (KTU 2010)

Konumsal verilerin saklanması genel olarak yaygın olarak kullanılan oracle, sqlserver, postgresql vb. veri tabanı yapılarında içinde koordinat bilgisinin yanı sıra projeksiyon sistemi ile saklanır. Günümüzde birçok veri tabanı yazılım üretici firma coğrafi bilgi sistemlerini destekleyecek duruma gelmiştir. Oluşturulan tabloda konum bilgisi 'Geometry' tipinde oluşturulan kolonlarda tutulmaktadır. Coğrafi bilgi sistemi desteği

bulunmayan veri tabanlarında ise konum bilgisi WKT formatında oluşturulmuş sözel veri olarak tutulabilir. Bunun yanı sıra bazı özel dosya formatlarında da konum bilgisi ve öznitelik bilgileri tutulabilir. Bunların başlıcaları SHP, TAB, GML, GEOJSON dur. SHP Esri firması tarafından geliştirilen bir dosya yapısıdır. Esri firmasının coğrafi bilgi sistemleri üretiminde uzun yıllardır yaptığı çalışmalar nedeni ile bu firma birçok kütüphane tarafından desteklenebilir hale gelmiştir. TAB MapInfo programı için geliştirilmiş bir dosya yapısıdır. Kullanım alanı SHP kadar geniş değildir. Genellikle MapInfo kütüphaneleri tarafından desteklenir. Fakat ülkemizde kullanım alanı yaygındır. GML genel olarak XML dosya formatıyla bire bir aynıdır. XML diliyle oluşturulmuş veri yapıları içinde konum bilgisini de barındırır. Kullanım oranı GEOJSON formatının yaygın olarak kullanılmaya başlaması ile birlikte azalmıştır. GEOJSON formatında ise bilgiler JSON formatında öznitelik ve konum bilgileriyle birlikte tutulur. Oluşturulan veri tabanı tablolarında aynı veri tipindeki veri toplamak verimlilik ve kullanım kolaylığı açısından daha uygundur. Arcgis, MapInfo vb. yazılımlar bu esas olmadan analiz yapamamaktadır.

## 2.3 Konumsal Veri Analizi

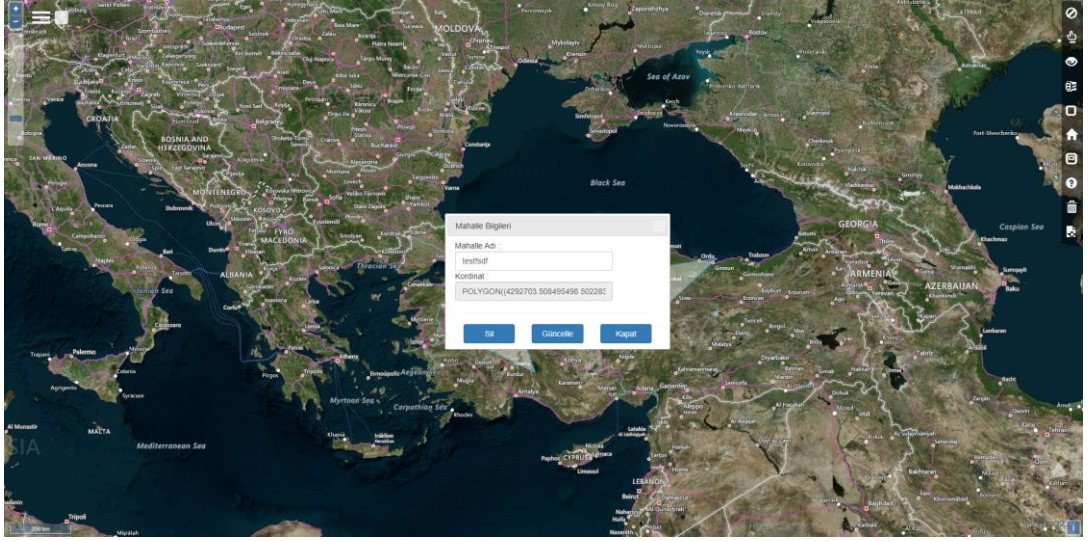
Konumsal verilerin analizi yapılırken verilerin birbirleriyle olan ilişkini anlamak önemlidir. Örneğin bir adres örneğini göz önüne alalım. Bir adres de il bilgisi, ilçe bilgisi, mahalle bilgisi, kapı bilgisi vb. öğeler içerir. İl, ilçe, mahalle verileri konumsal olarak poligon tipinde veri barındırırlar. Kapı ise nokta tipinde veri barındırır. Kapı mahalle sınırları içindedir. Mahalle ve kapı ise ilçe sınırları içinde kalır. İlçe, mahalle ve kapı ise ilçe sınırları içinde yer alır. Coğrafi bilgi sistemleri için bu konumsal ilişki önemlidir. Veriler arasındaki ilişki üç temel fonksiyon üzerine kurulmuştur. Bunlar Intersect(Kesişim), Within(içermek), Near(yakınlık) dır. Coğrafi analiz yapabilen yazılımlar genel olarak bu üç temel işlevden fazlasını barındırırlar. Özel durumlara göre bu üç işlevin iterasyonlarından türeyen farklı fonksiyonlar türetilmiştir. Web tabanlı coğrafi bilgi sistemlerinde genel olarak veri analizleri coğrafi bilgi sistemi desteği olan veri tabanlarında gerçekleştirilir. Örneğin mongoDB kullanan bir uygulama Intersect işlevini kullanabilmek için veri tabanında bulunan \$geoIntersects hazır fonksiyonunu kullanabilir. Bunun yanı sıra web tabanlı uygulamalarda çeşitli kütüphaneler kullanılarak da coğrafi veri analizi yapılabilir. Arcmapjs, turfjs vb. kütüphaneler buna örnek olabilir. Özellikle coğrafi data desteği bulunmayan veri

tabanı yapıları ile çalışan uygulamalar bu kütüphanelere ihtiyaç duyarlar. Diğer bir senaryoda ise coğrafi data desteği bulunan bir veri tabanı kullanılsa bile daha kompleks işlevler barındıran bir kütüphane kullanarak konumsal veri analiz işlemlerini gerçekleştirebilir. Örneğin Enerji sektöründe elektrik dağıtımını yapan bir firmanın web tabanlı coğrafi bilgi sistemini ele alalım. Envanter olarak direk, kablo, trafo vb. envanterleri veri tabanında barındırır. İhtiyaç dahilinde bir elektrik kesintisi planlandığını varsayarsak elektriği hangi trafodan kesildiğinde nerelerin etkilendiğini bilmek isteyeceklerdir. Bunu öğrenmenin yolu envanterler arasında bir topoloji kurmaktır. Bu topolojiyi kurmak için gelişmiş coğrafi bilgi sistemi kütüphanelerine ihtiyaç duyulacaktır. Bir diğer hususta konumsal hassasiyet garantisidir. Örneğin bir doğalgaz firmasının kullandığı web tabanlı coğrafi bilgi sistemini ele alalım. Doğal gaz boruları genellikle yer altında bulunurlar ve kazma esnasında tam olarak nereden geçtiği nerede olduğu tam bilinmelidir. Konumsal veri analizini veri tabanı üzerinden yapıldığını farz edelim. Günümüzde birçok veri tabanı üreticisi 10m ye kadar hassasiyet garantisi vermektedir. Bu nedenle doğalgaz borusunun kaydedilme aşamasında konumunun doğruluğu uzunluğu ve benzeri bütün nitelikleri yanlış olabilir, herhangi bir bakım işleminde can kaybına neden olabilecek sonuçlar ortaya çıkabilir. Bu yüzden hassasiyet toleransı daha düşük ve bununla ilgili sözleşmesinde oluşabilecek durumların garantisini yüklenen CBS kütüphanelerini kullanmak firmaların tercihi haline gelmektedir. Ülkemizde bununla ilgili en yüksek kullanım oranı Esri Firmasının üretmiş olduğu Arcmapjs ve MapInfo temelli Mapextreme kütüphaneleridir.

## Bölüm 3

# Web tabanlı Coğrafi Bilgi Sistemlerinin Temel Çalışma Prensipleri

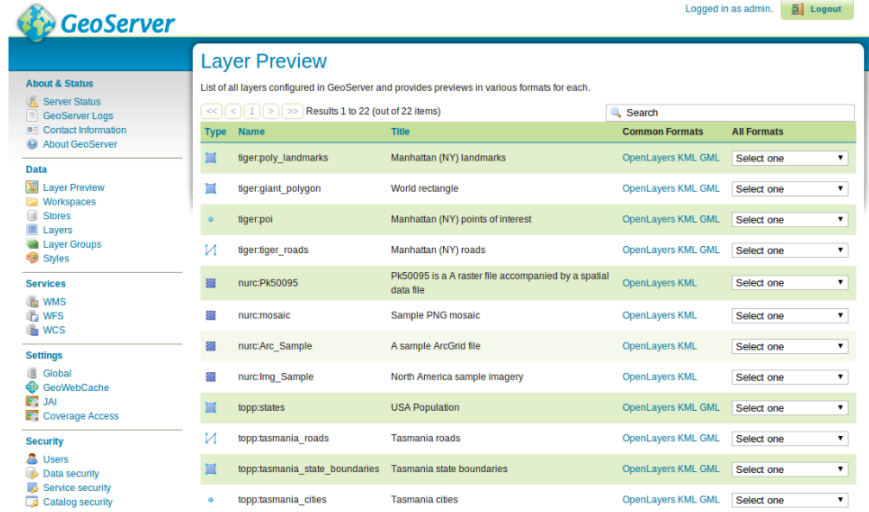
Web tabanlı coğrafi bilgi sistemleri temel prensip olarak genellikle tek sayfa çalışan uygulamalardır. Bunun nedeni arka planda raster veri barındıran bir harita altlığı kullanmasından kaynaklanmaktadır. Sayfa yenilenmesi sistemin en baştan yüklenmesi anlamına geleceği için genelde harita işlemleri harita altlığı üzerinde açılan açılır pencereler aracılığı ile gerçekleştirilir.



Şekil 3.1 Açılır Pencere

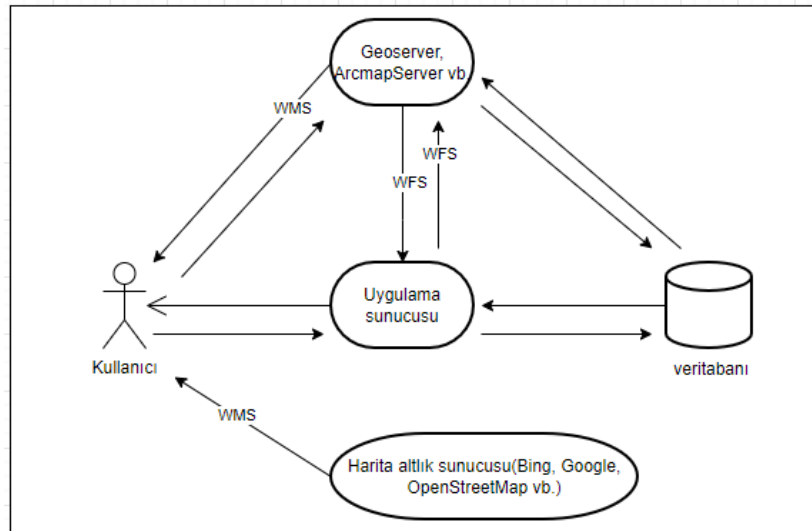
Uygulama da harita altlığı kullanmak konumun gerçek haritalarda tam olarak nerede bulunduğunu görmemiz açısından önemlidir. Harita altlıkları günümüzde Google, Bing, Esri, Openstreetmap gibi firmalar tarafından sağlanır. Firmalar kendi harita altlığı da üretip kendi serverlerin da wms servisler aracılığıyla uygulamada kullanabilirler. Harita üzerinde gösterilecek veriler vektörel veri olarak harita altlığının üstünde gösterilecek şekilde tasarlanır. Harita işlemlerinin tamamı için özel kütüphaneler kullanılır. En yaygın olarak kullanılan kütüphaneler Openlayers, Leaflet, Arcmapjs ve mapextremdir. Openlayers ve Leaflet opensource kütüphanelerdir. Bu

nedenle kullanım oranları çok yüksektir. Harita üzerinde gösterilecek vektörel veriler uygulama tarafından tarayıcı tarafına aktarılarak ya da ayrı bir wfs ya da wms servisler aracılığıyla istek gönderilerek elde edilebilir. Yoğun data barındıran uygulamalar genellikle wms ya da wfs servis kullanırlar. Bu durum uygulama sunucusunu rahatlatlattığı gibi data gösterim ayarlarını da ayrı bir arayüzde yapabilmelerini sağlar. Dünya üzerinde en yaygın kullanılan cbs servis sağlayıcı yazılımı Geoserver dır. Geoserver Opensource bir yazılım olup ücretli alternatiflerde mevcuttur.



Şekil 3.2 Geoserver

Geoserver kullanmayan uygulamalar ise direk veri tabanından elde edilen verileri harita üzerinde Openlayers, Lefleat gibi kütüphanelere data setini vererek sağlarlar. Aşağıdaki şekilde görsel olarak web cbs çalışma prensibi incelenebilir.



Şekil 3.3 Web Cbs Çalışma Şeması

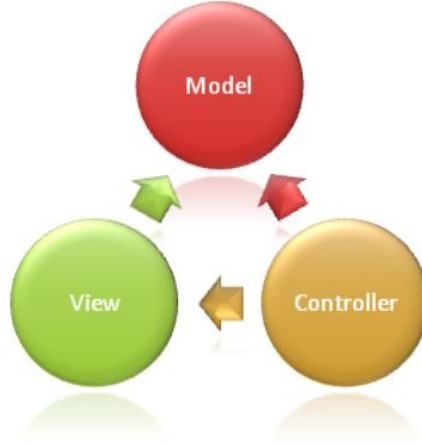
# Bölüm 4

## Uygulama projesi Altyapısı ve Kullanımı

### 4.1 Uygulama Projesi Altyapısı

Uygulama projesi asp.net mvc teknolojisi kullanılarak C# , Javascript, html dilleri kullanılarak yazıldı.

Model-View-Controller (MVC) mimari deseni bir uygulamayı üç ana bileşene ayırır: model, görünüm ve denetleyici. ASP.NET MVC çerçevesi, MVC tabanlı Web uygulamaları oluşturmak için ASP.NET Web Forms düzenine bir alternatif sağlar. ASP.NET MVC çerçevesi, (Web Forms tabanlı uygulamalarda olduğu gibi) ana sayfalar ve üyelik tabanlı kimlik doğrulaması gibi mevcut ASP.NET özellikleriyle tümleştirilen hafif, yüksek düzeyde test edilebilir bir sunu çerçevesidir. MVC çerçevesi System.Web.Mvc ad alanında tanımlanır ve System.Web ad alanının temel, desteklenen bir parçasıdır.(Microsoft Kaynak 6 2022)



Şekil 4.1.1 MVC. (Microsoft Kaynak 6)

Uygulama üzerinde iki adet controller oluşturuldu. LoginController ve HomeController uygulamada bulunun iki sayfa yani giriş ve anasayfa gerçekleşen fonksiyonlar bu controller larda yer almaktadır. Giriş işlemleri LoginController da anasayfa da ki tüm işlemleri içeren fonksiyonlarda HomeControllerda yer alır. Burada yer alan bazı fonksiyonlar için view'lar oluşturuldu. Örneğin BilgiAlPartialList

fonksiyonu için BilgiAlPartialList.cshtml view ı oluşturuldu ve model olarak BilgiAlPartialModel class ı ile nesne oluşturulup haberleşme sağlandı. Şekil 4.1.2 de BilgiAlPartialModel in kod blogunu görebilirsiniz.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Hasanileri_BitirmeProjesi_CbsUygulaması.Models
{
    3 references
    public class BilgiAlPartialModel
    {
        2 references
        public List<string> kapilist { get; set; }

        2 references
        public List<string> mahlist { get; set; }
    }
}
```

Şekil 4.1.2 Örnek Model

Javascript işlemlerini kolaylaştırabilmek için JQuery kütüphanesi kullanılmıştır. JQuery, John Resig tarafından 2006'da daha az yaz, daha fazlasını yap sloganıyla oluşturulmuş hızlı ve özlü bir JavaScript Kütüphanesidir (Tutorialspoint, 2015). Kütüphane hızla tanındı ve web uygulamaları oluşturmak için kullanılan en popüler Teknolojilerden biri haline gelmiştir. Uygulama geliştirme dünyasında etkili olan JQuery kütüphanesi, basit sözdizimi sayesinde bir zamanlar karmaşık olan işleri basit ve eğlenceli hale getirmektedir (Lindley, 2009). JQuery'yi her yerde, Twitter ve Facebook gibi oldukça popüler sitelerde, Yahoo! veya Google gibi arama motorlarında ve şık efektlerle dolu web sitelerinde görülmektedir (Beighley, 2010)

Harita işlemleri için Openlayers kütüphanesi tercih edilmiştir. Çok esnek, kullanıcı dostu ve anlaşılır bir javasctrip kütüphanesidir. Bu kütüphane sayesinde çok hızlı bir şekilde interaktif web tabanlı CBS uygulamaları geliştirilebilir. (ŞahinSoy, AK 2011) Openlayers uygulama harita işlemlerinin oturtulduğu ana kütüphanedir. Harita altlık gösterimi, harita şablonu, vektörel ve geojson verilerin gösterimi gibi birçok harita işlevi bu kütüphane odaklı gerçekleştirilir. Şekil 4.1.3 de Openlayers kütüphanesinin kullanım kod örneğini inceleyebilirsiniz.



```

var vectorSourceGeojson = new ol.source.Vector({
  format: new ol.format.GeoJSON(),
  url: 'GeoJson/gg.json'
});
var vectorLayerGeojson = new ol.layer.Vector({
  source: vectorSourceGeojson
});
map.addLayer(vectorLayerGeojson);
map.render();

```

Şekil 4.1.3 Openlayers Kütüphanesi kod örneği

Konumsal veri analizi için turfjs kütüphanesi tercih edildi. Openlayers Kütüphanesi harita işlemlerinde birçok kolaylık sağlamasına rağmen geometrik verilerin birbiriyle olan ilişkileri konusunda desteği bulunmamaktadır. Bu nedenle açık kaynak kodlu geliştirilen turfjs kütüphanesi kullanılmıştır. Şekil 4.1.4 de turfjs kütüphanesinde kesişim örnek kod u inceleyebilirsiniz.

```

var pt = turf.point(koordinata12);
var poly = turf.polygon(koordinata13);

var intersects = turf.intersect(poly, pt);
if (intersects != undefined) {
  sayac++;
  mahlist.push(item.mahalleAdi);
}

```

Şekil 4.1.4 Turfjs kod örneği

Veritabanı uygulaması olarak bir sqlite veritabanı dosyası tercih edilmiştir. Konumsal veriler sözel olarak wkt formatında tutulmuştur.

İsim	Tip	Şema
Tablolar (4)		
> Kapi		CREATE TABLE "Kapi" ( "Id" INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE, "KapiNo" TEXT, "MahalleAdi" TEXT, "Kkordinat" TEXT )
> Mahalle		CREATE TABLE "Mahalle" ( "Id" INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE, "MahalleAdi" TEXT, "MKordinat" TEXT )
> User		CREATE TABLE "User" ( "Id" INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE, "UserName" TEXT, "Password" INTEGER, "Role" INTEGER )
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
İndisler (0)		
Görünümler (0)		
Tetikleyiciler (0)		

Şekil 4.1.5 Veri tabanı şeması

Uygulama ve veri tabanı arasındaki iletişim ado.net teknolojisi ile sağlanmıştır. Sqlitextx sınıfında veri tabanı işlemleri için fonksiyonlar oluşturulmuştur. Şekil 4.1.6

de veri tabanına select sorgusu gönderip dönen listeyi datatable olarak dönen Sel isimli fonksiyonun kodlarını inceleyebilirsiniz.

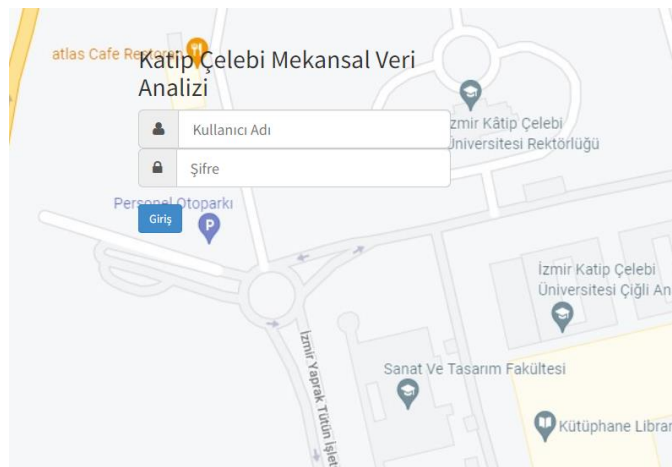
```
public DataTable sel(string sql)
{
    SQLiteConnection ctx = new SQLiteConnection(conn);
    SQLiteCommand command = new SQLiteCommand(sql, ctx);

    try
    {
        ctx.Open();
        DataTable dt = new DataTable();
        SQLiteDataAdapter da = new SQLiteDataAdapter(command);

        da.Fill(dt);
        ctx.Close();
        return dt;
    }
    catch (Exception ex)
    {
        ctx.Close();
        return null;
    }
}
```

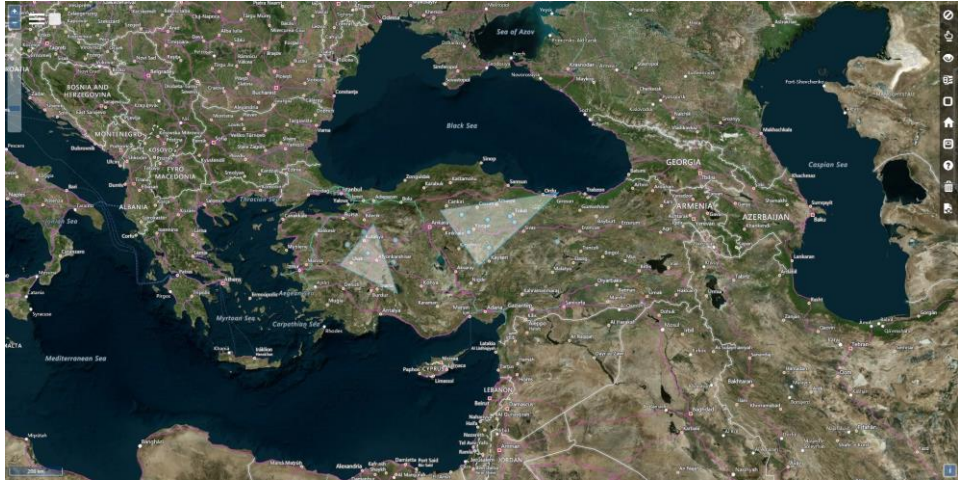
Şekil 4.1.6 Ado.net örnek kodu

Uygulama üzerinde adresler bölümde listeleme işlemleri için açık kaynak kodlu bir listeleme kütüphanesi olan datatables kullanılmıştır. Ayrıca görsel görünümü güzelleştirmek için bootstrap kütüphanesinden yararlanılmış. Pdf çıktı işlemleri için pdfjs kütüphanesi kullanılmıştır. Uygulama üzerinde Bing harita altlığı kullanılmıştır. Bing harita altlığı Microsoft tarafından developerlar için ücretsiz sunulana bir wms servistir. Uygulama web tabanlı bir coğrafi adresleme sistemi olarak tasarlandı. Kullanıcı girişi için bir giriş sistemi tasarlandı.



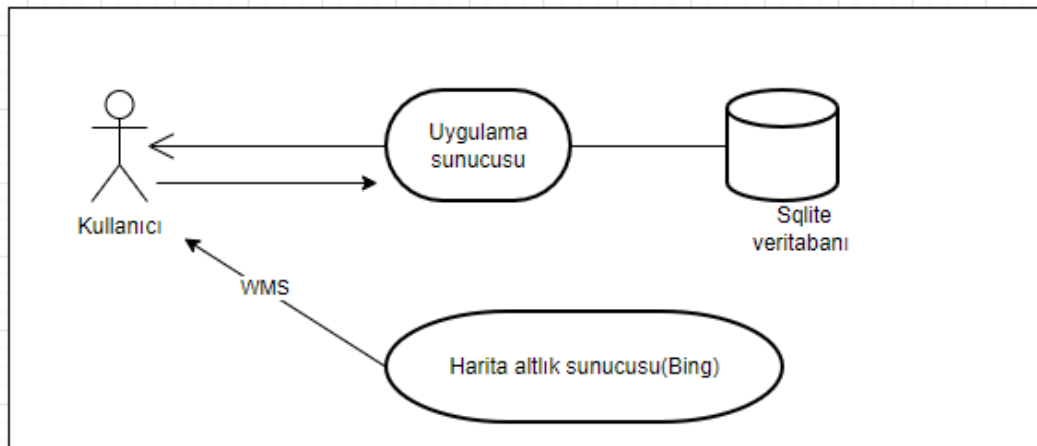
Şekil 4.1.7 Giriş Ekranı

Ön tanımlı olarak kullanıcı adı: 'hasanileri', şifresi ise '123' dür. Kullanıcı bilgileri doğru girildiğinde uygulama ana ekranına kullanıcı yönlendirilecektir. Eğer bilgiler doğru girilmezse kullanıcıya gerekli uyarılar gösterilip doğru giriş yapması istenilmektedir. Kullanıcı ana Ekranda Uygulama ile ilgili bütün işlevleri bu ekran üzerinden gerçekleştirecektir. Harita ekranında veri tabanında ki veriler harita ekranına yüklenir. Uygulama senaryosu olarak kullanıcı poligon olarak mahalle ve nokta olarak kapı çizimi yapabilmektedir. Kapı ve mahalle arasındaki konumsal ilişki turfjs kütüphanesinden yararlanılarak sağlanmıştır



Şekil 4.1.8 Ana ekran

Uygulamada bir coğrafi bilgi sistemi uygulamasında bulunması gereken temel işlevlerinin bulunması yönünde gayret gösterilmiştir. Kullanıcıya harita üzerinden bilgi alma, yüksek çözünürlüklü pdf çıktı. Uzunluk ve Alan ölçüm imkânı ve geojson formatında dosya yükleyip harita üzerinde dataların gösterilmesi gibi işlevler eklenmiştir. Uygulama çalışma şemasını aşağıdaki şekilde görüntüleyebilirsiniz.



## Şekil 4.1.9 Çalışma Şeması

### 4.2 Uygulama Kullanımı

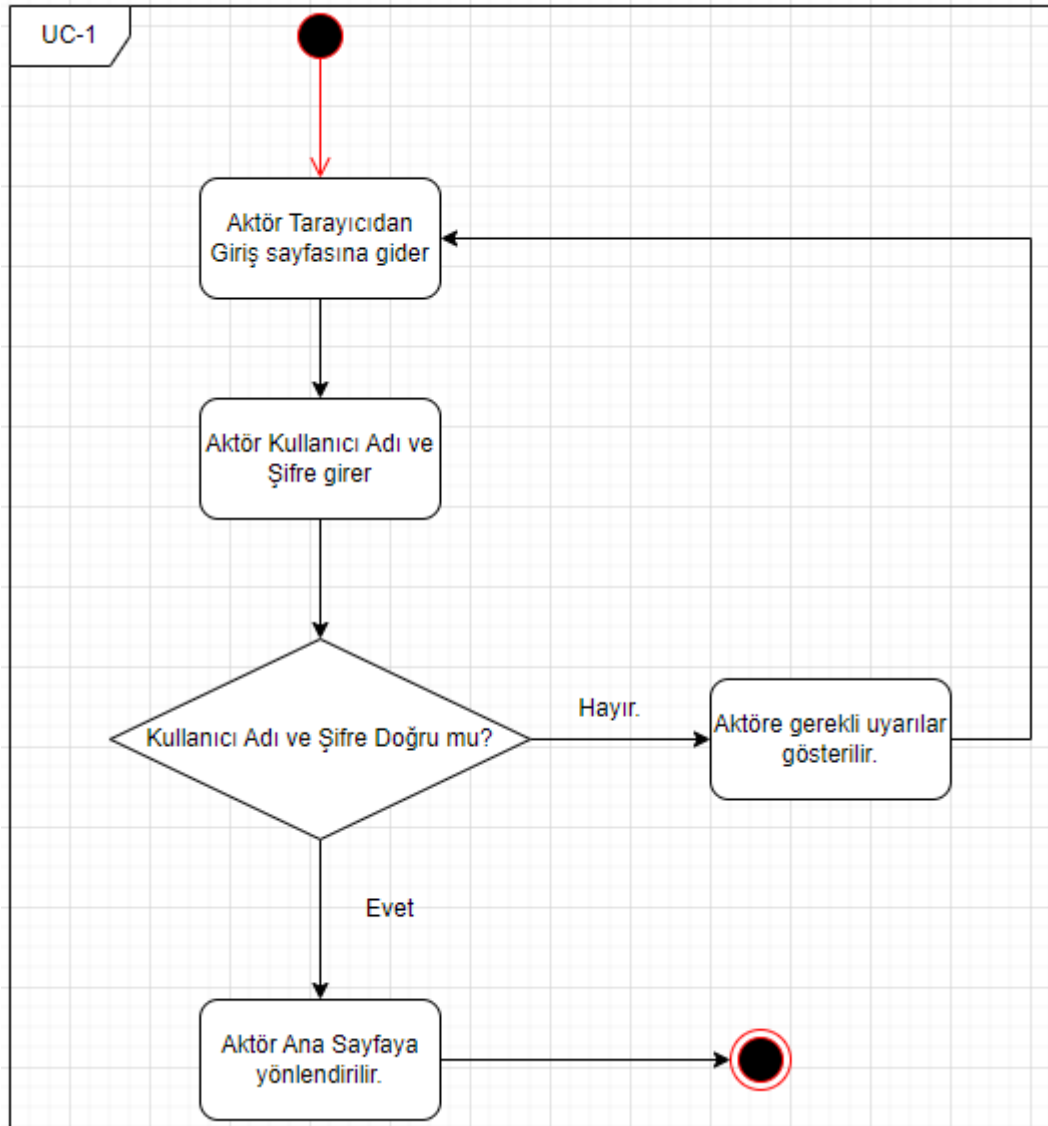
Uygulama üzerindeki fonksiyonel yapıları anlayabilmek ve kontrol edebilmek için ilgili fonksiyonlar Kullanım durum ( Use Case) diyagramları kullanılmıştır. Tablo 4.2.1 de diyagram listesini inceleyebilirsiniz

<b>Aktör</b>	<b>Aktör Hedefi</b>	<b>Use Case Id</b>
Kullanıcı1	Aktör sisteme giriş yapar.	UC-1
Kullanıcı1	Aktör haritada yakınlaşma, uzaklaşma işlemi yapar	UC-2
Kullanıcı1	Aktör mahalle çizimi yapar.	UC-3
Kullanıcı1	Aktör kapı çizimi yapar.	UC-4
Kullanıcı1	Aktör adresleri listeler.	UC-5
Kullanıcı1	Aktör adres listesinden adresin konumuna gider.	UC-6
Kullanıcı1	Aktör harita üzerinden bilgi alır.	UC-7
Kullanıcı1	Aktör kapı öznetelik bilgilerini günceller.	UC-8
Kullanıcı1	Aktör mahalle öznetelik bilgilerini günceller.	UC-9
Kullanıcı1	Aktör kapı siler.	UC-10
Kullanıcı1	Aktör mahalle siler.	UC-11
Kullanıcı1	Aktör veri tabanında bulunmayan kayıtları haritadan temizler.	UC-12
Kullanıcı1	Aktör pdf çıktı alır	UC-13
Kullanıcı1	Aktör haritada uzunluk ölçümü yapar	UC-14
Kullanıcı1	Aktör haritada alan ölçümü yapar.	UC-15
Kullanıcı1	Aktör haritada ilk görünümüne gider	UC-16
Kullanıcı1	Aktör haritada eller serbest modunu aktive eder.	UC-17
Kullanıcı1	Aktör haritaya Geojson formatında dosya yükler.	UC-18
Kullanıcı1	Aktör haritada seçtiği noktanın koordinat bilgisini görür.	UC-19
Kullanıcı1	Aktör Sistemden çıkış yapar.	UC-20

Tablo 4.2.1 Kullanım Durum Diyagramları

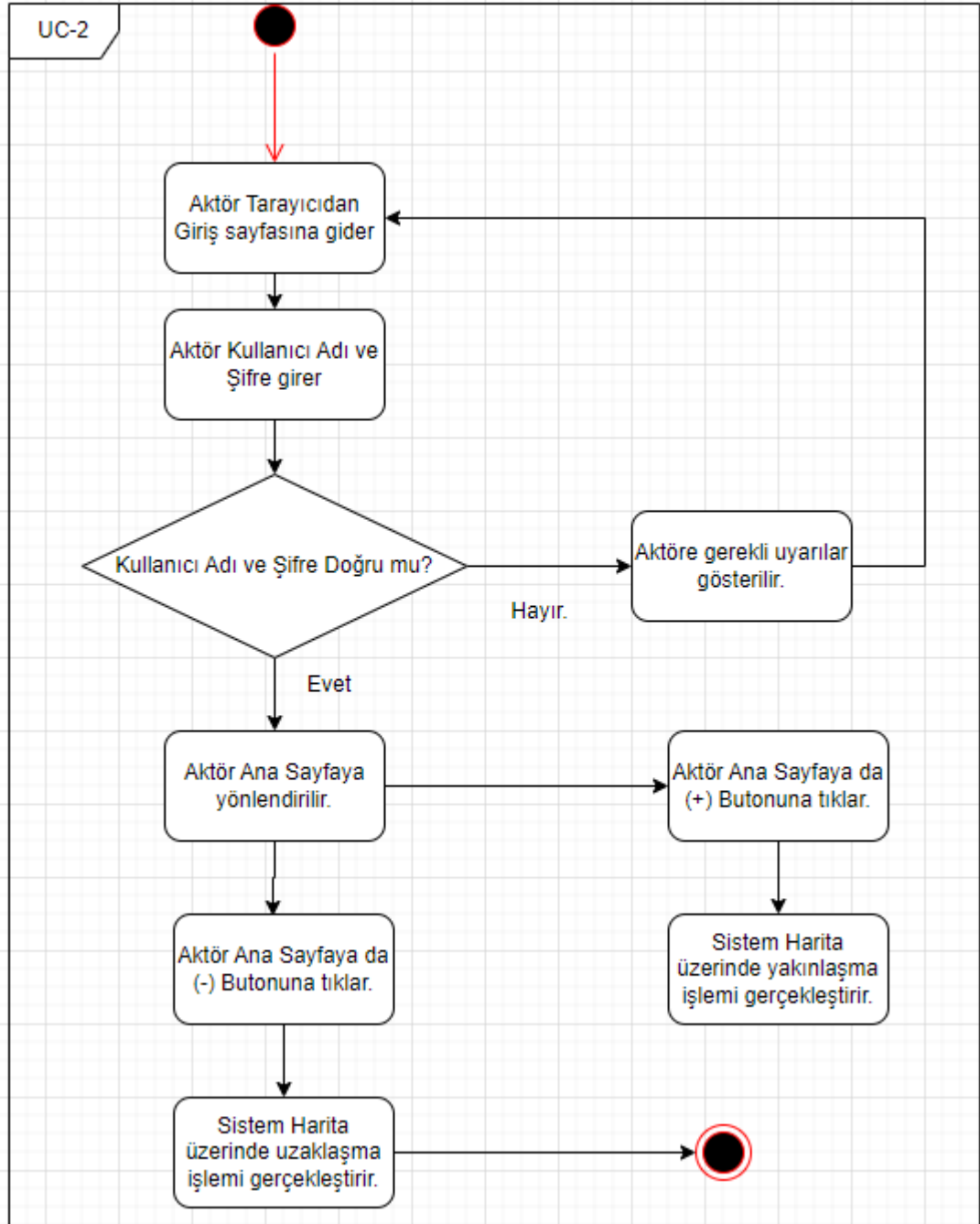
## 4.2.1 Uygulama Kullanım Diyagramları

### 4.2.1.1 Sisteme Giriş UC-1



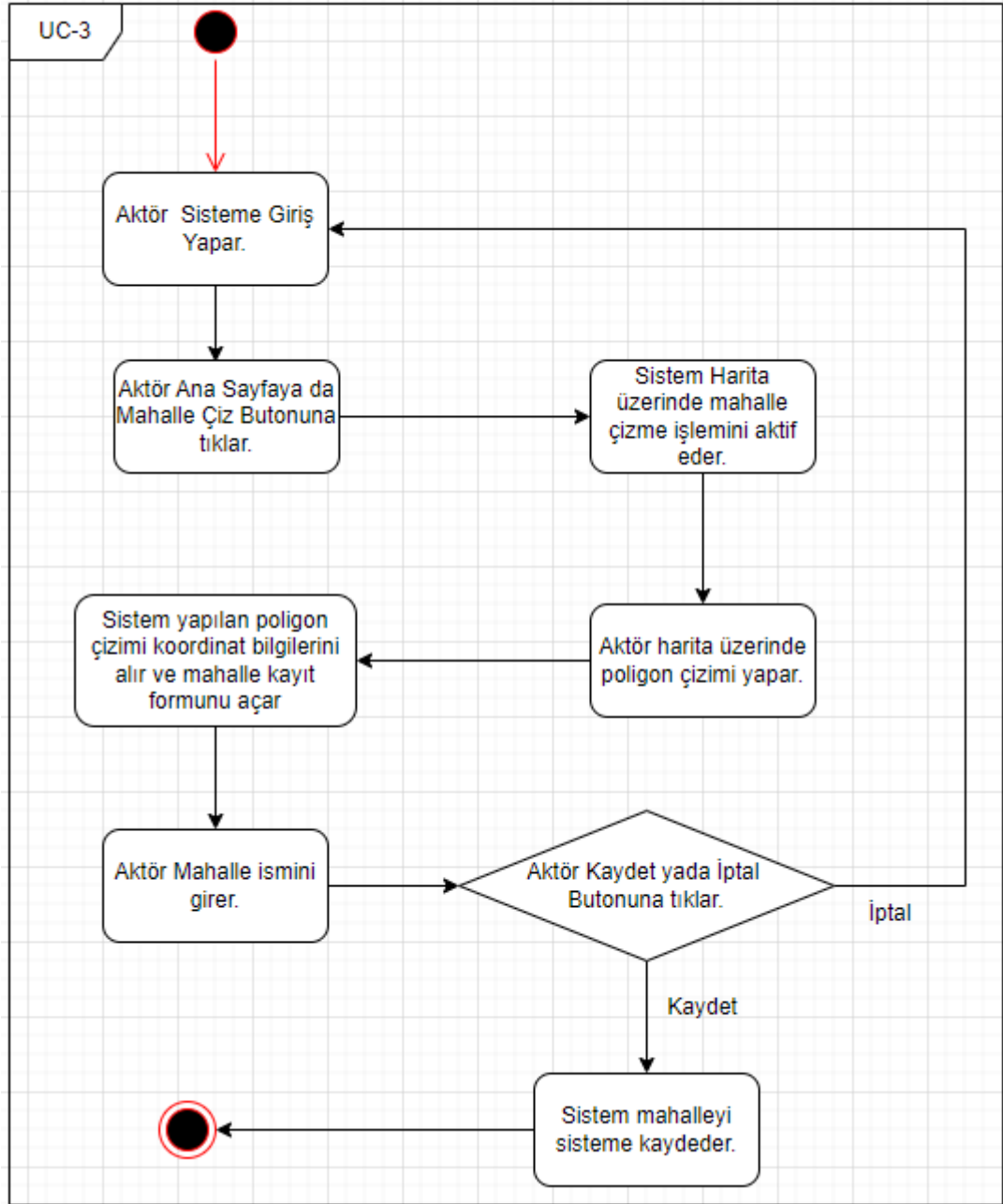
Şekil 4.2.1.1.1 Sisteme Giriş

#### 4.2.1.2 Harita Yakınlaşma, Uzaklaşma UC-2



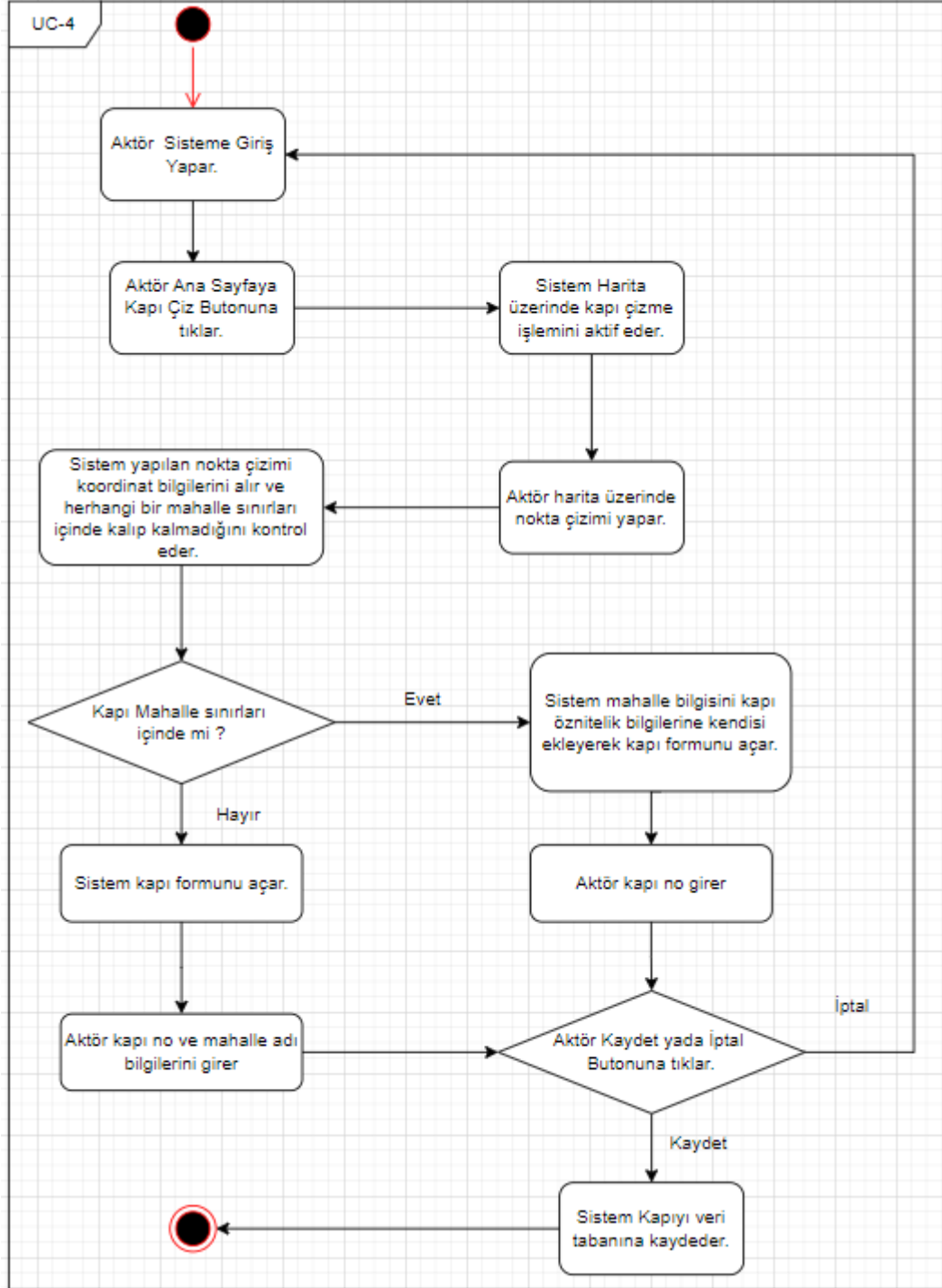
Şekil 4.2.1.1.2 Harita Yakınlaşma, Uzaklaşma

#### 4.2.1.2 Mahalle Çizimi UC-3



Şekil 4.2.1.1.3 Mahalle çizimi

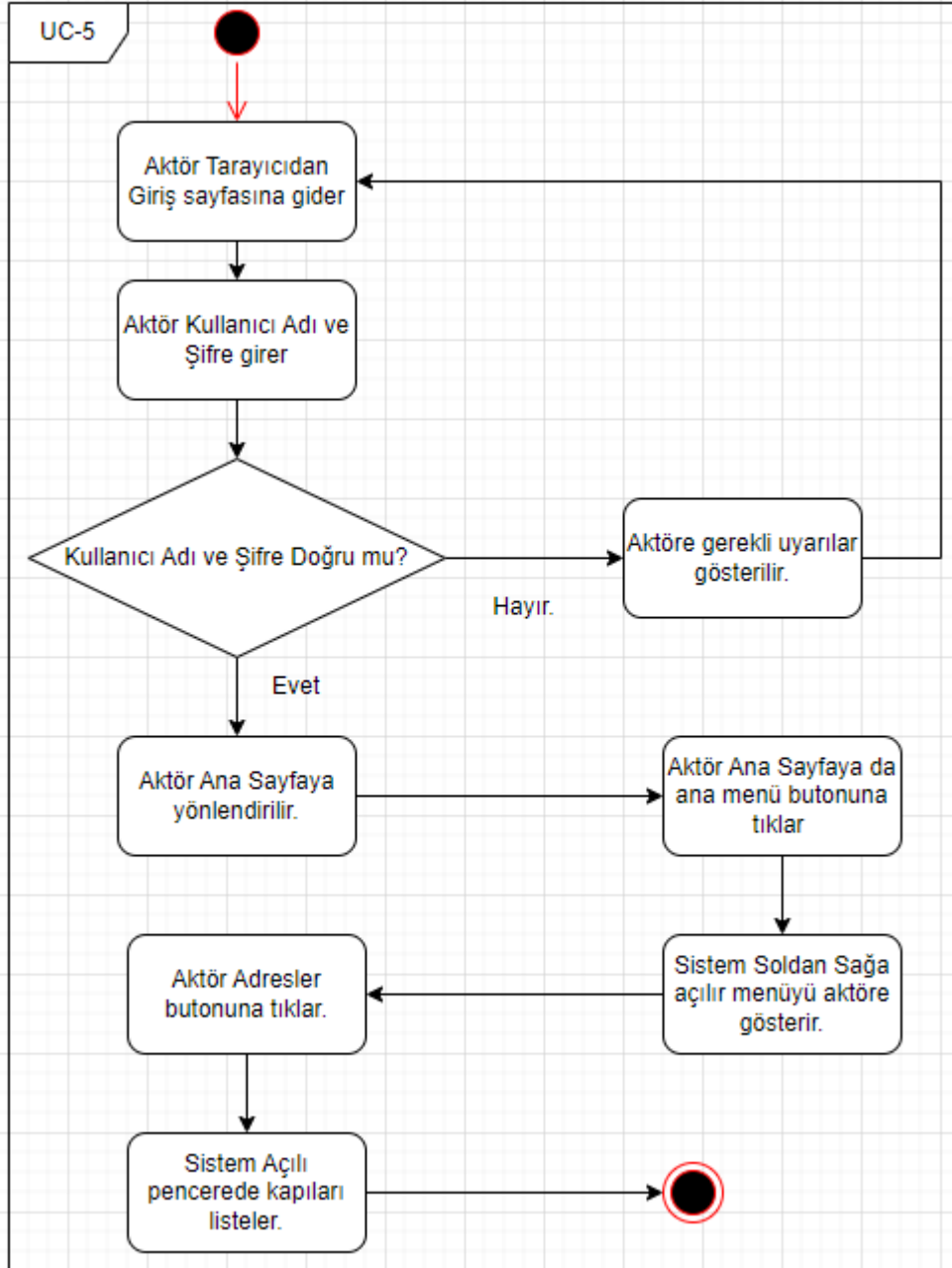
#### 4.2.1.4 Kapı Çizimi UC-4



Şekil 4.2.1.1.4 Kapı çizim

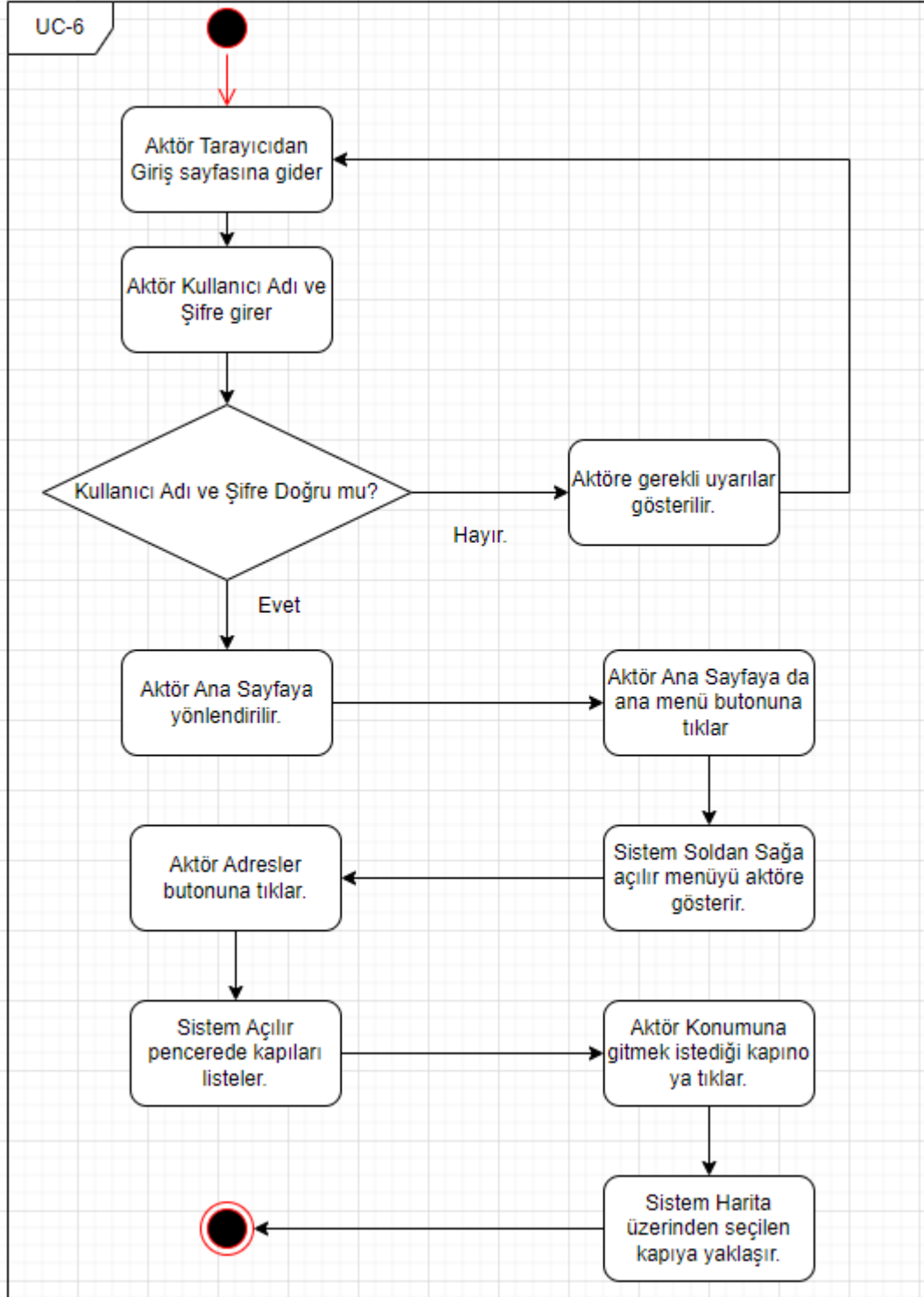


#### 4.2.1.5 Adres Listeleme UC-5



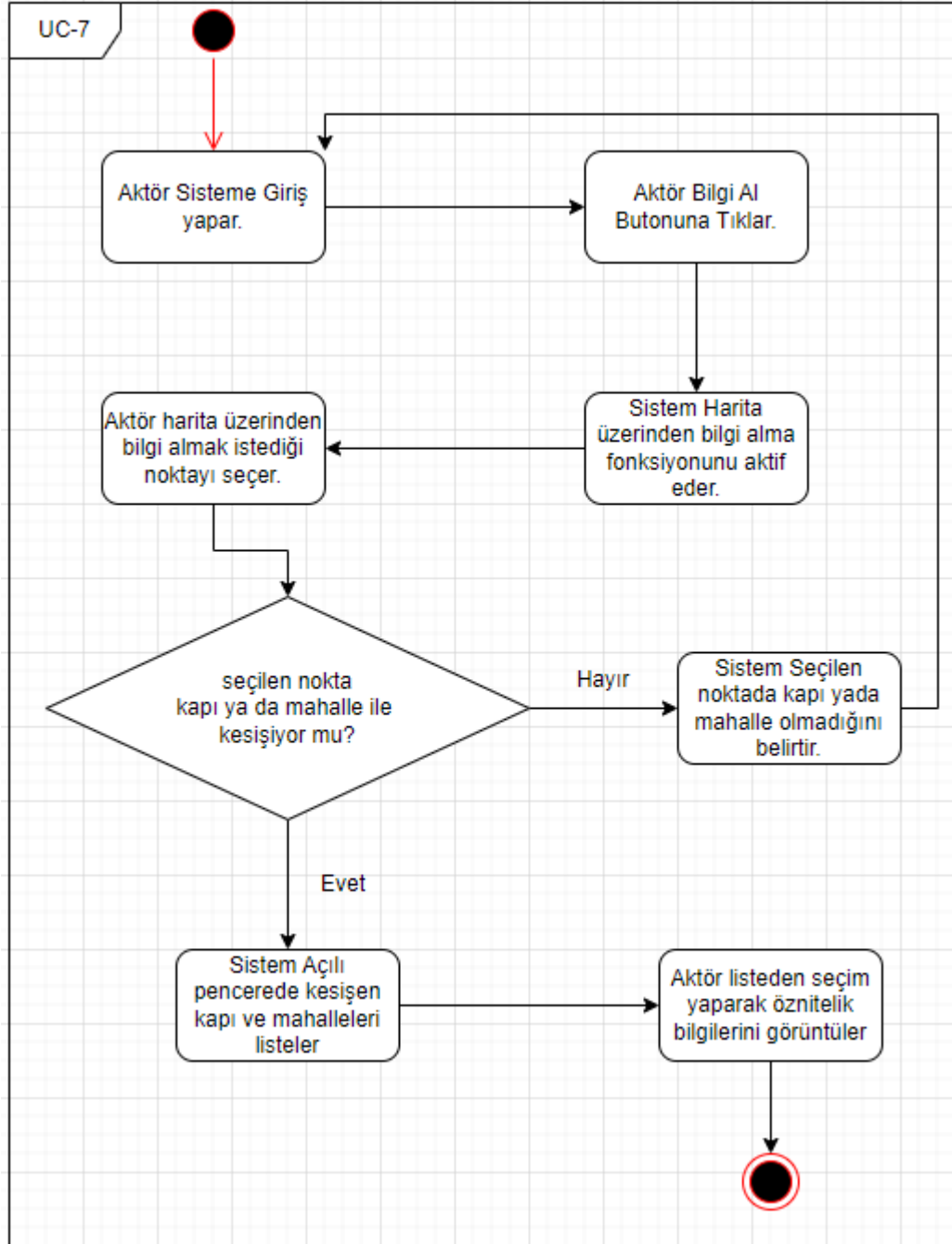
Şekil 4.2.1.1.5 Adres Listeleme

#### 4.2.1.6 Adres Listesinden Konuma gitme UC-6



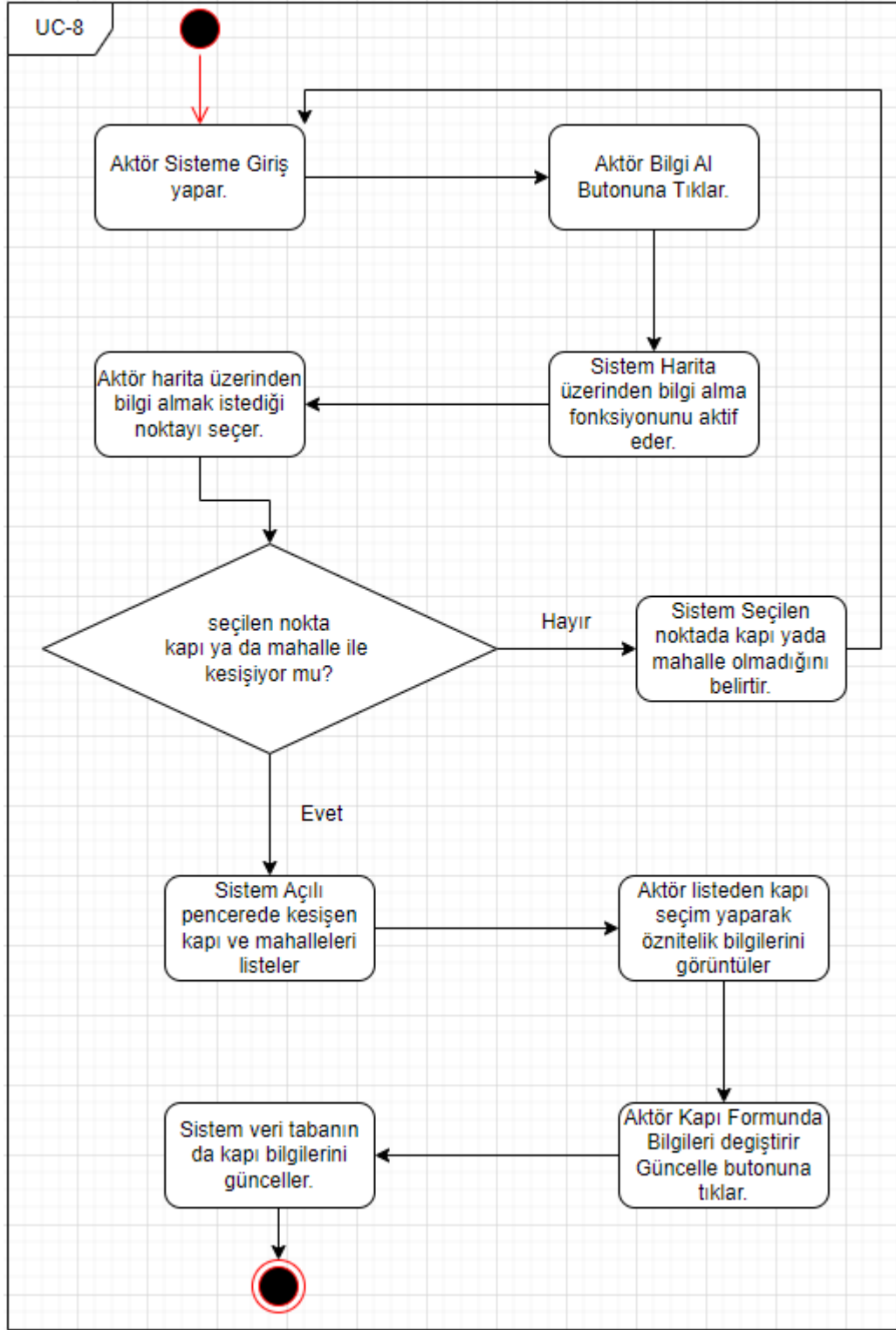
Şekil 4.2.1.1.6 Adres Listesinden konuma gitme

#### 4.2.1.7 Harita Üzerinden Bilgi Alma UC-7



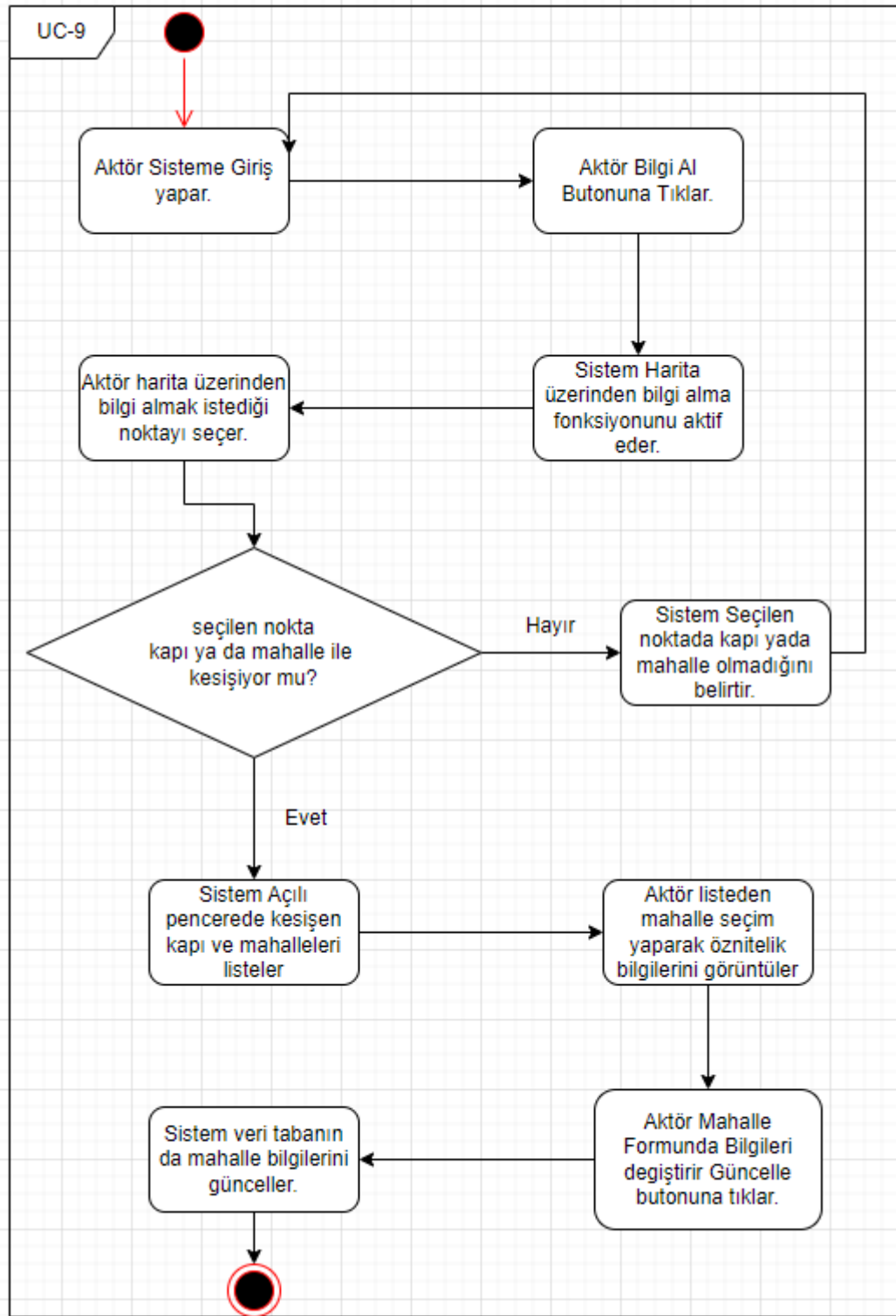
Şekil 4.2.1.1.7 Harita Üzerinden Bilgi Alma

#### 4.2.1.8 Kapı Bilgilerini Güncelleme UC-8



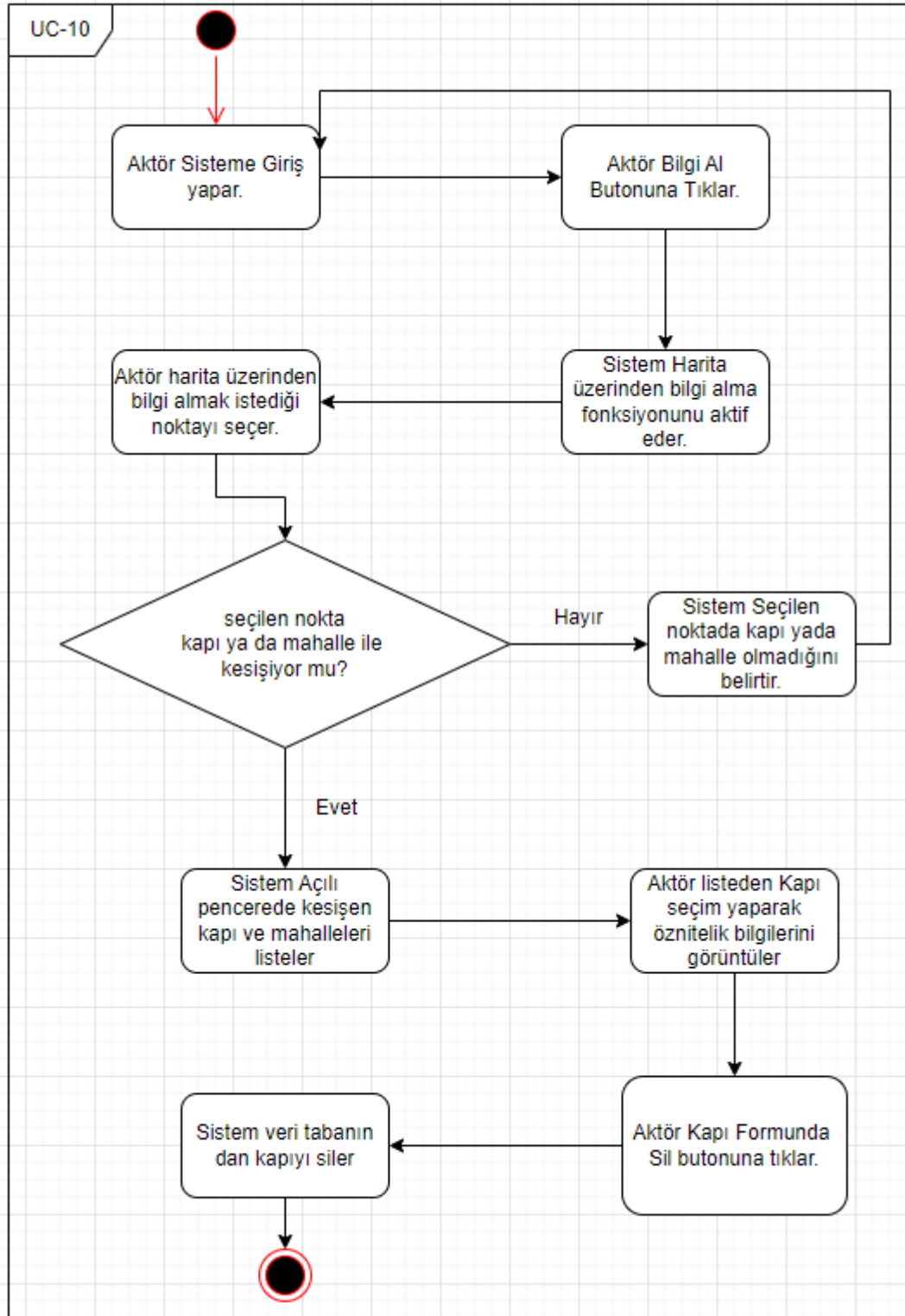
Şekil 4.2.1.1.8 Kapı Bilgilerini Güncelleme

#### 4.2.1.9 Mahalle Bilgilerini Güncelleme UC-9



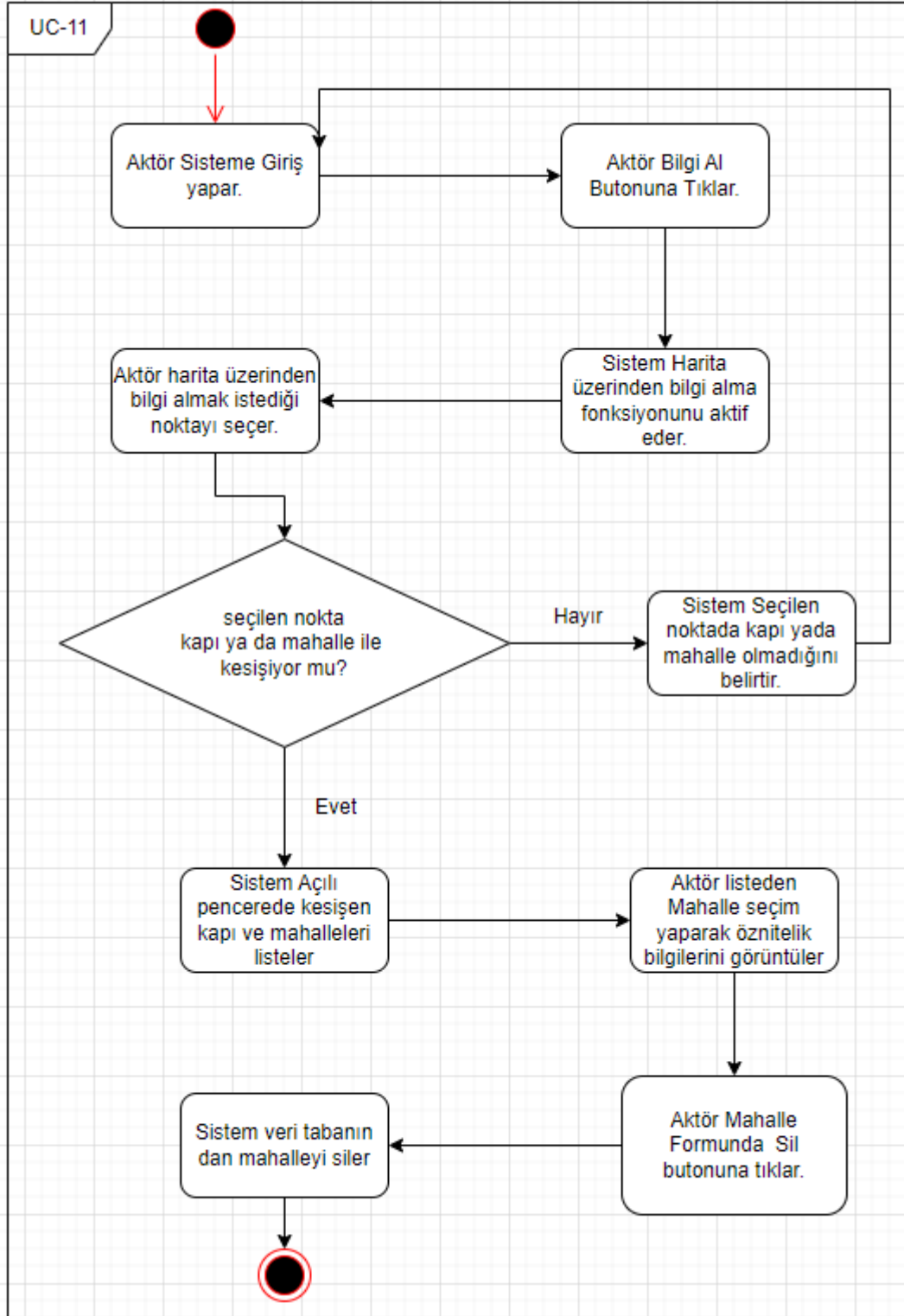
Şekil 4.2.1.1.9 Mahalle Bilgilerini Güncelleme

4.2.1.10 Kapı Silme UC-10



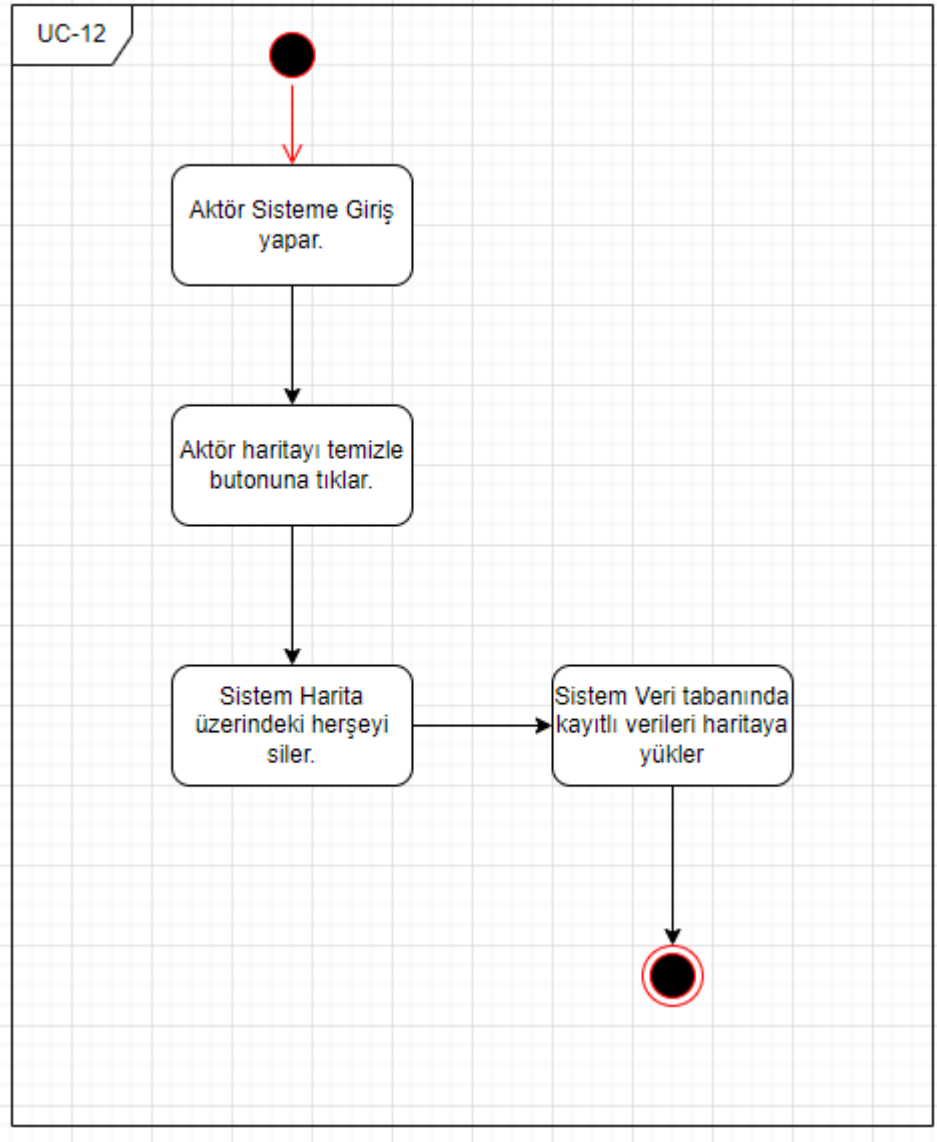
Şekil 4.2.1.1.10 Kapı Silme

4.2.1.11 Mahalle Silme UC-11



Şekil 4.2.1.1.11 Mahalle Silme

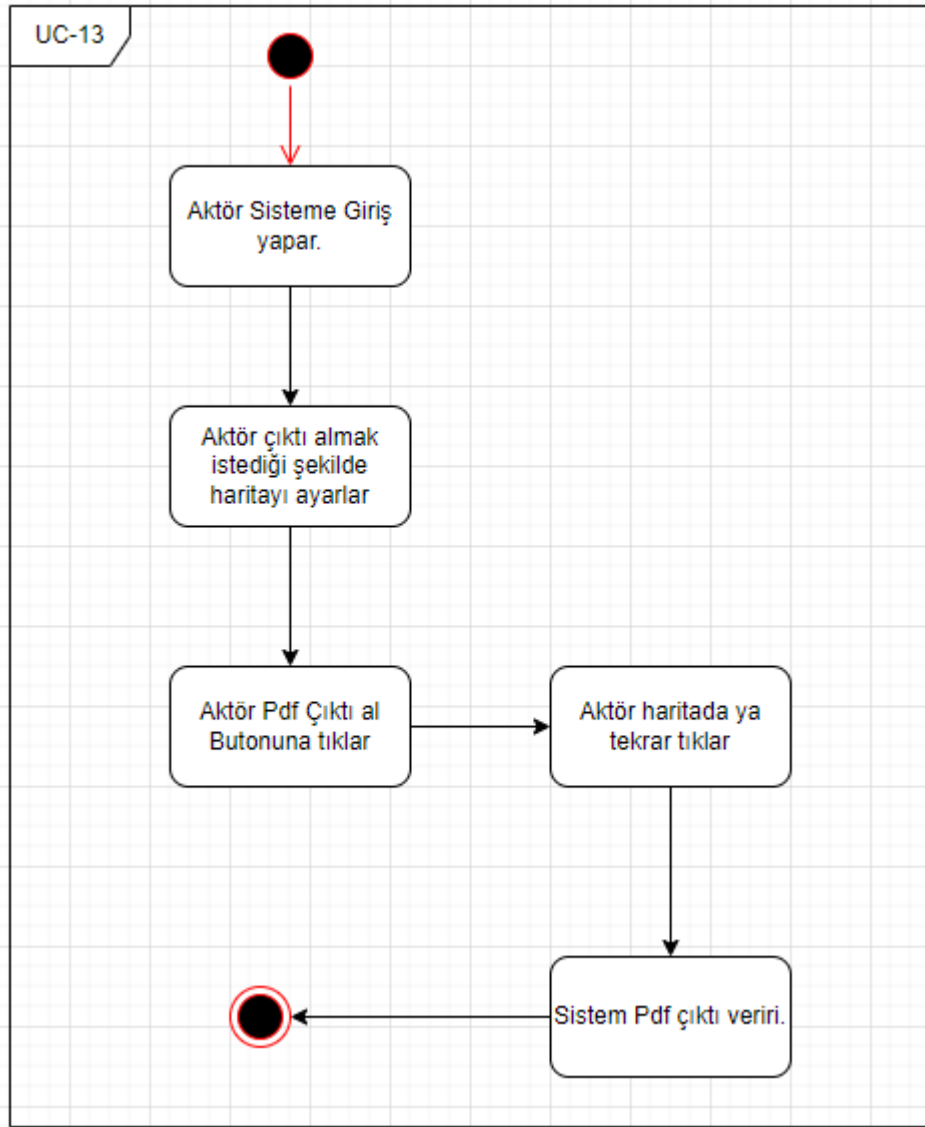
#### 4.2.1.12 Harita Temizleme UC-12



Şekil 4.2.1.1.12 Harita Temizleme

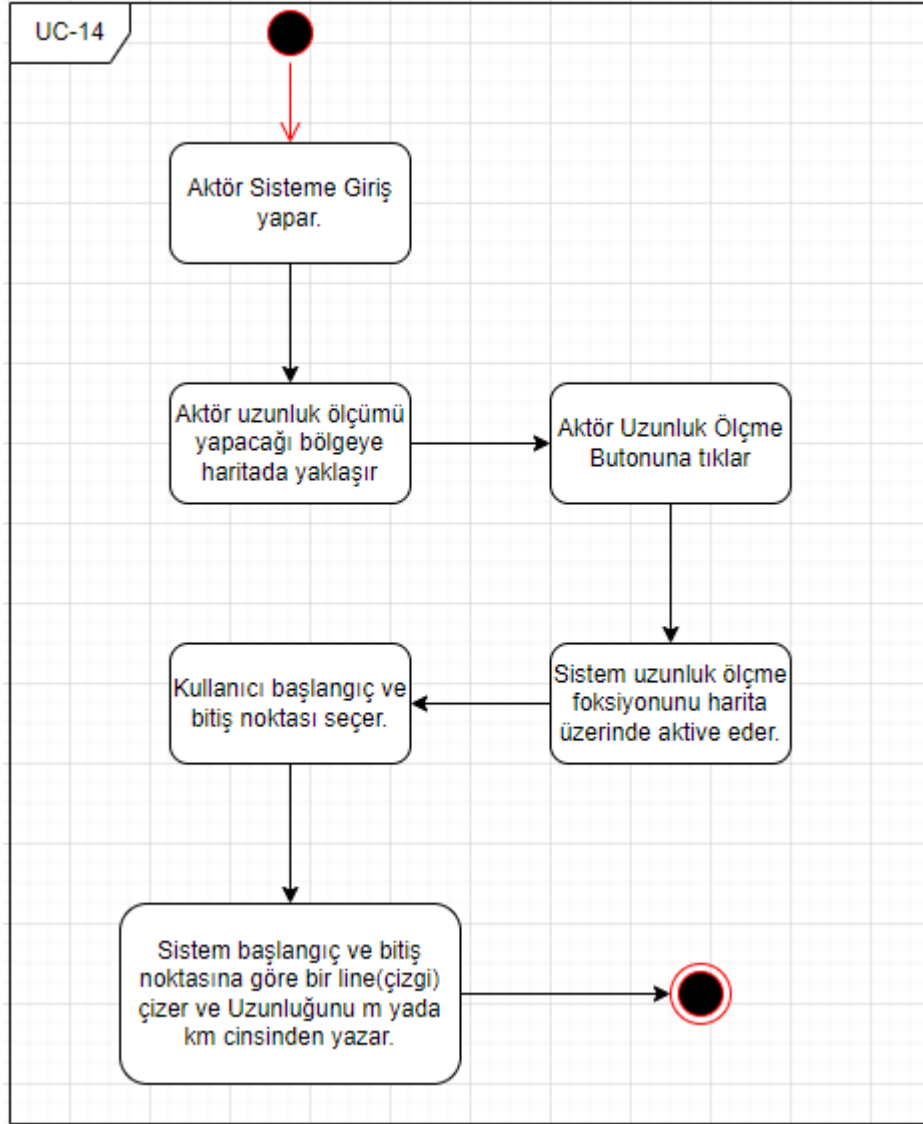


#### 4.2.1.13 Pdf Çıktı Alma UC-13



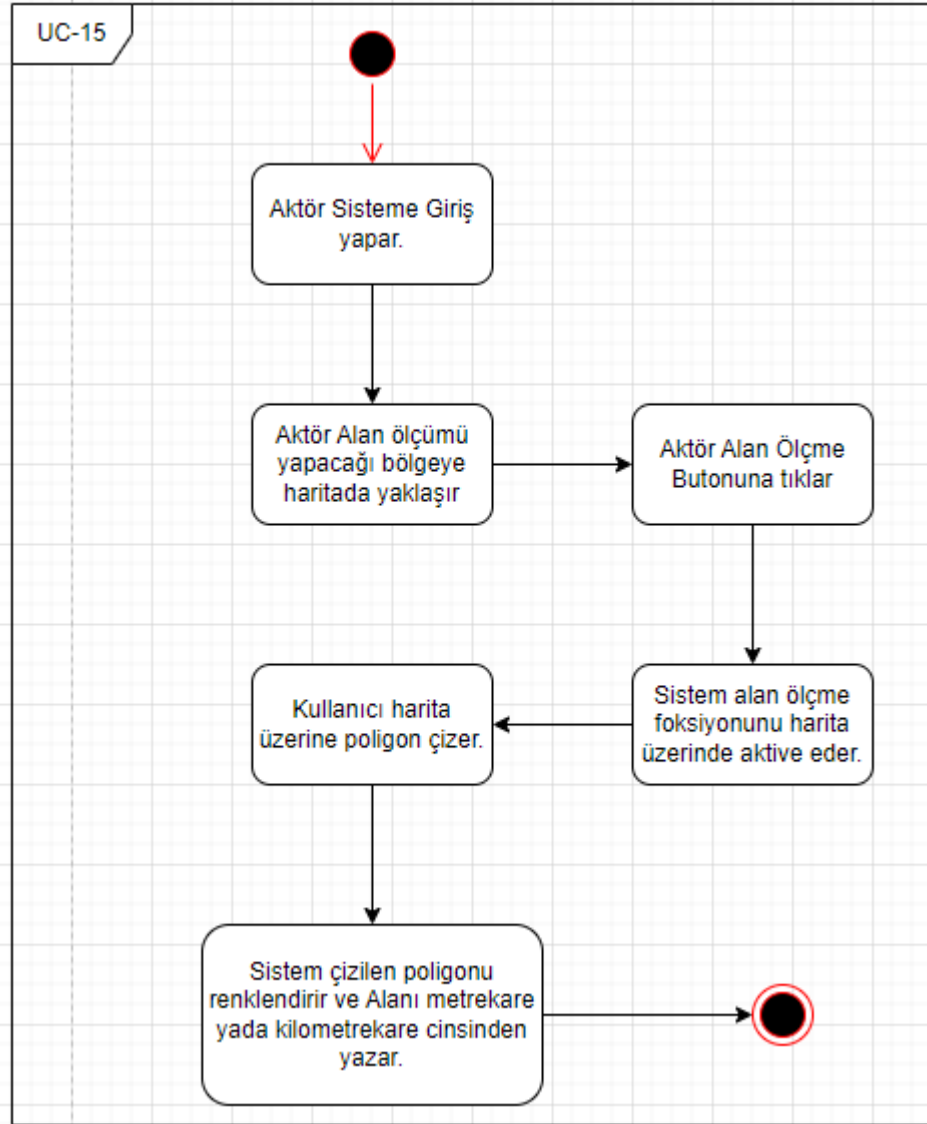
Şekil 4.2.1.1.13 Pdf Çıktı Alma

#### 4.2.1.14 Uzunluk Ölçme UC-14



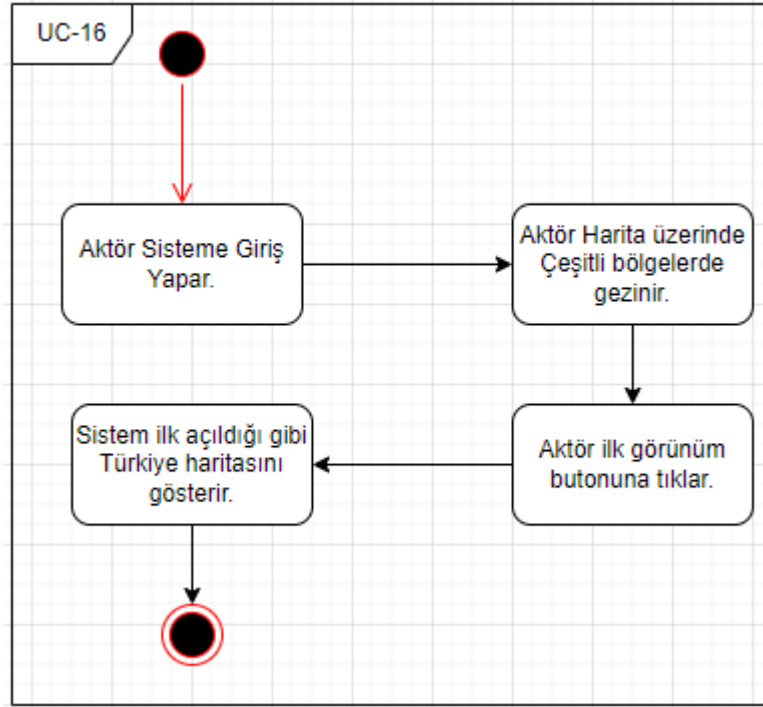
Şekil 4.2.1.1.14 Uzunluk Ölçme

#### 4.2.1.15 Alan Ölçme UC-15



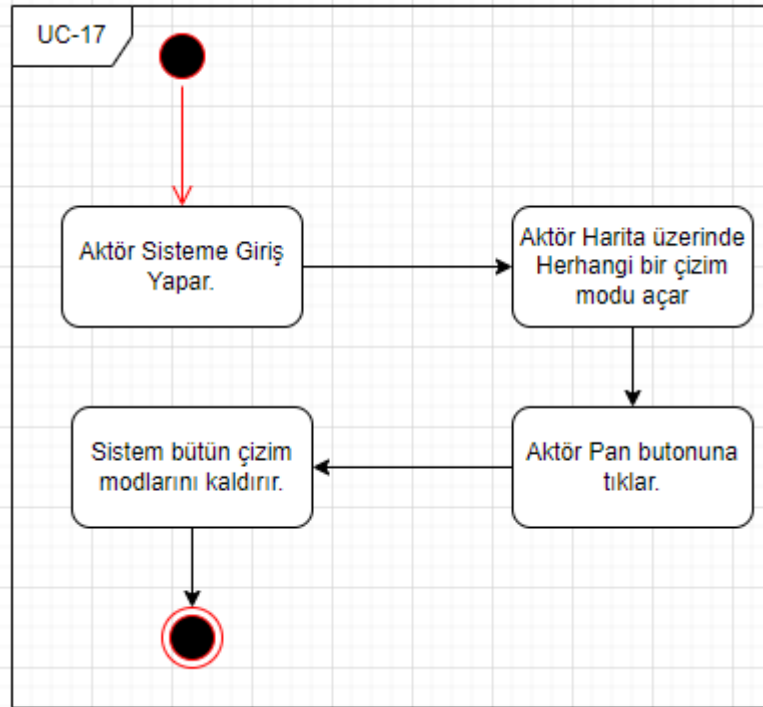
Şekil 4.2.1.1.15 Alan Ölçme

#### 4.2.1.16 İlk görünüme gitme UC-16



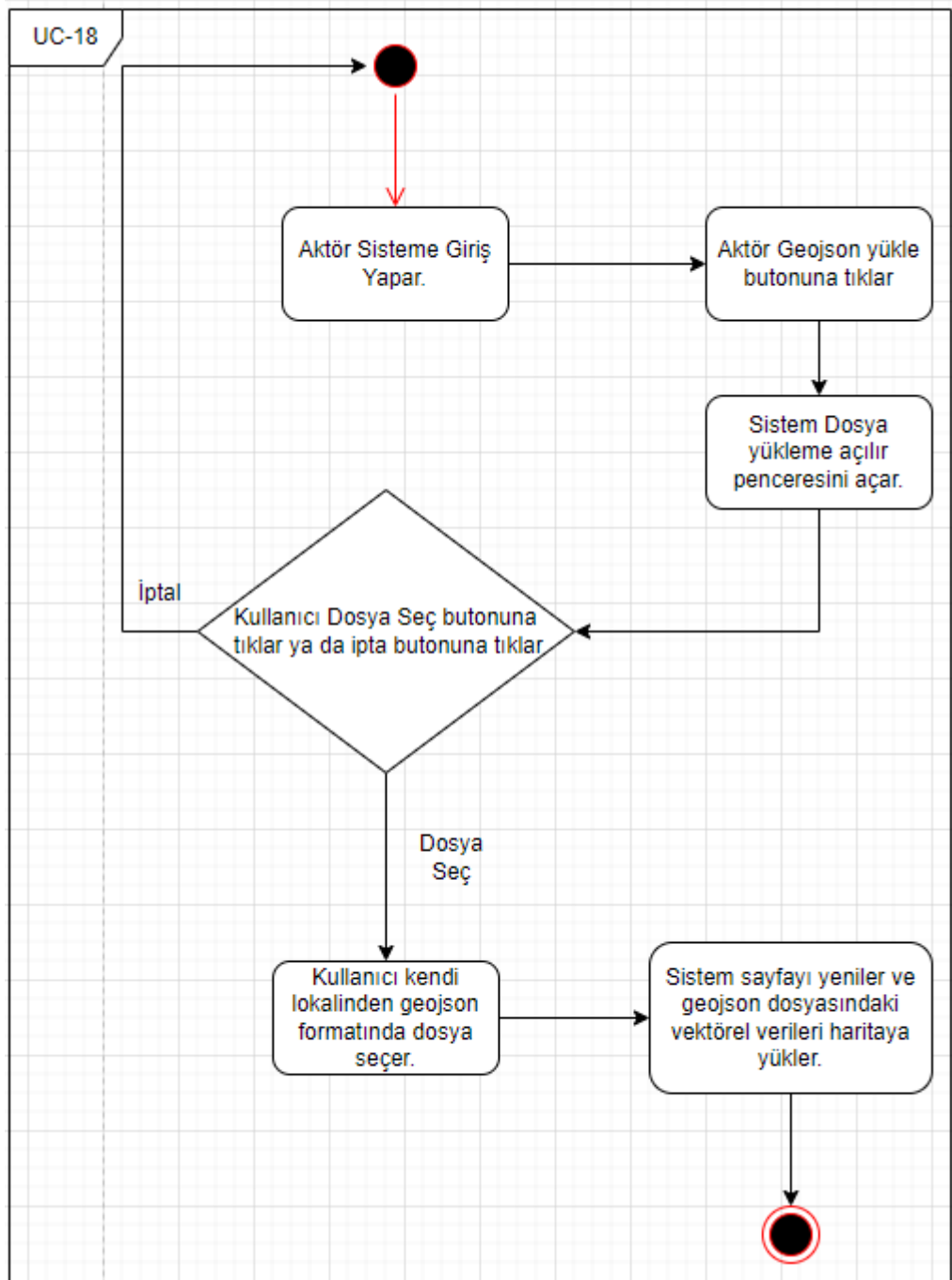
Şekil 4.2.1.1.16 İlk görünüme gitme

#### 4.2.1.17 Eller serbest modu UC-17



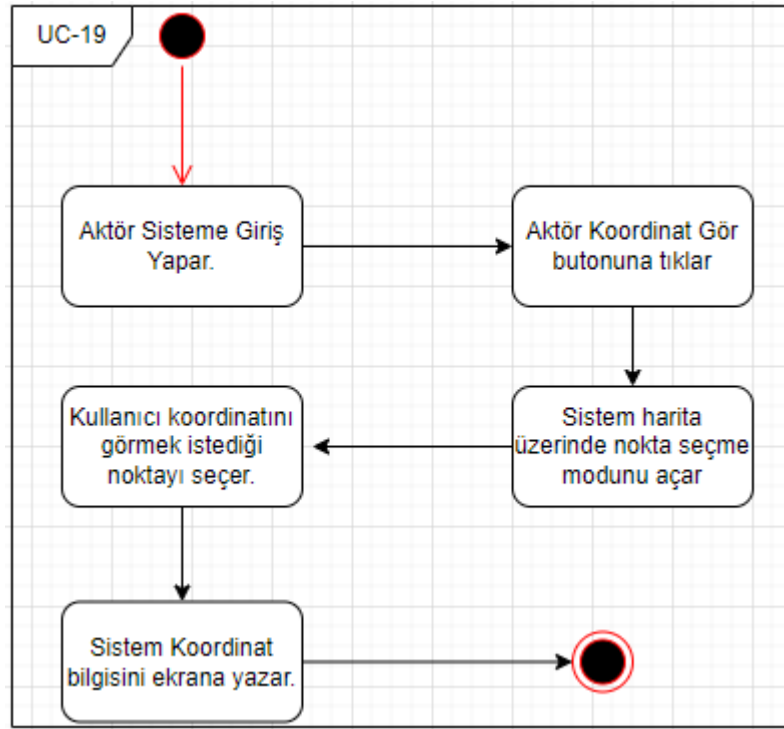
Şekil 4.2.1.1.17 Eller Serbest

#### 4.2.1.18 Geojson Formatında Dosya Yükleme UC-18



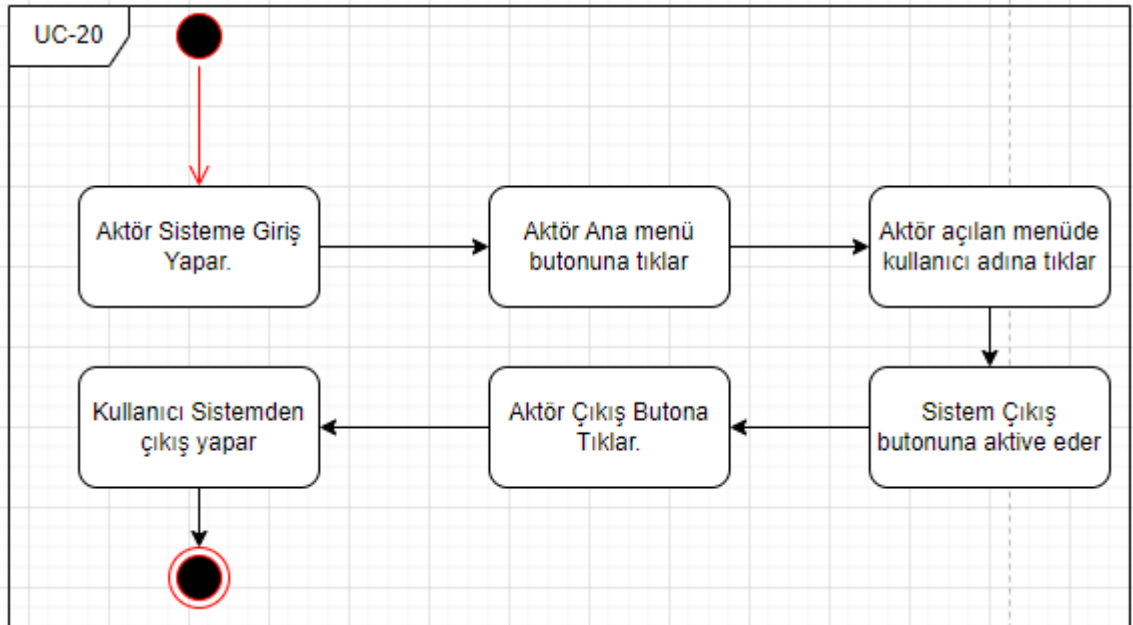
Şekil 4.2.1.1.18 Geojson Dosya yükleme

#### 4.2.1.19 Koordinat Bilgisi görme UC-19



Şekil 4.2.1.1.19 Koordinat Görme

#### 4.2.1.20 Sistemden Çıkış UC-20



Şekil 4.2.1.1.20 Sistemden Çıkış

# Kaynaklar

- [1] Başarsoft [internet]. Ankara ; 2022 [erişim tarihi 23.12.2022] .  
<https://www.basarsoft.com.tr/cografı-bilgi-sistemleri-cbs-nedir/>.
- [2] İrcan MR, Duman N. Use of Internet (WEB) Based Geographic Information Systems (GIS) Applications in High School Geography Lesson Teaching. GEFAD / GUJGEF40(3): 1051-1085(2020)
- [3] Gümüşay MÜ. WebTabanlı Coğrafi Bilgi Sistemi Uygulamaları. AKÜ FEMÜBİD 17 (2017) Özel Sayı (215-222)
- [4] Microsoft [internet]. Ankara;2022[erişim tarihi 23.12.2022].  
<https://learn.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver16>
- [5] KTU [internet]. Ankara;2010[erişim tarihi 24.12.2022].  
[https://www.ktu.edu.tr/dosyalar/ormanamenajmani\\_d14ba.pdf](https://www.ktu.edu.tr/dosyalar/ormanamenajmani_d14ba.pdf)
- [6] Microsoft [internet].Ankara;2022[erişim tarihi 09.01.2023].  
<https://learn.microsoft.com/tr-tr/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>
- [7] Tutorialspoint [internet]. Ankara;2015[erişim tarihi 09.01.2023] .  
<https://www.tutorialspoint.com/jquery/>
- [8] Lindley, C. (2009). JQuery Cookbook: Solutions ve examples for JQuery developers. CA, USA: O'Reilly Media, Inc
- [9] Beighley, L. (2010). jQuery for Dummies. Hoboken, NJ, USA : John Wiley & Sons Inc.
- [10] ŞahinSoy, AK. Açık kaynak kodlu cbs yazılımlarının imar planı verilerinin yönetilmesinde kullanımı İstanbul örneği (yüksek lisans tezi). İstanbul: İstanbul Teknik Üniversitesi; 2011. <https://web.itu.edu.tr/tahsin/PAPERBOX/M17.pdf>

# Özgeçmiş

Adı Soyadı: Hasan İleri  
E-mail (1): y210240003@ogr.ikcu.edu.tr  
E-mail (2): hasanileri@yandex.com

Eğitim:  
2005–2012: Terk Başkent Üniversitesi, Makine Müh. Bölümü L.  
2017–2019 Ahmet Yesevi Üniversitesi, Bilgisayar Programcılığı. Ö.L.  
2019-2021 Anadolu Üniversitesi, Yönetim Bilişim Sistemleri L.  
2021 – Devam İzmir Kâtip Çelebi Üniversitesi, Yazılım Mühendisliği Y.L.

İş Deneyimi:  
2014 – 2017 Asya Halı A.Ş.  
2017 – 2020 Başarsoft A.Ş.  
2020 – 2021 Başkent Gaz Dağıtım A.Ş.  
2021 – 2022 Fastsoft Bilişim Teknolojileri A.Ş.  
2022 – Devam Enerjisa A.Ş.