

ÖZET

WEB TABANLI SUNUCU TAKİP VE GÖRSELLEŞTİRME SİSTEMİ

Emin Soner TÜRK, Mustafa DENİZALTı

Lisans Tez Raporu,

Tez Danışmanı: Dr. Öğretim Üyesi Özgü Can

Haziran 2019, 65 sayfa

Günümüzde şirketler işlevlerini yerine getirmek ve müşteri ile ilgili ürün ve bilgilerin online olarak sunulmasını sağlamak amacıyla sunucu sistemlerini kullanırlar. Farklı işlemler için tutulan sunucular ve aralarındaki hiyerarşik ağ yapısı ve bağlantıları, sunucu sayısı arttıkça giderek karmaşık bir hal almaktadır.

Tez çalışmaları kapsamında; sunucular arasındaki ilişkiyi göstermek, hızlı analiz edilmesini sağlamak, sunucuların sahip olduğu uygulamaları ve üzerinde çalışan siteleri görmek, bunlarla birlikte yönetimini kolaylaştmak amacıyla görselleştirme yapılarak sunucuların sahip olduğu özellikler ve birbirleri arasındaki bağlantılar takip edilebilir hale getirilmektedir.

Sunucular ve aralarındaki ilişkiler; her bir sunucu için dinamik olarak oluşturulan hiyerarşik ağ yapısı sayesinde, sunucunun ilişkisel bağlantıları, barındırdığı siteler ve hiyerarşik yapıdaki konumu görselleştirilmiş olur.

Anahtar Kelimeler: Sunucu Yönetimi, Sunucu Bağıntıları, Hiyerarşik Ağ Yapı, Sunucu Bilgi Sistemi, Sunucu Takibi

ABSTRACT**WEB BASED SERVER MONITORING AND VISUALIZATION
SYSTEM**

Emin Soner TÜRK, Mustafa DENİZALTI

Supervisor: Asst. Prof. Dr. Özgü Can

June 2019, 65 pages

Nowadays, companies use server-named systems in order to perform their functions and to provide online products and information related to the customer. Servers that activated for different operations, hierarchical network structures and connections between them become more complex than the first case as the number of servers increases.

Within the scope of thesis studies; the system shows the relationship between servers and it analyses servers as fast as possible. Further, it lists the applications which installed on server and can able to show sites that working on servers. Common purpose of all this applications is facilitate the management of visualization by the server's features and connections.

The relationship between the servers and their relationship is dynamically generated by the hierarchical network structure for each server. The server's that relational connections, the sites which on server and its hierarchical position are visualized.

Keywords: Server Connections, Hierarchical Network Structure, Server Information System, Server Tracking, Server Management

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
KISALTMALAR DİZİNİ	v
1. GİRİŞ	1
1.1 Tezin Hedefi	3
2. LİTERATÜR ÇALIŞMASI.....	4
3. YÖNTEM VE TEKNOLOJİLER.....	6
3.1 ASP.NET MVC	6
3.1.2 View	7
3.1.3 Controller.....	7
3.2 Javascript	8
3.3 Veritabanı.....	8
3.4 Bootstrap	8
3.5 Kendo UI.....	9
3.6 JQuery	9
3.7 Vis.js	9
3.8 LINQ	10
3.9 Entity Framework Code First Development	11
4. ANALİZ	13
4.1 Bakış Açısı Gereksinimleri	13
4.2 Fonksiyonel Gereksinimler.....	14
4.3 Fonksiyonel Olmayan Gereksinimler.....	14
4.4 Use Case Diagram	15
5. TASARIM	17
5.1 Mimari Görünümü	17
5.2 Yazılım Mimarisi.....	17
5.3 Activity Diagram.....	18
5.4 Deployment Diagram	19

5.5 Prototip Geliştirme.....	20
6. GELİŞTİRİM	25
6.1 Sequence Diagramları	25
6.2 Config Sınıflar	28
6.4 Modal Sınıflar	34
6.5 Database Design/ER Diagram	34
7. SINAMA	36
7.1.1 Birim Testi.....	36
7.1.2 Entegrasyon Testi	36
7.2 Fonksiyonel Olmayan Testler	36
7.2.1 Hacim Testi	36
7.2.2 Performans Testi.....	37
7.2.3 Güvenlik Testi	37
7.2.4 Kurtarma Testi.....	37
7.2.6 Kullanılabilirlik Testi.....	37
7.2.7 Uyumluluk Testi	38
8. KULLANICI KILAVUZU.....	39
9. SONUÇ.....	56
9.1. Gelecek Çalışmalar.....	56
KAYNAKLAR DİZİNİ	57

KISALTMALAR DİZİNİ

Kısaltmalar

JS	JavaScript
MVC	Model View Controller
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
LINQ	Language Integrated Query
UI	User Interface
SQL	Structured Query Language
ER	Entity Relationship
XML	Extensible Markup Language
PHP	Hypertext Preprocessor
JSON	JavaScript Object Notation
CPU	Central Processing Unit

1. GİRİŞ

Web tabanlı sunucu takip ve görselleştirme sistemi; şirketlerin ve organizasyonların sahip oldukları sunucuları yönetebilmelerini, aralarındaki ilişkiyi yapısal diyagram şeklinde görebilmelerini sağlar. Sunucu yöneticileri için geliştirilen bu sistem, sistemin genel görünümünü yansıtarak sunucu haritalanması ve erişim kolaylığı sağlar. Sahip olunan sunucuların sistemsel özellikleri, üzerinde barındırdığı siteler ve yerel programlar, çevrimiçi (*online*) – çevrimdışı (*offline*) olma durumu ve diğer sunucularla ilişkiye sahip olduğu hiyerarşik yapısı görselleştirilir.

Bu görselleştirme ile birlikte sunucunun barındırdığı ve kapsadığı sunucular izlenerek olası bir çalışmama durumunda etkileyeceği sunucular; dolayısıyla etkilenen sitelerin gösterilmesi ile sunucu yönetimi sağlanır. İlişki düzenleme işlemleri için gerekli fonksiyonellik sağlanarak yeni bir ağ oluşturmak ve var olan yapıyı düzenleme işlemi kullanıcıya sağlanmaktadır. Ağ yapısındaki tüm sunucular barındırdıkları siteler ile listelenerek erişim kolaylığı sağlanmaktadır. Merkezi olarak sunucu ağı yönetimi amaçlanmaktadır.

Ağ yapısı dinamik olarak oluşturulur ve her bir site için veritabanındaki bilgi ve ilişki değişimi anlık olarak yansıtılır. Dinamik olarak yaratılmasının verdiği ayrıcalık ile sunuculara ping gönderilerek anlık durumları gösterilir. Yeni bir kullanıcının sistemin fonksiyonellliğini ve işleyişini kavraması için arayüzün; gerekli açıklamalar, ifadeler ve sembollerle olabildiğince kullanıcı dostu (*user friendly*) olması amaçlanmaktadır.

Bu çalışmada, sunucular arasındaki ilişkinin görselleştirilmesi ve bu ilişkilerin aktif olarak düzenlenmesinin sağlanması, sunucuların sistemsel özellikleri ve sunucuların sahip olduğu sitelerin listelenmesi amaçlanmaktadır.

Projede fonksiyonellik açısından gerekli görülen ve efektif olarak kullandığımız, hazır kod kütüphaneleri (*framework*) kullanılmıştır. Bununla birlikte; Microsoft Visual Studio geliştirme ortamında, C# programlama diliyle [1] ve .NET framework’ünün [2] MVC tasarım kalıbı ile geliştirilmiştir.

Verilerin listelendiği grid yapılarında Kendo UI [3] adlı javascript framework'yle, görselleştirme aşamasında ve ağ yapısının listelenmesi ise Vis.js [4] adlı javascript framework'ü ile yapılmıştır. Veritabanı yapısı ise Entity Framework Code First yapısı ile gerçekleştirilmiş olup, sorgular veritabanına LINQ [5] adlı kodlamayla entegre edilmiş sorgulama yöntemi ile atılarak veritabanındaki verilerin çekilmesi sağlanmıştır.

Bu çalışmanın organizasyonu şu şekildedir:

İkinci bölümde; bitirme tezi ile ilgili olarak literatürde yer alan benzer çalışmalar incelenmekte ve sektör araştırmaları yapılarak, tezin bu çalışmalardan farklılıklarını açıklanmaktadır.

Üçüncü bölümde; bitirme tezinde kullanılacak olan yöntem literatüre bağlı olarak anlatılmış, sistemde kullanılacak teknolojiler ve sistem kapsamında seçilen yaklaşım açıklanmaktadır.

Dördüncü bölümde; çalışmanın bkaş açısı gereksinimleri, fonksiyonel gereksinimler, fonksiyonel olmayan gereksinimler ve use case diyagramları detaylı olarak açıklanmaktadır.

Beşinci bölümde; çalışmanın mimari görünümü, yazılım mimarisi, çeşitli yazılım diyagramları ve prototip kısımları detaylı olarak açıklanmaktadır.

Altıncı bölümde; çalışmanın sequence diyagramları, çeşitli yazılım sınıfları ve database design diyagramları detaylı olarak açıklanmaktadır.

Yedinci bölümde; çalışmanın sınandığı fonksiyonel ve fonksiyonel olmayan testlerle birlikte, yapılmış olan uygulamaların detaylı açıklamaları yer almaktadır.

Sekizinci bölümde; çalışmanın çalışma senaryosu resimlerle detaylı olarak açıklanarak, kullanım klavuzu çıkarılmaktadır.

Dokuzuncu bölümde; çalışmadan çıkarılan sonuçlar ve sistemin özeti yer almaktadır.

1.1 Tezin Hedefi

Tez çalışması kapsamında, etki alanından bağımsız olarak geliştirilen model ile aşağıda yer alan maddelerin gerçekleştirilmesi hedeflenmektedir:

- Sunucular arasındaki ilişkileri görselleştirebilmek,
- Sunucunun sahip olduğu siteleri listeleyebilmek,
- Sunucular arası ilişki düzenleme işlemi yapabilmek,
- Sunucuların sahip olduğu donanımsal özellikleri listeleyebilmek.

Tez çalışması kapsamında, geliştirilen modelin kurumsal etki alanına uygulanması ile aşağıda yer alan maddelerin gerçekleştirilmesi hedeflenmektedir:

- Sisteme kayıtlı olan serverların listelenmesini ve analizini yapabilmek,
- Sisteme ait serverların bilgilerinin güvenliğini ve gizliliğini sağlayabilmek,
- İlgili serverin çökmesi durumunda diğer serverlara etkisinin öğrenilmesini ve önlem alınmasını sağlayabilmek,
- Server durumunu izleyebilmek,
- Çöken sunucuların ağ haritası incelenerek; diğer serverlarda ortaya çıkabilecek benzer sonuçların öngörülmesini sağlayabilmek.

2. LİTERATÜR ÇALIŞMASI

Sunucu bilgisayar donanım ekipmanlarından oluşan ve sunucu olarak kullanılan bilgisayarlardır. Kişisel bilgisayarlar gibi sunucu bilgisayarlar da monitör, sistem ünitesi, mouse ve klavyeden gibi donanımsal ekipmanlar içerebilir. Bu bilgisayarlar, ana ya da yönetici bilgisayarları diğer bilgisayarlara bağlama ile birlikte veri alışverini sağlar ve eş zamanlı olarak ağa bağlı her bir bilgisayar için ev sahipliği yapar. Bu nedenle sunucu bilgisayarlar ev sahibi ve istemci arasındaki ilişkiyi yansıtacak şekilde bilinir. Veri dağıtım ve yönetim merkezliği yapar. Özellikle bilgisayarlara erişim ve kullanıcı veritabanı yönetimi, belge düzenleme, yayınılama, web geliştirme ve video işleme uygulanmasını sağlayan özel yazılım uygulamalarının barınımını ve çalıştırılabilirliğini sağlar. Ayrıca, sunucuların hangi amaçla kullanılacağı da önemlidir. Örneğin, web hizmeti veren web sunucuları vardır. Donanım olarak büyük farklılıklar göstermemekle birlikte, sunucular kullanıldıkları alana göre özel yazılımlarla desteklenmektedir. Bunlara örnek olarak Apache, Microsoft IIS ve Abyss, Nginx ve Fastream IQ Web/FTP Server gösterilebilir. Dosya paylaşım, canlı yayın ve hatta çevrimiçi (*online*) oyun hizmeti verenleri de mevcuttur. Tüm bu sunucuların değişik sistem gereksinimleri olacaktır.

Kesinti (Downtime), bir bilgisayar, bilgisayar ağı, çevrimiçi program veya servisin yaşadığı donanım veya yazılım sorunları neticesinde hizmetini sona erdirmesi veya belli bir süreliğine kesilmesidir[7]. Kesinti birçok serviste görebilecek bir olayken web dünyasında sıkılıkla internet sunucularının kesintilerini aktarmak için kullandığımız bir terimdir. Ağ modemleri, internet sunucuları ve çevrimiçi servislerin büyük çoğunluğu servislerinin kullanılabilirlik erişim süresi hakkında kullanıcılar fikir vermek için kesinti veya çalışma süresi olarak yüzdelik dilimlerle belirtirler. Belirtilen bu oran müşterilere servis veya ürünün kalitesi ve ulaşılabilirliği hakkında da fikir sunmuş olur.

Tüm bunlara ek olarak, sunucunun barındırdığı sitelerin ve site bilgilerinin diğer sunucular tarafından da kullanılma durumu vardır. Yani bir sunucu diğerine ev sahipliği yapabilir. Bu bakımdan ele alındığı takdirde, bir sunucunun hizmet verememesi durumunda diğer sitelerin etkilenmesi söz konusu olabilir. Bu durum, büyük kurumsal firmaların ticari faaliyetleri için gerekli olan sitelerin veya onun

barındırdığı bazı özelliklerinin devre dışı kalmasına; dolayısıyla da firmaların maddi anlamda büyük, orta veya küçük ölçekli kayıplarına sebebiyet verebilir.

Literatürde, Sunucu İzleme ve Takip Sistemi (SİTS) [8] adlı bir program bulunmaktadır. Bu programda; iç saldırının tespiti için sunucuların hareketler izlenerek, şüpheli davranışların tespiti amaçlanmıştır. Ayrıca, sunucuların ajanların istenilen verileri veri tabanına kaydetmesi sonrasında ihtiyaç duyulan bilgileri ana ekranada yöneticiye sunmaktadır. Bu sayede, saldırı şüphesi oluşturan davranışlar sergileyen yetkili kullanıcılar, literatürden elde edilen bilgiler ışığında oluşturulan kurallar ve alarmlar yardımıyla tespit edilebilmektedir. Uygulamanın etkinliği, saldırıcıların muhtemel hareket tarzlarını içeren senaryolar doğrultusunda ortaya konmuştur. Uygulamanın kullanımıyla kurum içi saldırının tespitinin kolaylaşması, zararlı kullanıcılar üzerinde caydırıcılık yaratılması, diğer personel üzerinde ise farkındalık oluşturulması hedeflenmektedir.

Bu çalışmada geliştirilen uygulamada ise, kurum içi saldırılardan çok sunucuların yayınladığı sitelerin çökmesinin engellenmesi ve buna yönelik tedbirlerin alınması esas alınmıştır. Ayrıca her iki uygulamada da yönetim paneli ve sunucu izleme sistemi olmakla birlikte sunucuya kurulan uygulamalar da takip edilebilmektedir. Bu çalışmada literatürden farklı olarak, sunucuların aralarındaki ilişkilerin görselleştirilmesi ve ilişkilerin diğer sunucuları ve onların yayınladığı siteleri nasıl etkilediği analiz edilerek sitelerin çökmesi durumunda firmaların maddi veya hizmet anlamında kayba uğramasının önüne geçecek önlemler alınabilmektedir.

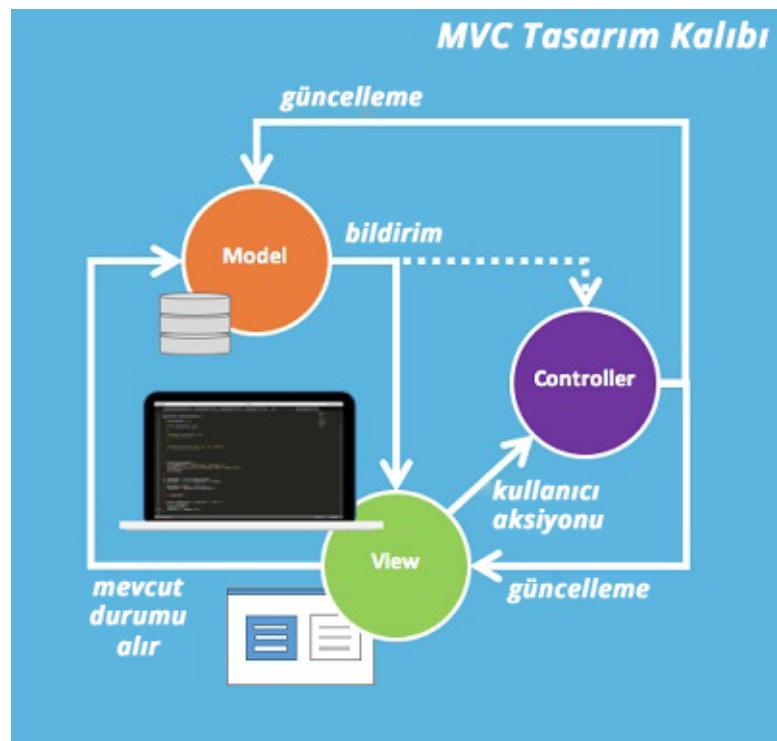
Yapılan tüm bu çalışmalar incelendiğinde, geliştirilen sistemin kurumsal veya ticari bir faaliyette bulunup; sitelerinin veya site servislerinin kullanım dışı kalmasını engellemek isteyen firmalara hitap etmektedir. Bununla birlikte, firmaların maddi kayıp veya iş kaybı anlamında zarara uğramalarını önleyerek; işlemlerini yaptığı site veya servislerin devamlılığını sağlamak esas amaç olarak edinilmiştir.

3. YÖNTEM VE TEKNOLOJİLER

Sistemin, bir web sitesinin bir parçası olarak, sistem yöneticileri ve kullanıcılarının kullanması amacıyla geliştirilmesi amaçlanmaktadır. Bu bağlamda, bu site için gerekli görülen ve kullanılan teknolojiler listelenmiş ve bu teknolojilerle ilgili kısa bilgiler verilmiştir.

3.1 ASP.NET MVC

Sistemin ASP.NET çatısı altında çalışan bir site olması öngörlmektedir. Yapısal pattern için Asp.NET MVC [2] adlı framework kullanılacaktır. ASP.NET MVC, MVC pattern’ini ASP.NET’e eklemek için Microsoft’un geliştirdiği framework’tür. Uygulama geliştirmede (özellikle web uygulaması geliştirmede) önemli yere sahip mimari desenlerden biridir. Şekil 3.1’de MVC çalışma mekanizmasını genel olarak gösteren şema gösterilmektedir [20]. Bu kısımda, veritabanından alınan model verileri; controller aracılığıyla çeşitli sebeplerle (manipüle edilmek gibi) kullanılmakta olup, view kısmında ise ekranda listelenmektedir.



Şekil 3.1 MVC Design Pattern Döngüsü

3.1.1 Model

MVC kalıbında model, uygulama verisinin veya durumunun saklandığı yerdir. Genellikle veritabanı veya xml/json dosyası formatındadır.

Model, veri katmanını (database, xml, json dosyası, vb.) uygulamadan izole eder, böylece diğer katmanlarda veri katmanının neresi olduğunun bilinmesine gerek kalmaz.

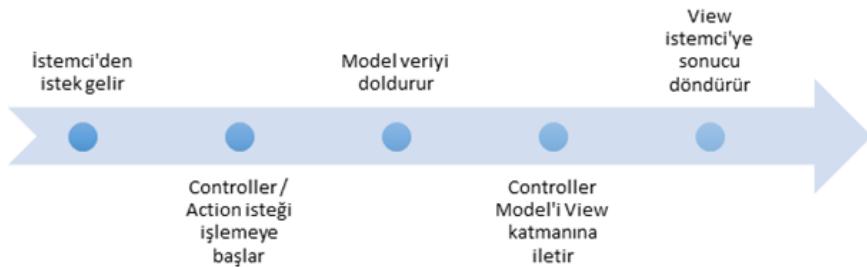
Model katmanı sıkılıkla Entity Framework, Nhibernate, LLBLGen, vb. gibi araçlar kullanılarak oluşturulur.

3.1.2 View

View, istemcinin (*client*) gördüğü arayüzü içeren katmandır. Genellikle model katmanındaki verinin kullanılması ile oluşturulur. View katmanın model ve controller katmanlarından ayrılması ile arayüz değişikliklerinin uygulamanın diğer katmanlarını değiştirmeye gerek kalmadan yapılabilmesi sağlanmıştır. View katmanında HTML5, CSS3 ve JavaScript gibi teknolojiler kullanmak mümkündür. Bu teknolojilerle masaüstü ve mobil tarayıcılarda çalışabilen uygulamalar geliştirilebilmektedir.

3.1.3 Controller

Controller; istemciden gelen isteği işlemek, model ve view katmanları arasında köprü olmak gibi görevleri yerine getirir. Controller içerisinde bir veya daha fazla Action olabilir, genellikle her Action bir web sayfası üretmek için kullanılır. Şekil 3.2'de MVC tasarım kalıplı bir sistemin örnek çalışma senaryosu gösterilmektedir [21]. İstemciden gelen isteği Controller işler. Model gerekli veriyi Controller'a döndürür. Controller View'a iletir.



Şekil 3.2 MVC Pattern Uygulanma Senaryosu

ASP.NET MVC sayesinde; .Net framework dilleri ile MVC kalıbı kullanılarak hızlı çalışan, test edilebilir, tekrar kullanılabilir parçaları olabilen bir web uygulaması geliştirilebilmesi amaçlanmaktadır.

3.2 Javascript

JavaScript, genellikle web tarayıcılarında kullanılmakta olan bir betik dilidir. JavaScript ile yazılan istemci tarafı betikler sayesinde tarayıcının kullanıcıyla etkileşimi, tarayıcının kontrol edilmesini, asenkron bir şekilde sunucu ile iletişime geçilmesini ve web sayfası içeriğinin değiştirilmesini sağlar. JavaScript, Node.js gibi platformlar sayesinde sunucu tarafında da yaygın olarak kullanılmaktadır.

3.3 Veritabanı

Veri tabanları aralarında belirli ilişki bulunduran bilgilerin depolandığı alanlardır. Veri artışı ile birlikte bilgisayarda bilgi depolama ve bilgiye erişim konularında güncel yöntemlere ihtiyaç duyulmuştur. Veritabanları; büyük miktardaki bilgileri depolamada geleneksel yöntem olan dosya-islem sisteme alternatif olarak geliştirilmiştir. Sistemin sahip olduğu sunucuların bilgilerinin, sunucular arası ilişkilerin, sisteme erişimde bulunan kullanıcıların erişim yetkisi gibi bilgilerin kayıt altında tutulması için veritabanı kullanılması amaçlanmaktadır. Tez projesinde veritabanı mimarisi Code First yaklaşımı kullanılarak tasarlanmıştır.

3.4 Bootstrap

Açık kaynak kodlu, ücretsiz bir CSS framework olan Bootstrap [11], tasarım aracıdır. Telefonlar, tabletler ve bilgisayarlar için farklı ve cihaz büyülüğüne

uygun şekilde sitenin gözükmekini sağlayan tasarımlar yapılabilmektedir. Bir site için gerekli olan tüm elementleri (form öğeleri, etiketler, tablolar, uyarı ve bilgi metinleri, navigasyon bar, sayfalandırma modülü, açılan menüler, farklı özellikte butonlar vb. birçok tasarım ögesini) içinde barındıran Bootstrap, tasarım yaparken bu hazır elementleri kullanarak bütün cihazlara uygun tasarımlar geliştirmeyi sağlar.

3.5 Kendo UI

Kendo UI framework’ü bünyesinde çeşitli teknolojileri barındırmaktadır. *HTML5 ve JavaScript teknolojilerini temel alan Microsoft .NET Framework, JavaServer Pages, PHP yapılarını server-side kısmında destekleyen bir UI Framework’tür* [12]. JavaScript kütüphanesi olan jQuery, AngularJS, ReactJS, KnockoutJS gibi alt yapıların kullanımını desteklemektedir.

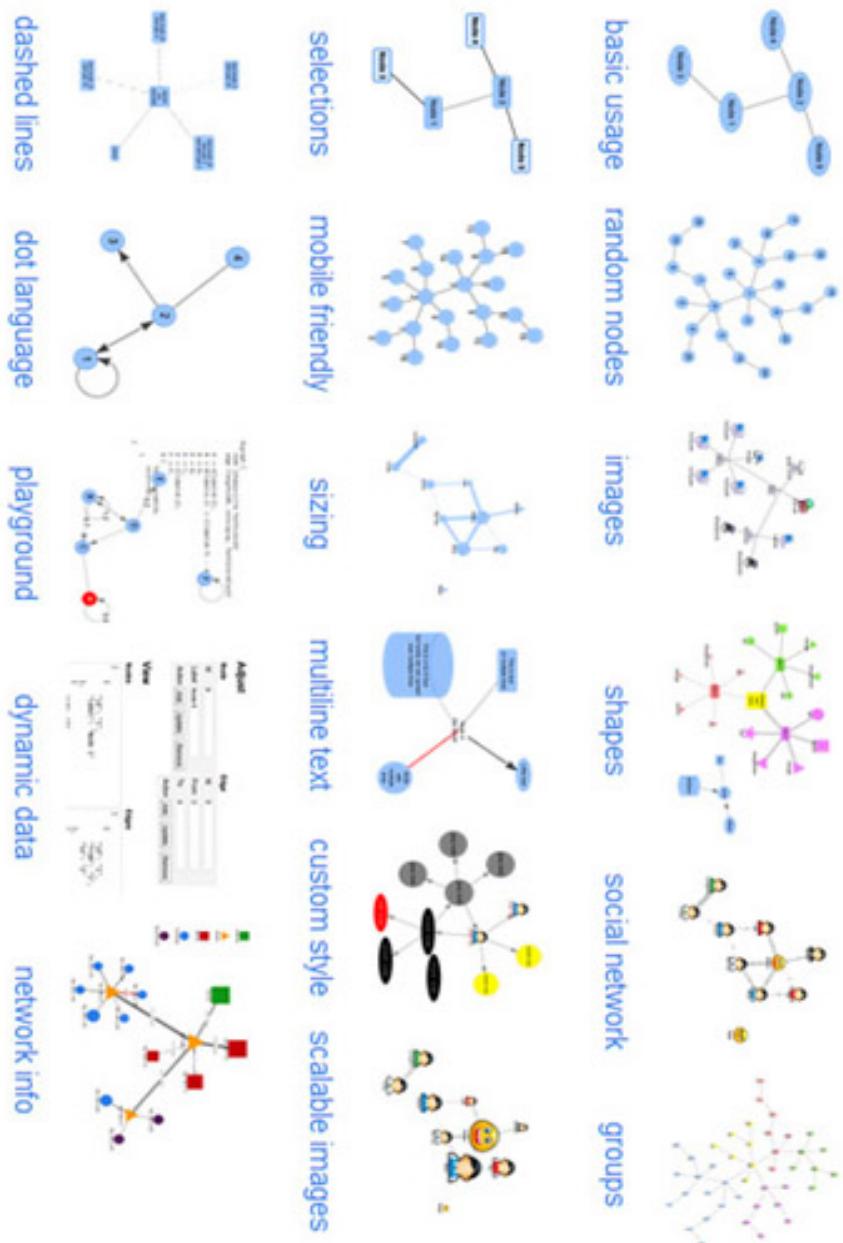
En önemli özelliği birçok arayüz olarak kullanılan Grid, Autocomplete input, ListView, TreeList, DatePicker/DateTimePicker gibi (70+ UI Bileşenleri) bileşenleri içinde barındıran bir Framework olup; yazılımcının bu parçaları bir daha yazması yerine; bunları kullanıp, bu parçalar üzerinde kişiselleştirilme yapmasını sağlamıştır. Mimari bir yapı oluşturulurken Kendo UI, ön yüz tarafında en çok tercih edilen ürünlerden birisidir.

3.6 JQuery

jQuery bir javascript kütüphanesidir. Başka bir ifadeyle; *jQuery, "daha az yazın, daha fazlasını yapın" sloganına sahip bir JavaScript kütüphanesidir* [13] . JQuery'nin amacı, web sitelerinde JavaScript'i kullanmayı kolaylaştırmaktır. Bununla birlikte, birçok satır JavaScript kodu gerektiren birden fazla ortak görevi yerine getirebilmekte ve bunları tek bir kod satırıyla çağırılmamızı sağlamaktadır.

3.7 Vis.js

Vis.js; dinamik, tarayıcı tabanlı bir görselleştirme kütüphanesidir. Büyük miktardaki dinamik verileri işlemek ve verilerin manipülasyonunu ve etkileşimini sağlamak için tasarlanmıştır. Kütüphane Şekil 3.3’te gösterilen DataSet, Timeline, Network, Graph2d ve Graph3d vb. bileşenlerden oluşur [22].

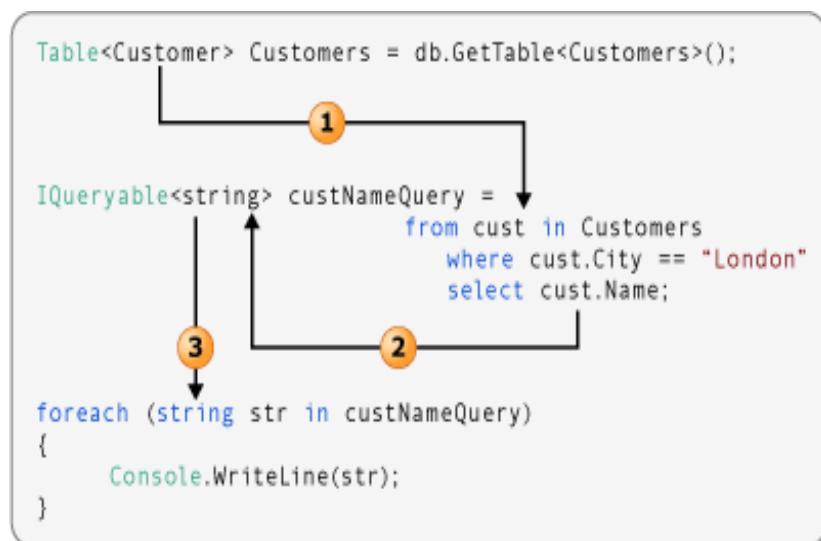


Şekil 3.3 Vis.js görselleştirme kütüphanesiyle oluşturulmuş örnek yapılar

3.8 LINQ

Microsoft'un Visual Studio 2008 ve .NET Framework 3.5 ile beraber kullanıma sunduğu LINQ, İngilizcesi Language Integrated Query, Türkçesi Programlama Diliyle Bütünleştirilmiş / Entegre Edilmiş Sorgu olan; Haskell, XML, HTML ve

SQL programlama dilleri gibi bildirimsel yazım şekli (sözdizimi, syntax) kullanan bir bileşen / teknolojidir. LINQ, programlama diline SQL (Yapısal Sorgu Dili - Veritabanındaki kayıtlarla ilgili işlemleri yapmak için kullanılan bir bildirimsel programlama dili.) biçimini veri sorgulama yeteneği ekler. LINQ ile SQL Server Veritabanları, XML belgeleri, ADO.NET Veri Kümeleri (Dataset) ve hafızada bulunan koleksiyon türlerindeki verileri sorgulayabilirsiniz. Şekil 3.4'te örnek bir linq sorgusu gösterilmektedir [10]. Bu şeitin 1. kısmında, tablodan alınan customer adlı class yapısına sahip veriler, databasedeki customers verilerinden tablo halinde alınmaktadır. 2. kısmında ise bir SQL sorgusu gibi; programlama dili aracılığıyla (*compiler*), LINQ Framework'ü kullanılarak istenen özelliklere sahip verilerin filtrelenmesi sağlanmış olup, ardından da 3. kısmında ekrana yazdırılması işlemi yapılmaktadır.

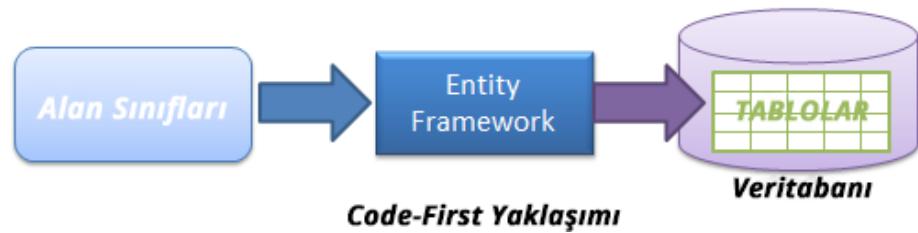


Şekil 3.4 Örnek bir LINQ sorgusu

3.9 Entity Framework Code First Development

MVC projelerinde veritabanı işlemlerinin yapılması için klasik olarak kullanılan Microsoft SQL Server veritabanı oluşturma işlemi yerine Entity Framework yapısını kullanarak veritabanı yapısını MVC projesi içerisinde bulunan Model klasörünün içerisinde oluşturma işlemi Code First olarak adlandırılmaktadır. Şekil 3.5'te görüldüğü üzere model kısmında oluşturulmuş olan sınıflar, entity framework aracılığıyla veritabanındaki tablolara otomatik şekilde

dönüştürülmektedir [23]. Veritabanındaki tablolar için uygun şekilde nesneler oluşturan ve aralarındaki ilişkiyi kuran teknolojidir.



Şekil 3.5 Code First yaklaşımı

4. ANALİZ

4.1 Bakış Açısı Gereksinimleri

Sistemimizde 2 farklı kullanıcı tipi bulunmaktadır. Bunlar:

1. Normal kullanıcılar,
2. Yönetici tipi kullanıcılardır.

Normal kullanıcılar; sistemi, normal bir şekilde kayıtlı sunucu bilgilerini manipüle etmeden; sadece sunucu bilgilerini görüntüleme amacıyla sistemi kullanmaktadır.

Yönetici tipi kullanıcılar ise; sistemdeki sunucu ve kullanıcı bilgilerini manipüle ederek, bunlara müdahalelerde bulunabildiği gibi; aynı zamanda var olan bilgiyi de görüntüleyebilmektedirler. Kullanıcı tiplerinin, yapmak istedikleri bakış açısı gereksinimleri:

Normal kullanıcı:

- Sunucu listesine erişmek ister,
- Sunucunun ilişkilerinin bulunduğu ağı görmek ister,
- Sunucunun barındırdığı site listesini görmek ister,
- Sunucuya erişim olup olmadığını görmek ister,
- Sunucuda kurulu olan uygulamaları görmek ister,
- Sunucunun donanımsal bilgilerini görmek ister.

Yönetici:

- Sunucu listesine erişmek ister,
- Sunucunun ilişkilerinin bulunduğu ağı görmek ister,
- Sunucunun barındırdığı site listesini görmek ister,
- Sunucu listesine yeni sunucu ekleyebilmek ve silebilmek ister,
- Var olan sunucuya ait bilgileri güncellemek ister,
- Sunucuda kurulu olan uygulamaları görmek ister,
- Sunucunun donanımsal bilgilerini görmek ister,
- Sunucuya erişim olup olmadığını görmek ister,
- Sunucuya, diğer sunucularla arasında olan ilişkileri ekleyebilmek ve sunucunun var olan ilişkilerini düzenleyebilmek ister.

4.2 Fonksiyonel Gereksinimler

Sistemin gerektirdiği, fonksiyonel olan birçok gereksinim vardır. Bunlar temel olarak:

- Sistem, kullanıcıya sunucu listesini başarılı bir şekilde göstermelidir,
- Sistem; kullanıcının yetkisini (yönetici veya normal kullanıcı) tanıyalırmalıdır,
- Sistem; istenen ilişkisel sunucu ağını ekranda göstermelidir,
- Sistem; sunucu üzerindeki site listesini bir kısıtlama yok ise, gösterebilir;
- Sistem; yöneticinin ilişki ekleme, düzenlemesine izin vermelidir;
- Sistem; yöneticinin yeni sunucu ekleme ve var olan sunucuları düzenlemesine izin vermelidir,
- Sistem; yöneticinin kullanıcılarla yöneticilik özelliği vermesine izin vermelidir;
- Yöneticiler; sunuculara ait donanımsal bilgilere erişebilmelidir,
- Kullanıcılar; sunucular üzerinde kurulu programların listesine erişebilmelidir,
- Sistem; ilişki ağında sunucuların bağlı olup olmadığını göstermelidir,
- Sistem; sunucu bilgilerinin yedeğini tutmalıdır,
- Sistem, ilişki ağının fotoğraf olarak indirilebilir olmasını sağlamalı ve izin vermelidir.
- Sistem sunucu listesinin excel dosyası olarak indirilebilir olmasını sağlamalıdır.

4.3 Fonksiyonel Olmayan Gereksinimler

Sistemin gerektirdiği, fonksiyonel olmayan birçok gereksinim vardır. Bunlar temel olarak:

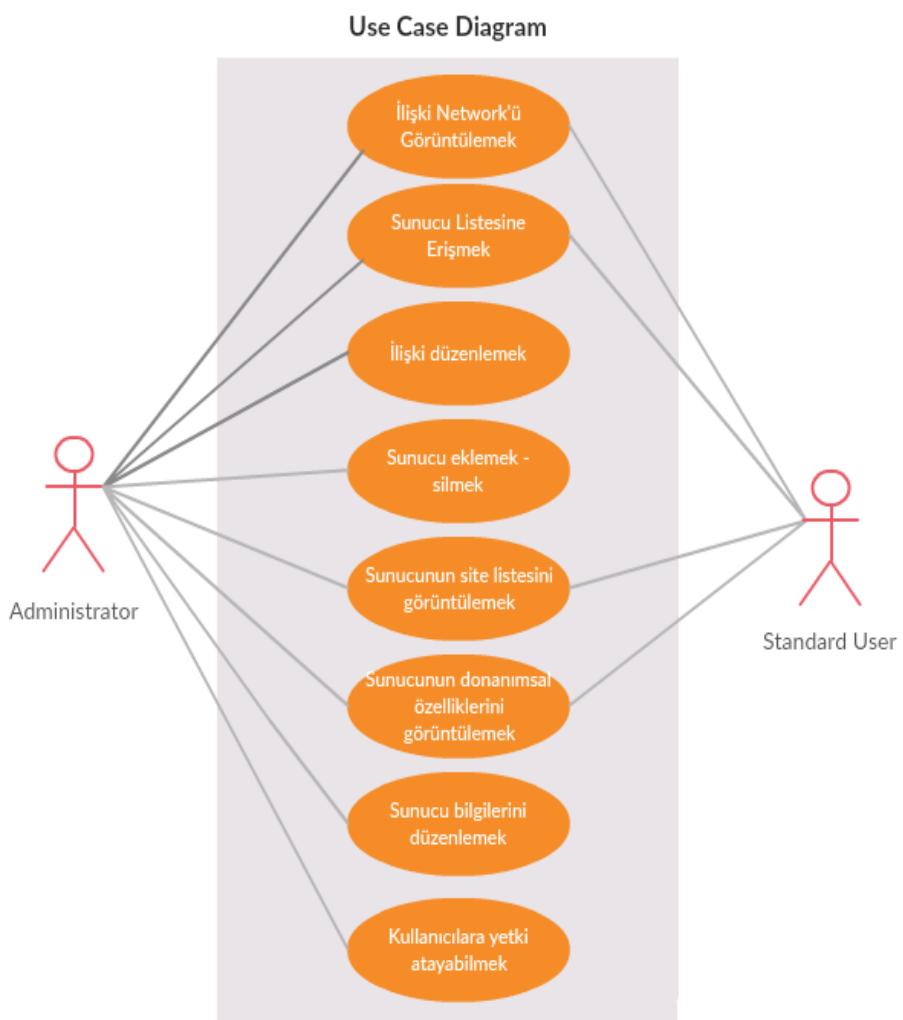
- Sunucu ve kullanıcı verilerini içeren bir veritabanı sistemi olmalı,
- Tepki süresi 15 saniyenin altında olmalı,
- Yeni bir kullanıcı uygulamayı nasıl kullanacağını 10 saniye içerisinde öğrenebilmelidir,
- Sistem tüm tarayıcıları desteklemelidir,
- Sistemin kabul edilebilir arıza oranı 1: 1000'in altında olmalıdır,
- Herhangi bir günde 06.00 ile 23.50 arasında, toplam sistem arıza süresi 10 dakikayı geçmemelidir,

- Sistem Visual Studio ortamında; Asp.net MVC design patterni kullanılarak, C# dili ile geliştirilmelidir.

Bu gereksinimlere bakarak; kullanıcı istekleri ön planda tutulup, analizi yapıldıktan sonra sistem tasarıımı oluşturulma yoluna gidilmiş ve yazılım mimarisi oluşturulmuştur.

4.4 Use Case Diagram

Use case diyagramı amacı, kullanıcının bir sistemle etkileşime girebileceği farklı yolları göstermektir. Yazılım mimarisi tasarlanırken, ilk başta istek analizi yapılmış ve kullanıcı tiplerine göre şekildeki use case diagram çıkartılmıştır.



Şekil 4.1 Projenin kullanıcı gruplarına göre Use Case'leri

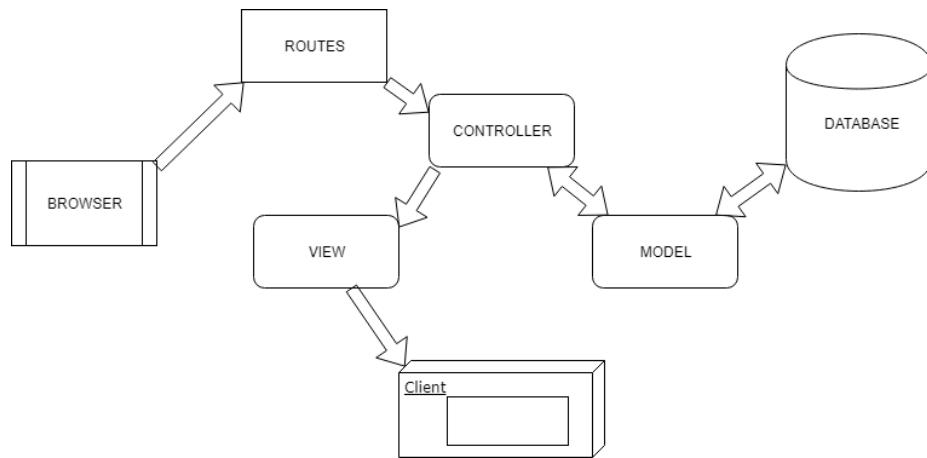
Kullanıcılar; sistemdeki rollerine göre, şekil 4.1'de belirtilen sistem fonksiyonlarını yapmak istemektedirler. Yöneticiler, normal kullanıcıların sahip olduğu fonksiyonlara ek olarak sistem yönetimi için gerekli fonksiyonlara da sahiptir. Yöneticiler, kullanıcıların erişebildikleri fonksiyonları düzenleyebilmektedir.

5. TASARIM

5.1 Mimari Görünümü

Projenin mimari görünümü Şekil 5.1'de gösterilmiştir. Bu şekilde, uygulamada kullanılacak verileri tutmak; ihtiyaç duyulan nesne modelleri, bu web sayfalar ve model ile görüntüler arasında bağlantı sağlayan denetleyicileri tutmak için veritabanı bulunur. Browserden alınan veri Controller'a ulaşır. Model katmanı aracılığıyla Database'den gerekli veriler getirilir. Controller aracılığıyla View katmanına veriler aktarılır ve Client'a veri sağlama işlemi tamamlanmış olur.

Ayrıca aynı şekilde; browser, route, client, view, controller, model ve database kısımlarının koordineli bir şekilde çalışmasıyla, veri kullanıcıya dinamik bir şekilde tekrar tekrar gösterilmektedir.



Şekil 5.1 Mimari Görünümü

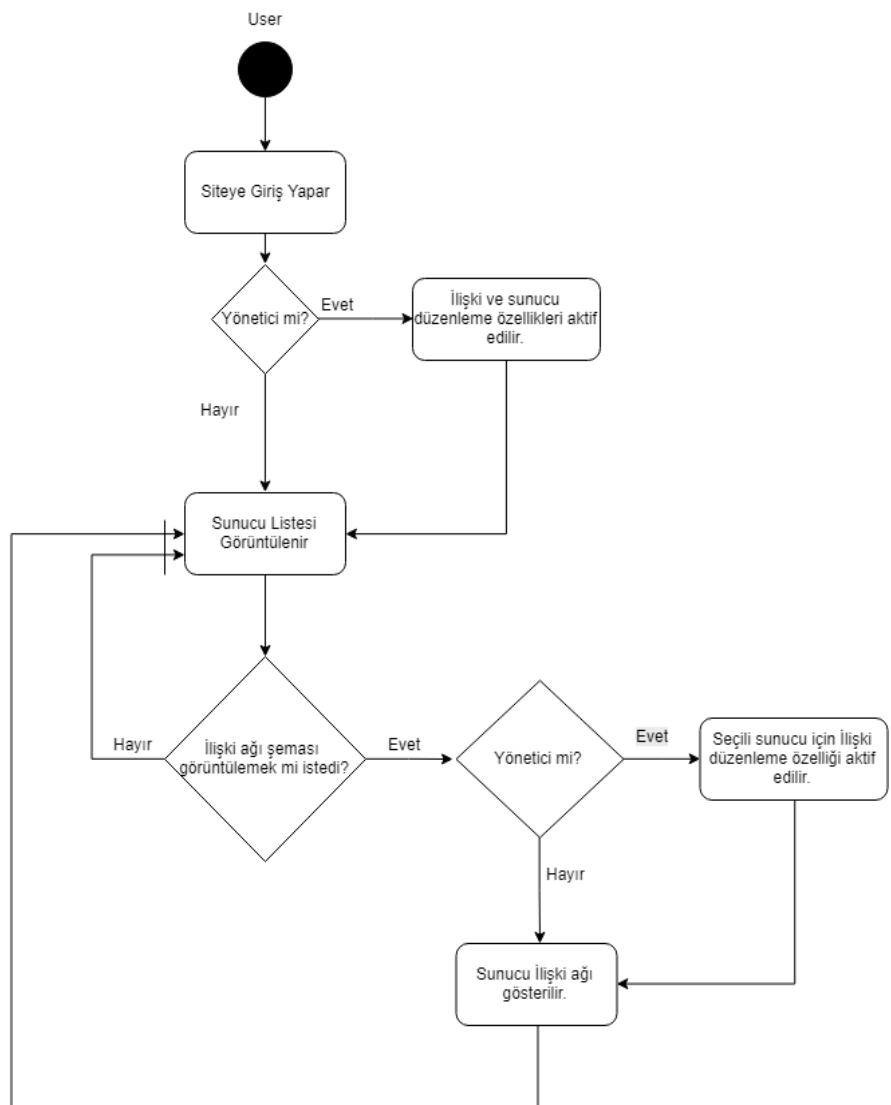
5.2 Yazılım Mimarisi

Sistemin yazılım mimarisi; Activity Diagram ve Deployment Diagram kullanılarak tasarlanmıştır.

5.3 Activity Diagram

Activity diagram, sistemin dinamik yönlerini tanımlamak için UML diyagramındaki bir diğer önemli davranış diyagramıdır. Aktivite şeması, temel olarak bir aktiviteden diğer aktiviteye olan akışı modelleyen ileri bir akış şeması sürümürdür.

Bu çalışma kapsamında gerçekleştirmeye başlanmadan önce, activity diagram üzerinden sistemin nasıl çalışacağına dair bir taslak çıkarılmıştır. Şekil 5.2’te olay döngüsü gösterilmektedir.



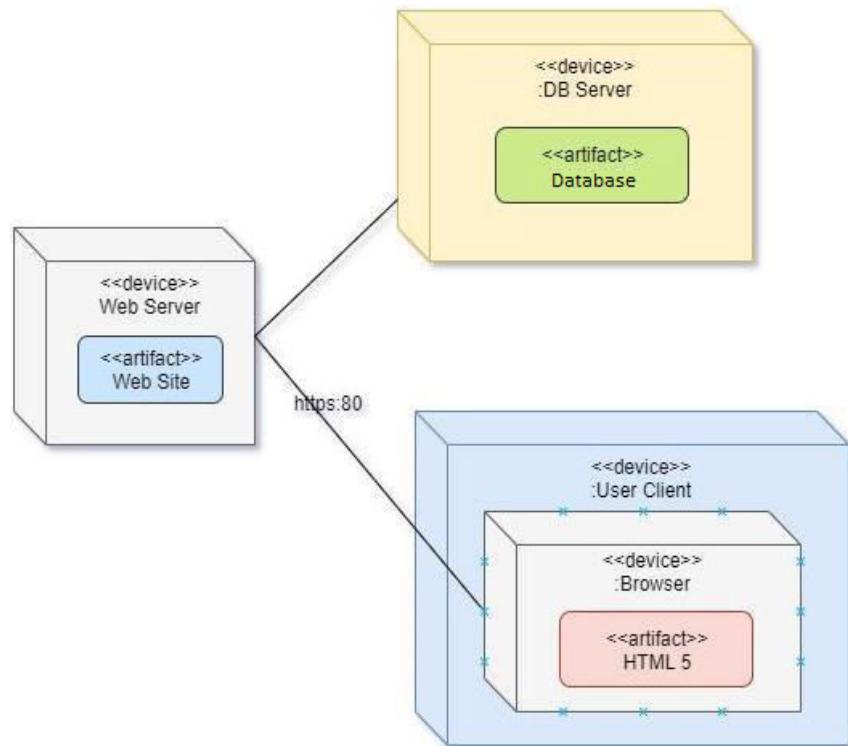
Şekil 5.2 Activity diyagramı

Şekil 5.2'de verilen şemada, çalışmanın activity diyagramı gösterilmektedir.

Bu kısımda, kullanıcının siteye giriş yapması ile sistem aktif olmaktadır. Bu işlemden sonra giriş yapan bu kullanıcının yönetici olup olmadığını sorgulanır. Eğer kullanıcı aynı zamanda yönetici ise sunucu düzenleme özelliği aktif edilir. Yönetici değil ise, kullanıcıya normal sunucu listesini görüntülenir. Ağ şeması görüntüleme isteğinin gelmesi durumunda ağ şeması görüntülenir, buna ek olarak kullanıcı yönetici ise ilişki düzenleme özelliği aktif hale getirilir. Bu işlem sonucunda, sistem sunucu listesini tekrar görüntüler ve başarılı senaryo böylelikle tamamlanmış olur.

5.4 Deployment Diagram

Deployment diyagramlar sistemin donanım ve yazılım ilişkilerinin fiziksel gösterimidir. Deployment diyagramlar bağlantılar ve nodelardan oluşur. Node sistemin donanım parçasını ifade eder. Sistemin donanım topolojisini gösterir. Mimari spesifikasyonun bir parçası olarak yapılandırılır. Şekil 5.3'te sistemin sunucu kısmında database kısmı, web server tarafında web site dosyaları, user client tarafında ise verilerin tarayıcı yoluyla istemciye传递 iletimi gösterilmektedir.



Şekil 5.3 Projenin deployment diyagramı

5.5 Prototip Geliştirme

Yazılım prototiplemesi, yazılım uygulamalarının prototiplerini oluşturma aktivitesidir, yani geliştirilmekte olan yazılım programının eksik versiyonları da denilmektedir. Yazılım geliştirmede ortaya çıkabilecek bir aktivitedir. Bir prototip tipik olarak nihai ürünün sadece birkaç yönünü simüle eder ve tamamen farklı olabilir.

Prototiplemenin birçok faydası vardır. yazılım tasarımcısı, projenin başındaki kullanıcılarından değerli geri bildirimler alabilir. Müşteri, yapılan yazılımın, yazılım programının yapılandırıldığı yazılımın özelliklerine uyup uymadığını karşılaştırabilir. Ayrıca, yazılım mühendisine başlangıçtaki proje tahminlerinin doğruluğu ve önerilen son tarihlerin ve kilometre taşlarının başarılı bir şekilde karşılanıp karşılanamayacağı konusunda bazı bilgiler sağlar.

Şekil 5.4'de sunucu listesi yönetici girişyle listelenmiştir. Kullanıcı buradan ilgili özelliğin simgesine tıklayarak ilgili sayfanın açılmasını ve ilgili işlemlerin yapılmasını sağlayabilir.

Bağlı sunucular özelliğine sahip simgeye erişim yaptığımız zaman, ilgili örnek sunucunun ağ görüntüsü ve sunucuya ilişkili diğer sunucular görüntülenmektedir. Şekil 5.5'de görüldüğü gibi, alt kısmda Sunucu IP ve Sunucu Adı kolonlarına sahip, site bilgilerini içeren Kendo UI Grid yapısı görüntülenmektedir.

Sunucu performans bilgileri sekmesine erişim yapıldığı zaman ise, ilgili sunucunun CPU, Disk, Memory gibi özellikleri yeni açılan ekranda şekil 5.6'da görüldüğü gibi listelenmektedir.

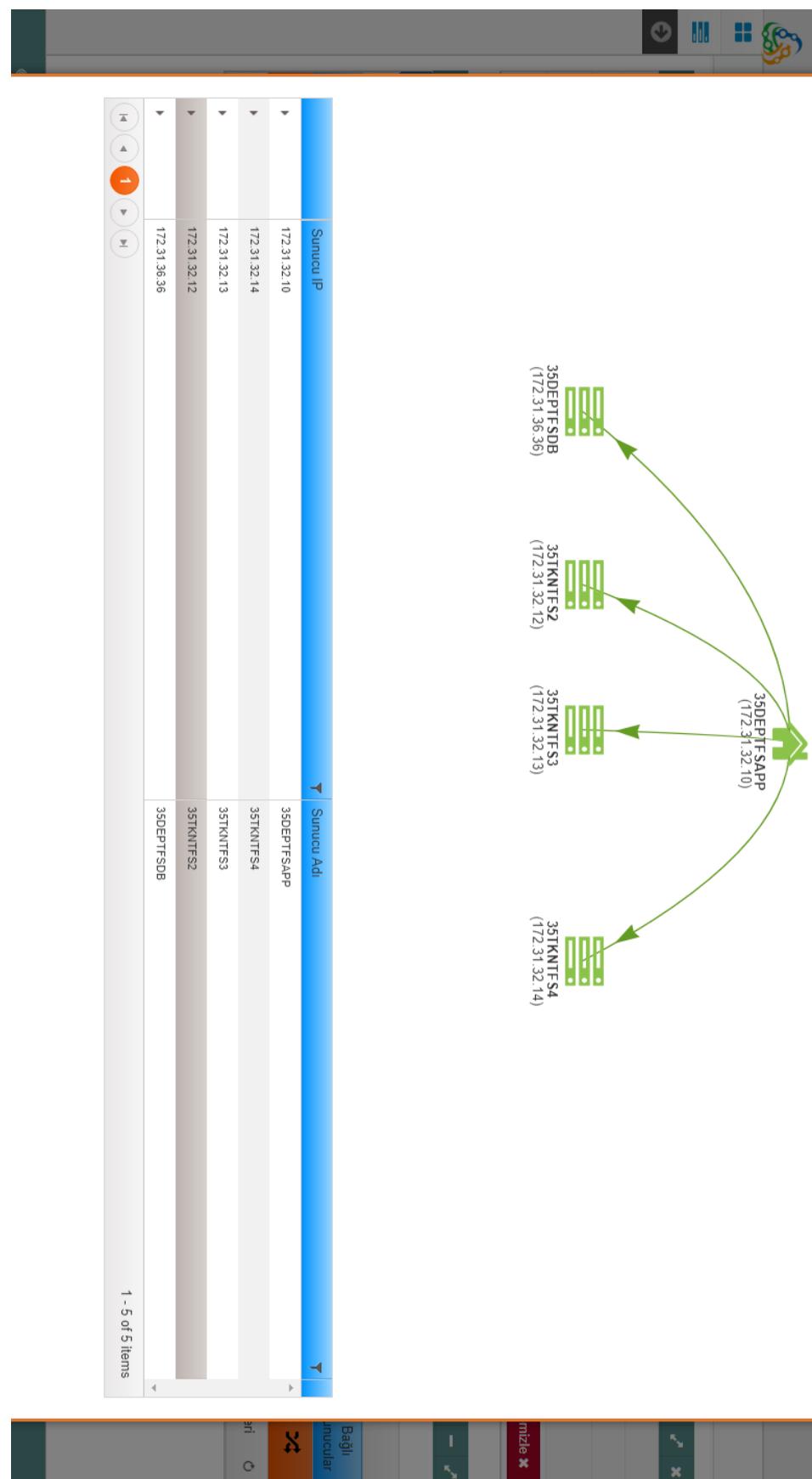
The screenshot shows the BIZWATCH Server application interface. At the top, there are navigation icons for Home, Servers, Applications, and Help. Below the header, there's a search bar labeled "Search Criteria" with dropdowns for "Server Tipi" (Selected), "Lokasyon" (Selected), and "İşletim Sistemi" (Selected). To the right of the search bar are buttons for "Ara Q" (Search) and "Temizle ✖".

The main area displays a table titled "List" with two tabs: "Yeni Server" (selected) and "Güncelle". A "Excel'e aktar" (Export to Excel) button is located above the table. The table has columns: Lokasyon, Server Tipi, İşletim Sistemi, Server IP, Makine Adı, Açıklama, Sorumlu Ekle, Veri Duzanla, Veri Sil, Veri Ping, Veri Stüler, Kurulu Uygulamalar, Performans, and Bağlı Sunucular.

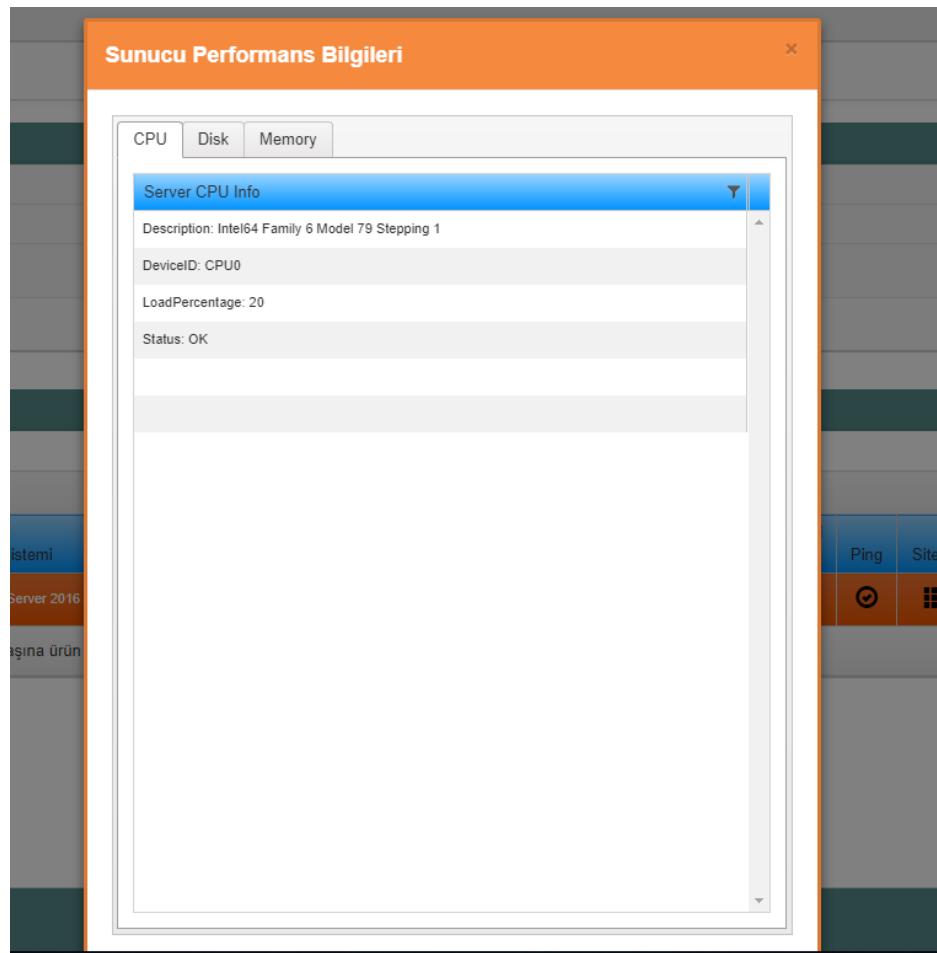
The table contains the following data:

Lokasyon	Server Tipi	İşletim Sistemi	Server IP	Makine Adı	Açıklama	Sorumlu Ekle	Veri Duzanla	Veri Sil	Veri Ping	Veri Stüler	Kurulu Uygulamalar	Performans	Bağlı Sunucular
DEPARKIBIMAR	Canlı	Linux	10.10.10.10	34sibtest3									
DEPARKIBIMAR	Test	Windows Server 2008 R2 Enterprise Edition	10.35.104.40	35TKNIBIMARS	Binar Kurulumumam bulunduğu server dir.								
DEPARKIBIMAR	Canlı	Windows Server 2012 R2 Enterprise Edition	10.35.104.47	35DEPRRSRV	Print Server								
DEPARKIBIMAR	Developer	Windows Server 2012 R2 Enterprise Edition	10.35.104.71	35depr01	Depark için oluşturumış olan Terminal Server - YNA Terminal Server								

Şekil 5.4 Sunucu Listesinin Genel Görünümü



Şekil 5.5 Sunucu ile İlişki Sunucuların Ağ Görüntüsü ve Bağlantılı Siteler Listesi



Şekil 5.6 Örnek Bir Sunucunun Donanım Bilgileri

Ağ görüntüsünün hemen altındaki kolonda listelenen, ilgili sunucu IP'li sunuculardan herhangi bir tanesine erişim yapıldığı zaman, sunucu içerisinde barındırılan site bilgilerinin listesi şekil 5.7'de gösterildiği gibidir.

Sunucu IP	Sunucu Adı
172.31.32.10	35DEPTFSAPP
IIS Sites	
Default Web Site	
deneime	
Team Foundation Server	
TFSAPI	
TFSNotifyService	
TFSReportBI	
TFSWICChangedEvent	
WICChanged	
1 - 8 of 8 items	
172.31.32.14	35TKNTFS4
172.31.32.13	35TKNTFS3

Şekil 5.7 Örnek Bir Sunucuda Barınan Örnek Site Listesi

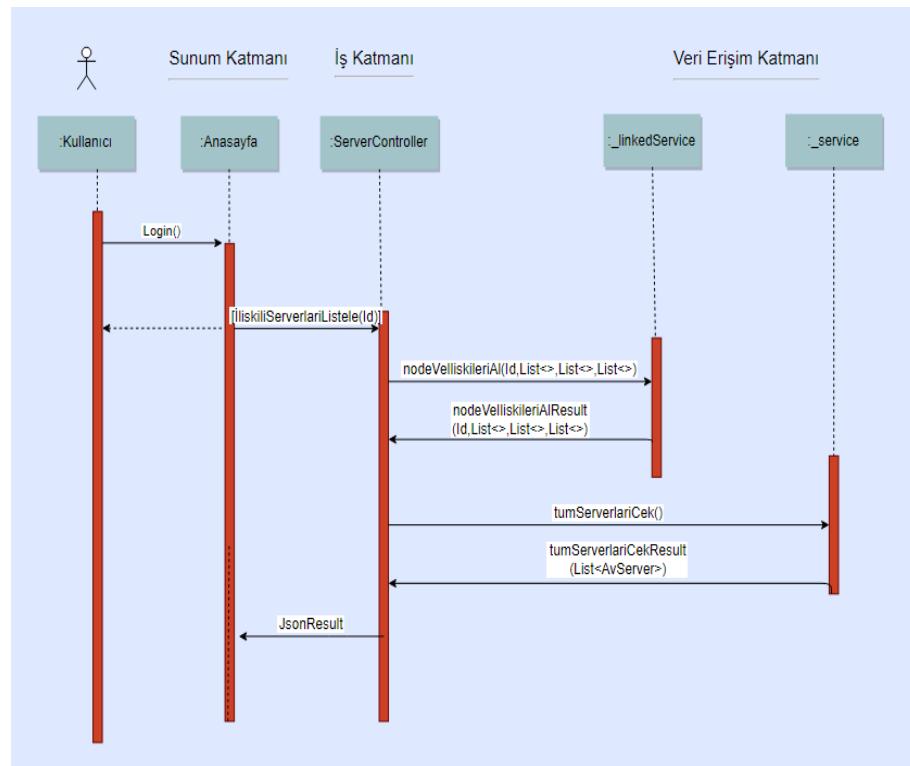
6. GELİŞTİRİM

Proje geliştirilirken, Sequence Diyagramlar, Config Sınıflar, Controller Sınıflar, Modal Sınıflar, Database Design/ER Diyagram gibi yapılara gereksinim duyulmuştur.()

6.1 Sequence Diagramları

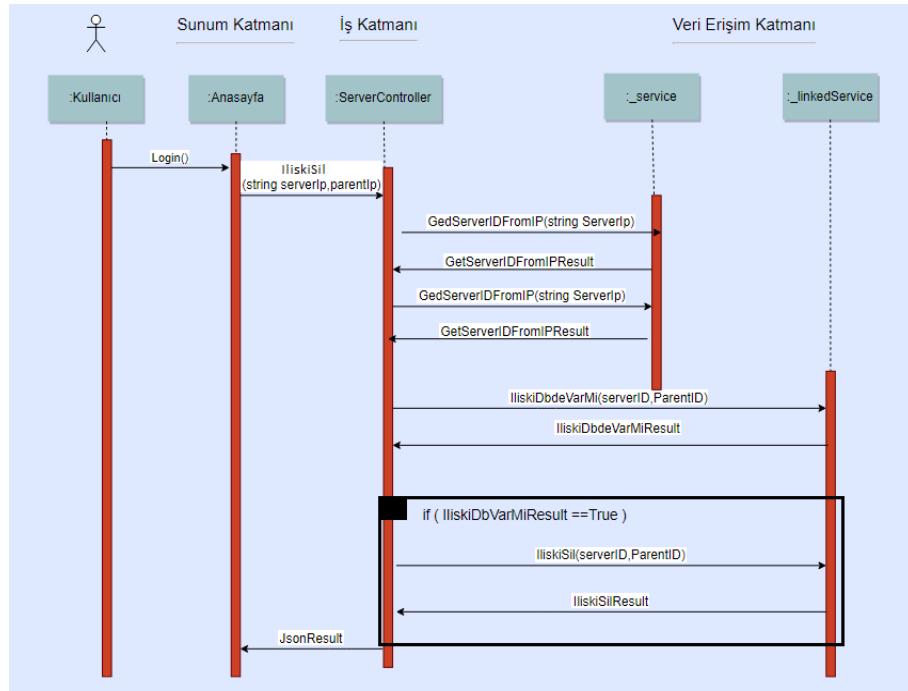
Sequence diagram, zaman dizisinde düzenlenen nesne etkileşimlerini gösterir. Senaryoda yer alan nesneleri ve sınıfları ve senaryonun işlevsellliğini gerçekleştirmek için gereken nesneler arasında değiştirilen mesajların sırasını gösterir. Sıra diyagramları, geliştirilmekte olan sistemin mantıksal görünümünde kullanım durum gerçekleştirmeleri ile ilişkilendirilir. Şekil 6.1, 6.2 ve 6.3'de sistemi örnekleyen şemalar yer almaktadır.

Şekil 6.1'de kullanıcı, sunum katmanındaki login fonksiyonunu çağrımaktadır. Bu çağrıının sonucunda, iş katmanında bulunan serverController controller sınıfı; seçilen IP'li sunucunun id'sini alarak İlliskiliServerlariListele fonksiyonunu çağrımaktadır. Bu controller veri erişim katmanında bulunan _linkedService classına ait olan nodeveİlliskileriAl fonksiyonunu çağırarak ilgili sunucuya ve onun uzantılarının sahip olduğu tüm sunucuları ve ilişkileri döndürür. Ardından _service class'ına ait olan tumServerlariCek fonksiyonu ile veritabanında kayıtlı olan tüm sunucu verileri _service classından döndürülerek serverController classında JSON veri tipinde görüntülenmesi sağlanmaktadır.



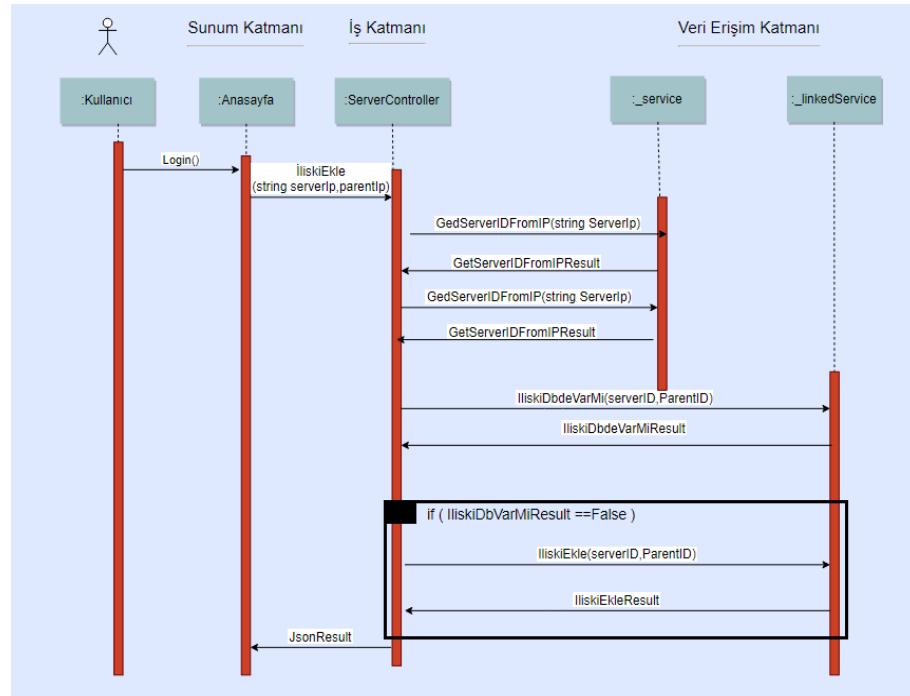
Şekil 6.1 Sunucunun ilişki ağını görüntüleme işlemi diyagramı

Şekil 6.2’de kullanıcı ilk başta sunum katmanındaki anasayfadan sisteme giriş yapmaktadır. Hemen ardından iş katmanındaki `ServerController` classıyla ilişkisiil fonksiyonu çağrılarak ilgili ip değerleri alınmıştır. `ServerController` classı `_service` classının `GetServerIDFromIP` fonksiyonuyla ana sunucu ve ilişkisi bulunan serverların id değerlerinden ip değerlerini döndürmektedir. Ip değerlerinin alınmasının ardından ilişkinin tekrardan veritabanına kaydedilmemesi amacıyla `_linkedService` classının `IliskiDbdeVarMi` fonksiyonuyla ilişkinin veritabanında olup olmadığı sonucuna göre, ilişki silinir ya da silinmez. Eğer ilişki veritabanında varsa, `_linkedService` classının `Iliskisiil` fonksiyonu çağrılarak ilişki silinir. Silinen ilişkinin sonucuna göre ekranda listeleme işlemi yapılır.



Şekil 6.2 Sunucuya ait ilişki silme işlemi diyagramı

Şekil 6.3’de kullanıcı ilk başta sunum katmanındaki anasayfadan sisteme giriş yapmaktadır. Hemen ardından iş katmanındaki `ServerController` classıyla `İliskiEkle` fonksiyonu çağrılarak ilgili ip değerleri alınmıştır. `ServerController` classı, `_service` classının `GetServerIDFromIP` fonksiyonuyla ana server ve ilişkisi bulunan serverların id değerlerinden ip değerlerini döndürmektedir. Ip değerlerinin alınmasının ardından ilişkinin tekrardan veritabanına kaydedilmemesi amacıyla `_linkedService` classının `İliskiDbdeVarMi` fonksiyonuyla ilişkinin veritabanında olup olmadığı sonucuna göre, ilişki eklenir ya da eklenmez. Eğer ilişki veritabanında yoksa, `_linkedService` classının `İliskiEkle` fonksiyonu çağrılarak ilişki eklenir. Eklenen ilişkinin sonucuna göre ekranda listeleme işlemi yapılır.

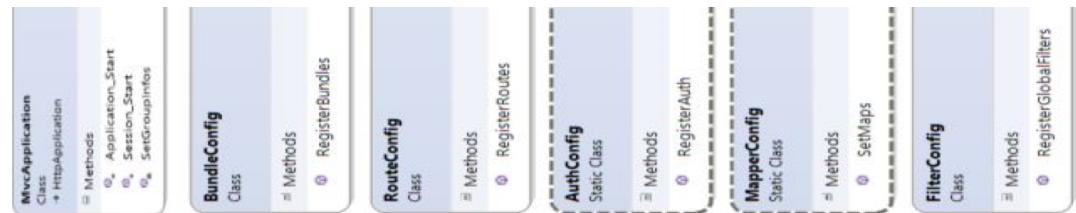


Şekil 6.3 Sunucuya yeni ilişki ekleme diyagramı

İlgili programın yazılım mimarisi: Config sınıfları, Controller sınıfları, Modal sınıfları ve Mvc Application sınıflardan oluşmaktadır.

6.2 Config Sınıflar

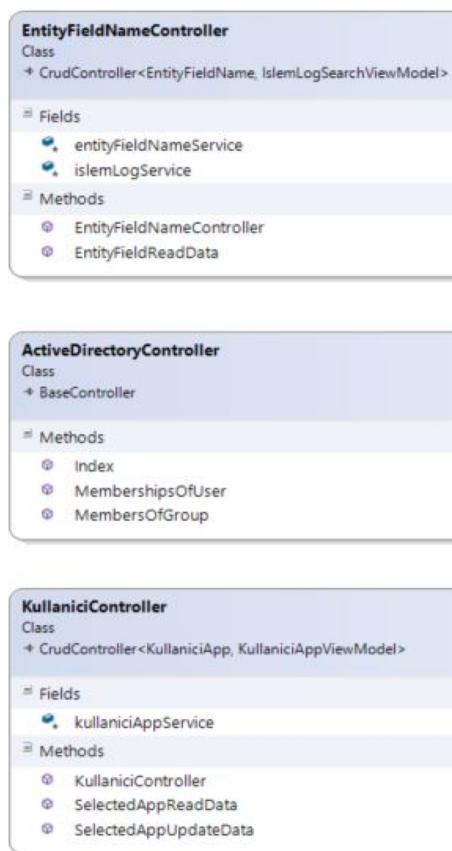
Visual Studio'nun MVC tasarım kalibini web tabanlı uygulamalar için geliştirdiği ASP.NET MVC projesinin içinde gelen programsal sınıflarıdır. Şekil 6.4'de yer alan sınıflar (class), MVC tasarım kalibinin uygulanması için Visual Studio tarafından oluşturulmuş olan hazır sınıf yapılarıdır.



Şekil 6.4 Projenin içinde yer alan config sınıflar

6.3 Controller Sınıflar

İlgili program kodunun, modal classlarının uygun şekillerde kullanılarak; ilgili işlemin gerçekleştirilmesini sağlayan arayüz yazılımlarıdır. Şekil 6.5, 6.6, 6.7, 6.8, 6.9 içerisinde çeşitli modelleri kullanan controller yapıları listelenmiştir. Çalışma kapsamında geliştirilen şekil 6.7 içerisinde yer alan `ServerController` içerisinde `getIISiteNamesFromIP` fonksiyonu belirli IP'ye göre sunucunun barındırdığı siteleri veritabanından çekmekte, `GetIIServerID` sunucunun ilişkiye sahip olup olmadığını test etme işlemini yapmaktadır. `IliskiEkle` fonksiyonu ilgili ilişkisiyi veritabanına ekleme işlemini, `IliskisiSil` fonksiyonu ilgili ilişkisiyi veritabanından silme işlemini yapar. `IliskiliServerlariListele` fonksiyonu, ilişki ağı görüntüleme işlemi için sunucuların ilişkileri ve birbirleri arasındaki hiyerarşik katman sıralamasını dinamik olarak oluşturarak ilişki ağı görüntüleme işlemi için veri seti listesi oluşturmaktadır. `tumIliskileriAl` fonksiyonu sunucunun sahip olduğu tüm ilişkilerin listesinin alınmasını sağlamaktadır.



Şekil 6.5 Projenin içinde yer alan Controller sınıfları

HomeController

Class
+ BaseController

= Methods

- ⊕ About
- ⊕ ChangeCulture
- ⊕ Contact
- ⊕ HomeController
- ⊕ Index

IslemLogController

Class
+ Crud5Controller<IslemLog, IslemLogViewModel, IslemLogSearchViewModel, IslemLogViewModel>

= Fields

- ⊕ islemLogService

= Methods

- ⊕ CustomReadDataForServerSelect
- ⊕ GetGridData
- ⊕ IslemLogController

ReportServerController

Class
+ Crud5Controller<ReportServer, ReportServerViewModel, ReportServerSearchViewModel, ReportServerViewModel>

= Fields

- ⊕ dtoMapper
- ⊕ reportServerService

= Methods

- ⊕ CustomReadDataForServerSelect
- ⊕ GetGridData
- ⊕ ReportServerController

Şekil 6.6 Projenin içinde yer alan controller sınıflar

EntityFieldLogController

Class
→ CrudController<EntityFieldLog, EntityFieldLogViewModel>

Fields

- entityFieldLogService

Methods

- EntityFieldLogController

ServerController

Class
→ CrudController<AvServer, ServerCrudViewModel, ServerSearchViewModel, ServerEditViewModel>

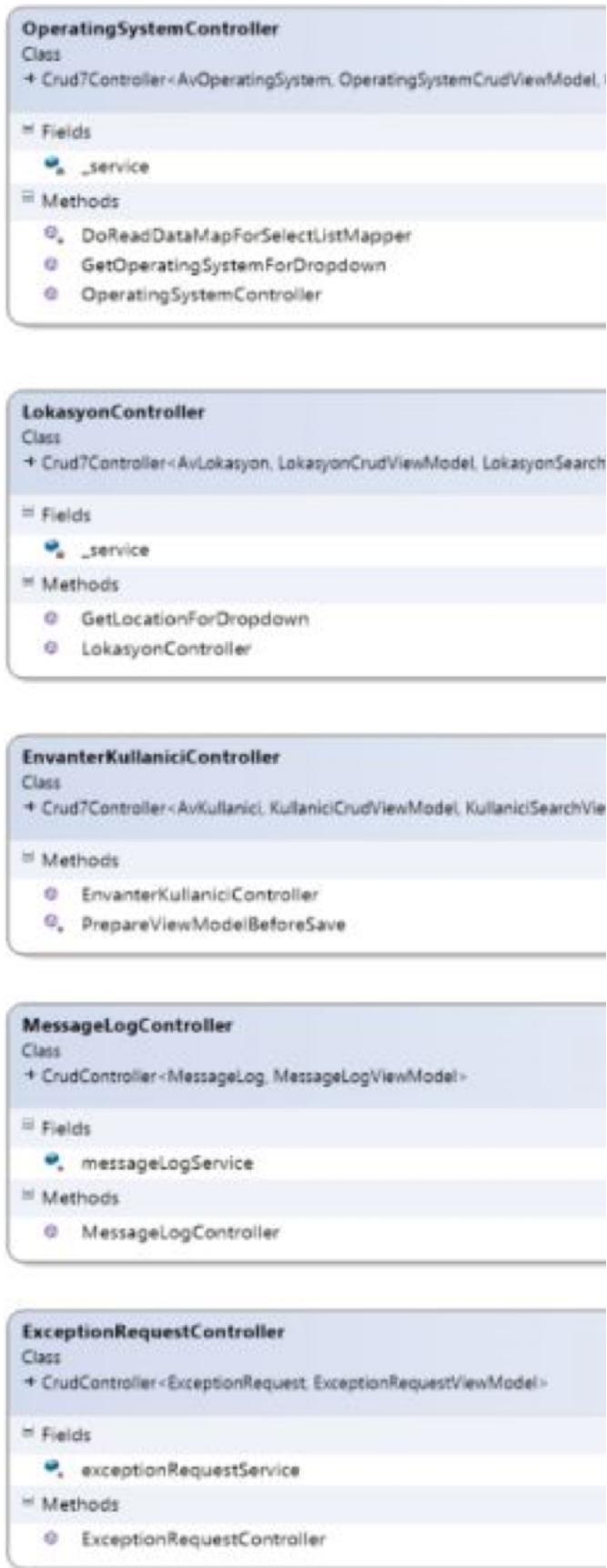
Fields

- _appService
- _linkedService
- _JokasyonAppService
- _operatingSystemAppService
- _serverTipiAppService
- _service
- _yoneticilServerService
- allServers
- IliskiliSunucuList
- ServerDenemeList
- ServerList
- slimeSayaciChild
- slimeSayaciParent

Methods

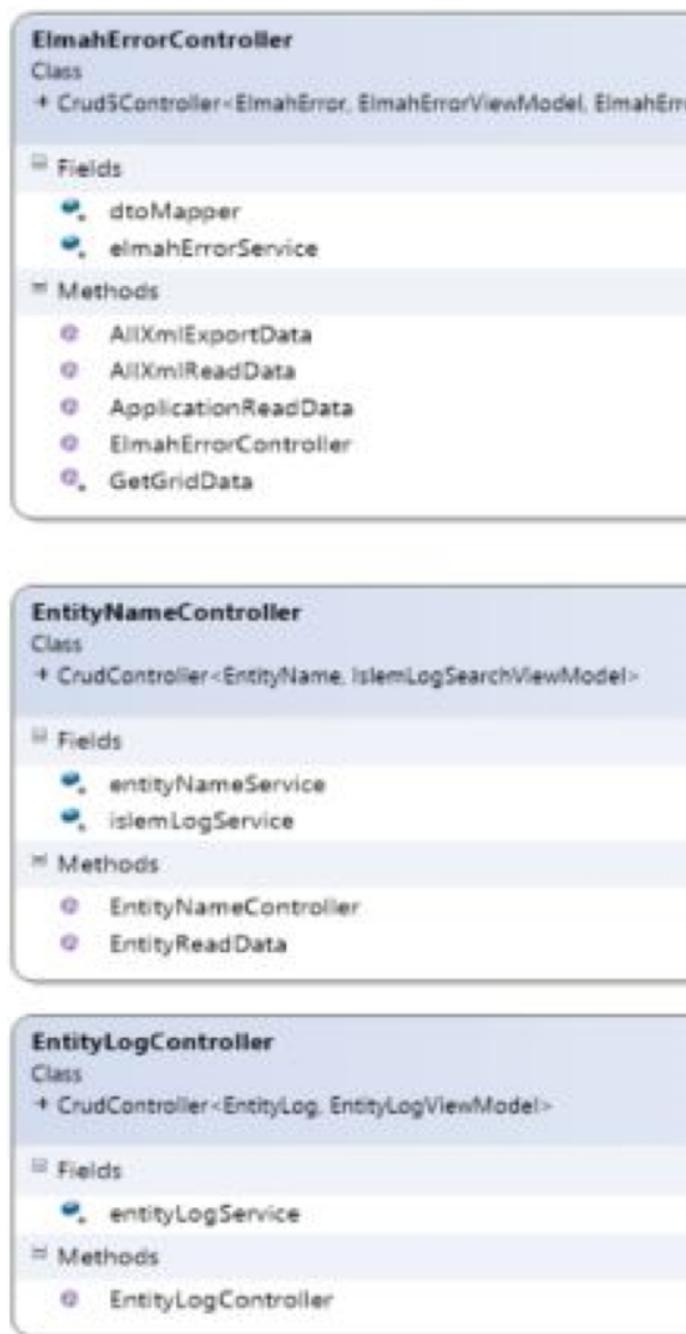
- AddServer
- Convert.ToDecimalPointFormatFromCommaFormat
- DeleteServer
- DoEntityMapBeforeCreate
- DoEntityMapBeforeUpdate
- DoReadDataMapForListMapper
- getAppName
- GetGridData
- getIISBindings
- getISSiteNames
- getISSiteNamesFromIP
- GetIliskiliServerID
- getInstalledApplications
- getServerInfo
- IliskiEkle
- IliskiSil
- IliskiliServerlariListele
- Ping
- PrepareViewModelBeforeSearch
- refreshServerInfo
- ServerController
- serverlariListele
- tumIliskilenAl
- UpdatePing

Sekil 6.7 Projenin içinde yer alan controller sınıflar



Şekil 6.8 Projenin içinde yer alan controller sınıflar

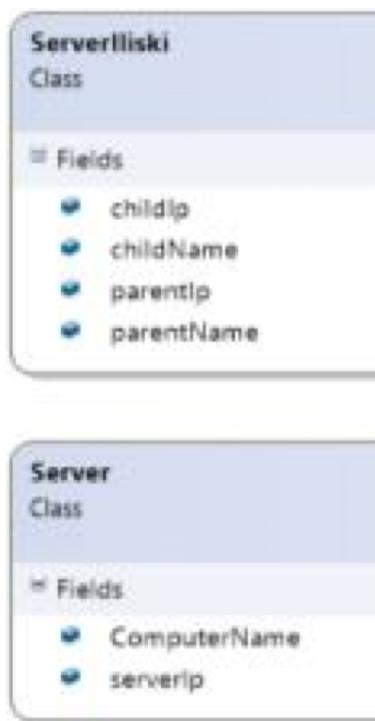
Burada, Server Controller sınıflarının içerisinde bulunan `iliskiliServerlariListele` fonksiyonu ile `_linkedService` isimli class'a ait olan `SERVER_ILISKILI_SUNUCU` adlı tabloya bağlanarak; ilgili ip'ye sahip değer ve onun sahip olduğu ilişkiler bulunup, bir liste halinde döndürülmektedir. Dönüş değerler ise ön kısımda Vis.js kütüphanesi yardımıyla ağ görüntüsü olarak ekrana çizilmektedir. Projenin ana bölümünü bu yapı oluşturmaktadır.



Şekil 6.9 Projenin içinde yer alan controller sınıfları

6.4 Modal Sınıflar

Modal, Kullanıcılara bilgi vermek yada kullanıcılarından bilgi almak için jquery [5] framework’ünün sağladığı dialog pencereleridir. Veritabanından ilgili verilerin çekildikten sonra soyutlanması ve modal yapısında kullanımı için konulan yapılardır. Bu classlar ile modal yapısında gösterilecek ve geri bildirimi yapılacak veriler için sunum ortamı sağlanmış olur. Şekil 6.10’da projede kullanılmakta olan modal sınıflar listelenmiştir. Serverİlişki sınıfı ilişkisi bulunan sunucunun ve ilişkili sunucusunun IP’sini ve adını tutmaktadır. Bununla birlikte Server sınıfında sunucunun ComputerName ve serverIp değerleri tutulmaktadır.

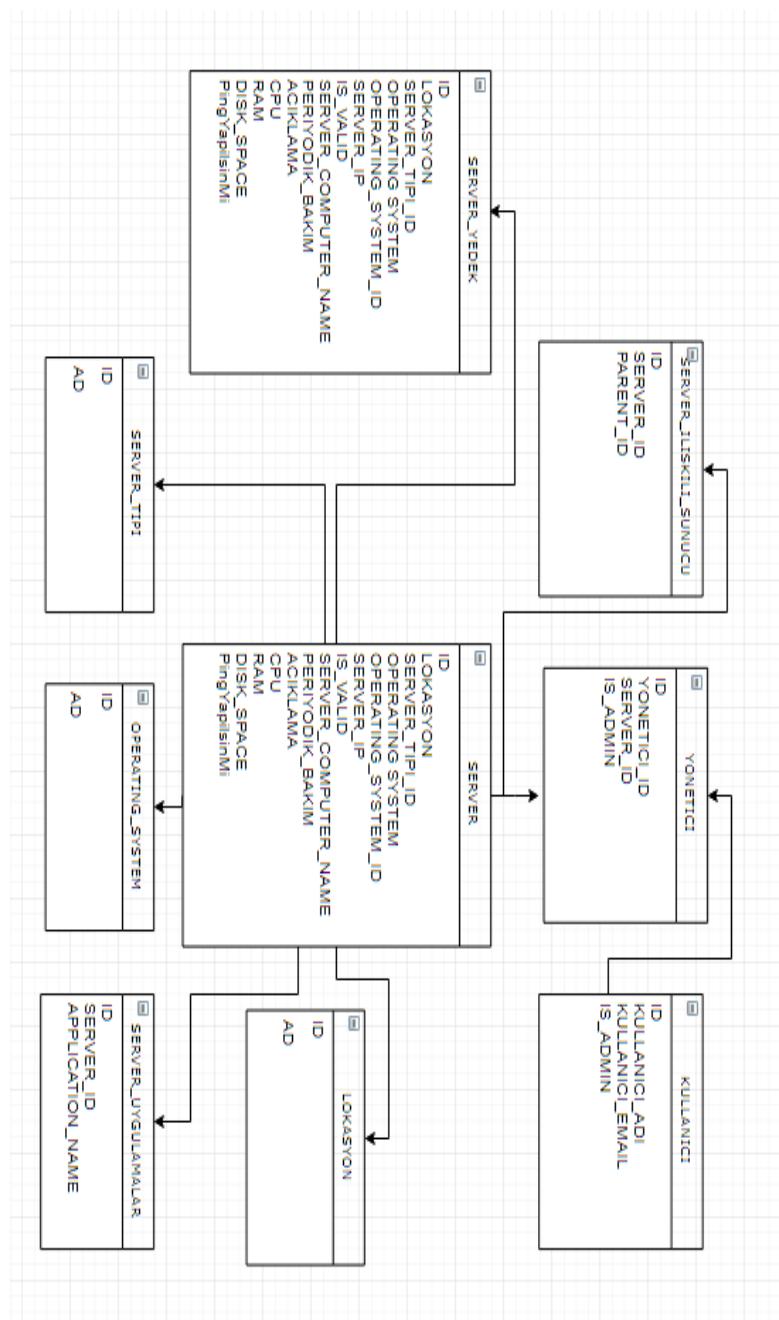


Şekil 6.10 Projenin içinde yer alan modal yapısı sınıfları

6.5 Database Design/ER Diagram

Bir varlık-ilişki (Entity Relationship, ER) diyagramı olarak da adlandırılan bir entity relationship modeli, varlıkların ve birbirleriyle ilişkilerinin grafiksel bir temsilidir. Tipik olarak veri tabanları ve bilgi sistemleri içindeki verilerin düzenlenmesi için hesaplamada kullanılır. Bir varlık bir veri parçasıdır. Verilerin depolandığı bir nesne veya kavramdır. Şekil 6.11’de projenin veritabanı mimarisinin ER diyagramı görüntülenmektedir. SERVER tablosu genel sunucu

özelliklerini, SERVER_ILISKILI_SUNUCU tablosu hangi serverlar arasında ilişki olduğunu, YONETICI tablosu bir kullanıcının yönetici olup olmadığını, KULLANICI tablosu kullanıcı bilgilerini, SERVER_YEDEK server verilerinin yedegini, SERVER_TIPİ sunucunun tipini, LOKASYON tablosu sunucunun hangi konumda olduğunu, OPERATING_SYSTEM tablosu serverda hangi işletim sisteminin yüklü olduğunu, SERVER_UGULAMALAR tablosu serverda hangi uygulamaların kurulu olduğu bilgisini tutmaktadır.



Sekil 6.11 Projenin ER diyagramı

7. SINAMA

Bu bölümde sisteme uygulanan testler yer almaktadır. Testler ile ortaya çıkan sonuçlar ve sistemin genel davranışları hakkında da bilgi verilmektedir.

7.1 Fonksiyonel Testler

Bu bölümde sistemin işlevselliğine yönelik yapılan testler yer almaktadır. Alt başlıklarda yapılan testler ve sonuçları incelenmiştir.

7.1.1 Birim Testi

Sistemin her bir bileşenin ayrı ayrı test edildiği test türüdür. Amaç yazılımın her biriminin tasarlandığı gibi çalıştığından emin olmaktadır. Buna göre sistem için yazılan fonksiyonlara birim testi uygulanmış olup, sistemin olabildiğince tutarlı davranış sergilemesi sağlanmaya çalışılmıştır.

7.1.2 Entegrasyon Testi

Entegre üniteler arasındaki hataları bulabilmek için yapılan testtir. Sistemdeki entegre olan ünitelerin grup halinde çalışma durumu test edilerek, sistemin olabildiğince tutarlı davranışması amaçlanmıştır.

7.2 Fonksiyonel Olmayan Testler

Bu bölümde sistemin nasıl çalıştığını dair, işlevsel olmayan testler yer almaktadır.

7.2.1 Hacim Testi

Hacim (*Volume*) testi, yazılımın çok büyük miktarda veriyle sınandığı bir testtir. Projede, sistem içinde barındırdığı sunuculara başarılı bir şekilde erişebilirken, sunucu kayıt miktarının artırılması sistem için bir problem teşkil etmemektedir.

7.2.2 Performans Testi

Yazılımın beklenen iş yükü altında iyi performans gösternesini sağlayan bir testtir. Sistem stabil bir şekilde çalışmaktadır. Proje içerisindeki bağlı sunucular sekmesi ortalama 5sn gibi bir sürede yanıt vermektedirler.

7.2.3 Güvenlik Testi

Sistemin güvenlik açılarını açığa çıkararak, verilerin ve kaynakların yabancı kişilerin eline geçmesini önlemek üzerine yapılan testlerdir. Program başlangıçta kullanıcı adı ve şifre istemektedir. Bu işlemleri sertifika ile yapmaktadır. Eğer sistem kayıtlanmamış bir bilgisayarda çalıştırılırsa, program açılmayacaktır. Bu güvenlik testleri programa uygulanmış olup, başka bilgisayarlardan ilgili sunucuların bilgilerinin alınmasının önüne geçilmiştir. Ayrıca, güvenliği yüksek derece de olan sunucularda olabilmektedir. Bu gibi sunucularda da, program ilgili yüksek güvenlikli sunucunun bilgilerini alamamaktadır. Bilgileri alınamayan sunucularda da bu işlemlerin test edilmesi sağlanmıştır.

7.2.4 Kurtarma Testi

Kurtarma (Recovery) Testi, sistemde beklenmeyen bir durum olduğunda oluşabilecek veri ve zaman kaybının ölçülebilmesi, tekrardan eski haline getirebilme durumunu test eden bir test türüdür. Sisteme kayıtlı sunucu bilgileri, her yeni veri eklendiğinde yedeklenerek oluşabilecek herhangi bir beklenmedik bir durumda veya veri kaybı yaşandığında sistem eski haline yedeklenen veriler sayesinde hemen getirilebilmektedir.

7.2.6 Kullanılabilirlik Testi

Kullanıcı Deneyimi testi olarak da adlandırılan bu test, son kullanıcının programı kullanarak ortaya çıkabilecek kusurları test etmeye yönelik bir işlemidir. Sistemi kullanma potansiyeline sahip kullanıcılarla gerekli testler yapılarak, kusurların olabildiğince en aza indirgenmesi ve sistemi iyileştirme işlemi yapılmıştır.

7.2.7 Uyumluluk Testi

Uyumluluk testi sistemi geliştirirken öngörülen standartları, sistemin karşılayıp karşılamadığını doğrulamak ve sistemi geliştirmek için yapılan test türüdür. Sistemi tasarlarken öngördüğümüz durumların gerçekleştirildiğinden emin olmak için gözden geçirme işlemi yapılmış ve bu yönde sistem geliştirilmiştir.

8. KULLANICI KILAVUZU

Proje çalıştırıldığı zaman ilk olarak şekil 8.1'deki gibi bir ekran gelmektedir. Burada uygulamaya kayıtlı olan sunucular listelenmiştir. Admin girişi yapılarak girilen bu uygulama ekranında ilişkilerin eklenmesi ve silinmesi işlemini yapan temel düzenleme özelliklerini aktif buton olarak gelmektedir.

Ana ekran üzerindeki ilişki düzenleme butonuna basıldığı takdirde, tüm sunucular için ilişki ekleme ve silme işlemleriyle ilgili Şekil 8.2'deki ekran karşımıza gelmektedir. Bu kısımda ilişki eklenir ya da silinir.

İlişki ekleme butonuna basıldığı takdirde, çalışma bizden ata (*parent*) ve çocuk (*child*) IP'lerini girmemizi istemektedir. Şekil 8.3'te görüldüğü gibi seç sekmesine erişim yapılarak bu IP'lerin seçilmesi işlemi yapılmaktadır.

IP'leri seçmek için seç sekmesine eriştiği zaman; hem ata hem de çocuk için Şekil 8.4'teki ekran, ilgili IP'lerin seçilmesi işleminin yapılmasını istemektedir.

İlgili sunucunun IP'si seçildikten sonra Şekil 8.3'teki ekranda, ata ve çocuk IP'leri listenelecektir. Bu kısımda eklenen bilgiler kontrol edildikten sonra; ilişki ekle butonuna basıldığı takdirde, ilgili IP'lere sahip bilgiler veritabanı tarafında sisteme işlenecektir.

İlişki silme işleminde, Şekil 8.2'deki ilişki silme butonuna erişim yapıldığında sistemdeki tüm ilişkiler listenecektir. Buradan silinmesi istenen ilişki seçilip, onaylandıktan sonra ilgili ilişki veritabanı tarafında temizlenmiş olacaktır. Bu işlemler tüm sunucular için yapılmaktadır.

SİZWATCH

ANA VERİ

ENVANTER

Server

Search Criteria

Server Tipi	Seçiniz	Server IP
Lokasyon	Seçiniz	Makina Adı
İletim Sistemi	Seçiniz	Açıklama

List

Yeni Server **Güncelle**

Excel'e Aktar

Günlük Düzene

Ara **Tanımla**

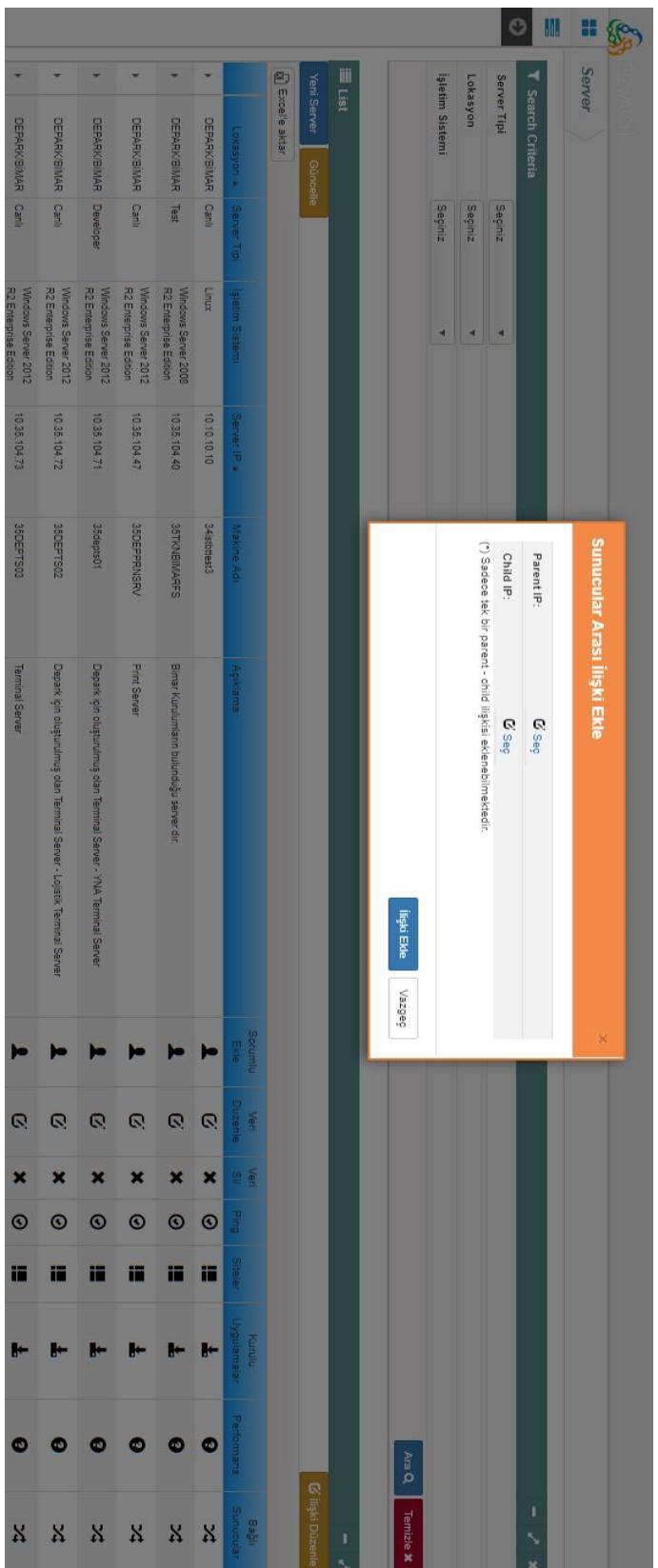
Lokasyon	Server Tipi	İletim Sistemi	Server IP	Makine Adı	Açıklama	Sorumlu Eki	Veri Dengeleme	Veri Sil	Ping	Sistem Uyumlama	Kurulu	Bağılı Sunucular	Performans	Sınıflandırma
DEPARKBMAR	Canlı	Linux	10.10.10.10	34S08T03	Birler Kurulumun bulunduğu serverdir.	1	✓	✗	○	■	■	2	2	2
DEPARKBMAR	Test	Windows Server 2008 R2 Enterprise Edition	10.35.104.40	35DEPPRNSRV	Print Server	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	Canlı	Windows Server 2012 R2 Enterprise Edition	10.35.104.47	35DEPPRNSRV	Dekrk. için oluşturulmuş olan Terminal Server - YMA Terminal Server	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	Developer	Windows Server 2012 R2 Enterprise Edition	10.35.104.71	35DEP0501	Dekrk. için oluşturulmuş olan Terminal Server - YMA Terminal Server	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	Canlı	Windows Server 2012 R2 Enterprise Edition	10.35.104.72	35DEP1TS02	Terminal Server	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	Canlı	Windows Server 2012 R2 Enterprise Edition	10.35.104.73	35DEP1TS03	Terminal Server	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	Canlı	Windows Server 2012 R2 Enterprise Edition	10.35.104.95	35DEP0502	TFS de ki Sharepoint verilerinin tutulduğu sunucudur.	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	VFC	Windows 7	10.35.105.129	PARTTIME01-VFC	Sayer-PartTime için gerekli sanal PC-1	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	VFC	Windows 7	10.35.105.130	bizalkicev-vpc	BMAR Yazılım - Sanal VPC - Brak 2010 ile şartname için gerekli olan sanal pc - Sanal VPC - Kullanıcı PC'lerin formülasyona gitmesi ve ya aktif kullanılmış拼搏en durumları için yazılımı olan VPC 4	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	VFC	Windows 7	10.35.105.131	PARTTIME02-VFC	Sayer-PartTime için gerekli sanal PC-2	1	✓	✗	○	■	■	1	2	2
DEPARKBMAR	VFC	Windows 7	10.35.105.132	PARTTIME03-VFC	Sayer-PartTime için gerekli sanal PC-3	1	✓	✗	○	■	■	1	2	2
			10.35.105.133	PARTTIME04-VFC	Sayer-PartTime için gerekli sanal PC-4	1	✓	✗	○	■	■	1	2	2

Incorrect header

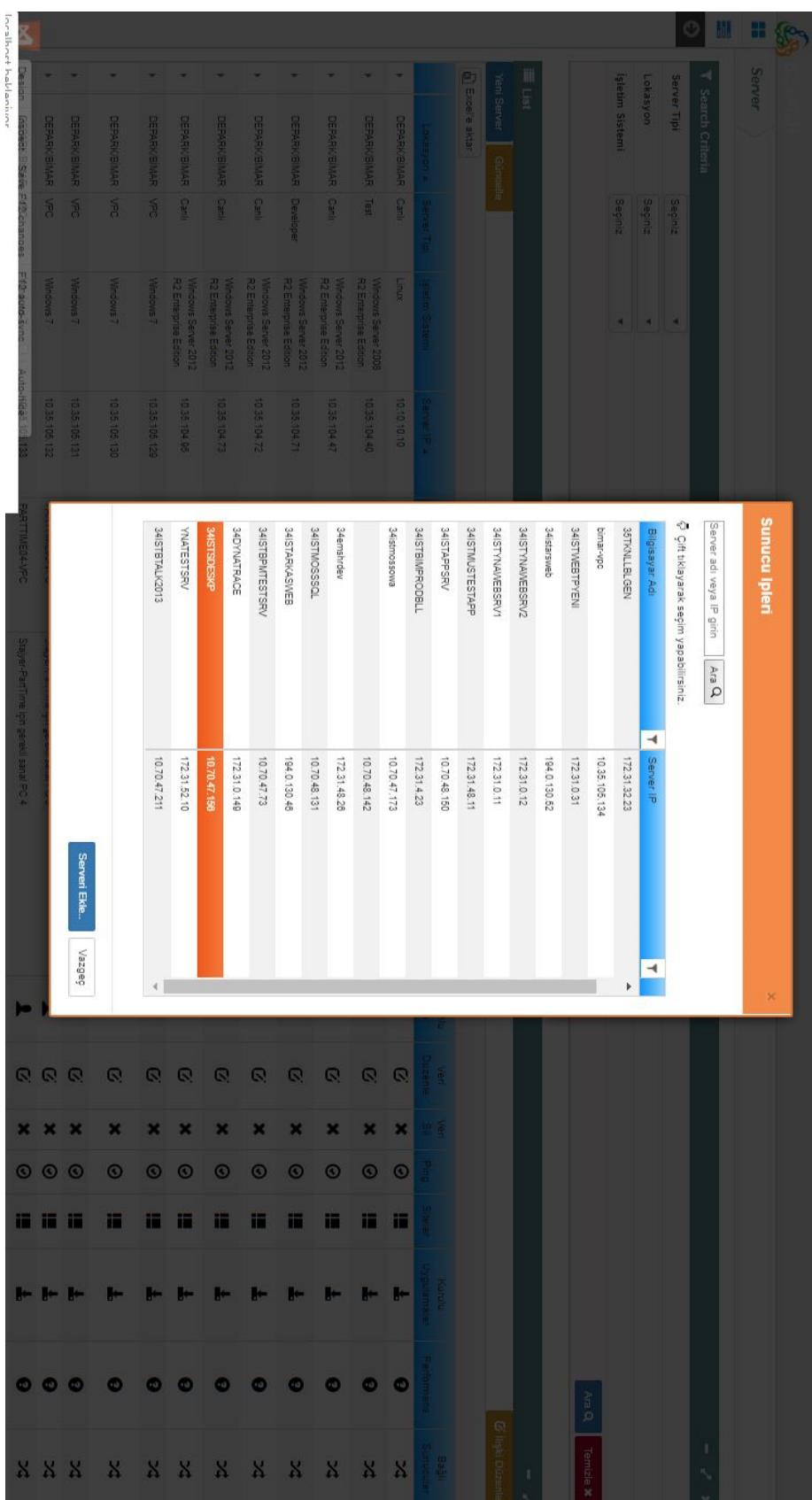
Şekil 8.1 Proje giriş ekranı



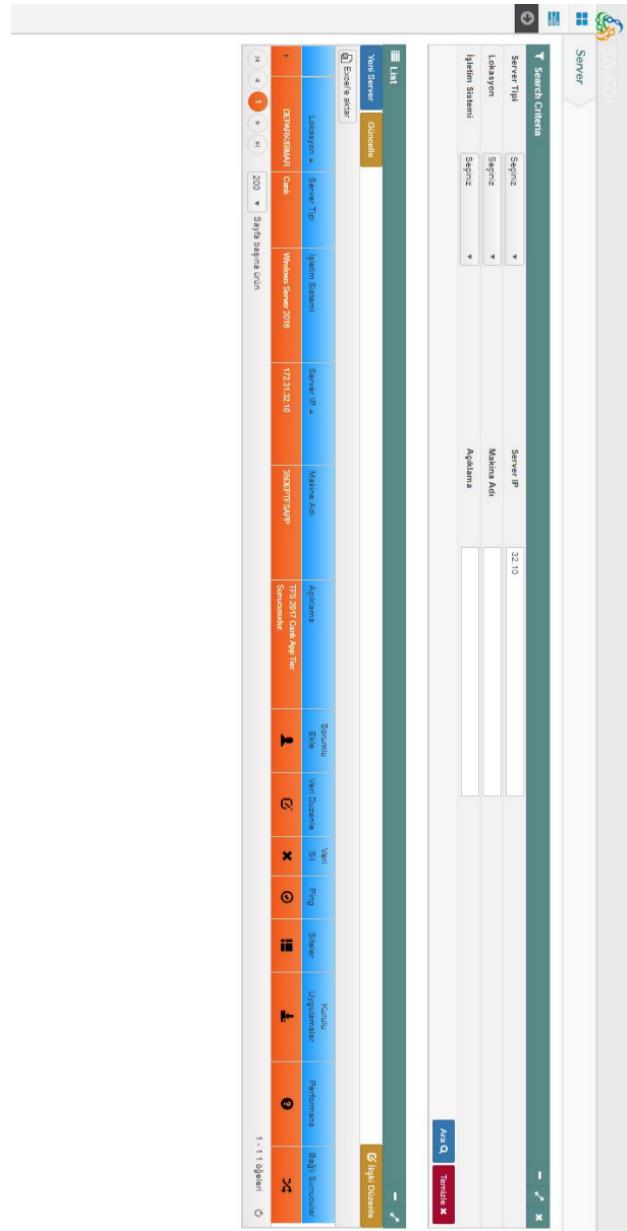
Şekil 8.2 Sunucular arası ilişki düzenleme ekranı



Şekil 8.3 Sunucular arası ilişki ekleme ekranı



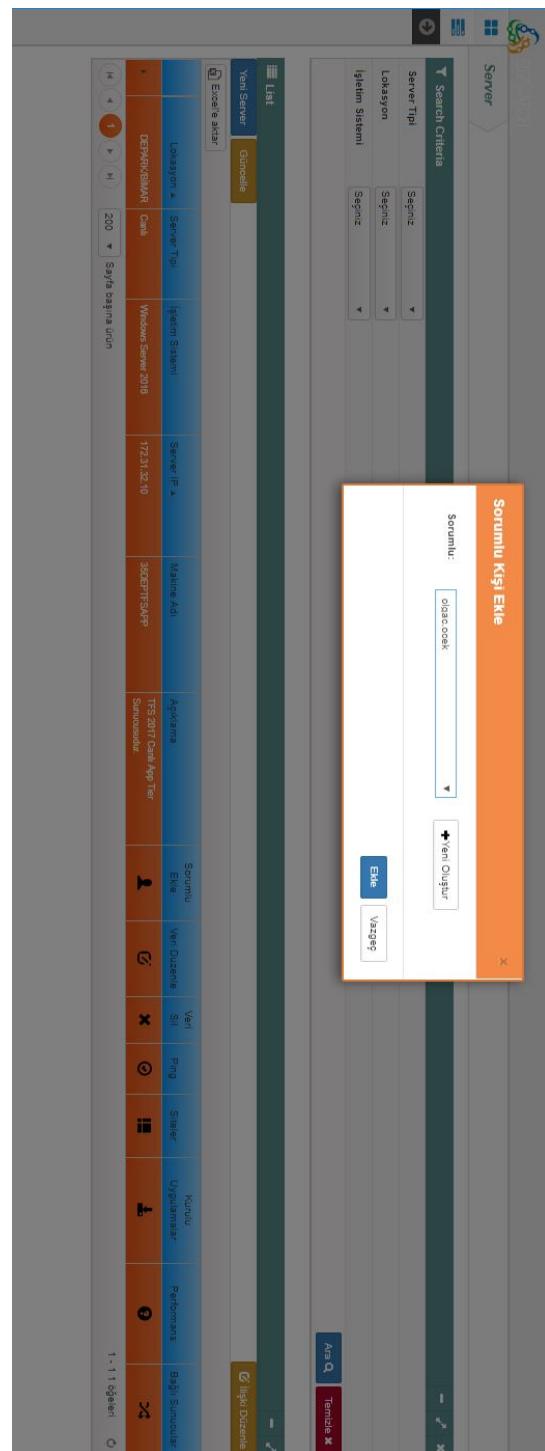
Sekil 8.4 Parent ya da Child IP ekleme ekranı



Şekil 8.5 Belirli bir IP'ye göre makine arama ekranı

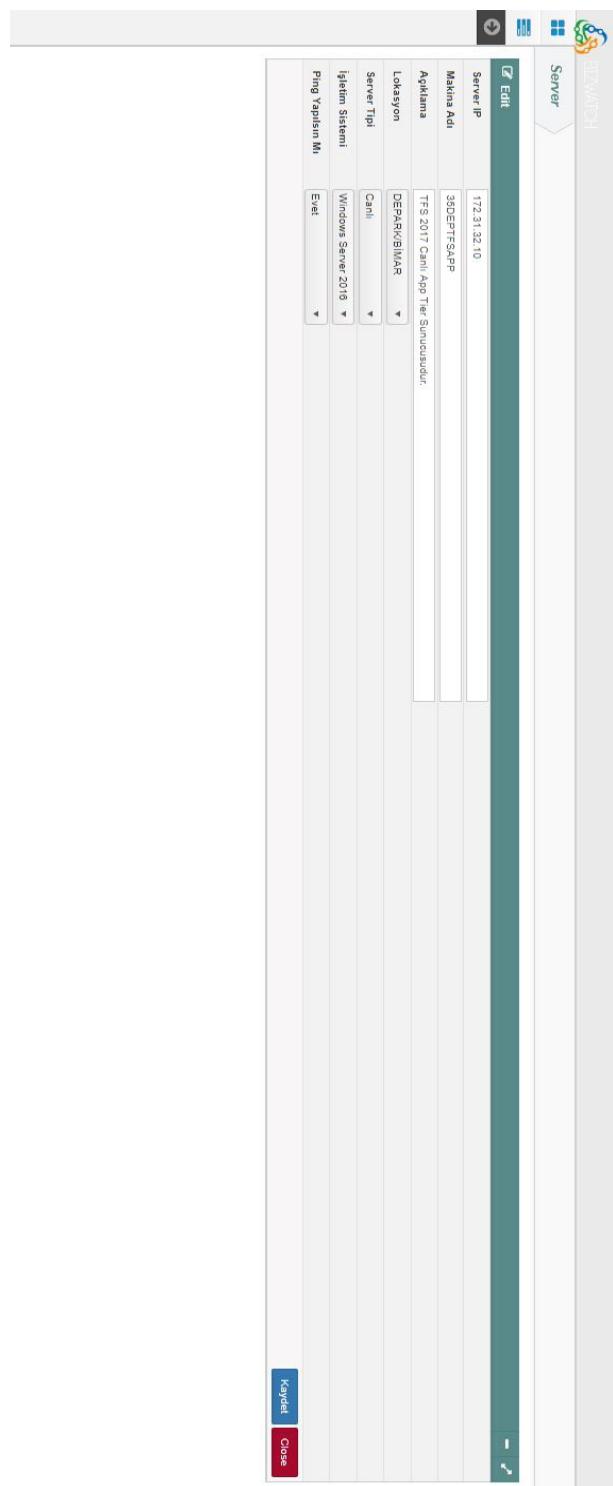
Şekil 8.5'teki örnek olarak aranmış olan makinenin; sorumlu ekleme, ilgili makinenin verilerini düzenleme, ilgili makinenin verilerini silme, ilgili makineye ping atma, makinedeki kurulu uygulamalar, makine performans bilgileri ve makineye bağlı sunucuların detaylı olarak incelenmesi yapılmaktadır.

Şekil 8.5'te sorumlu ekleme sekmesine erişim yapıldığı takdirde; Şekil 8.6'daki gibi bir ekran karşımıza çıkmaktadır. Uygulama, ilgili sorumlu kişinin seçilen sunucuya atanması işlemini bu sekme aracılığıyla yapmaktadır.



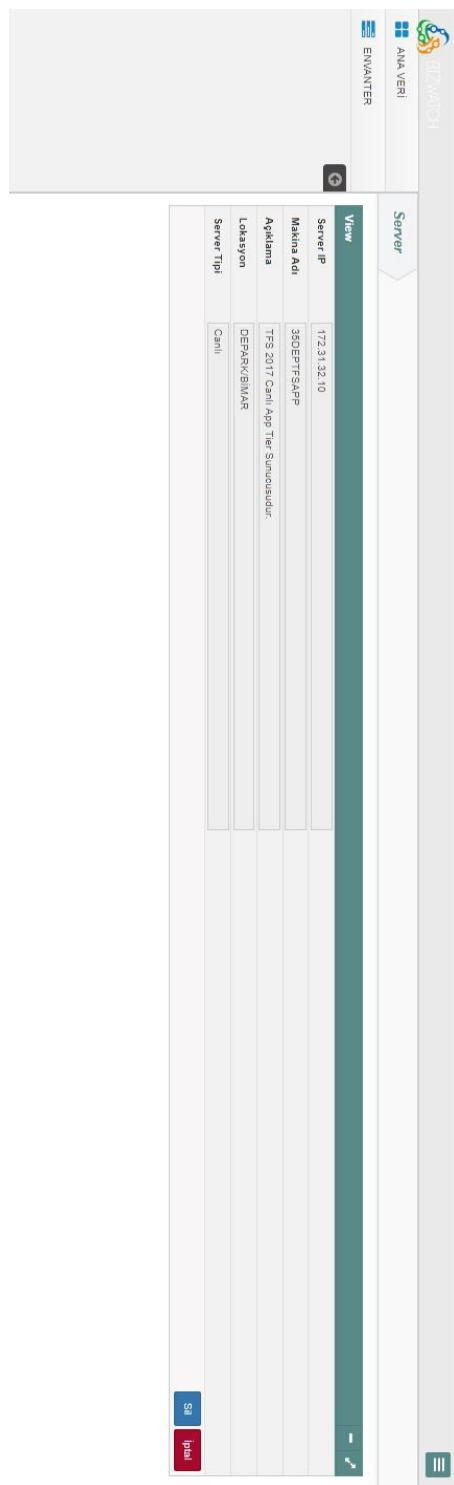
Şekil 8.6 Sorumlu ekleme ekranı

Şekil 8.5'deki veri düzenleme sekmesine erişim yapıldığı zaman, Şekil 8.7'deki ekran ile karşılaşılmaktadır. Bu sekmeden ilgili sunucunun ip, makine adı, açıklama, lokasyon, server tipi, işletim sistemi, ping atılma gibi özelliklerini düzenlenebilmektedir.



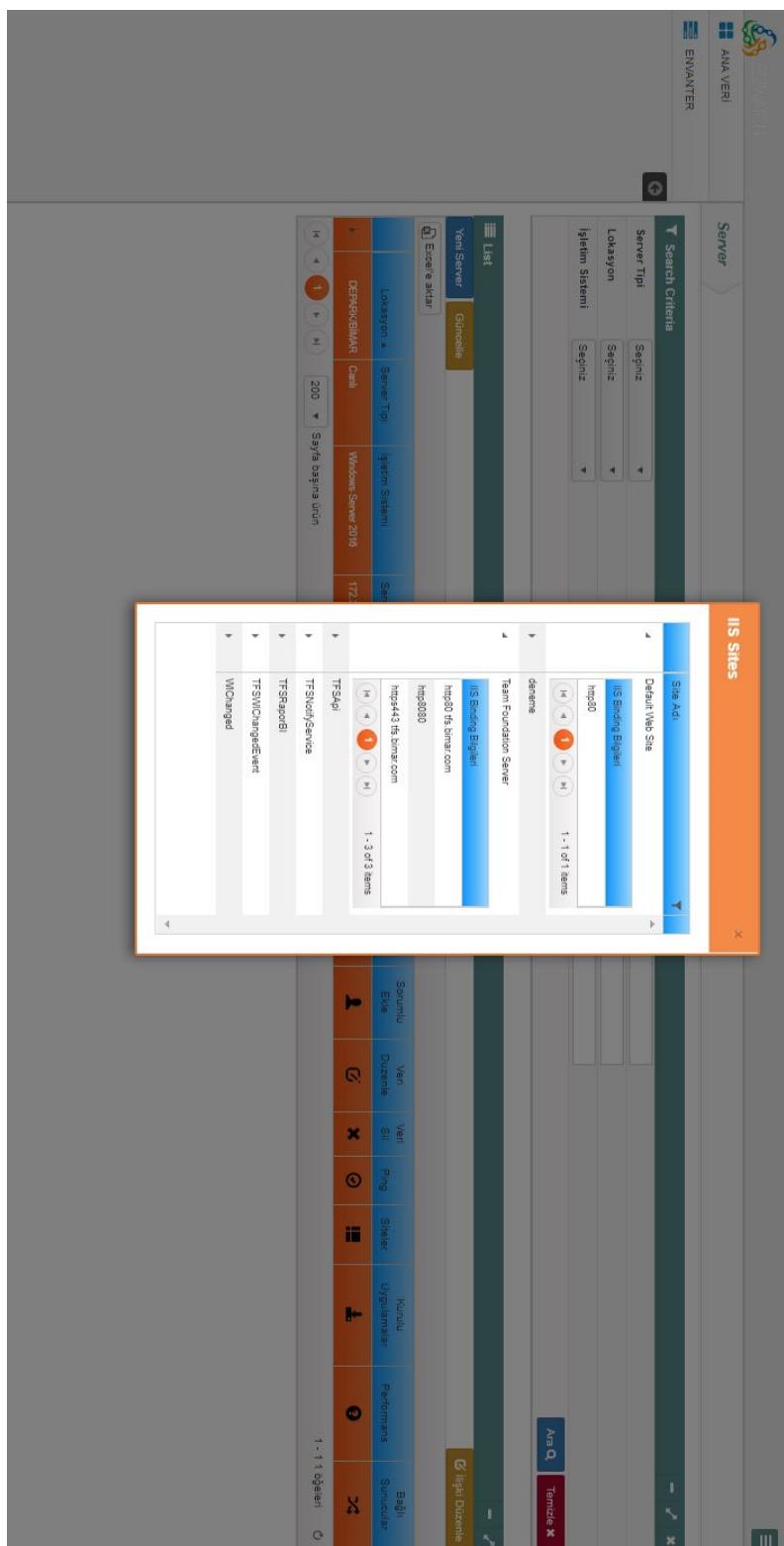
Şekil 8.7 Veri düzenleme sekmesi

Şekil 8.5'deki veri silme sekmesine erişim yapıldığı zaman, Şekil 8.8'deki ekran ile karşılaşılmaktadır. Bu sekmeden ilgili sunucunun IP, makine adı, açıklama, lokasyon, server tipi gibi özellikleri temizlenebilmektedir.



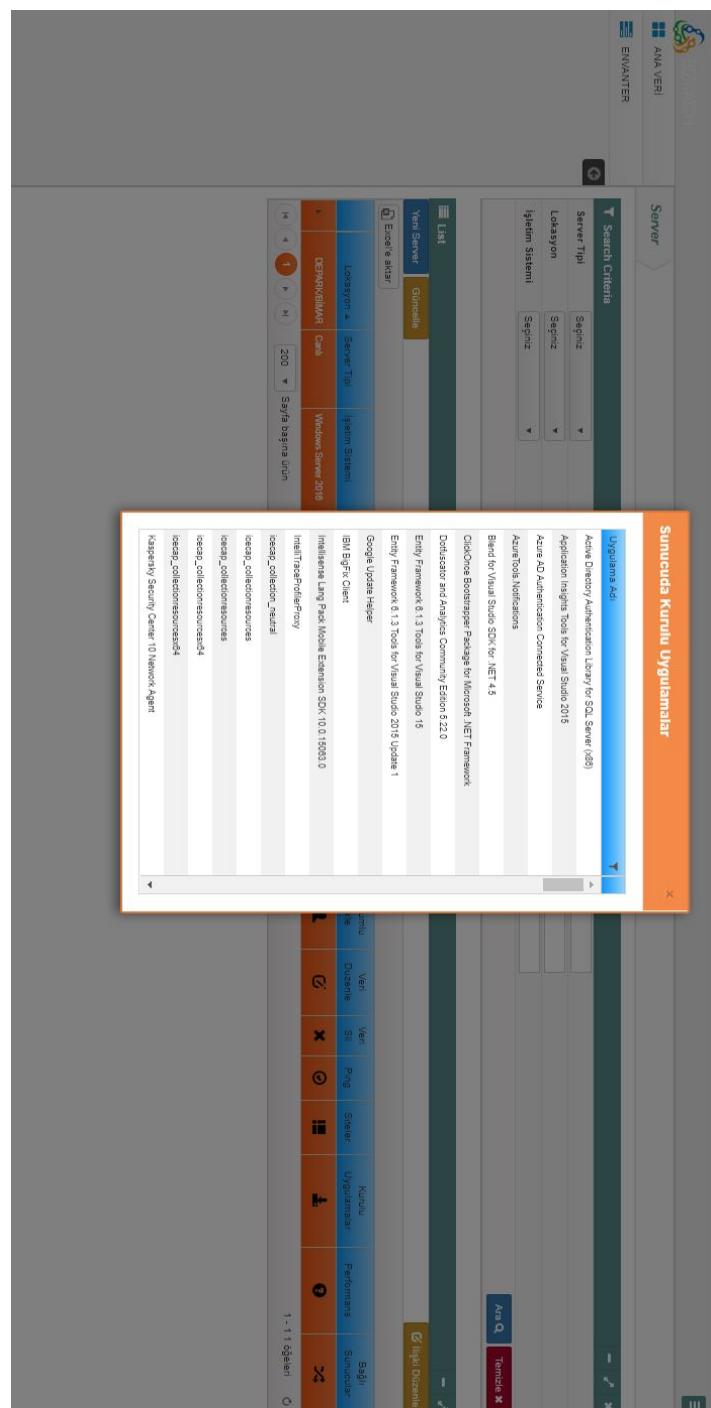
Sekil 8.8 Veri silme sekmesi

Şekil 8.5'teki siteler sekmesine erişim yapıldığı zaman, Şekil 8.9'daki ekran ile karşılaşılmaktadır. Bu sekme içerisinde, ilgili sunucunun barındırdığı site ve servis adları detaylı olarak analiz edilebilmektedir.



Şekil 8.9 Siteler sekmesi

Şekil 8.5'teki kurulu uygulamalar sekmesine erişim yapıldığı zaman, Şekil 8.10'daki ekran ile karşılaşılmaktadır. Bu sekme içerisinde, ilgili sunucu içerisindeki kurulu olan uygulamalar listelenmektedir.

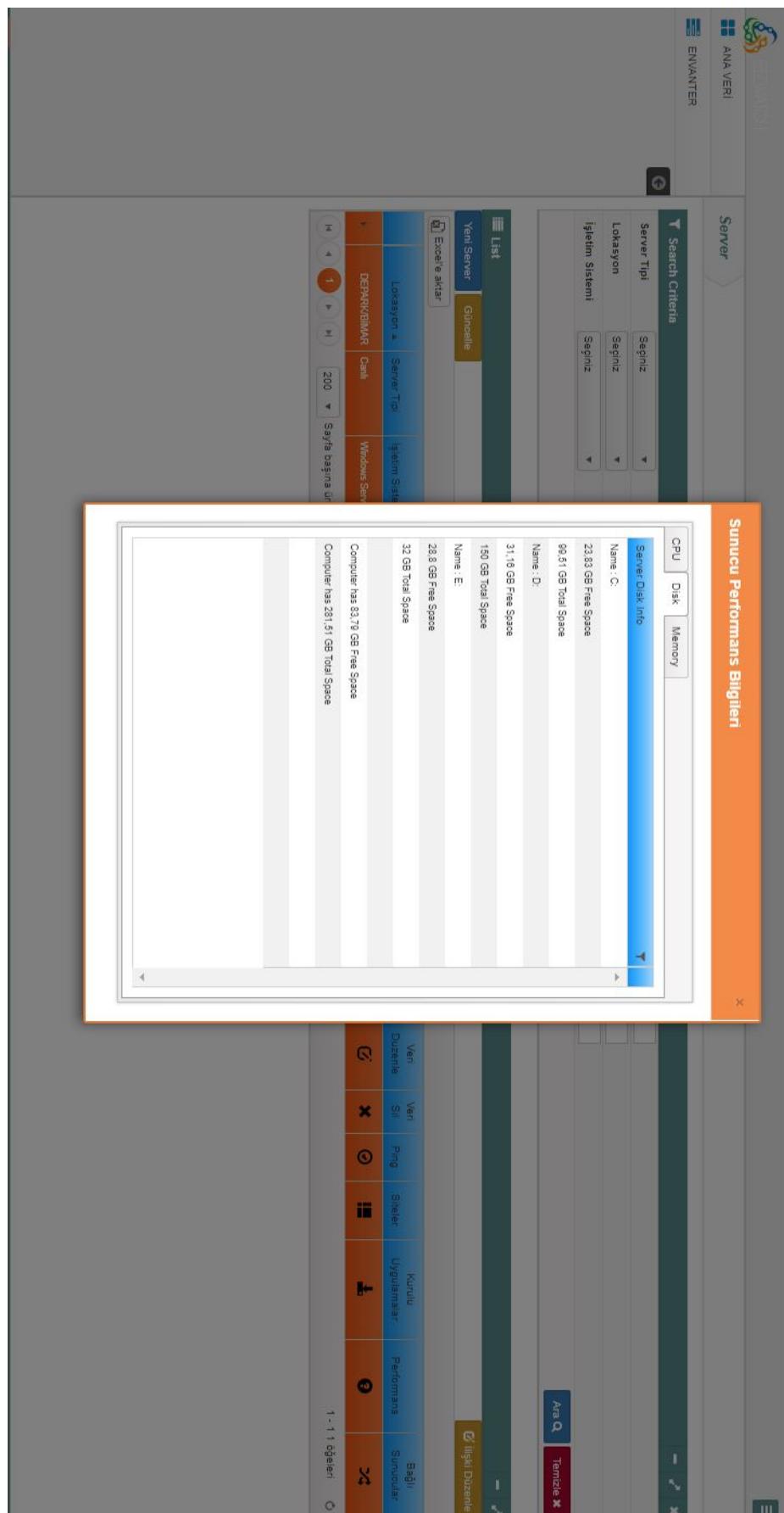


Şekil 8.10 Kurulu uygulamalar sekmesi

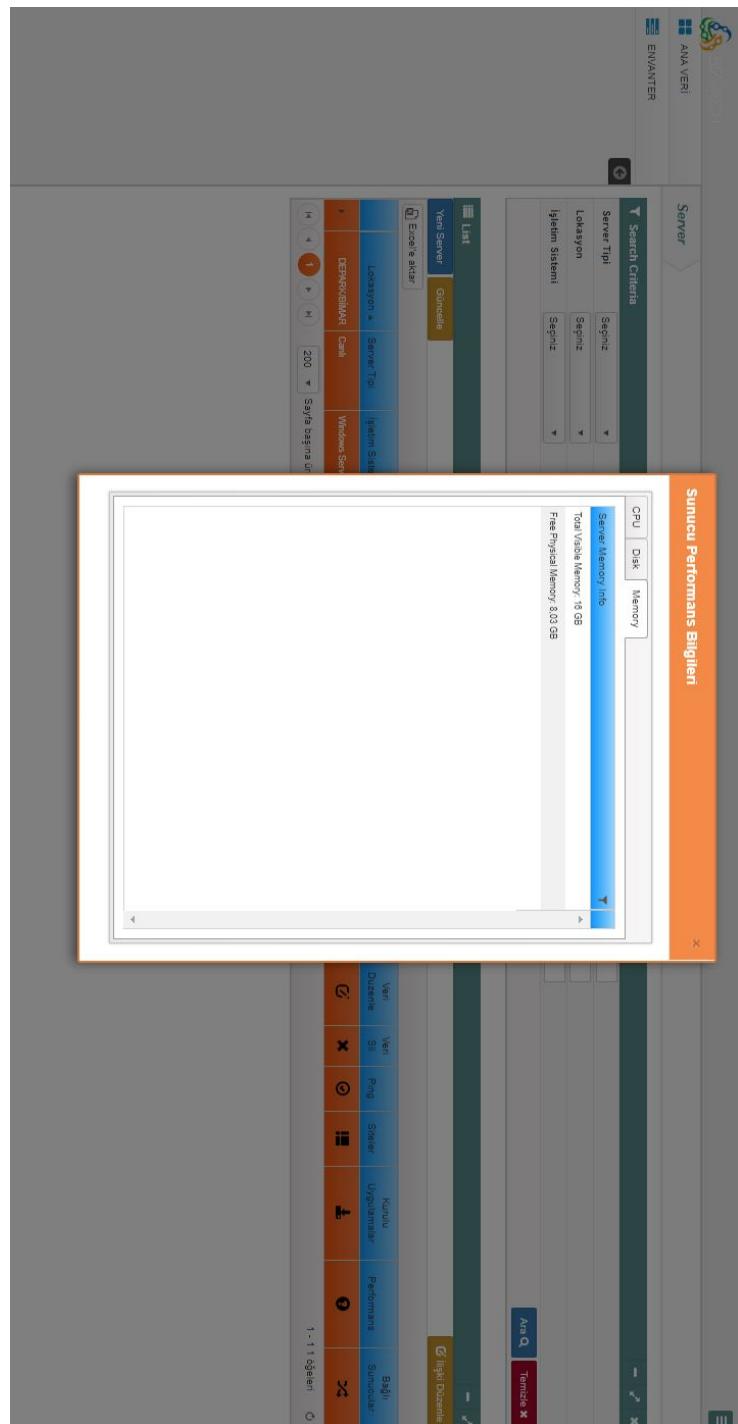
Şekil 8.5'teki performans sekmesine erişim yapıldığı zaman, Şekil 8.11'deki ekran ile karşılaşılmaktadır. Bu sekme içerisinde, ilgili sunucunun CPU özellikleri Şekil 8.11'deki, disk özellikleri Şekil 8.12'deki ve memory özellikleri Şekil 8.13'deki gibi gösterilmektedir.

The screenshot shows a server management interface with a navigation bar at the top. The main area is titled "SUNUCU PERFORMANS BİLGİLERİ" (Server Performance Information). Below this, there are tabs for "CPU", "Disk", and "Memory". The "CPU" tab is currently selected and highlighted in orange. On the left side, there is a search criteria panel with fields for "Server Tipi" (Server Type), "Lokasyon" (Location), and "İşletim Sistemi" (Operating System), each with dropdown menus. To the right of the search panel, there is a summary box containing information about the server's CPU: "Server CPU Info", "Description: Intel® Core™ i7-7700K Processor 4.20GHz", "DeviceID: CPU0", "LastFerrocene: 5", and "Status: Ok". At the bottom of the CPU tab, there are buttons for "Arıza" (Fault) and "Temizle" (Clean).

Şekil 8.11 Performans sekmesi içerisindeki CPU sekmesi

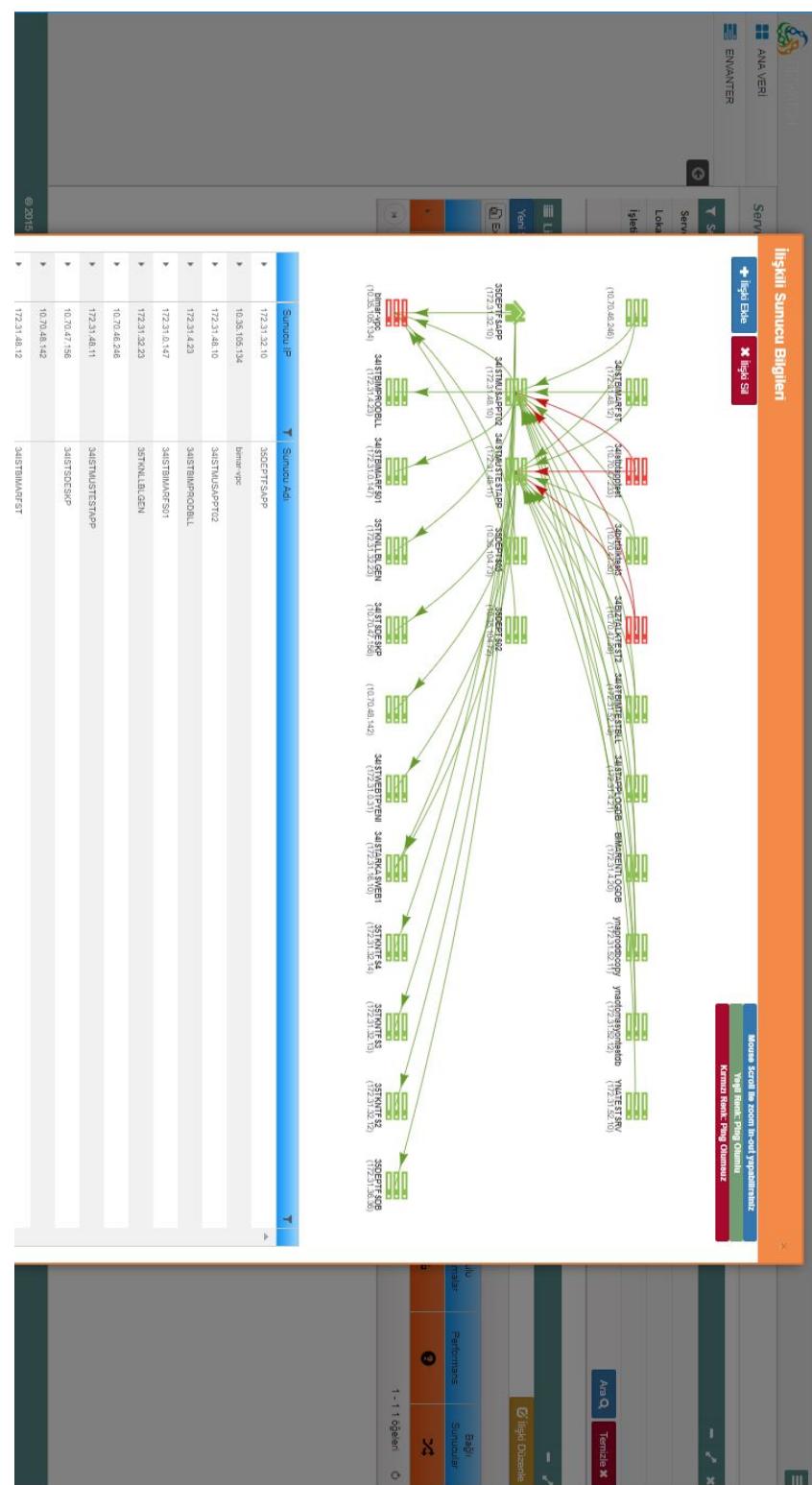


Şekil 8.12 Performans sekmesi içerisindeki Disk sekmesi



Şekil 8.13 Performans sekmesi içerisindeki memory sekmesi

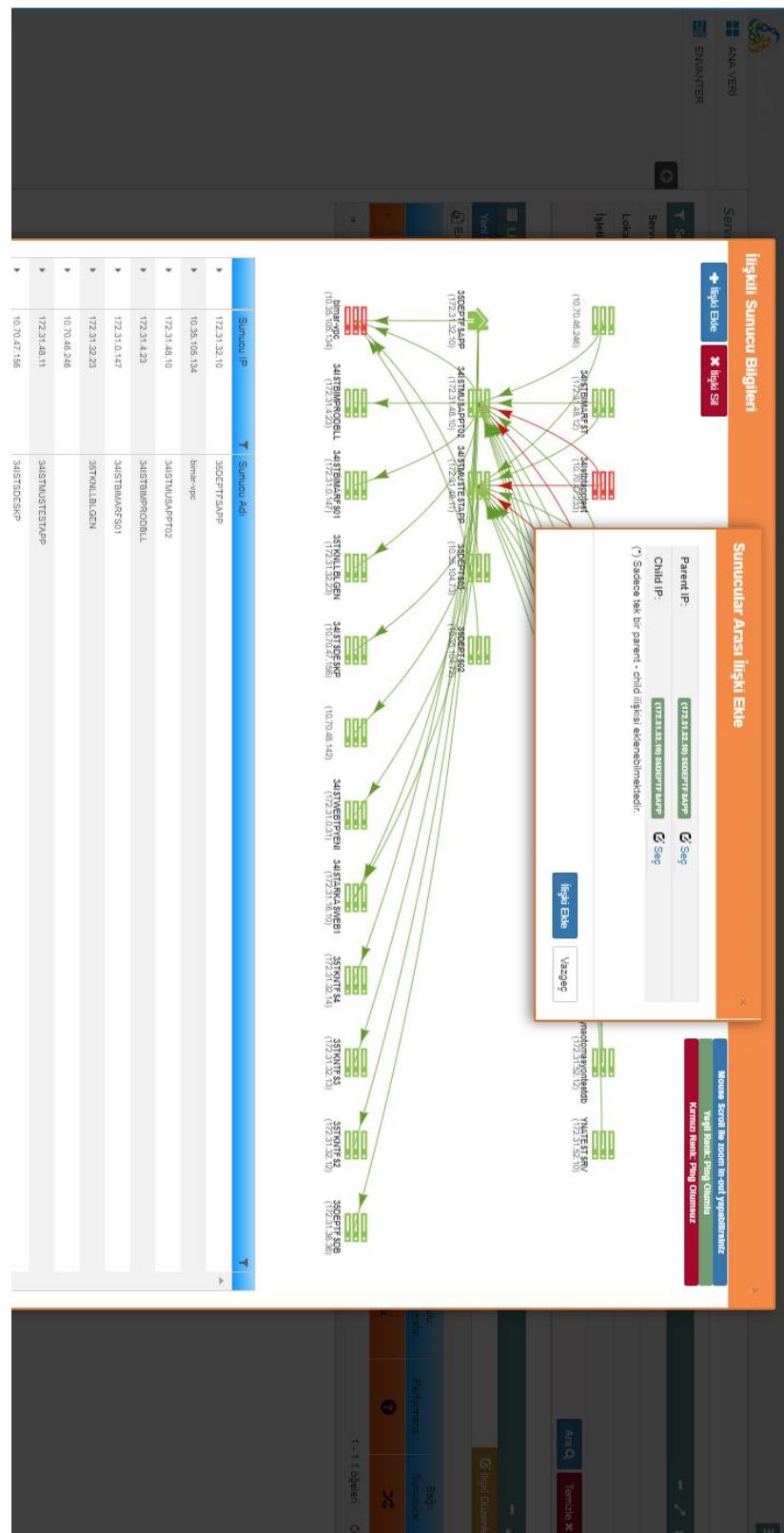
Şekil 8.5'teki bağlı sunucular sekmesine erişim yapıldığı zaman, Şekil 8.14'deki ekran ile karşılaşılmaktadır. Bu sekme içerisinde, ilgili sunucunun bağlı olduğu sunucular bir ağ yapısı olarak üst kısımda, bu ağın içerisinde olan sunucuların barındırdığı site ve servis adresleri de alt kısımda tablo şeklinde yer almaktadır. Sol üst kısımda ise, ilgili sunucuya ilişki ekleme ve ilgili sunucudan ilişki silme işlemleri yapılabilmektedir.



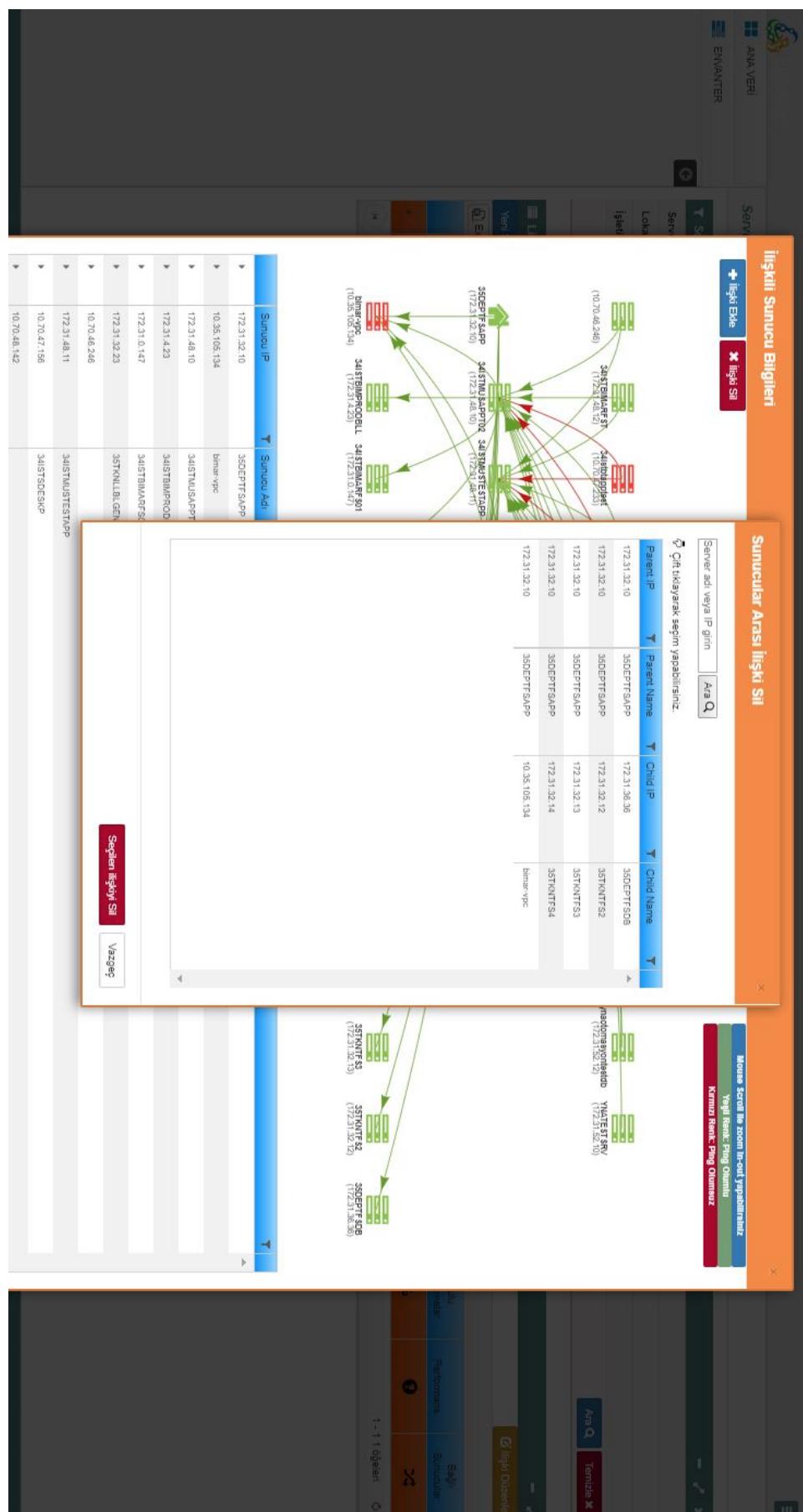
Şekil 8.14 Bağlı sunucular sekmesi

Şekil 8.14'deki; sol üst kısmında bulunan ilişki ekle sekmesine erişim yapıldığında, Şekil 8.15'deki ilişki ekleme ekranı gelmektedir. Şekil 8.3'deki işlemlerinin tekrarı yapıldığı takdirde, ilgili sunucuya yeni ilişki eklenmesi işlemi gerçekleştirilebilmektedir.

Şekil 8.14'deki; sol üst kısmında bulunan ilişki sil sekmesine erişim yapıldığında, Şekil 8.16'daki ilişki silme ekranı gelmektedir. İlgili sunucunun; mevcut ilişkisi listeden seçilip silindiği takdirde, ilişki veritabanı tarafından da tamamen silinmiş olmaktadır.



Şekil 8.15 İlişki ekleme sekmesi



Sekil 8.16 İlişki silme sekmesi

9. SONUÇ

Özetle bu çalışmada; bir sistem içerisinde bulunan sunucuların yönetimi tek bir portal üzerinden sağlanmış olup, sunucuların birbirleri arasında oluşturdukları hiyerarşik ağ yapısı çıkartılarak; herhangi bir sunucuda arıza olması durumunda sorunun etkisi ve kapsamı belirlenebilmektir.

Çalışmanın özgün değerleri şu şekildedir:

- Sunucular tek elden yönetilebilir olmakta,
- Sunucuların arasındaki ilişkilerin görselleştirilmesi sağlanmaktadır.

Çalışmanın yaygın etkileri şu şekildedir:

- Rekabet gücünün artırcı etkisi bulunmakta,
- Lisanslanabilir,
- Yeni projelerin üretilmesine olanak tanımakta,
- Dışa bağımlılığı azaltmaktadır.

Çalışmanın uygulanabilirliği şu şekildedir:

- Fizibilitesi yüksek bir uygulamadır,
- Maximum kazancı esas alınarak geliştirilmiştir. Bakımı için fazla masraf gerekmemektedir.

9.1. Gelecek Çalışmalar

Bitirme tezi kapsamında yapılan çalışmanın, geliştirilebilir yönleri olarak şu maddeler düşünülmektedir:

- Çalışma web tabanlı olmanın dışında, farklı platformlar için de geliştirilebilir olarak görülmektedir.
- Çalışma kapsamında 2 boyutlu olan sunucu ağ görselleştirmesi, 3 boyutlu ve daha gelişmiş bir şekilde geliştirilebilir olarak görülmektedir.

KAYNAKLAR DİZİNİ

- [1] C# Programlama Klavuzu, <https://docs.microsoft.com/tr-tr/dotnet/csharp/programming-guide/> (Son Erişim 18.06.2019)
- [2] .NET Dökümantasyonu, https://docs.microsoft.com/tr-tr/dotnet (Son Erişim 18.06.2019)
- [3] Telerik.com, Kendo-ui Dökümantasyonu, <https://docs.telerik.com/kendo-ui/> (Son Erişim 18.06.2019)
- [4] Visjs.org, Visjs Framework Dökümantasyonu, <http://visjs.org/docs/network/> (Son Erişim 18.06.2018)
- [5] C#'de LINQ'e Başlarken,
<https://docs.microsoft.com/tr-tr/dotnet/csharp/programming-guide/concepts/linq/getting-started-with-linq> (Son Erişim 18.06.2019)
- [7] Downtime , <https://wmaraci.com/nedir/downtime> (Son Erişim 03.06.2019)
- [8] **Halil İbrahim Ulus**, Kurum İçİ Saldırılara Karşı Sunucu İzleme ve Takip Sistemi, Temmuz 2017
- [9] LINQ Query, <https://docs.microsoft.com/tr-tr/dotnet/csharp/programming-guide/concepts/linq/type-relationships-in-linq-query-operations>
(Son Erişim 18.06.2019)
- [10] LINQ Query Operations, <https://docs.microsoft.com/tr-tr/dotnet/csharp/programming-guide/concepts/linq/type-relationships-in-linq-query-operations> (Son Erişim 18.06.2019)
- [11] Bootstrap, <https://getbootstrap.com/docs/4.2/getting-started/introduction/> (Son Erişim 18.06.2019)
- [12] Kendo UI HTML Framework Nedir?, <http://www.burakavci.com.tr/2017/01/kendo-framework-javascript.html> (Son Erişim 18.06.2019)

KAYNAKLAR DİZİNİ (devam)

- [13] Jquery.com, Jquery Dökümantasyonu, <https://api.jquery.com/> (Son Erişim 18.06.2019)
- [14] **Can, Ö. Akbaş**, M. F. (2014). Kurumsal Ağ ve Sistem Güvenliği Politikalarının Önemi Ve Bir Durum Çalışması. TÜBAV Bilim Dergisi, 7 (2), 16-31.
- [15] **Richard A. Gahan, John Hickey**, Server monitoring and failover mechanism, 31.10.2002
- [16] Working With MultiColumnComboBox with .Net MVC - KendoUI, <https://www.c-sharpcorner.com> (Son Erişim 18.06.2019)
- [17] Javascript Tutorial, <https://www.w3schools.com/js/> (Son Erişim 18.06.2019)
- [18] UX Design, <https://www.interaction-design.org/literature/topics/ux-design> (Son Erişim 18.06.2019)
- [19] ASP.NET MVC Gelişmiş Konular, <https://docs.microsoft.com/tr-tr/aspnet/mvc/overview/advanced/> (Son Erişim 18.06.2019)
- [20] No more MVC frameworks, <https://www.infoq.com/articles/no-more-mvc-frameworks/> (Son Erişim 18.06.2019)
- [21] ASP NET MVC Nedir, Ne işe yarar? , <http://www.borakasmer.com/asp-net-mvc-nedir-ne-ise-yarar/> (Son Erişim 18.06.2019)
- [22] Network Examples, https://visjs.org/network_examples.html (Son Erişim 18.06.2019)
- [23] What is code first? , <https://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx> (Son Erişim 18.06.2019)