



BASİT ARAMA MOTORU
(BM495 BİLGİSAYAR PROJESİ)

Samet YILMAZ

GAZİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

Ocak 2018

ÖZET

Bu projede internet kullanıcıları için web platformuna yönelik bir arama motoru geliştirilmiştir. Bu alanda daha önce yapılmış olan arama motorları incelenmiştir. Uygulamalar üzerinde fizibilite çalışması yapılmış olup, avantaj ve dezavantajları göz önüne alınarak ‘Basit Arama Motoru’ uygulaması geliştirilmeye başlanmıştır.

Projenin geliştirilme aşamaları detaylı ve açık bir biçimde raporda anlatılmaktadır. Yapılan sistemde kullanılan algoritmaların sonuçları ve test aşamaları bulunmaktadır.

Sayfa Adedi 20

Danışman : Prof. Dr. M. Ali AKCAYOL

TEŞEKKÜR

Proje geliştirme aşamasında bölümümüzde eğitim veren çok saygıdeğer hocalarımıza ve asistan hocalarımıza, proje geliştirme aşamasında, bana yardımcı olan ve desteğini esirgemeyen, anlayışlı tavrı ile beni sürekli motive eden saygı değer asistan hocamız Arş. Gör. Anıl UTKU hocamız ile danışman hocam olan Prof. Dr. M. Ali AKCAYOL hocamıza teşekkürlerimi sunarım.

İçindekiler

1. Projenin Gerçekleştirilmesi	2
1.1. Proje Özeti	2
1.2. Proje Hedefi	2
1.3. Projede Mevcut Aşama	3
1.3.1. Tasarım Kararları	3
1.3.1.1. Veritabanı Tasarımı	3
1.3.1.2. Yazılım Modüller	4
1.3.1.3. İstisnai Durumların İdaresi	9
1.3.2. Test	9
1.3.2.1. Test Edilen Özellikler	9
1.3.2.2. Test Sonuçları	9
1.3.2.3. Teste Yönelik Belirtilmemiş Değişiklikler	14
2. Durum Değerlendirmesi	14
2.1. Test Durumları	14
2.2. Gereksinim Değişiklikleri	15
2.3. Projenin Gerçekleştirilmesi	15
2.4. Projenin Sonuçlandırılması	15
3. Projenin Katkısı	16
3.1. Ekip Üyelerine Katkısı	16
3.2. Bölüme Katkısı	16
3.3. Ülkemize Katkısı	16
4. Projeye Yönelik Gelecek Çalışmaları	17
5. Referanslar	18

1. Projenin Gerçekleştirilmesi

1.1. Proje Özeti

Günümüzde internet üzerinde en çok kullanılan teknolojilerden biri olarak geliştirilmeye başlayan 'Basit Arama Motoru' internet üzerinden yapılabilecek aramalara cevap vermek için düşük maliyetli bir sistem olarak geliştirildi. Bu sistemde kullanılan basit ara yüzü sayesinde arama motoru kullanıcıların rahatlıkla kullanabileceği bir sistemdir. Web crawler ve arama algoritmalarıyla kullanılabilecek botlar geliştirilerek sistemin daha hızlı yanıt vermesi ve yapılacak aramalara istenilen hassasiyet düzeyinde yanıtları vermesi için çözümler ile projenin daha iyi bir sonuç üretmesi gereklilik haline getirilmiştir. Bu çalışmada arama algoritmaları ve internet siteleri içeriklerini ayırtmak için kullanılan sistemler geliştirerek arama motoru olarak kullanabileceğimiz bir yapı oluşturulmuştur.

1.2. Proje Hedefi

Basit arama motoru internet üzerindeki bilgi birikiminin artması ve web sayfalarının sayılarının her geçen gün artışıyla beraber sitelere ulaşılması için arama motorları bir ihtiyaçtan çıkıp gereklilik haline gelmiştir. Yapmış olduğum proje ile bu ihtiyacı karşılamak ve kendi geliştirmiş olduğum arama motorunu insanların bilgi almak için kullanabileceği bir sistem haline getirmektir. Ayrıca yapmış olduğum proje geliştirilmeye açık bir sistem olduğundan insanlara daha faydalı bir platform haline getirilmeye çalışılmıştır. Bu sistemi kullanan kullanıcıların artmasıyla birlikte daha farklı entegreler yapılarak geliştirilebilir. Programın ilk sürümü olarak oluşturduğum sistemin kullanıcılara özgü, onların isteklerini hızlı ve doğru bir şekilde yanıtlamasını amaçlanmıştır.

1.3. Projede Mevcut Aşama

1.3.1. Tasarım Kararları

Arama motorları, internet üzerinde bulunan bilgiyi aramanıza yardımcı olan temel araçlardır. Gelişmiş arama motorları olmasaydı gün geçtikçe devasa şekilde büyüyen Internet ağı üzerinde ki bilgiyi bulabilmek, elinizde belirli bir URL adresi de yok ise, gerçekten çok zor olurdu. İnternet'in genelinde arama yapan motorlar, aslında ağı doğrudan taramazlar. Her biri, sunucular üzerinde bulunan milyarlarca web sayfasından seçilerek oluşturulan ve web sayfalarının metin biçimlerinin tamamını içeren bir veri tabanını tararlar. Yani bir arama motoruyla ağı taradığınızda, bir anlamda, gerçek web sayfasının bir miktar bayatlamış bir kopyasını aramaktasınız. Ancak arama motorlarının arama sonucunda buldukları bağlantılara (linklere) tıkladığınızda, elbette o sayfanın güncel halini asıl sunucudan almış oluruz.

Temel olarak üç tip arama motoru vardır: örümcekler (spiders) ile çalışanlar, insanların iletileri ile çalışanlar ve bu iki tipin bir bileşimi ile çalışanlar. Bu proje iki tipin birleşimi olarak tasarlanmış ve her kısım için farklı tasarımlar oluşturulmuştur. Bu tasarımların kararları başlıklar halinde detaylı bir şekilde anlatılmıştır.

1.3.1.1. Veritabanı Tasarımı

Yapılan projemiz bir arama motoru olduğundan dolayı sistemin çalışabilmesi için internet üzerindeki web sayfaları hakkındaki bilgileri depolamak büyük bir veri tabanı demektir. Yapılan sistem açısından veri tabanı önemli bir yere sahiptir. Bu yüzden veri tabanı tasarlanmadan önce yapacak olduğumuz sistem için web sayfalarından alınacak olan bilgilerin neler olacağına karar verildi. Bizim için gerekli olan bilgiler web sayfalarının internet üzerinde yayınlar iken oluşturulan başlıklar ve SEO çalışmaları yapılırken arama motorlarının web sayfasını tanıması için her web sayfasını tanımlayan kelimeler, en önemli kısımlardan biri olarak tanımlaya bileceğimiz web sayfalarının içeriklerini yerleştirdikleri body kısmında bulunan textleri veri olarak kullanmaktayız. Veri olarak kullanacağımız kısımların text uzunlukları farklı olduğu için MYSQL üzerinde kayıt işlemi yapılırken bu uzunluklar göz önüne alınarak veri tablosu

oluşturulmuştur. Yapmış olduğum aramalar bu tablo üzerinde gerçekleştirilmektedir. Oluşturulmuş web crawler yapım bu veri tabanı için web sayfalarının içinde bulunan bilgileri bu veri tabanı üzerine kayıt işlemi yapmaktadır. Yapılan proje de arama işlemi ve veri kayıt işlemi aynı anda farklı sınıflar sayesinde yapılmaktadır. Yani veri tabanı içinde arama yaparken başka bir sınıfta bu veri tabanı üzerine veri saklama işlemi yapmaktadır. Bu işlemleri yaptığım kayıt işlemlerinin tablosu Şekil-1 de görüldüğü gibidir.



The screenshot shows a database management interface with a table structure view. The table has four columns: #, Adı, Türü, Karşılaştırma, Öznitelikler, Boş, Varsayılan, Açıklamalar, Ekstra, and Eylem. The rows represent the table's fields: id (int(11)), link (longtext utf8mb4_turkish_ci), title (longtext utf8mb4_turkish_ci), and body (longtext utf8mb4_turkish_ci). Each row has a set of icons for actions like edit, delete, and insert.

#	Adı	Türü	Karşılaştırma	Öznitelikler	Boş	Varsayılan	Açıklamalar	Ekstra	Eylem
1	id	int(11)			Hayır	Yok		AUTO_INCREMENT	Değiştir Kaldır Birincil Benzersiz Index Daha fazla
2	link	longtext utf8mb4_turkish_ci			Hayır	Yok			Değiştir Kaldır Birincil Benzersiz Index Daha fazla
3	title	longtext utf8mb4_turkish_ci			Hayır	Yok			Değiştir Kaldır Birincil Benzersiz Index Daha fazla
4	body	longtext utf8mb4_turkish_ci			Hayır	Yok			Değiştir Kaldır Birincil Benzersiz Index Daha fazla

Şekil-1 Veri tablosu

Şekil-1 de görüldüğü üzere uzunlukların ne kadar olacağı bilinmediğinden boyutu büyük tutulmuştur. Bu veri tabanını kullanmak için xampp platformundan yararlanılmıştır. Xampp, ücretsiz olup açık kaynak kodlu geliştirilen çalışmaları, kolay ve hızlı bir şekilde yerel bir web sunucu oluşturan yazılım paketidir. Web sunucusunu kolay bir şekilde oluşturmamızı sağlıyor dememiz ki neden ise bir web sunucusu için gerekli olan bütün bileşenleri paket halinde bize sunmasıdır. Bu platform sayesinde oluşturmuş olduğum veri tabanını rahat bir şekilde kullanabilmekteyim.

1.3.1.2. Yazılım Modüller

Yazılım olarak açık kaynaklardan oluşan bir dil olan java dili seçilmiştir. Bu proje için javanın seçilme sebebi web sayfaları içinden bilgi alma ve bu verileri işlemek için geliştirilen kütüphanelerin daha fazla işlem yapabilmesidir. Bu proje açısından büyük bir avantajdır. Yapılacak olan proje yazılım olarak 3 farklı kriter üzerinden ele alınabilir.

Bu kriterlerden ilki bize web sayfaları hakkında bilgi toplayacak olan bir yapı olan web crawler yapısı olarak tanımlanabilir. İkinci olarak kullanıcıların göreceği ve arama işlemlerini gerçekleştireceği arayüz olarak tanımlanmıştır. Üçüncü ve son olarak kullanıcıların arama yapmak için kullanacağı arayüze bağlı olan asıl arama işlemini gerçekleştirecek olan arama algoritma sınıfından oluşmaktadır. Bu sınıflar detaylı olarak alt başlıklar halinde incelenmiştir.

Arayüz

Arayüz olarak diğer arama motorlarına çok benzer sade ve kullanımı kolay bir görünüm elde edilmiştir. Bu şekilde bir arayüz tasarlanmasının sebebi kullanıcıların rahat bir şekilde arayüzü anlayabilmesidir. Yapılan arayüz kısmına ek olarak her kullanıcının daha önceki yaptığı aramalara hızlı bir şekilde ulaşabilmesini sağlamak için bir javascript kodu eklenmiştir. Böylelikle önceki aramalara bakarak daha hızlı giriş yapmaları sağlanmıştır. Yapılan aramaların sonuçları farklı bir sayfada sıralanmıştır. Bu arayüz tasarımsal olarak Şekil-2 ve Şekil-2.1' te gösterimi mevcuttur.

GAZİ ÜNİVERSİTESİ

Aranacak kelimeyi yazınız..

Arama

Şekil-2 Ana Sayfa

GAZİ ÜNİVERSİTESİ

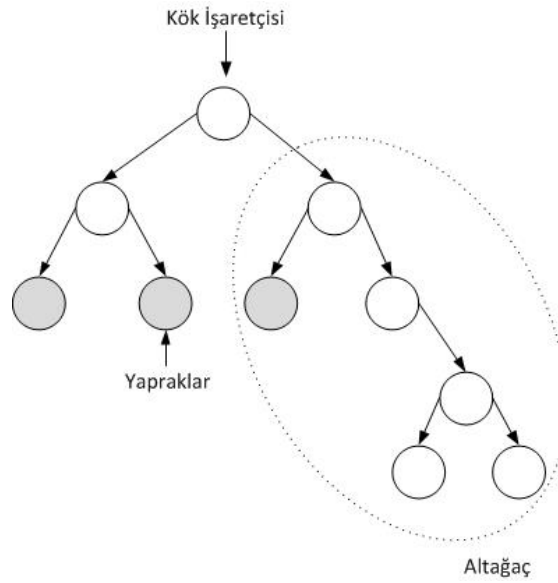
f
facebook

Arama

Şekil-2.1 Anasayfa eklentisi

Web Crawler

Bu sınıf bizim en önemli sınıflarımızdan biridir. Sebebi ise bu sınıf bizim için internet üzerinde bulunan her sitenin içeriği, SEO ve title gibi site hakkında bilgiyi alarak veri tabanı kısmında bu bilgilerin saklanmasını sağlamaktadır. Web sayfaları aslında bir ağaç yapısı içermektedir.[Şekil-2.2]



Şekil-2.2 Ağaç yapısı

Girmiş olduğumuz bir web sayfasının url'sini bu ağaç üzerindeki kök işaretçisi olarak düşünebiliriz. Web crawler yapısı bu kök işaretçisini referans alarak tarama ve bilgi

alma işlemi yapmaktadır. Bu işlemi yaparken yapraklar kısmının oluşturan diğer linkleri bir listeye alarak kök işaretçisinden sonra yapraklarda bulunan farklı linkleri de gezer ve bu web sayfaları üzerindeki bilgileri de bizim için kullanılabilir bir hale getirmektedir. Ayrıca web crawler bu işlemleri yaparken gezmiş olduğu linklere bir daha gitmeyerek tekrarı ortadan kaldırmaktadır. Bu şekilde web sayfalarındaki bilgileri almaktadır. Fakat web sayfaları farklı dillerde kodlama içermektedir. Bu yüzden bu web kazıyıcısı bilgileri alırken tag diye adlandırdığımız kısımları html kodlamalarından ayırarak bizim kullanabileceğimiz bir şekilde sadece web sayfaların bilgilerinin alınmasını sağlamaktadır. Bu işlem bize hem veri tabanındaki bilgilerin uzunluğunun kısaltılmasını hem de arama yaparken veri tabanındaki bilgilerin anlaşılabilirliğini sağlamış olmaktadır.

Ayrıca bu işlem yapılırken anlaşılmasını kolaylaştırmak için 'Zemberek' adı verilen bir kütüphanesinden faydalanılmaktadır. Bu kütüphane açık kodlu Türkçe Doğal dil işleme kütüphanesi ve OpenOffice eklentisidir. Bu sayede alınan veriler üzerindeki herhangi farklı bilginin bizim arama sınıfımız için anlaşılabilirliği artmış olmaktadır.

Fakat web crawler bu işlemleri yapabilmesi için bir tane root olarak giriş yapılacak olan bir web url'sine ihtiyaç duymaktadır. Bu işlemleri sadece bu web sayfası ve yaprak olan url'ler üzerinde gerçekleştirebilmektedir. Bu yapının devamlılığını sağlamak için farklı url'ler bulacak bir sınıfa ihtiyaç duyulmaktadır. Bu işlem de oluşturulmak için iki benzer algoritma olan BFS(Breadth First Search) ve DFS(Depth First Search)'den yararlanılmıştır.

BFS(Breadth First Search), ağaç veya grafik veri yapılarını incelemek veya araştırabilmek için kullanılan bir algoritmadır. Temel prensip olarak en yakın düğümleri ziyaret etmeyi hedefler. En yakındaki düğümler taranarak uzaktaki düğümlere gidilir. Yakın olan düğümler öncelikli olarak ziyaret edileceği için bizim bir veri yapısı kullanmamızı gerektirmektedir. Bu sınıf ile farklı url'lere ulaşılabilmekte ve bu bulunan linkleri web crawler sınıfımıza root değeri olarak yollanarak siteler hakkındaki bilgiyi rahat bir şekilde elde edebilmektedir.

DFS(Depth First Search) ,algoritması da BFS algoritmasıyla aynı amacı sahip bir algoritmadır. Ancak çalışma mantığı farklıdır. BFS algoritması temelde en yakın komşuları ziyaret etmeyi hedeflemektedir. DFS ise gidebildiği yere kadar gitmeyi, daha

sonra geri dönerek grafi dolaşmayı hedefler. Yani BFS komşuları ziyaret etmeyi hedeflerken DFS ise sürekli olarak komşunun komşusuna gitmeyi amaçlamaktadır.

Sonuç olarak bu iki algoritmaların bulundukları sınıflar projeye hizmet etmek için farklı url'lere ulaşılabilmekte ve bu bulunan linkleri web crawler sınıfımıza root değeri olarak göndermekte ve siteler hakkındaki bilgiyi rahat bir şekilde elde edebilmektedir. Bu algoritma uygulanırken tekrarı önlemek için kuyruktaki url'eler bir liste yardımı ile tutulmuş ve tekrarı gelmemesi için önlem alınmıştır. Bu sınıflar sayesinde oluşturulan veri tabanı üzerinde arama işlemi yapacak olan bir de arama sınıfı bulunmaktadır.

Arama Sınıfı

Bu sınıf kullanıcıların yapmak istedikleri aramalara yanıt vermesi için tasarlanmış olan bir sınıftır. Kullanıcıların girmiş olduğu kelime veya kelime öbeklerinden hangi web sayfasına ulaşmak istediğini veri tabanından bularak ve belirli bir kurala göre sıraya yerleştirerek onların ulaşmasını sağlamak için tasarlanmıştır. Arama motorları internetteki sayfaların ne kadar ziyaretçi aldığını, içeriklerinin ne kadar kaliteli olduğunu ve bizim o sayfaya hangi olasılıkla ulaşabileceğimizi ölçmek için her sayfaya bir değer atar. Sayfa değeri veya sayfa popülaritesi diye Türkçeleştirilen PageRank, bu değere verilen addır. Sayfa değeri hesaplaması belli bir algoritmaya göre yapılmaktadır. Bu algoritma internet sayfalarının birbirleriyle olan ilişkilerini ölçebildiği gibi, aslında birbirleriyle bir şekilde ilişkili olan herhangi bir veri yığınınına da uygulanabilir. Sayfa değeri bir sayfanın değerini hesaplarken, o sayfaya diğer sayfalardan verilen bağlantıları göz önüne alır. Ancak bağlantı veren sayfaların çok ziyaret edilmesi onların verdikleri bağlantıların değerini artırır; çünkü bu sayfalardan yola çıkarak diğer bir sayfaya ulaşma olasılığımız daha yüksektir. Fakat benim yapmış olduğum arama motoru bir server üzerinde çalışmadığı için bu PageRank değerleri elimde bulunmamaktadır. Bu yüzden şu anda localde çalıştığı için bu değerleri göz ardı ederek bir arama algoritması uygulanmıştır. Bu algoritma oluşturulurken bu değerler yerine algoritma içerisinde başlık, açıklama ve içerik kısmına farklı ağırlıklar verilerek arama işleminin yapılması sağlamaya çalışılmıştır.

1.3.1.3. İstisnai Durumların İdaresi

Yapılmış olan proje bir arama motoru olduğundan dolayı server üzerinde çalışması gerekmektedir. Ancak bu sistemi kurabileceğim herhangi bir servera sahip olmadığımından dolayı yapılan işlemlerin hepsi localde çalışmaktadır. Bu yüzden oluşturmuş olduğum veri kaynakları kısıtlıdır. Elimde bulunan kaynaklar ile arama yapılabilecek bir sistem yapmış olsam da bu sistem yapılacak aramalara kısıtlı cevaplar vermektedir. Bu yüzden bu sistemin yararlılığı açısından server üzerinde testler tekrarlanarak yapılacak olan iyileştirmeler ile daha iyi sonuçlar alınabilecektir.

1.3.2. Test

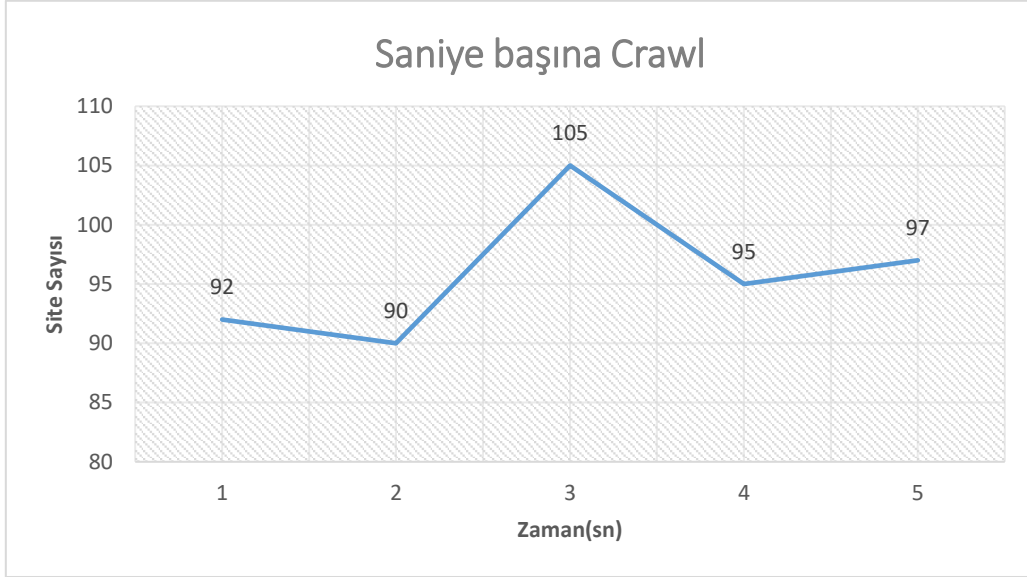
1.3.2.1. Test Edilen Özellikler

Yapmış olduğum proje localde çalıştığından dolayı stres testleri gibi testler uygulanamamıştır. Localde uygulanmış olan testler web crawler hızı, BFS ve DFS algoritmalarının hızı, Precision, recall ve f – measure'den oluşmaktadır. Bu testler için localde elde edilen sonuçlar 1.3.2.2 Test Sonuçları başlığı altında verilmektedir. Ayrıca projenin localde çalışması ve PageRank değerlerinin elle girişinin yapılması precision ve f-measure değerlerinin doğru olup olamayacağı hakkında net bir bilgi vermemek ile birlikte bu testler içi kesin sonuçların elde edilebilmesi için yapılan sistemin hayata geçirilip belirli bir süre kullanıldıktan sonra testlerin tekrarlanması doğruluk açısından önemlidir. Arama motorları internet ağı üzerinde çalışan sistemler olduğundan dolayı bu sistemin her testinin server üzerinde tekrar yapılması doğrulukları açısından gerekli görülmektedir.

1.3.2.2. Test Sonuçları

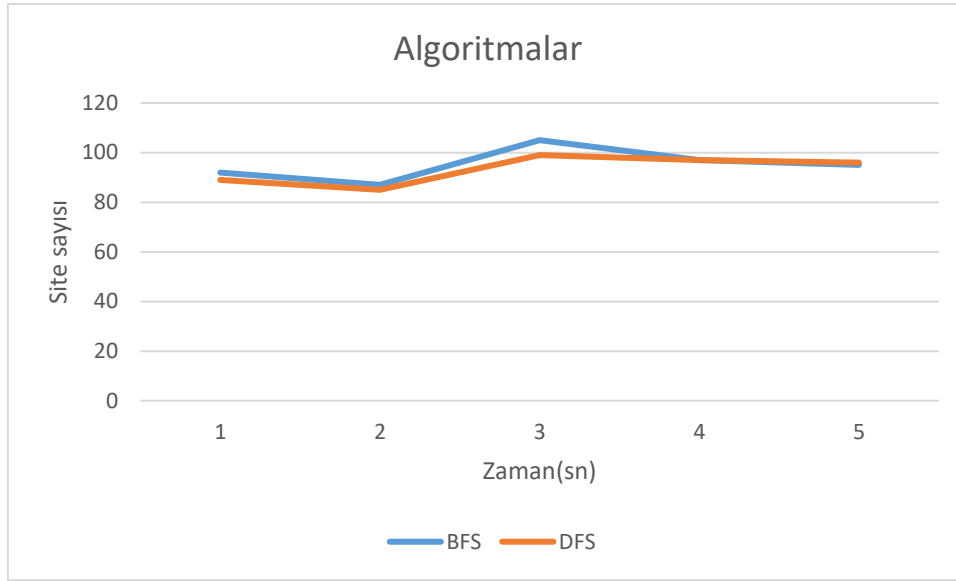
Yapılan testlerden ilki olan web crawlerın web sayfalarının içerisindeki bilgileri almadaki hızı incelenmiştir. Yapılan web crawlerın sonuçlarını almak için test amaçlı bir root olarak elle bir web sayfası giriş olarak verildi. Root olarak verilen web sayfasının içerisindeki linklerin fazla olması gerektiğinden yapılan incelemede '<http://www.hurriyet.com.tr/>' içerisinde fazla link barındırdığı tespit edildi ve test amaçlı

root linki olarak kullanıldı. Testin sonucunda elde edilen grafik Şekil-3 de görüldüğü gibi farklılıklar göstermektedir. Bu farklılıklarının oluşmasının sebebi ise ziyaret edilen web sayfalarının tekrar gelmesi ve bu sayfaların linklerinin elenmesinde harcanan süre ile alakalıdır.



Şekil-3 Web Crawler Test Sonucu

Diğer hız testi olarak yapılan testlerde web crawler için root değerlerini bulan ve bunları crawl edilmesi için gerekli sınıfa yollama işlemlerini gerçekleştiren iki farklı algoritma olan BFS ve DFS algoritmaların bulundukları sınıflardır. Bu sınıflara hız testi uygulanmasının sebebi ise sınıfların bizim crawl işlemini yapacak olduğumuz ana kök olarak değerlendirdiğimiz linkleri oluşturmalarıdır. Bu işlem web sayfalarının linklerini crawl etme süremize büyük bir etki oluşturmaktadır. Bu işlem yapılırken algoritmaların farklı olması sebebiyle ayrı olarak incelenmiştir, fakat aynı amaca hizmet etikleri için yapılan test sonuçların ortak verileri birlikte incelenmiş ve hangi algoritmanın projenin gelişimi için uygun olduğuna karar verilmeye çalışılmıştır. Bu işlem için ilk olarak bu algoritmaların buldukları web sayfalarının ne kadar sürede buldukları incelenmiştir.[Şekil-3.1]



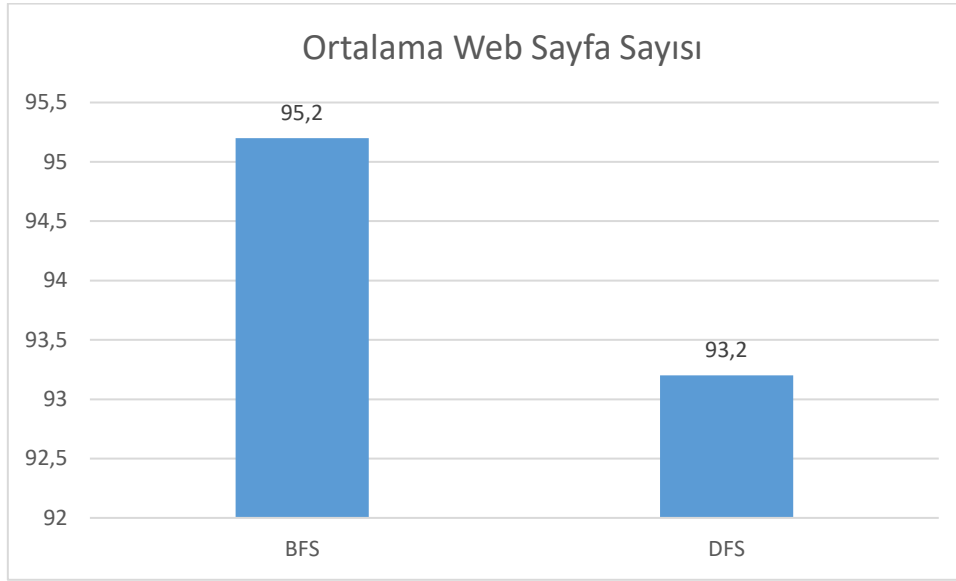
Şekil-3.1 BFS ve DFS Algoritmaları

Burada bulunan grafiğe baktığımızda web sayfalarının hem aynı algoritma hem de iki algoritma üzerindeki değerlerin farklı olduğu görülmektedir. Aynı algoritmanın süre içerisinde farklı sayıda web sayfasına gitmiş olmasının sebebi gezilen linklerin birbirinden farklı olması gerektiği için aynı linklerin sayılmamasıdır. Algoritmaların arasında farklılık oluşmasının sebebi ise BFS algoritmasının yapısı gereği önce komşularına giderek bu işlemi tamamlaması gerektiği fakat DFS algoritmasının ilk olarak son child'a giderek buradan ana köke geri dönüş yapmaya çalışmasıdır. Bu farklılığı daha iyi anlayabilmek için gidilen web sayfalarının sayısı tablo üzerinde gösterilmiştir.[Tablo-3.1]

Algoritmaların grafik verileri					
Zaman(sn)	1	2	3	4	5
BFS	92	87	105	97	95
DFS	89	85	99	97	96

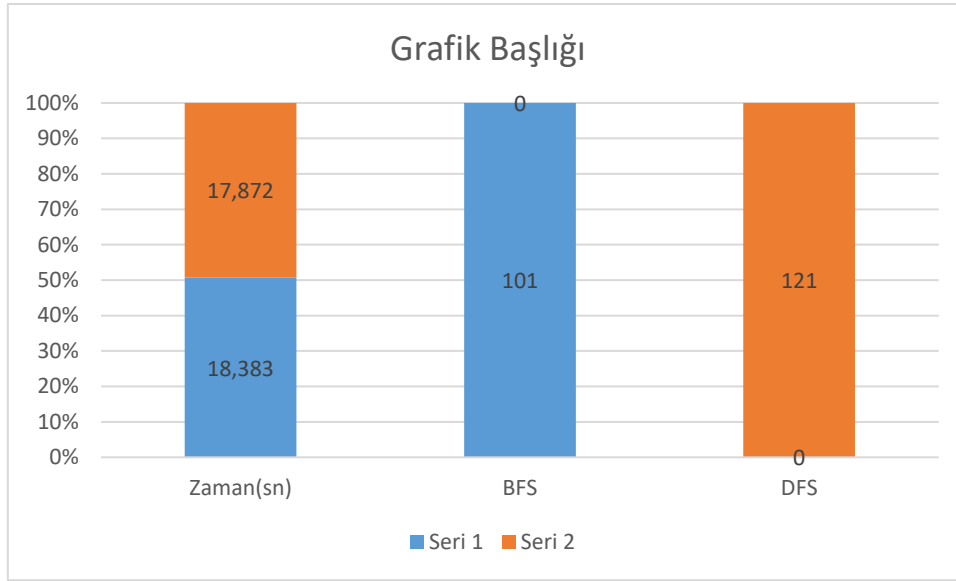
Tablo-3 Zamana bağlı web sayfa sayısı

Bu tablodan anlaşıldığı gibi aynı sürelerde her zaman aynı değerdeki web sayfaları gezilmemiştir. Bu değerler baz alınarak 5 sn içerisindeki ortalama değerleri Şekil-3.2 gösterilmiştir.



Şekil-3.2 Algoritmaların ortalama değerleri

Bu değerler arasındaki fark az gibi gözükse de yapmış olduğumuz proje açısından crawl etme hızına etki etmektedir. Sistem açısından bu hız çok önemli olduğunu daha öncesinde de vurguladığımız için bu işlem sonucundan BFS algoritması iyi gözükmemektedir. Fakat bu testi yaparken kullanmış olduğumuz root urlsi olan <https://www.facebook.com/> üzerinde uzun süre çalışması sonucunda algoritmalar arasında büyük bir fark olduğu anlaşılmıştır. BFS algoritması DFS algoritmasına göre daha uzun sürede daha az linke ulaşım sağlamıştır. Bu ayrıntı göz önüne alınarak farklı linkler üzerinde denemeler yapıldı ve bu farklılık diğer linkler üzerinde de gerçekleşti.[Şekil-3.3]



Şekil-3.3 Belirli root üzerindeki süreye göre bulunan linkler

Oluşan bu farkın crawl üzerinde nasıl bir farklılık meydana getirdiği incelendi. Bu inceleme sırasında iki algoritma için belirli bir süre içerisinde bizim için kaç web sayfasının crawl işleminden geçtiğine bakıldı. Yapılan işlem sonuçları tam olarak oluşturulamamıştır. Sebebi ise programın localde çalışması için kullanılan cihazı fazla ısınarak hata vermesidir. Önceki sonuçlar göz önüne alındığında da webcrawler sınıfına gidecek olan web sayfaları sayısı arasında fark olması crawl edilecek olan web sayfalarının fazla olacağı aşikardır.

Bu testlerin yanında arama işlemini yapacak olan sınıf içinde precision, recall ve f-measure testleri uygulanmaya çalışılmıştır. Bu testlerin yapılmasının istenmesi, getirilen sonuçların ne kadar bir doğruluk ve veri boyutu artınca nasıl bir sonuç ortaya çıkacağını göstermek için uygulanmıştır. Buradaki precision(kesinlik kavramı) genelde p harfi ile gösterilir ve getirilen bilgidaki doğru sonuçların, getirilen bilginin tamamına oranı olarak hesaplanır. Recall(Hassasiyet) kavramı da genelde r harfi ile gösterilir ve getirilen doğru sonuçların, getirilmesi gereken doğru sonuçlara oranı ile hesaplanır. F-measure ise bu iki matematiksel işleminden elde edilen sonucun harmonik ortalaması olarak tanımlanabilir. Bulunan bu değerlerin kesin olarak doğru bir sonuç olarak ele alınmamasından dolayı yapılan işlemler sonucunda şuanda kullanılan arama motorlarına göre çok düşük bir hassasiyete sahip olduğu görülmektedir. Bu hassasiyet oranının yükselmesi için önceden bahsedildiği gibi server üzerinde PageRank değerleri atanmasıyla daha sağlıklı diyebileceğimiz bir sonuca ulaşmamız mümkündür.

1.3.2.3. Teste Yönelik Belirtilmemiş Değişiklikler

Test işlemlerinde daha öncesinde yapılmamış olan percision, recall ve f-measure testleri eklenerek test işlemleri çoğaltılmış ve yapılan sistem için geliştirilmesi gereken kısımlar hakkında daha detaylı bilgi edinilmeye çalışılmıştır. Testler ile birlikte yapılan arama sonuçlarının şuanda en çok kullanılan arama motorlarına göre çok düşük değerler alınmasından dolayı projenin daha nasıl katkılar sağlanarak geliştirilebileceğine bakılmış ve bunlar üzerine değişiklikler hayata geçirilmeye çalışılmıştır.

2. Durum Değerlendirmesi

Proje açısından gerekli görülen durum değerlendirmeleri başlıklar halinde sunulmuştur. Değerlendirmeler yapılırken projenin mevcut durumu göz önüne alınmıştır.

2.1. Test Durumları

Basit arama motoru adı altında yürütülen projenin test durumları daha önceki 1.3.2 Test başlığı altında detaylı bir şekilde anlatılmıştır. Orada da belirtildiği üzere yapmış olduğum proje bir web arama motorudur. Bu yüzden yapılan sistemin testlerinden daha iyi verim ve sonuçlar elde etmek için bir server üzerinde gerçekleştirilmesi gerekmektedir. Bu işlem gerçekleştirilmeden önce PageRank algoritması da eklenerek daha net sonuçlar elde edilebilir. Yapılan test durumlarının hepsi localhosttaki çalışmalardan elde edilen sonuçlardır. Bu yüzden yapılan projenin gerçek çalışma ortamı olan internet üzerindeki sonuçları hakkında bir bilgi elimde bulunmamaktadır. Localhost'daki sonuçlar göz önüne alındığında ise BFS-DFS ve web crawler sistemlerinin sonuçları kötü olmamakla birlikte geliştirilebilir bir yapıya sahip olmaları projeden daha iyi sonuçlar elde edilebileceği göstermektedir. Elde edilen sonuçlar göz önüne alındığındaki arama hassasiyeti ve doğruluk oranı düşük olmasından dolayı şuan ki ihtiyaca tam olarak yanıt verememektedir. Fakat günümüzde kullanılan arama motorları ile kıyaslandığında sonuçlar iyi görülmese de sistemin geliştirilebilir olması büyük bir avantajdır.

2.2. Gereksinim Değişiklikleri

Arama motoru olarak kullanılan sistemler bir server üzerinde çalışmaktadır. Bu sistemler için iki olanak vardır: kullanıcıya hızlı yanıt verme veya geç ve doğruluk oranı yüksek sonuçlar göstermelidir. Günümüzde hız büyük bir öneme sahip olması nedeniyle arama motorları kullanıcılara hızlı ve doğruluk oranı da yüksek olan sonuçlar göstermeye çalışmaktadır. Bu işlemi yapmak için maliyeti yüksek tutmaları gerekmektedir. Çünkü internet üzerinde bulunan web sayfalarının sayısı her geçen gün katlanmaktadır. Bu web sayfalarının taramak ve taranan sayfalarının bilgilerinin saklanması için büyük bir veri tabanı gerekmektedir. Bu verileri saklamak için çok sayıda sunucuya ihtiyaç duyulmaktadır. Projenin adı 'Basit Arama Motoru' olarak görülse de yapılmaya çalışılan sistem bir arama motoru olduğundan projenin geliştirilmesi için bu yapılara sahip olması gerekmektedir.

2.3. Projenin Gerçekleştirilmesi

Projem olan 'Basit Arama Motoru' internet kullanıcılarının en çok ihtiyaç duydukları bir web uygulamalarından bir tanesidir. Ayrıca bu proje geliştirilebilir bir sistem olduğundan dolayı yapılacak olan destekler ile birlikte daha da büyük bir sistem haline getirebilecektir. Projenin başlangıcından itibaren yapılan geliştirilmeler ile birlikte bu alandaki ihtiyacı gelişmiş arama motorları gibi sağlayamasa bile geliştirilebilecek bir yapı halini almıştır.

2.4. Projenin Sonuçlandırılması

Basit arama motoru olarak tasarlanan sistem şuanda internet üzerinde kullanımda olan arama motorlarına benzer olarak yapılmıştır. Bu sistem yapılan aramalara cevap verebilen bir sistemdir. Fakat bu sistemlerde en önemli faktörler arasında yer alan kullanıcıların yaptığı arama sonuçlarına hızlı bir şekilde cevap vermesi olarak gösterilebilir. Benim tasarlamış olduğum sistem kullanılan arama motorları kadar ne yazık ki hızlı değildir ama yapılan sistemin güvenilirlik oranı yüksek ve gelişmeye açık bir yapıya sahiptir.

3. Projenin Katkıları

3.1. Ekip Üyelerine Katkıları

Yapmış olduğum proje aslında bir ekip çalışması olması gerekirken ekip arkadaşımın hiçbir katkısı olmadan şahsım ve danışman hocamın katkılarıyla yürütülmüştür.

Bu projenin bana en büyük katkısı arama motorların çalışma mantığı olan istenilen web sitelerinin arama kontrol algoritmasının nasıl gerçekleştiğini kavramamdır. Başka bir katkısı ise verilerin kullanıcıya sunulmadan önce arama motoru tarafından nasıl değerlendirildiğini görmüş olmaktayım. Bu tarz sistemlerin verileri incelerken nelere dikkat ettikleri ve web sayfalarındaki bilgileri nasıl ayrıştırdıkları ile bu işlemi yaparken kullandıkları örümcek yapılarının nasıl bir işleyişe sahip olduğunu detayları ile öğrenmiş bulunmaktayım. İnternet üzerindeki bilgilerin ve web sayfalarının gün geçtikçe artması sebebiyle bu tarz sistemlere olan gerekliliği görmüş ve bu sistemlerin nasıl çalıştığını anlamış bulunmaktayım.

3.2. Bölüme Katkıları

İnternet üzerindeki web sayfaları gün geçtikçe çoğalması ile arama motorlarına duyulan ihtiyaç artmaktadır. Benim yapmış olduğum projede bu ihtiyacı karşılamak açısından önemli ve geliştirilebilir bir yapıya sahip olduğundan dolayı desteklenerek bölümümüz ve çoğu kullanıcının ihtiyacının karşılayabilecek bir proje halini alabilecek yapıya sahiptir.

3.3. Ülkemize Katkıları

Geliştirmeye çalıştığım sistem şuanda internet kullanıcıları tarafından en çok ihtiyaç duyulan sistemler arasında yer almaktadır. Bu sistemi geliştirerek öncelikle ülke çapında sonrasında ise dünya çapında kullanılabilecek sistem olarak geliştirilebilir. Böyle bir sistem ülke açısında gelişmemize büyük bir katkı sağlayacaktır.

4. Projeye Yönelik Gelecek Çalışmaları

Arama motorları, günümüzde internet kullanıcıların en çok kullandıkları sistemler arasında yer alması sebebiyle bu proje gelecek vaad eden bir yapıya sahiptir. Bu şekilde bir yapısı olması sayesinde geliştirmeye açıktır. Yapılacak olan destekler sayesinde kendimize ait olan bir arama motoruna sahip olabiliriz. Bu arama motoru ayrıca bize kullanıcıların yaptıkları aramalara dayanarak ne tarz ihtiyaçları olduğunu belirlemek için kullanılabilir bir sistemdir.

5. Referanslar

1. A multithreaded web crawler and text search engine. Yüksek lisans tezi. Aralık 2017. Kaynak: <http://openaccess.dogus.edu.tr/handle/11376/59#sthash.19q2AQGT.P08PwTQw.dpbs>
2. Web Crawling & Web Scraping Nasıl Yapılmalı? Aralık 2017. Kaynak: <https://kodcu.com/2013/03/web-crawling-web-scraping-nasil-yapilmali-idi/>
3. Zemberek Kütüphanesini Kullanmak Aralık 2017. Kaynak: <http://sariyildiz.net/zemberek-kutuphanesini-kullanmak/>
4. Search Engine Algorithm Basics. Aralık 2017. Kaynak: <https://moz.com/blog/search-engine-algorithm-basics>
5. Jsoup HTML parser Aralık 2017. Kaynak: <https://aboullaitte.me/jsoup-html-parser-tutorial-examples/>
6. Java Programlama Dili Kullanarak Web Sitelerini Crawl Etme. Aralık 2017. Kaynak: <https://kodcu.com/2012/12/java-programlama-dili-kullanarak-web-sitelerini-crawl-etme-tum-linklerini-ziyaret-etme/>
7. Search Result Clustering Studies in Turkish. Aralık 2017. Kaynak: <http://www.kemik.yildiz.edu.tr/data/File/publications/Text%20Classification%20Clustering/Turkce%20Arama%20Motoru%20Sonucu%20Kumeleme%20Calismalari.pdf>
8. Test çeşitleri. Aralık 2017. Kaynak: <http://www.orhanyener.net/yazilim-testi-2/yazilim-test-cesitleri/>
9. Introduction to Web scraping with Java. Aralık 2017. Kaynak: <https://ksah.in/introduction-to-web-scraping-with-java/>
10. Focused Web Crawling Algorithms. Aralık 2017. Kaynak: <http://www.jcomputers.us/vol10/jcp1004-04.pdf>
11. THE SHARK-SEARCH ALGORITHM. Aralık 2017. Kaynak: <http://www.cs.cmu.edu/~dpelleg/bin/360.html#ref-figure2>
12. PageRank. Aralık 2017. Kaynak: <http://jung.sourceforge.net/doc/api/edu/uci/ics/jung/algorithms/importance/PageRank.html>
13. The PageRank Algorithm. Aralık 2017. Kaynak: <http://graphstream-project.org/doc/Algorithms/PageRank/>
14. Page Rank Algorithm and Implementation. Aralık 2017. Kaynak: <https://www.geeksforgeeks.org/page-rank-algorithm-implementation/>
15. Arama motorları nasıl çalışır? Aralık 2017. Kaynak: <http://cisn.odtu.edu.tr/2004-10/arama.php>
16. PageRank (Sayfa Değeri) Algoritması. Aralık 2017. Kaynak: <http://e-berqi.com/y/pagerank-algoritmasi/>
17. WHAT IS SEARCH ENGINE ALGORITHM? Aralık 2017. Kaynak: <http://www.brickmarketing.com/define-search-engine-algorithm.htm>
18. F1 Değerlendirme (F1-Scoring). Aralık 2017. Kaynak: <http://bilgisayarkavramlari.sadievrenseker.com/2010/09/30/f1-degerlendirme-f1-scoring/>
19. Developing a search algorithm. Aralık 2017. Kaynak: <https://softwareengineering.stackexchange.com/questions/166702/developing-a-search-algorithm>

20. Creating a simple search algorithm with PHP and MySQL. Aralık 2017. Kaynak: <http://idiallo.com/blog/php-mysql-search-algorithm>
21. Arama Motorları ve Algoritmalar: Keşfedilen Arama Motoru Algoritmaları. Aralık 2017.
Kaynak: <http://www.seochat.com/c/a/search-engine-optimization-help/search-engines-and-algorithms-search-engine-algorithms-explored>
22. Xampp nedir? Aralık 2017. Kaynak: <https://www.webagi.net/2017/01/08/xampp-nedir-xampp-nasil-kurulur-resimli-anlatim/>
23. Zemberek (Yazılım) nedir? Aralık 2017. Kaynak: [https://www.turkcebilgi.com/zemberek_\(yaz%C4%B1l%C4%B1m\)](https://www.turkcebilgi.com/zemberek_(yaz%C4%B1l%C4%B1m))
24. BFS Algorithm. Aralık 2017. Kaynak: <https://www.quora.com/How-BFS-algorithm-helps-in-web-crawling>