

Tam Metin Arama Kullanılarak Kullanıcı Deneyiminin İyileştirilmesi

Çağdaş ÜSFEKES, Kürşat İNCE

HAVELSAN A.Ş., Ankara, Türkiye

{cusfekes, kince}@havelsan.com.tr

Özetçe: Kurumsal arama altyapıları, yapısal verinin yanı sıra yapısal olmayan metin tabanlı verinin aranmasını sağlayarak arama etkinliğini arttırmaktadır. Doğru kurgulanmış/uyarlanmış bir kurumsal arama altyapısı kullanıcı deneyimine katkı sağlamakta ve verimliliği arttırmaktadır. Bu çalışmada, HAVELSAN'da kurulu bulunan uygulama yaşam döngüsü yönetim sistemi (Geliştirme Ortamı - GO) özelinde iş öğelerinin geliştirilen arama altyapısı ile kullanıcı sorgulamalarına açılması ve bu altyapının kullanıcı deneyimlerine/verimliliğine katkısı anlatılmaktadır. Tam metin arama uygulaması (OKAPI BM25) ve kurumsal özelleştirmeler için HVL-GO verisinin barındığı SQL sunucusu kullanılmıştır. Yapılan geliştirmeler sonrasında, kullanıcı bilgisayarlarından yapılan iş öğesi arama ve ilişkilendirme işlemlerinde hız sağlandığı görülmüştür.

Anahtar Kelimeler: Kurumsal arama, tam metin arama, OKAPI BM25, uygulama yaşam döngüsü yönetimi, iş öğesi ilişkilendirme

Abstract: Enterprise search infrastructure supports searching in structural data and non-structural data with high efficiency. A well designed enterprise search infrastructure increases productivity and contributes user experience. In this work, we describe querying work items with advanced search and effect to user experience with application lifecycle management system which is located in HAVELSAN. SQL Server is used in full-text search algorithm (OKAPI BM25) and enterprise customizations. After these customizations searching and associating work items from user computers are going to faster.

Keywords: Enterprise search, full-text search, OKAPI BM25, application lifecycle management, associate work items

1 Giriş

Kurumsal uygulamaların artışına paralel olarak artan veri hacmi, bu verinin sınıflandırılması, saklanması ve gerektiği zaman çağırılması sorunlarını ortaya çıkarmıştır. Kurumsal uygulamalar tarafından üretilen yapısal verilerin yanında kurumsal süreçler ile

doküman, rapor, resim vb. gibi yapısal olmayan verilerin de sıklıkla üretildiği görülmektedir. Kurumsal uygulamaların sayısı ve çeşitliliğine ek olarak farklı veri türlerinin ilişkişel veritabanı ve içerik veritabanı gibi farklı ortamlarda saklanması verinin geri çağırılmasını zorlaştırmaktadır. Kurumsal arama yeteneđi, kurumsal bilgiye hızlı erişim, bilginin yeniden kullanımın sağlanması, üretim emeđin tekrar etmemesi, “mevcutta ne var” farkındalığının artırılması, birlikte çalışmayı arttırması, kurum içine aradığı kişiyi, uzmanı bulma amaçlarına hizmet etmektedir.

Arama uygulamaları, verinin türüne göre kullanıcılara farklı arayüzler sunmaktadır. Yapısal verilerde arama yaparken daha çok alan üzerinden arama / sorgu sihirbazı kullanılırken, yapısal olmayan verilerde tam metin arama tercih edilmektedir. Sezgisel arama ve öneri sistemleri ise sezgisel arayüzleri kullanıcılara arama kelimeleri önermekte ve bilgiye erişimi hızlandırmaktadır.

Bu bildiriye, HAVELSAN’ın kurumsal uygulamalarından biri olan HAVELSAN Geliştirme Ortamı (HVL-GO)’na eklenen iş öđesi öneri sistemi ve ilişkilendirme özelliğinin geliştirilmesi ve bu özelliğın kullanıcı deneyimine etkileri anlatılmaktadır.

Bildiriye, sezgisel arama ve öneri sistemleri ile ilgili genel bilgi Bölüm 2’de, HVL-GO mimarisi ve mevcut arama özellikleri Bölüm 2’de, HVL-GO’ya eklenen iş öđesi arama ve ilişkilendirme Bölüm 4’te, değerlendirme ve devam eden çalışmalar Bölüm 5’te anlatılmaktadır.

2 Sezgisel Arama ve Öneri Sistemleri

Standart arama yöntemleri kullanıcının çalışmak için kullanıcının sorgu kelimeleri girerek arama yapmalarını önermektedir. Bu yöntemde arama motorları “arama” düğmesine basıldıktan sonra sorguya eşleşen bilgileri kullanıcıya sunmaktadır.

Günümüzün internet arama motorları ve bunların kurum içi uygulamaları, kullanıcı deneyimini iyileştirmek ve aranan bilgiyi en kolay şekilde sergilemek için donanmış durumdadır. Bu mekanizma içerisinde, kullanıcı sorgu kelimelerinin izlenmesi ve kullanıcının arama sonuçlarına verdiği tepki, kullanıcı davranışlarının takibi yer bulmaktadır. Bunun sonucu olarak da sezgisel arama kabiliyeti sunmak mümkün olmaktadır. Sezgisel arama sırasında arama motorları kullanıcı sorgu kelimeleri girmeye başladığında bu kelimeleri tamamlamakta, kelimeler için cümlecik önerileri sunmakta hatta sorgu sonuçlarını henüz kullanıcı kelimeleri yazarken sergilemektedir. Böylelikle kullanıcı, sorgu kelimelerine ek olarak cümlecikler tabanlı sorgu yapmayı ve hatta sorgu kelimesi ile ilgili “başka hangi cümlecikler” kullanılabileceğini görerek sorgularını şekillendirmektedir.

Öneri sistemleri, kullanıcı bağlamını ve kullanıcı davranışını izleyerek kullanıcılara yeni önerilerde bulunan sistemlerdir. Kullanıcının sorgu kelimelerine göre yerel reklamları gösteren Google Adwords [1], kullanıcıların daha önce dinlediği şarkılara göre yeni şarkı öneren Spotify [2] örnek olarak gösterilebilir. Bu sistemler, kullanıcı geçmiş zamanda en çok ne dinlenmiş, kimleri izlemeye almış kimlerle arkadaş vb. gibi kullanıcı bilgilerinden, kullanıcı eğilimlerini ortaya çıkarmakta ve kullanıcılara yeni öneriler sunmaktadır.

3 HAVELSAN Geliştirme Ortamı

HAVELSAN kurumsal uygulamalarından biri olan HVL-GO, kuruma tümleşik uygulama yaşam döngüsü yönetimi sunarak yazılım geliştirme sürecinin otomasyonunu sağlamaktadır.

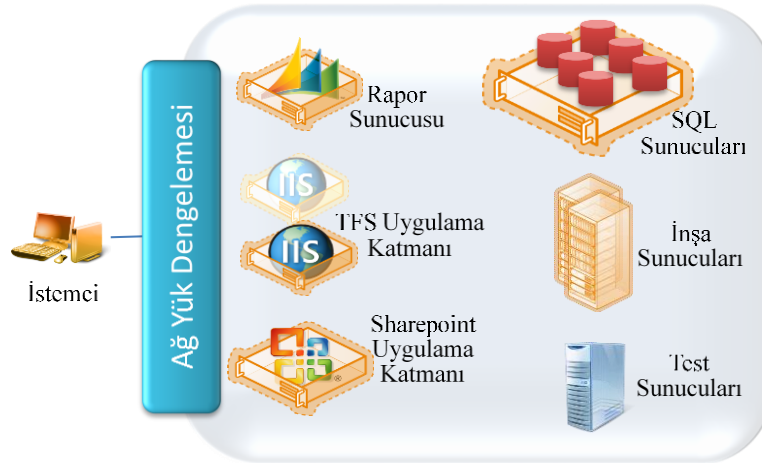
Uygulama yazılımlarının fikir aşamasından itibaren geliştirme, dağıtım ve bakım süreçlerinin tamamı Uygulama Yaşam Döngüsü Yönetimi (UYY) olarak tanımlanmaktadır [3]. Uygulama yaşam döngüsü (UYY), yazılım uygulamalarına ait bütünsel yaşam döngüsü bakışı ve geliştirme sürecinin koordinasyonunu sağlamak ve süreç çıktılarını yönetmek amacı ile ortaya çıkmıştır. UYY, yazılım çıktılarının yüksek kalitede ve hızlı şekilde teslim edilmesi için, yaşam döngüsü süreçlerinin otomasyonunu sağlayarak, yazılım geliştirme sürecine izlenebilirlik ve güçlü raporlama yeteneği kazandırmaktadır [4]. HAVELSAN Geliştirme Ortamı, kurum içi UYY bilincini arttırmak ve UYY otomasyonu sağlamak için geliştirilen kurumsal bir sistemdir.

3.1 HVL-GO Mimarisi

HVL-GO projelere gereksinim yönetimi, tasarım yönetimi, inşa ve yayın yönetimi, test yönetimi, değişiklik yönetimi, konfigürasyon yönetimi, görev yönetimi, proje izleme ve bilgi havuzu fonksiyonları sunmaktadır.

UYY fonksiyonları, HVL-GO kapsamında yer alan hazır ticari ürünlerle sağlanmaktadır. UYY altyapısı olarak MS Team Foundation Server [5] (TFS) kurumda yaygın olarak kullanılmaktadır. MS TFS, yazılım dünyasına iş ögesi yönetimi, inşa ve yayın yönetimi, test yönetimi, yazılım konfigürasyon yönetimi, proje izleme ve raporlama özellikleri sunmaktadır.

HVL-GO genel mimarisi Şekil 1’de gösterilmektedir.

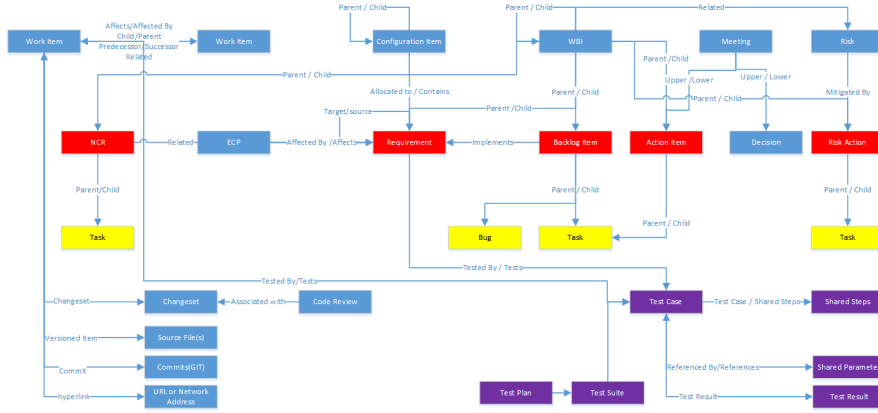


Şekil 1 HVL-GO Mimarisi

3.2 HVL-GO İş Ögesi Yönetimi

MS TFS'in kurumsal uygulamalara en büyük katkısı iş ögesi yönetimi ile gerçekleşmektedir. Nitekim UYY'nin görev yönetimi, gereksinim yönetimi ve değişiklik yönetimi özellikleri iş ögeleri ile sağlanmaktadır. MS TFS'in bir diğer katkısı, UYY sürecinin izlenebilirliğin sağlanmasıdır. İş ögelerinin ve diğer özelliklere ait nesnelerin birbirleri ile ilişkilendirilmesi ile bir proje/ürün kapsamında gereksinimden test sürecinin sonuna kadar tüm geliştirme süreci izlenebilmektedir.

HAVELSAN kapsamında UYY altyapısında yapılan ayarlamalarla elde edilen ilişki haritası **Şekil 2**'de gösterilmiştir.



Şekil 2 TFS İlişki Haritası

UYY mimarisinde gereksinimler, test durumları, hatalar, görevler gibi farklı ögeler "Work Item (İş Ögesi)" adındaki tek bir model olarak tutulmaktadır. **Şekil 2**'de bu yapı gösterilmiştir.

UYY altyapısındaki bu iş ögeleri hiyerarşik ve hiyerarşik olmayan çeşitli ilişki tipleriyle birbirlerine bağlanabilir. Bu ilişki ve iş ögesi tiplerinin çeşidi kullanıcı tarafından da artırılabilir. Gerekli olduğu durumlarda yeni iş ögesi tipi veya iş ögesi ilişki tipi yaratılabilir.

3.3 HVL-GO Yerleşik Arama Özellikleri

HVL-GO'da kayıtlı iş ögelerinin sorgulanması için MS TFS hızlı arama ve sorgu sihirbazı işlevleri sunmaktadır.

HVL-GO Web istemcisinde yer alan hızlı arama kutusu, iş ögelerinin tekil tanımlayıcı numara ile çağırılmalarına imkân sağlar. Açılan ekranda çağırılan iş ögesi ile ilgili düzenleme yapmak mümkündür. Anahtar kelime kullanılarak hızlı arama yapılması durumunda, anahtar kelimeler yapısal sorguya dönüştürülerek eşleşen iş ögeleri sorgu sonucu olarak gösterilir. Bu durumda kullanıcı her bir iş ögesini açmak sureti ile işlem yapabilir. İş ögesi numarasının bilinmesi veya iş ögesi tanımında yer alan "ayırt edici" kelimelerin hızlı aramada kullanılması doğrudan erişimi mümkün hale getirmektedir.

İş ögesi numarasını bilmek her zaman mümkün olmayabilir. Doğru anahtar kelimelerin seçilmemesi durumunda da arama binlerce sonuç dönebilir. Bu durumlar için daha detaylı bir arama fonksiyonuna ihtiyaç vardır.

HVL-GO Web istemcisi, sorgu sihirbazı kullanılarak iş öğelerini tanımlayan alanlar üzerinden detaylı/yapısal arama yapılmasına imkân sağlamaktadır. Bu alanlar iş öğelerine ait ortak alanlar olabildiği gibi uyarılma ile eklenmiş özel alanlar da olabilir. Sorgu, alt sorgu ile desteklendiğinde iki seviyeli veya ağaç görünümünde sorgu sonuçları elde etmek mümkündür. Sorguda kullanılan alanlara ve sorgu bağlaçlarının kullanımına bağlı olarak sorgu sonucunda onlarca iş ögesi dönüyor olabilir.

Yerleşik aramanın sezgisel arama veya iş ögesi önerisi kabiliyeti bulunmamaktadır. Kullanıcı deneyiminin artırılması için iş ögesi öneri sistemi geliştirilerek HVL-GO Web arayüzüne eklenmiştir.

4 İş Ögesi Öneri Sistemi Geliştirilmesi

UYY araçlarının gücü, sahip oldukları iş öğelerinde ve bu öğeler arasında kurulan izlenebilirlik ilişkilerinden gelmektedir [6]. HVL-GO sistemi yaygınlaşma sürecinde kullanıcılar, iş öğeleri arasındaki ilişkinin daha kolay sağlanabilmesini istemişlerdir. HVL-GO'daki yerleşik arama özelliği belirli ölçüde bunu sağlamakla birlikte, kullanıcı deneyimi ve sorgu sonuçlarının yorumlanması açısından zorluklar barındırmaktadır. Arama ile iş öğelerinin bulunması ve ilişkilendirilmesi kullanıcılara çok zaman kaybettirmektedir.

UYY fonksiyonlarının iyileştirilmesi [7] kapsamında UYY altyapısına uzantı olarak geliştirilen yeni arama özelliği ile kullanıcılara hem iş ögesi öneri sistemi hem de ileriki aşamalarda sezgisel aramaya dönüşebilecek tam metin arama altyapısı sağlanmıştır.

Bu bölümün devamında iş ögesi öneri sistemi geliştirme detayları yer almaktadır.

4.1 Sık Kullanılan İş Ögesi Alanları

Her iş ögesi kendi içerisinde farklı alanlardan ve bu alanlar üzerindeki yetkiye bağlı operasyonlardan oluşur. Bu alanların hepsi bir ortak havuzda bulunur. Her iş ögesine gereken alanlar bu havuzdan seçilerek eklenir. Bir alan birden çok iş ögesinde bulunabilir. Tıpkı iş öğeleri ve iş ögesi ilişkileri gibi yeni alanlarda kullanıcılar tarafından tanımlanabilir. Bu alanlardan en sık kullanılanlar **Tablo 1**'de gösterilmiştir.

Tablo 1 Sık Kullanılan İş Ögesi Alanları

Referans Adı	Alan Adı	Veri Tipi
System.Id	İş ögesi kimlik numarası	Integer
System.Title	İş ögesi başlığı	String
System.WorkItemType	İş ögesi tipi	String
System.Description	İş ögesi açıklaması	HTML
System.AreaPath	İş ögesi alanı	String
System.TeamProject	İş ögesi takımı	String
...

MS TFS kendi üzerinde kayıtlı olan iş öğelerini bu alanlar yardımı ile sorgulamaya yarayan WIQL (Work Item Query Language) adında SQL benzeri bir dil kullanır. WIQL ile SQL üzerinde tanımlı olan pek çok metod kullanılabilir böylece iş öğeleri ve gerekirse ilişkileri bu sayede sorgulanabilir.

İş öğelerini oluşturan tüm alanların SQL üzerinde tanımlı bir veri tipi vardır. Sorgulama yapılırken bu veri tiplerine göre filtreleme yapılır ve büyüklük, küçüklük vb. operatörler kullanılır. En sık kullanılan alanların veri tipleri **Tablo 1**'de gösterilmiştir.

Bu makalede geliştirilen uygulamada MS TFS üzerindeki iş öğesi alanlarından İş öğesi kimlik numarası, İş öğesi başlığı ve İş öğesi açıklaması alanları kullanılmıştır.

- **İş Öğesi Kimlik Numarası:** İş öğesi her yaratıldığında sistem tarafından verilen ayırıcı numaradır. 1'den başlar ve her yeni iş öğesi kaydedildiğinde değeri bir artar.
- **İş Öğesi Başlığı:** İş öğesi içeriğinin en net olarak belirtildiği alandır. En fazla 256 karakter uzunluğunda veri girilebilir. Tüm iş öğelerinde bulunur.
- **İş Öğesi Açıklaması:** İş öğesi içeriği ile ilgili başlık kısmına yazılamayacak kadar uzun olan veya detay bilgileri içeren alandır. Veri tipi HTML (zengin metin) olduğu için içeriğine her türlü resim, link bilgisi, font ve renk içeren yazı eklenebilir. Her iş öğesi tipinde bulunur.

Bu çalışmada iş öğesi öneri sistemi geliştirilirken iş öğesi başlığı ve iş öğesi açıklama alanları tercih edilmiştir. Çünkü bir iş öğesinin içeriğinin ne olduğunu ve içeriği ile ilgili detayları belirten alanlar bu alanlardır. İş öğesi kimlik numarası ise ayırıcı birer değer olması amacıyla kullanılmıştır.

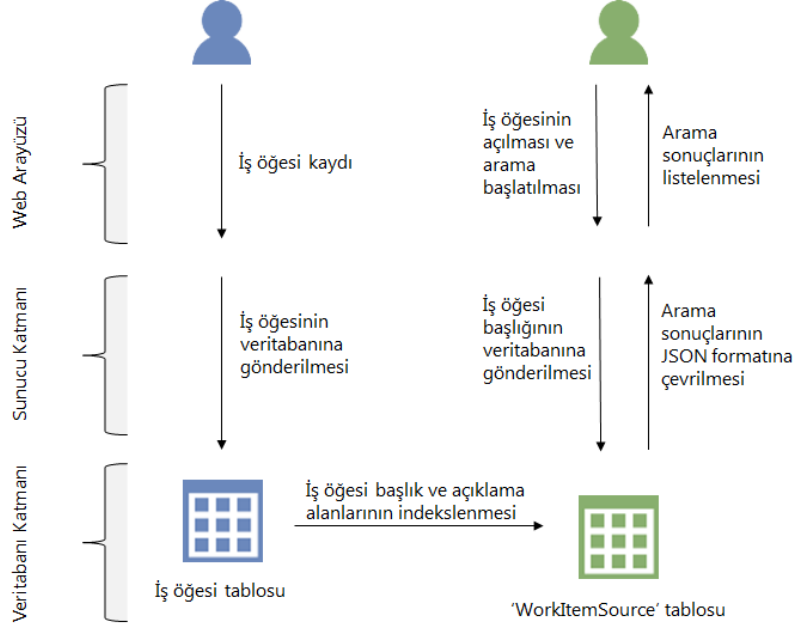
4.2 TFS Uyarlamaları

İş öğesi öneri sistemi MS TFS arayüzüne ek bir bileşen olarak geliştirilmiştir. MS TFS iş öğelerinin kullanıcılar tarafından olabildiğince esnek bir şekilde modifiye edilmesine olanak sağlamaktadır. Kullanıcılar tarafından geliştirilen uzantılar (extension) iş öğesi ekranında istenen yere eklenebilir. İş öğesi üzerindeki bütün alanlar, rollere tanımlanan yetkiler ve geliştirilen uzantılar XML formatında tutulur. İş öğesi puanlama uygulamasına ait XML içeriği şöyledir.

```
<Control FieldName="HVL.ALM.WorkItemSuggestor" Type="Control"
Label="Work Item Suggestor" LabelPosition="Left" />
```

Uygulamanın web arayüzünün geliştirilmesinde TypeScript [11] programlama dili kullanılmıştır. TypeScript, derlendiği zaman çıktı olarak JavaScript kodu veren ve JavaScript kodunu C programlama dili benzeri bir yaklaşımla yazmaya yarayan bir programlama dilidir.

Uygulamanın genel akışı **Şekil 3**'de gösterilmektedir. Kullanıcı yeni bir iş öğesi kaydettiğinde iş öğesi tablosuna tüm alanlar kaydedilir. Bu sırada bir tetikleyici (trigger) yardımı ile iş öğesinin kimlik numarası (ID), başlık ve açıklama alanları "WorkItemSource" tablosuna kaydedilir ve indekslenir. Kullanıcı, kaydedilen iş öğesi formunu açtığı anda ilgili iş öğesinin başlık bilgisi veritabanına gönderilerek öneri iş öğeleri için arama yapılır. Arama sonuçları JSON formatına çevrilerek iş öğesi formu üzerinde listelenir.



Şekil 3 Uygulama Akışı

4.3 SQL Server Uyarlamaları

MS TFS üzerinde tutulan proje ait verilerin incelenmesinden sonra iş ögesinin kimlik numarası, başlık ve açıklama bilgileri "WorkItemSource" adlı ayrı bir veritabanı tablosu yaratılarak bu tabloya aktarılmış ve indekslenmiştir. Ayrıca her yeni iş ögesi yaratıldığında bu iş ögesinin kimlik numarası, başlığı ve açıklaması da bir tetikleyici (trigger) yardımıyla "WorkItemSource" adlı tabloya aktarılmıştır.

4.3.1 Veri Tablosu

İndeksleme yapmak için oluşturulan "WorkItemSource" tablosunun şeması aşağıda gösterilmiştir.

```
CREATE TABLE [dbo].[hvl_WorkItemTitle] (
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [WorkItemId] [int] NOT NULL,
    [Title] [nchar](256) NOT NULL,
    [Description] [nvarchar](max) NOT NULL,
    [TitleDescription] [nvarchar](max) NOT NULL,
    CONSTRAINT [PK_hvl_WorkItemTitle] PRIMARY KEY CLUSTERED (
        [Id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
    ON, FILLFACTOR = 80) ON [PRIMARY]
```

) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

Kod Listesi 1 WorkItemSource Tablosu Şeması

Tabloya iş ögesinin kimlik numarası, başlık bilgisi ve açıklama alanları kaydedilmiştir. Son olarak da başlık ve açıklama alanları aralarına birer boşluk karakteri konarak birleştirilmiş ve tek bir kolon olarak kaydedilmiştir. Bunun sebebi “WorkItemSource” tablosu üzerinde çalışacak olan “Work item ranking” uygulamasının başlık ve açıklama alanlarında 0 ile 1 arasında değişen ağırlık oranlarına göre arama yapabilmesini sağlamaktır. Arama algoritması ile ilgili detaylı bilgiler 4.4 bölümde yer almaktadır.

4.3.2 Etkisiz Kelime Listesi

Etkisiz kelimeler (stopwords), cümle içinde bir anlam taşıyan ancak kullanım frekansları göz önüne alındığında tam metin aramaya değer katmayan kelimelerdir. Bu kelimeler indeksleme işlemi sırasında göz ardı edilirler. Bu çalışmada [8]’daki etkisiz kelime listesi kullanılmıştır.

4.4 Tam Metin Arama

Tam metin arama için MS SQL Server ile yerleşik olarak gelen tam metin arama hizmetinden yararlanılmıştır [9].

İş ögesi araması yapılırken ve sonuçlar puanlanırken veritabanı seviyesinde uyarlanmış olan bir algoritma kullanır. Veritabanı seviyesinde indekslenmiş olan “Title”, “Description” ve “TitleDescription” sütunları üzerinde tam metin arama yardımı ile hızlı bir şekilde arama ve sonuçları olabilecek en yakın öğeden en uzak öğeye doğru sıralayarak listelemek mümkündür.

Veritabanı üzerinde tam metin arama mimarisi MS SQL Server 2008 öncesinde bağımsız olarak çalışmaktaydı fakat sonraki sürümlerde bu yapı veritabanı mimarisi ile birleştirilmiş ve “sql full-text filter daemon launcher” adı altında hizmet vermeye devam etmiştir. Tam metin arama mimarisinde kullanılan veri tipleri, Char, Varchar, Nchar, Nvarchar, Text, Ntext, Image, Xml, varbinary(max), FileStream olarak sıralanmaktadır.

Tam metin arama mimarisi üç ana adımda oluşturulur. Bunlar:

- Tam metin kataloğu oluşturma
- Tam metin indeks oluşturma
- Tam metin indeks içeriğinin oluşturulması

Yukarıdaki üç aşamayı da başarılı bir şekilde bitirdikten sonra ilgili veritabanı tablosunun ilgili kolon veya kolonlarında tam metin arama ile arama yapmak mümkündür. Tam metin arama üzerinde dört farklı SQL metodu kullanmak mümkündür: Contains, ContainsTable, Freetext, FreetextTable

Bu dört metottan “Contains” ve “Freetext” metotları sadece ilgili algoritmaya göre arama sonuçlarını döndürür. Fakat “ContainsTable” ve “FreetextTable” metotları ilgili anahtar kelimeye göre dönen her arama sonucuna dayandığı algoritmaya göre bir sıralama puanı (rank) verir. Arama sonuçları puan bilgisine göre sıralayarak aramaya en

uygun sonuçların başlarda listelenmesini sağlamış olur. Bu makalede “FreetextTable” metodu kullanılmıştır. Çünkü bu metoda göre bir kelime aranırken bu kelimenin aynı-sının geçtiği sonuçlarla birlikte kelimenin çoğul vb. takılarının düşmüş yalın halinin geçtiği sonuçlarda gelecektir.

FreetextTable metot tanımı [9] aşağıda gösterilmiştir:

```
FREETEXTTABLE (table , { column_name | (column_list) | * }
               , 'freetext_string'
               [ , LANGUAGE language_term ]
               [ , top_n_by_rank ] )
```

Kod Listesi 2 Freetexttable Kullanım Deseni

Metot tanımındaki yer tutucuların tanımı aşağıda yer almaktadır:

- **table:** freetexttable metodu ile arama yapılacak olan veritabanı tablosudur.
- **column_name:** SQL cümleciğindeki FROM tümcesinden sonra kullanılacak olan en az bir veya birden çok kolonun tanımlandığı değişkendir.
- **column_list:** Birden çok kolon üzerinde freetexttable metodu ile arama yapılacağı zaman kullanılan değişkendir. Kolon isimleri aralarına virgül konarak ve parantez ile açılıp kapatılarak bu değişkenle tanımlanır. Eğer özel bir dil değişkeni tanımlanmamış ise tüm kolonlar için aynı ve varsayılan dil değişkeni geçerli kabul edilir. Kolon tipleri char, nchar, varchar, text, ntext, image, xml, varbinary veya varbinary(max) olabilir.
- **freetext_string:** İndekslenmiş kolonlarda aranacak olan kelime veya kelimelerin tanımlandığı değişkendir. Bütün bir cümle ile de tek seferde arama yapılabilir. Arama sonunda, aramanın yapıldığı bütün cümleyle eşleşen ya da cümle içindeki bir kelime ile eşleşen sonuçlar hesaplanan puan ile birlikte geri döner. Freetexttable metodunda contains metodunun aksine engellenecek kelimeler listesi dikkate alınır. Örneğin tam metin arama için tanımlanan dil İngilizce ise “and” vb. gibi kelimeler etkisiz kelimeler listesinde görülür ve aramaya dâhil edilmez.
- **language_term:** İndekslenmiş verinin üzerinde arama yaparken dikkat edilmesi gereken dil bilgisi kuralları, harici kelimeler listesi vb. Özelliklerin tanımlanması için kullanılan değişkendir. Bu dil değişkeni sayısal, string veya hexadecimal olarak tanımlanabilir.
- **top_n_by_rank:** freetexttable metodu ile dönen arama sonuçlarında her satırın bir rank değeri bulunur. Bu sonuçlardan belli bir puandan büyük olanlar filtrelenmek isteniyorsa “n” adlı sayısal (integer) değişken kullanılarak bu filtreleme yapılır.

Tam metin arama MS SQL Server’a eklenen alt yordam (stored procedure) ile sağlanmaktadır. Freetexttable’in alt yordamda kullanımını aşağıdaki gibidir:

```
...
SELECT WorkItemId, Title, Description, FT.[rank]*@titleWeight
FROM WorkItemSource as WS inner join freetexttable(WorkItemSource, Title, @keyword) as FT on WS.WorkItemId=FT.[key]
Union all
```

```

SELECT WorkItemId, Title, Description,
FT.[rank]*@descriptionWeight FROM WorkItemSource as WS inner
join freetexttable(WorkItemSource, Description, @keyword) as FT
on WS.WorkItemId=FT.[key]
...

```

Kod Listesi 3 Arama Altyordamı

SQL kod parçasında kullanılan parametreler şu şekildedir:

- @keyword: Aranan metin.
- @titleWeight: Başlık bilgisi için verilen ağırlık değeridir. Uygulamada 0,67 olarak kullanılmaktadır.
- @descriptionWeight: Açıklama bilgisi için verilen ağırlık değeridir. Uygulamada 0,33 olarak kullanılmaktadır.

MS SQL Server tam metin aramada OKAPI BM25 algoritmasını kullanmaktadır. Bu algoritma ile ilgili bilgiler aşağıdaki bölümde yer almaktadır.

4.5 OKAPI BM25

Freetexttable metodu, arama sonuçlarının aranan anahtar kelime veya kelimelerle olan yakınlık ilişkisini bulabilmek için OKAPI BM25 puanlama algoritmasını kullanır [10]. OKAPI BM25 arama motorları tarafından anahtar kelimelerle aranan dokümanların arasındaki ilgi derecesini belirli bir aralıktaki sayısal değerlere oturtmak için tasarlanmıştır. Algoritmanın temeli 1970 ve 1980'lerde Stephen E. Robertson, Karen Spärck Jones ve diğerleri tarafından geliştirilen olasılıksal ilgi modeli (Probabilistic relevance model) üzerine dayanır. OKAPI BM25 algoritmasının matematiksel formülü (1) ve (2)'de gösterildiği gibidir.

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}, \quad (1)$$

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}, \quad (2)$$

OKAPI BM25 algoritmasındaki değişkenler aşağıdaki gibi tanımlanmaktadır:

- **Q**: Arama sorgusu
- **q₁, q₂, ..., q_n**: Arama sorgusu içerisindeki anahtar kelimeler
- **D**: Doküman
- **|D|**: Dokümanın kelime sayısı bakımından uzunluğu
- **f(q_i, D)**: Anahtar kelimenin (q_i) doküman içinde geçme sıklığı
- **k₁, b**: Optimizasyon sırasında gerektiğinde değeri değiştirilebilen serbest parametreler. (1,2 <= k₁ <= 2,0 ve b = 0,75)

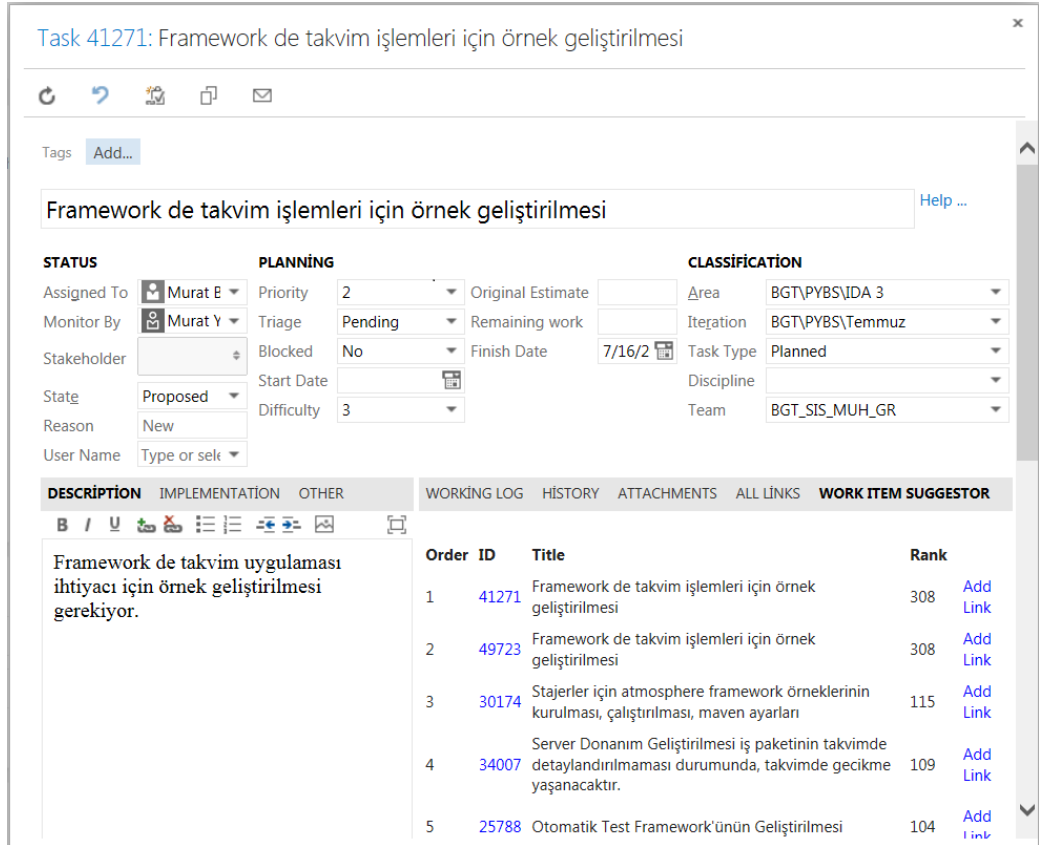
- **avgdl**: Dokümanların ortalama uzunluğu
- **N**: Kaynaktaki toplam doküman sayısı
- **n (q_i)**: q_i anahtar kelimesinin geçtiği toplam doküman sayısı

OKAPI BM25 algoritması ve freetexttable metodu iş ögesi öneri sisteminin temelini oluşturmaktadır.

4.6 İş Ögesi Öneri Sistemi Uygulaması

İş ögesi öneri sistemi HVL-GO üzerindeki iş öğelerine uzantı olarak yazılmıştır. Geliştirilen uygulama, HVL-GO ile birlikte web tabanlı olarak çalışmaktadır. Bu uygulama sayesinde HVL-GO üzerinde bir iş ögesi yaratıldığında veya hazır kayıtlı bir iş ögesi açıldığında ekran üzerindeki “İş Ögesi Önerici” alanında bu iş ögesiyle ilgili olabilecek diğer iş öğeleri başlık ve açıklama bilgisine göre aranır ve sıralama puanı (rank) değerlerine göre sıralanarak listelenir. Bu listeden istenen iş ögesi de üzerine tıklanarak mevcut iş ögesi ile ilişki kurulmuş olur.

Geliştirilen öneri sisteminin örnek ekran görüntüsü Şekil 4’de gösterilmektedir.



Task 41271: Framework de takvim işlemleri için örnek geliştirilmesi

Tags: Add...

Framework de takvim işlemleri için örnek geliştirilmesi [Help ...](#)

STATUS		PLANNING			CLASSIFICATION		
Assigned To	Murat E	Priority	2	Original Estimate		Area	BGT\PYBS\IDA 3
Monitor By	Murat Y	Triage	Pending	Remaining work		Iteration	BGT\PYBS\Temmuz
Stakeholder		Blocked	No	Finish Date	7/16/2	Task Type	Planned
State	Proposed	Start Date		Discipline		Team	BGT_SIS_MUH_GR
Reason	New	Difficulty	3				
User Name	Type or select						

DESCRIPTION	IMPLEMENTATION	OTHER	WORKING LOG	HISTORY	ATTACHMENTS	ALL LINKS	WORK ITEM SUGGESTOR
Framework de takvim uygulaması ihtiyacı için örnek geliştirilmesi gerekiyor.							
Order	ID	Title	Rank				
1	41271	Framework de takvim işlemleri için örnek geliştirilmesi	308				Add Link
2	49723	Framework de takvim işlemleri için örnek geliştirilmesi	308				Add Link
3	30174	Stajyerler için atmosphere framework örneklerinin kurulması, çalıştırılması, maven ayarları	115				Add Link
4	34007	Server Donanım Geliştirilmesi iş paketinin takvimde detaylandırılmaması durumunda, takvimde gecikme yaşanacaktır.	109				Add Link
5	25788	Otomatik Test Framework'ünün Geliştirilmesi	104				Add Link

Şekil 4 İş Ögesi Öneri Sistemi Ekran Görüntüsü

5 Değerlendirme ve Devam Eden Çalışmalar

Kurumlarda artan veri hacmi, kurum bilgi sistemleri genelinde yapılacak kurumsal ara- maları zorunlu hale getirmektedir. Ancak artan veri hacmi ve kullanıcı beklentileri ara- maların sezgisel olmasını, kullanıcıyı yönlendirmesini gerektirmektedir. Arama sistem- leri bunun da bir adım ötesine geçerek öneri sistemleri sunmaktadır.

Bu bildiriye, uygulama yaşam döngüsü yönetimi altyapısı sağlayan HVL-GO ku- rumsal uygulamasına eklenen iş öğelerinin öneri sistemi geliştirilmesi adımları anlatıl- mıştır. İş öğesi önerisi, HVL-GO mimarisinde yer alan MS TFS uzantısı olarak gelişti- rilmiş ve MS TFS ve MS SQL Server uyarlamaları ile gerçekleştirilmiştir. Kullanıcıla- rın yapılan geliştirmeye ilk yaklaşımları öneri sisteminin ve iş öğesi ilişkilendirmenin kolaylaştığı yönündedir.

Çalışmanın bundan sonraki aşamasında sezgisel arama altyapısının tamamlanarak devreye alınması ve yeni arama işlevlerinin kullanıcı deneyimine ve etkinliğine katkı- sının ölçülmesi planlanmıştır.

Kaynaklar

1. Google Adwords Reklam Hizmeti, <https://www.google.com/adwords/>
2. Spotify Müzik Hizmeti, <http://www.spotify.com/>
3. Chappell, D. "What is Application Lifecycle Management", Chappell & Associates 2008
4. Macit, Y., Tüzün E., İnce K., Aytekin A. İ: Büyük Ölçekli Bir Organizasyonda Uygulama Yaşam Döngüsü Yönetimi Uygulama Deneyimi, 8. Ulusal Yazılım Mühendisliği Sempozyumu (2014)
5. MS Team Foundation Server Ürün Sayfası, <https://www.visualstudio.com/tr-tr/products/tfs-overview-vs.aspx>
6. Jane, C-H.; Olly, G.; Jane, H.H.; Patrick, M.; Andrea, Z. "Software traceability: trends and future directions", Proceedings of the on Future of Software
7. Klespitz, J., Biró, M., Kovács, L., "Aspects of improvement of software development li- fecycle management," Computational Intelligence and Informatics (CINTI), 2015 16th IEEE International Symposium on, Budapest, 2015, pp. 323-327.
8. Türkçe Etkisiz Kelimeler Listesi 1.1, [http://www.turkceogretimi.com/Genel-Konular/ar- ticle/541-turkce-etkisiz-kelimeler-stop-words-listesi-11/35](http://www.turkceogretimi.com/Genel-Konular/article/541-turkce-etkisiz-kelimeler-stop-words-listesi-11/35)
9. MS SQL Server Tam Metin Arama Kullanımı, [https://msdn.microsoft.com/en-us/lib- rary/ms177652\(v=sql.110\).aspx](https://msdn.microsoft.com/en-us/lib- rary/ms177652(v=sql.110).aspx)
10. OKAPI BM25 Wikipedia Sayfası, https://en.wikipedia.org/wiki/Okapi_BM25
11. Typescript Language, <https://www.typescriptlang.org/>