**BLM3110**

Bilgisayar Mühendisliğine Özel Konular

**Doç. Dr. Göksel BİRİCİK**

Proje Faz-1

Hazırlayanlar:

16011105-Buğra KARACA

15011111-Eray Cinci

16011107-Halid Çifci

15011034-Ünal Asil

16011100-Enes Keskin

17011606-Halid Beheşti Sipahioğlu

**Program Tanımı**

Bu program platform bağımsız bir uygulamadır. Girdi olarak kullanıcıdan bir text alır ve bu textin içinden anahtar kelimeler çıkarır. Ardından bu kelimelerden sıralı bir index çıktısı oluşturur.

**İşlevsel Gereksinimler**

Nesneye dayalı mimari ile gerçeklediğimiz bu programda Engine ve sentence classlarında oluşturduğumuz üye alanlarını ve metotları gui(main) classında engine ve sentence türünde oluşturduğumuz nesneler ile kullanılabilir hale getirdik.

Program çalıştığında açılan pencerede Url giriniz yazan kutuya url adresi, metin giriniz yazan kutuya indislenecek kelimelrin oluştuğu cümleler ya da kelimeler girildiğinde oluştur butonu vasıtasıyla sıralı dairesel kaydırılmış kelime versiyonları çıktı sekmesinde görülebilmektedir. url indislemeleri de url indisleri sekmesinde görülebilmektedir.

**Gui class metotları:**

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**try** {

Gui frame = **new** Gui();

frame.setSize(1500, 600);

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

}); }

Main metodu ile run metodu çağrılarak frame nesnesinin kullanıcı tarafından görülebilir hale gelmesi sağlanmıştır.

**public** Gui() {

setTitle("Metin Indeksleme");

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

JTabbedPane tappedPane = **new** JTabbedPane();

panel1 = **new** JPanel(**new** BorderLayout());

panel2= **new** JPanel();

panel3= **new** JPanel();

text = **new** JTextField("Url giriniz");

text2 = **new** JTextField("Metin giriniz");

olustur= **new** JButton("Olustur");

panel1.add(text,BorderLayout.***PAGE\_START***);

panel1.add(text2,BorderLayout.***CENTER***);

panel1.add(olustur,BorderLayout.***PAGE\_END***);

panel2.setLayout(**new** GridLayout(20,1));

panel3.setLayout(**new** GridLayout(20,1));

tappedPane.add(panel1,"Metin Girisi");

tappedPane.add(panel2,"Cıktı");

tappedPane.add(panel3,"URL Indisleme");

add(tappedPane);

olustur.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String ex2 = text.getText();

*ex*=text2.getText();

String [] parts=*ex*.split("\\.");

**for**(**int** i=0;i<parts.length;i++) {

String[] words = parts[i].split("\\s+");

**for** (**int** j = 0; j< words.length; j++) {

// You may want to check for a non-word character before blindly

// performing a replacement

// It may also be necessary to adjust the character class

words[j] = words[j].replaceAll("[^\\w]", " ");

}

Sentence wordiex = **new** Sentence(words);

*sentarray*[i]=wordiex;

*sentencecount*++;

}

Engine searchEngine = **new** Engine();

**for**(**int** i=0;i<*sentencecount*;i++) {

searchEngine.addSentence(*sentarray*[i]);

}

//searchEngine.listItems();

searchEngine.sort();

/\*Çıktı sayfasında for içinde print işlemini buraya yap\*/

**for**(**int** i=0;i<searchEngine.getCountSort();i++) {

JTextField text3 = **new** JTextField();

text3.setBackground(Color.***gray***);

text3.setText(searchEngine.listIndex(i));

panel2.add(text3);

panel2.revalidate();

panel2.repaint();

}

/\*Url sayfasında for içinde print işlemini buraya yap\*/

**for**(**int** i=0;i<searchEngine.getCountSort();i++) {JTextField text4 = **new** JTextField();

text4.setBackground(Color.***gray***);

text4.setText(ex2+"/"+searchEngine.listIndex(i));

panel3.add(text4);

panel3.revalidate();

panel3.repaint();

}

}});

}

}

Bu metodun içinde jframe elemanları(jpanel, jtappedpane, jtextfield, jbutton) oluşturulmuş ve metin girişleri kullanıcıdan alınmıştır. Oluştur butonuna actionlistener eklenmiş ve bu actionlistener içinde kullanıcıdan alınan cümleler engine clasından oluşturulan searchengine nesnesi ile sıralama ve yazdırma işlemleri yapılmıştır. Sentence classından oluşturulan wordiex nesnesi ile cümleler ve kelimelerin ayrıştırılması sağlanmış ve çıktı ve url indisleme sekmelerine çıktı olarak yazdırılması sağlanmıştır.

**Engine class metotları:**

**public** **void** sortArrayCreate() {

**for**(i=0;i<count;i++) {

**for**(j=0;j<sentenceArray[i].getArraylength();j++) {

sortArray[countsort]="";

sortArray[countsort]=sortArray[countsort].concat(sentenceArray[i].sentenceArrayVar[j]);

countsort++;

}

}

Sentence clasından oluşturulan sentencer nesnesinin elemanlarının sıralı olarak birbirine eklenmesi sağlanmıştır.

**public** **void** sort() {

sortArrayCreate();

**char** temp[]= **new** **char**[countsort];

**for**(i=0;i<countsort;i++) {

temp[i]=sortArray[i].charAt(1);

**int** x=temp[i];

**if**(x<97) {

x=x+32;

temp[i]=(**char**)x;

}

}

**int** n = countsort;

// One by one move boundary of unsorted subarray

**for** (i = 0; i < n-1; i++)

{

// Find the minimum element in unsorted array

**int** min\_idx = i;

**for** (j = i+1; j < n; j++) {

**int** s = temp[j];

**int** o = temp[min\_idx];

**if** (s< o) {

min\_idx = j;

}

}

// Swap the found minimum element with the first

// element

**char** temp2 = temp[min\_idx];

temp[min\_idx] =temp[i];

temp[i] = temp2;

String temp3 = sortArray[min\_idx];

sortArray[min\_idx]=sortArray[i];

sortArray[i]=temp3;

}

}

Bu metotta gerekli sort işlemi yapılmıştır. (Alfabetik artan sıra)

**public** **void** addSentence(Sentence sentencex) {

sentenceArray[count]=sentencex;

count++;

}

Kelime sayısını belirleyen metot.

**public** String listIndex(**int** i) {

**return** sortArray[i];

}

Sıralı cümleleri yazdıran metot.

**public** **int** getCountSort() {

**return** countsort;

}

Count üyealanının gui tarafından erişilmesini sağlayan metot.

**Sentence class metotları:**

**public** **int** getArraylength() {

**return** arraylength;

}

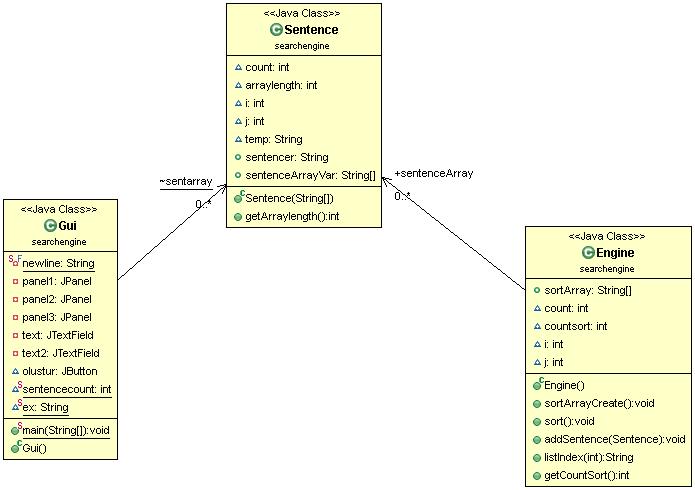
Arraylenght üye alanının return eden fonksiyon.

**İşlevsel Olmayan Gereksinimler**

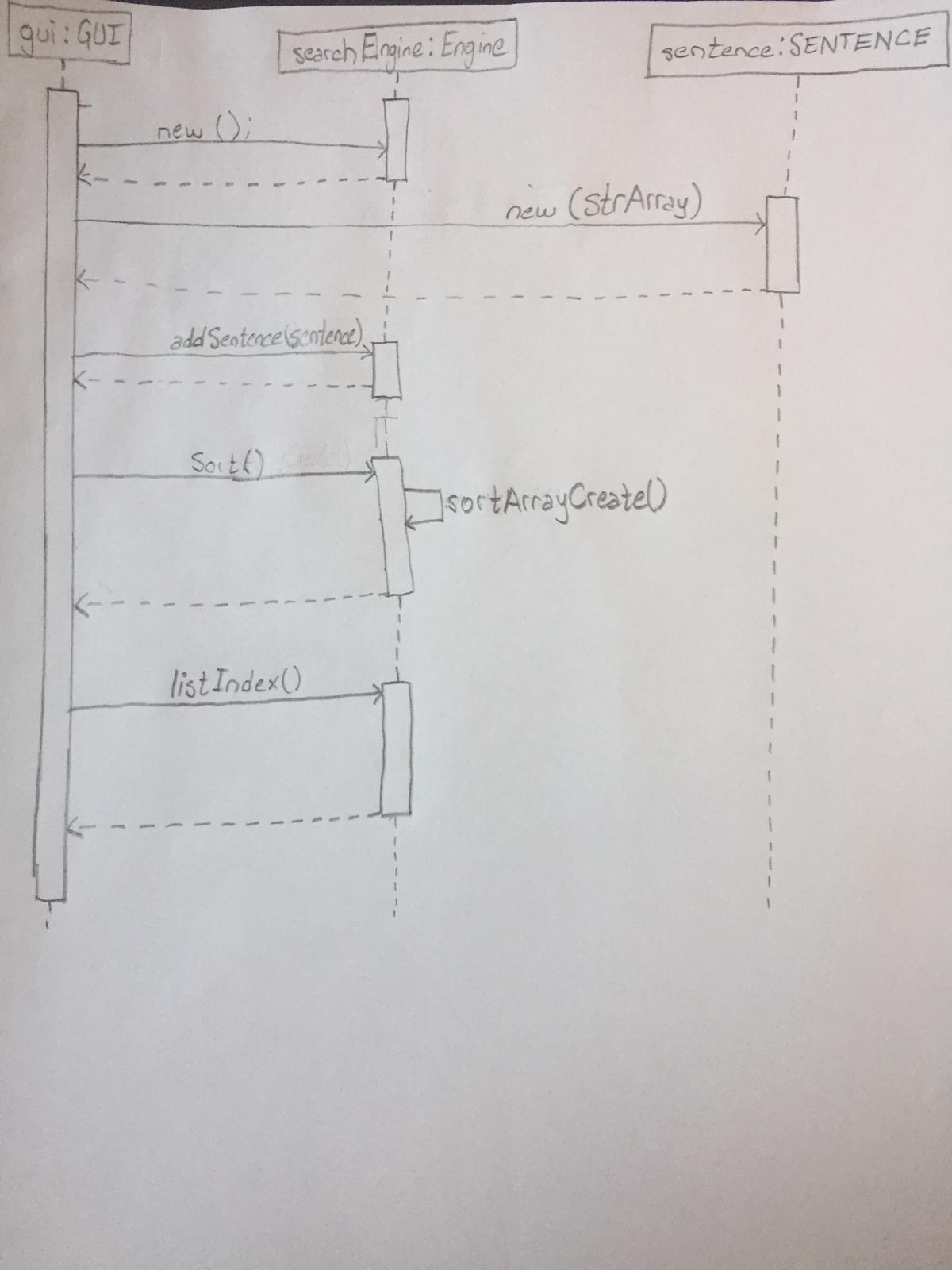
Hafızada kapladığı yer:8 Kb

Programın Çalışma Hızı:1ms

Uml Diagram:

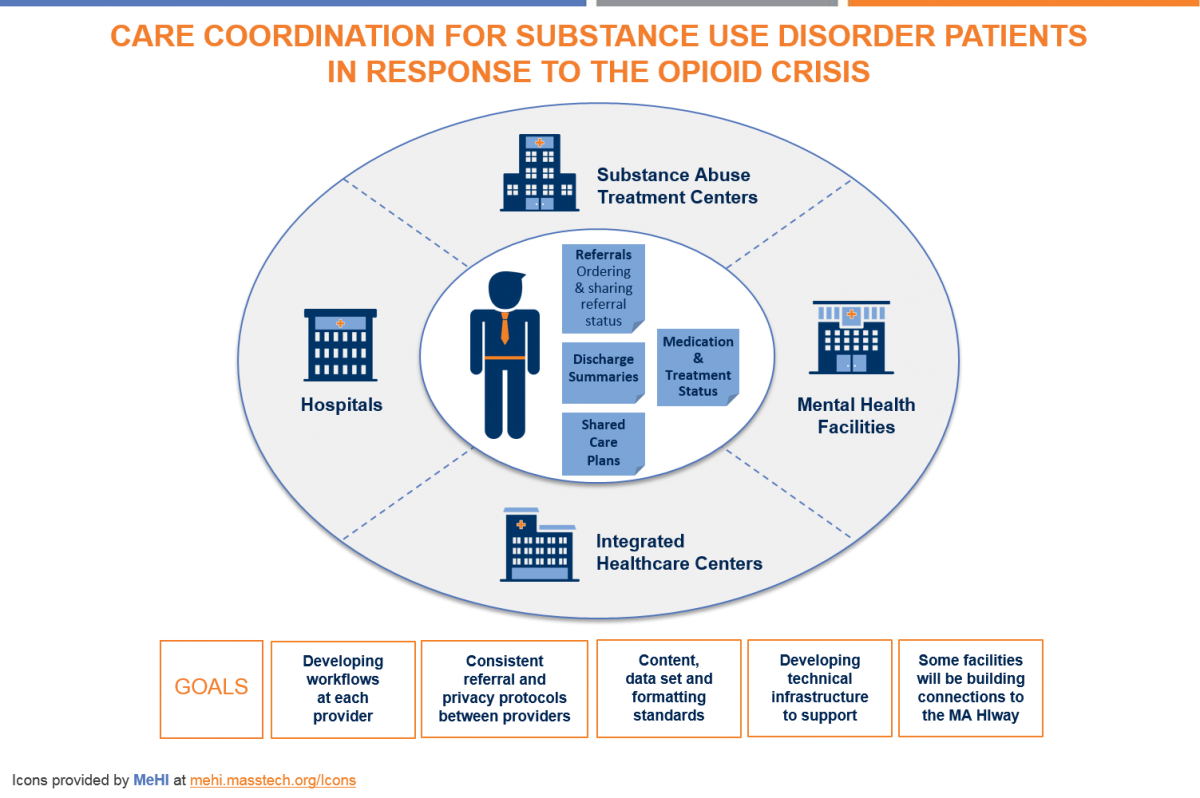
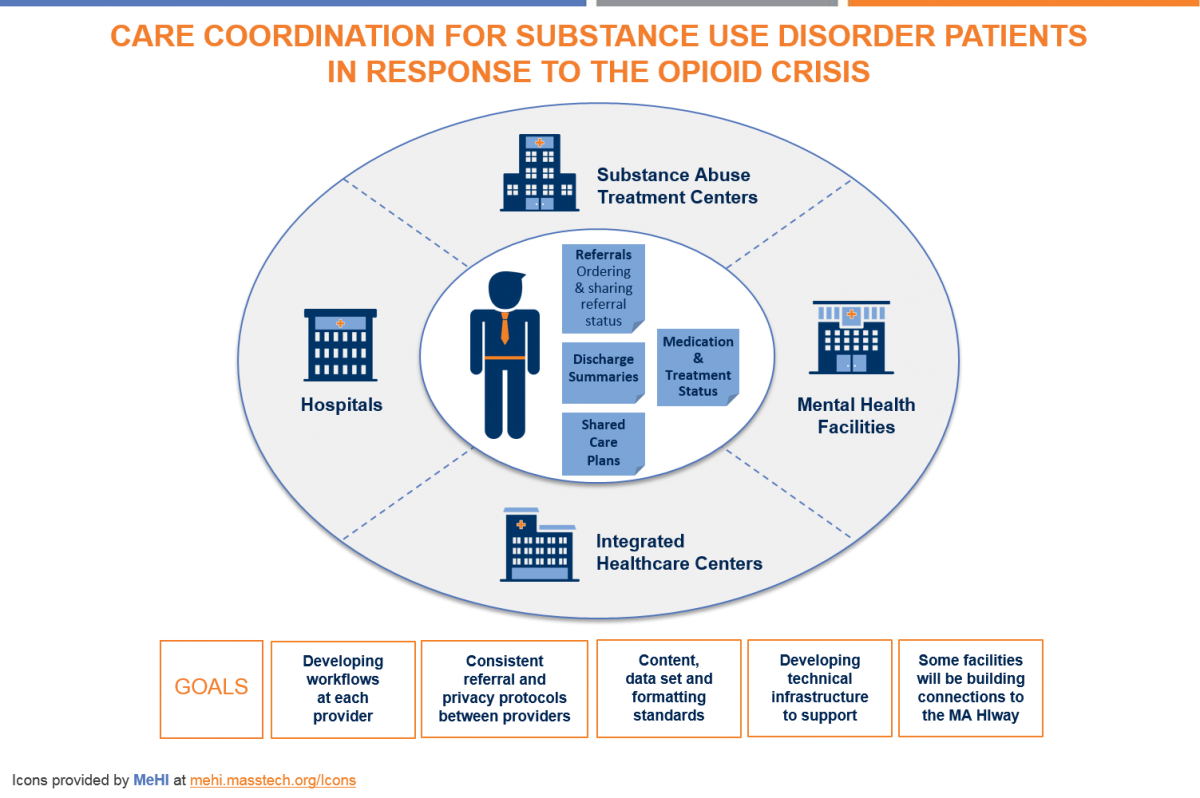


Sequence Diagram:

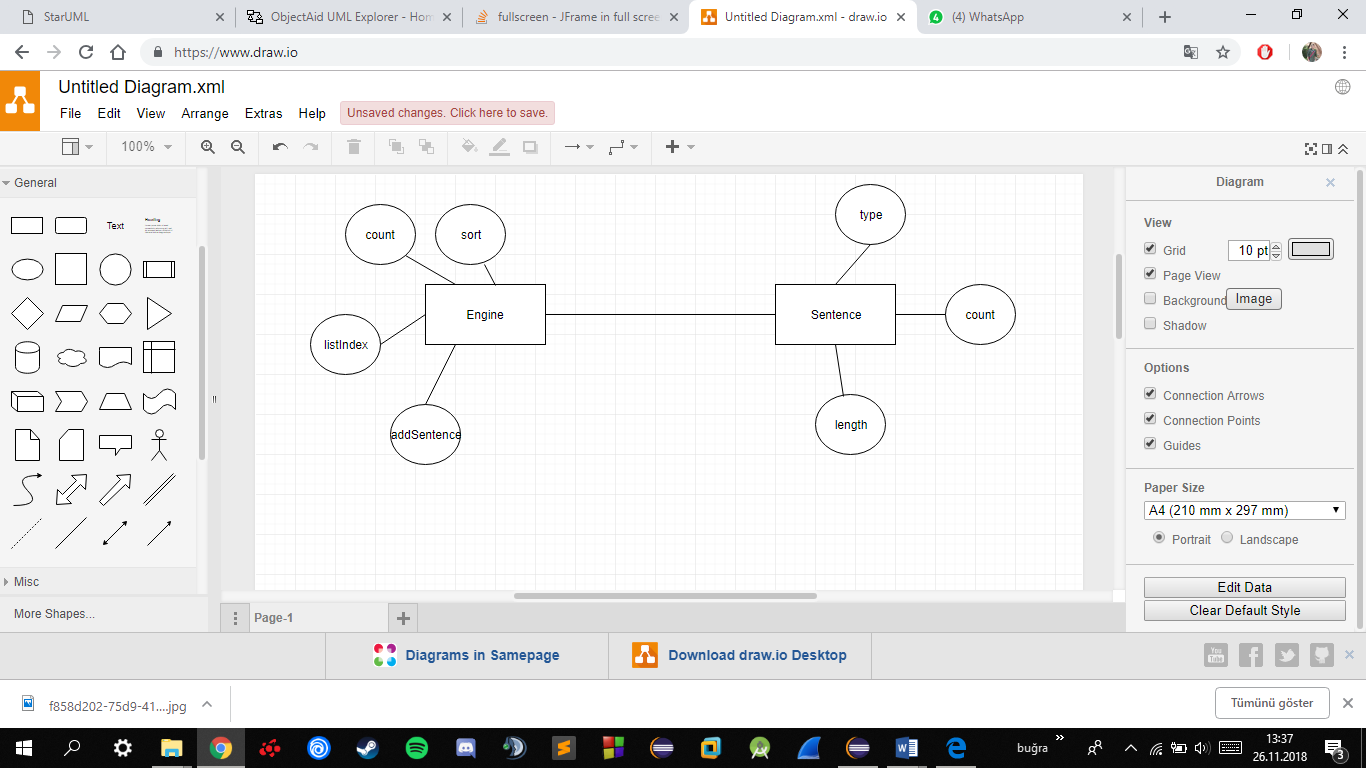


Use Case Diagram:

Kullanıcı

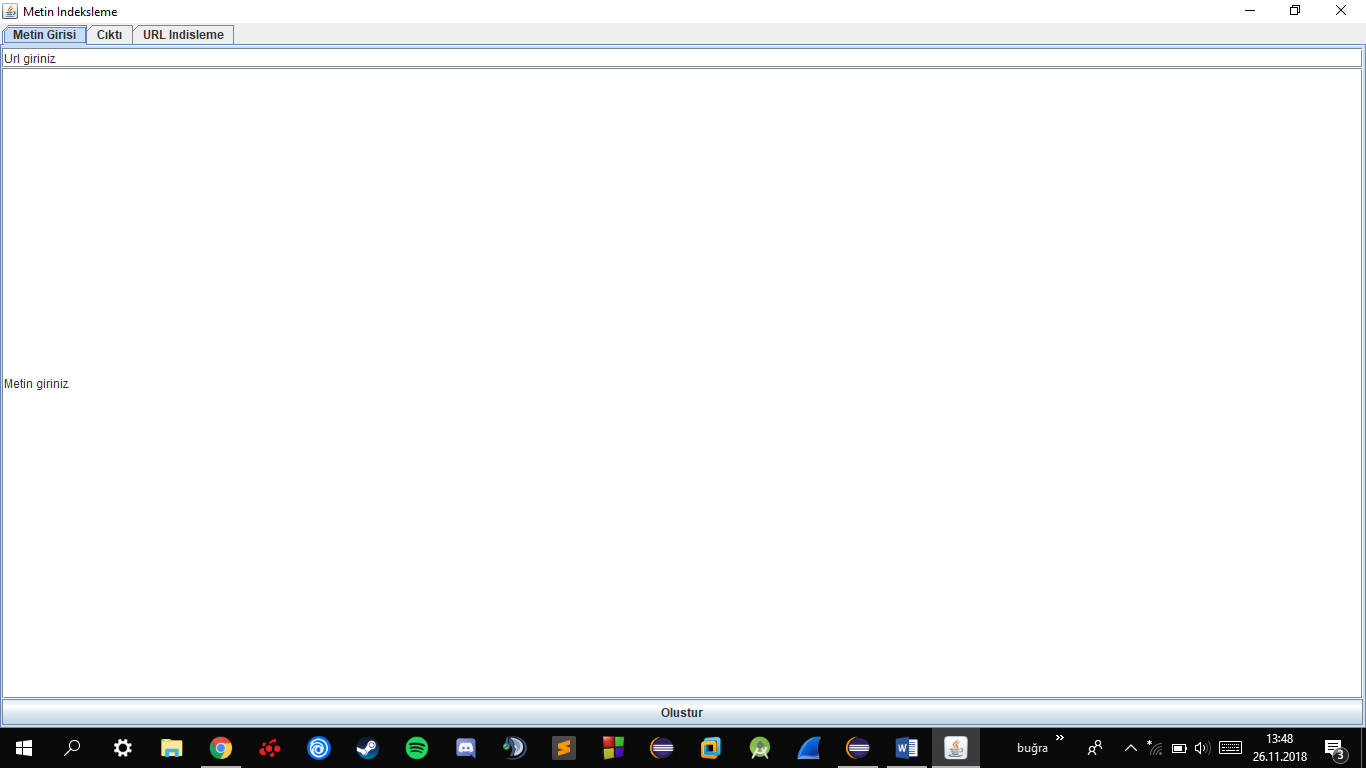


ER Diagram



# Kullanıcı El kitabı

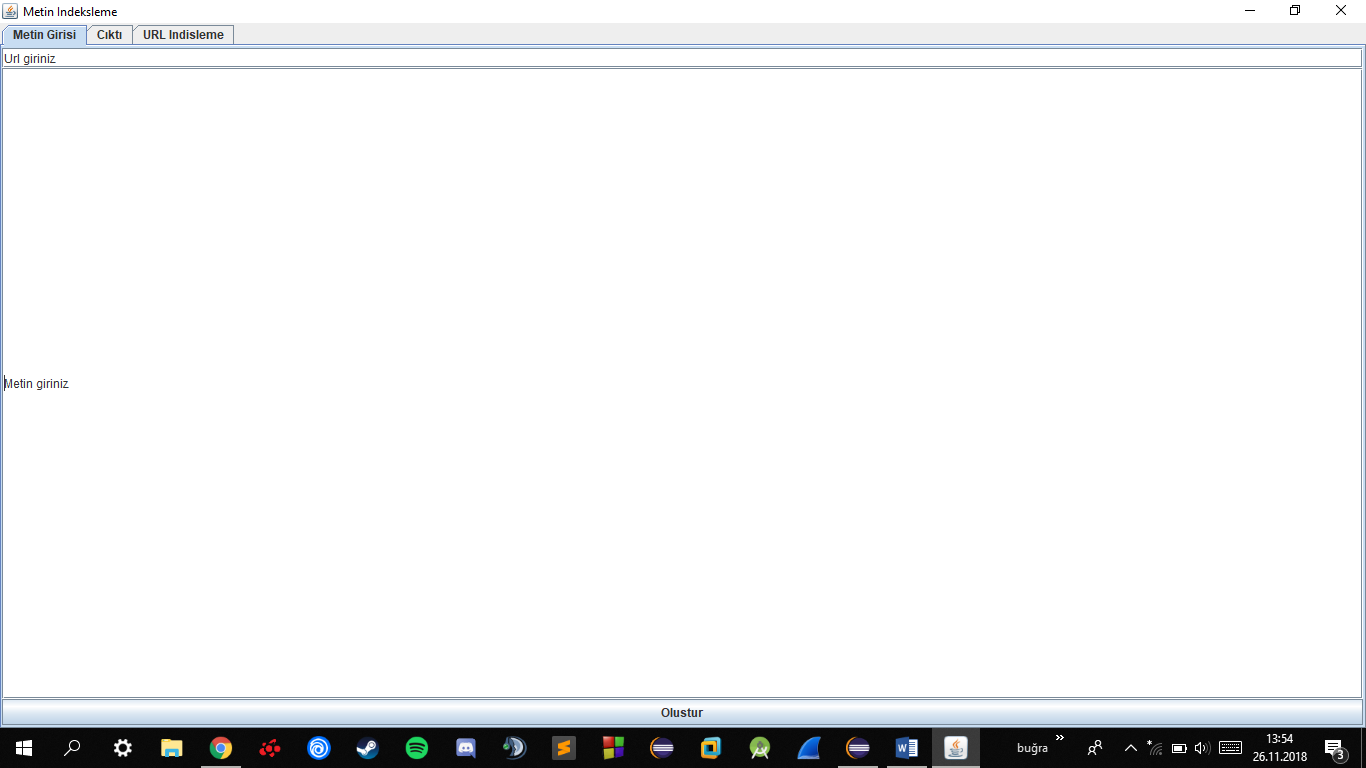
Sisteme runnable jar dosyasını çalıştırarak erişim sağlanır. Kullanıcıdan 2 adet input vermesi istenir. Bunlardan birincisi index adresi, ikincisi ise anahtar kelimelerinin çıkarılmasını istediği metindir. Bu işlemler sonunda “oluştur” butonuna basarak index listesinin oluşturulmasını sağlar. “Çıktı” sekmesinden alfabetik sıralı sonuçları görür. “Url indisleme” sekmesinde de bu indislerin url formatına uygun halini görür.Metin ve Url girişi öncesi kullanıcının Metin giriniz ve url giriniz yazılarını silmesi gerekmektedir.



2-Metin girişi

3-Oluştur Butonu

1-URL girişi



6-URL indisleme çıktıları sekmesi

5-Çıktı sekmesi

4-Metin girişi sekmesi