Keyword counter

Domaći 1

1 Pregled zadatka

Keyword counter sistem treba da podrži brojanje pojavljivanja ključnih reči koje su unapred zadate u korpusima različitog tipa. Ovo prebrojavanje treba da se obavlja konkurentno, uz mogućnost dodavanja novih korpusa kao i pregleda rezultata prebrojavanja za pojedinačne korpuse i sumiranje rezultata. Za potrebe domaćeg zadatka će biti neophodno implementirati rad nad ASCII kodiranim tekstualnim datotekama i HTML datotekama koje se nalaze na web-u. Ključne reči se u ovim korpusima broje samo kada stoje samostalno, ne i kada su deo drugih reči, i pretraga treba da bude case-sensitive. Sistem treba u potpunosti implementirati Java programskim jezikom.

Sistem je podeljen na nekoliko komponenti, s time da treba da bude moguće dodavati nove komponente u budućnosti relativno lako. Zahtevi za komponente i njihova međusobna komunikacija su dati u odeljku 2.

Sistem treba da graciozno razrešava problematične situacije pri radu. Bitno je da sistem informiše korisnika kada dođe do problema, kao i da nikada ne kolabira u potpunosti. Zahtevi vezani za razrešavanje grešaka u sistemu su dati u odeljku 3.

Korisnik interaguje sa sistemom preko komandne linije (CLI), unošenjem komandi. Pored toga, rad sistema će se podešavati preko konfiguracione datoteke. Opis ove datoteke, kao i spisak komandi, njihovih parametara, rezultata i mogućih grešaka je dat u odeljku 4. U ovom odeljku će biti dato i nekoliko primera korišćenja sistema.

Bodovanje zadatka, kao i instrukcije za predaju zadatka su dati u odeljku 5.

2 Opis sistema

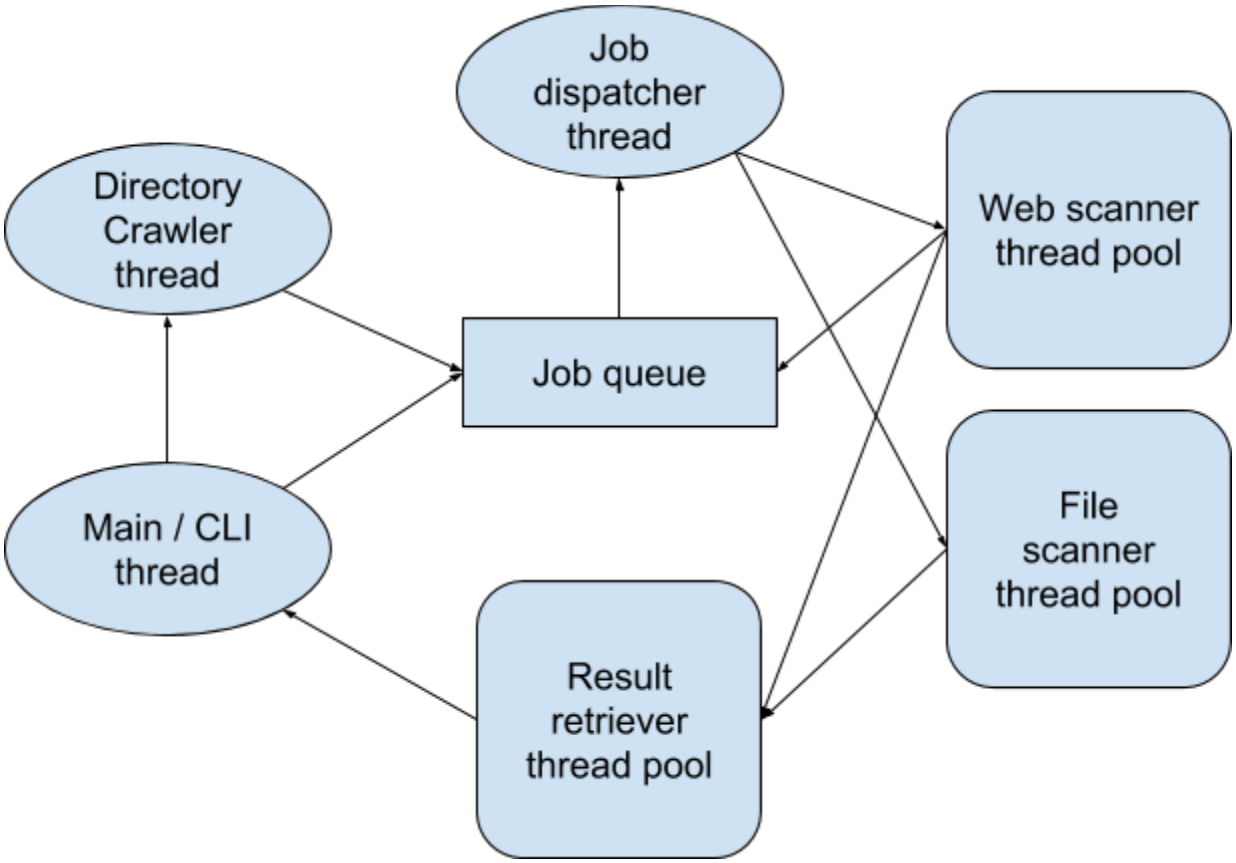
Sistem se sastoji iz tri komponente koje su zasnovane na thread pool-u, i nekoliko pomoćnih komponenti koje se izvršavaju svaka u svojoj niti. Svoj thread pool imaju:

- Web Scanner
- File Scanner
- Result Retriever

Dok se u pojedinačnim nitima izvršavaju:

- Main / CLI
- Job dispatcher
- Directory crawler

Pored ovih aktivnih komponenti, postoji i jedan deljeni red koji se koristi za zadavanje novih poslova i njihovo pokretanje. Sistem se grafički može predstaviti na sledeći način:



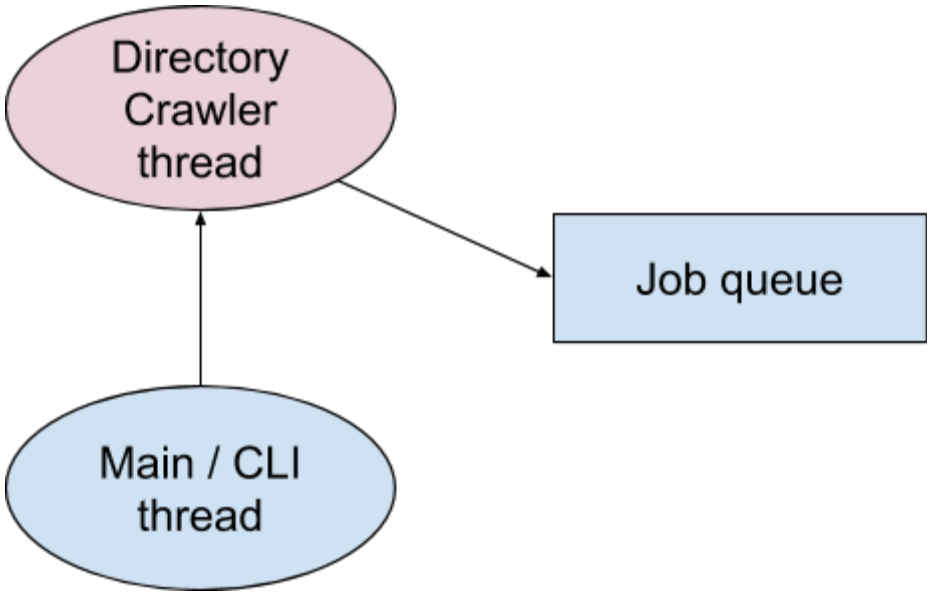
2.1 Directory crawler thread

Ova komponenta obilazi navedeni skup direktorijuma na disku, i traži u njima poddirektorijume koji predstavljaju tekstualne korpuse. Main komponenta će pomoću posebne komande da navodi koje direktorijume directory crawler treba da obilazi, a koji poddirektorijumi predstavljaju korpus će biti zadato u konfiguracionoj datoteci pomoću prefiksa za ime direktorijuma.

Directory crawler treba da pretražuje zadati direktorijum (i da ulazi u poddirektorijume), sve dok ne pronađe direktorijum čije ime počinje zadatim prefiksom. Kada pronađe takav direktorijum, on pretpostavlja da je u pitanju korpus i kreira novi Job za skeniranje tog direktorijuma i smešta ga u Job queue. Directory crawler u isto vreme treba da za sve datoteke u direktorjiumu zabeleži njihovu “last modified” vrednost. Ako su svi fajlovi u direktorijumu već imali zapisanu “last modified” vrednost, i trenutna vrednost je ista kao prethodno zapisana, onda ne treba započeti novi Job.

Primer direktorijumske strukture sa nekoliko tekstualnih korpusa je dat u arhivi kids_d1_data_primer.zip uz ovaj dokument. Unutar te arhive se nalazi i objašnjenje očekivanog ponašanja sistema pri radu sa tim podacima.

Directory crawler nakon skeniranja zadatih direktorijuma pauzira na neko vreme (trajanje dato u konfiguracionoj datoteci) nakon čega ponovo obilazi isti skup direktorijuma.



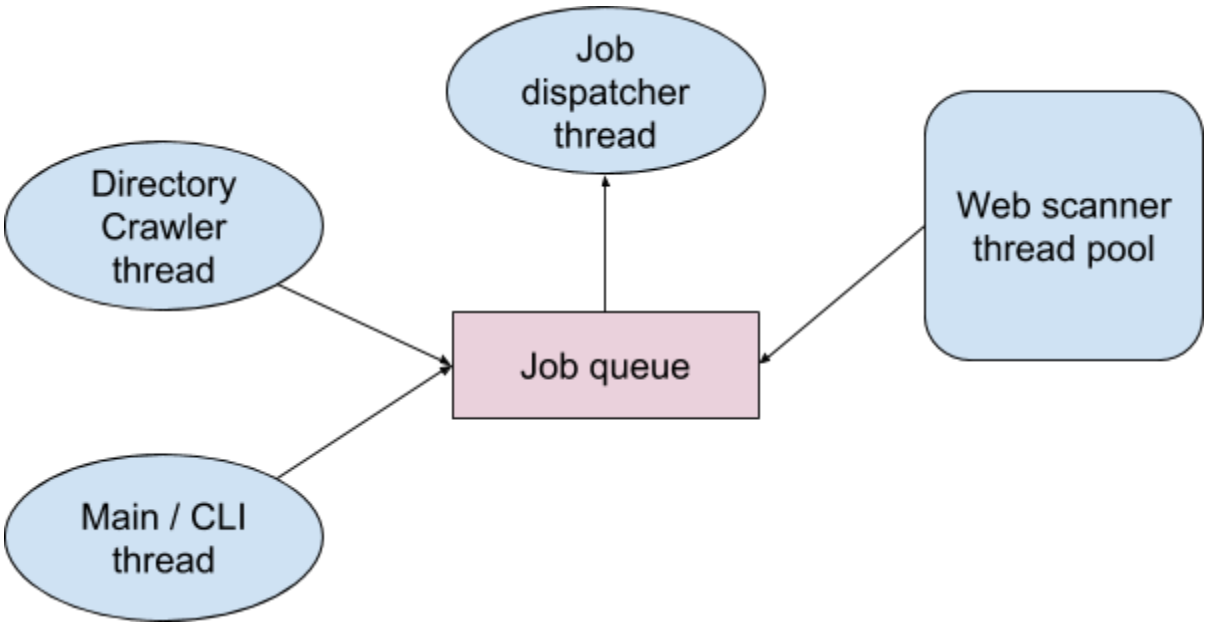
2.2 Job queue

Job queue je deljeni blokirajući red koji sadrži opise poslova koje treba startovati. Jedan posao bi mogao (ali ne mora) da se opiše sa ovakvim interfejsom:

```
public interface ScanningJob {

      ScanType getType();

      String getQuery();

      Future<Map<String, Integer>> initiate();

}
```

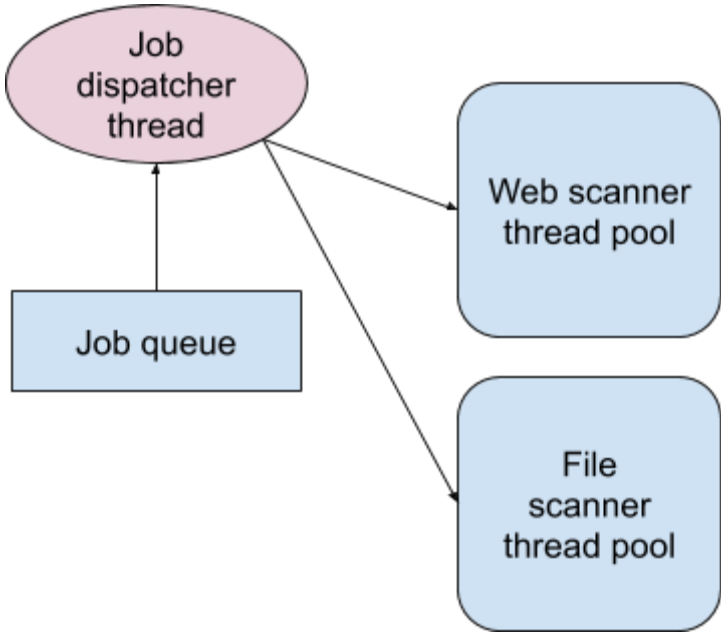
Gde je `ScanType` enumeracija (`FILE` / `WEB`) koja navodi tip posla, `query` je upit sa kojim će se dohvatati rezultati rada ovog posla preko CLI, i `initiate` je metoda koja će započeti posao unutar odgovarajućeg thread pool-a. Rezultat `initiate()` metode je `Future` objekat koji predstavlja posao prebrojavanja za taj korpus, i čiji rezultat je mapa koja ima tačno onoliko ključeva koliko ima ključnih reči, i za svaku od njih vezan broj pojavljivanja u datom korpusu.

Ideja je da bilo koja komponenta može da piše u ovaj red, ali trenutno to su Main / CLI (pri startovanju prebrojavanja za Web korpus), Directory crawler (pri startovanju prebrojavanja za tekstualni korpus) i Web scanner (kada nađe novi URL unutar HTML koda). Ovaj queue čita samo Job dispatcher komponenta.



2.3 Job dispatcher thread

Job dispatcher predstavlja nit koja čeka pojavljivanje nekog Job-a, i onda ga delegira odgovarajućem thread pool sistemu (generalno gledano, Job dispatcher može da izvršava poslove i na drugačiji način, ali i File i Web poslovi će se obavljati unutar thread pool-a). Ova nit treba da se blokira ako nema stavki u redu, a ne da opterećuje procesor konstantnim propitivanjem reda.



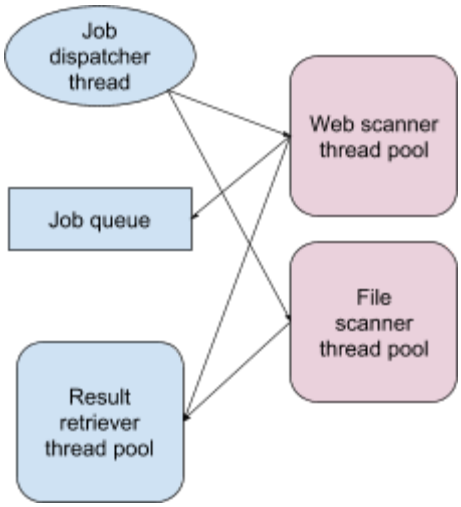
2.4 Web / File scanner thread pool

Pri startovanju Web scanner posla se navodi jedna HTML stranica kao polazna tačka, kao i broj skokova koji će biti napravljen sa te stranice (kada se posao isprva kreira, broj skokova se uzima iz konfiguracione datoteke). Ova komponenta u okviru jednog posla treba da uradi dve stvari:

1. Prebroji pojavljivanja ključnih reči za datu stranicu i prijavi to Result retriever komponenti.
2. Ako je zadati broj skokova za ovaj posao veći od nule, onda treba da pretraži HTML dokument za druge URL-ove, i da kreira nove poslove za pregledanje tih URL-ova. Svi tako kreirani novi poslovi će imati broj skokova koji je umanjen za 1 u odnosu na trenutni. Ovi poslovi se upisuju u Job queue. Ako je pronađeni URL već skeniran, onda ga preskočiti. Skup skeniranih URL-ova treba da se automatski briše po isteku perioda koji se navodi u konfiguracionoj datoteci.

Napomena: za skidanje HTML stranica, kao i njihovo parsiranje (radi pronalaženja linkova) je dozvoljena, i preporučuje se, upotreba neke biblioteke, kao npr. Jsoup, koja taj proces olakšava. Nije neophodno implementirati parsiranje HTML koda, niti ručno razrešavanje URL-ova unutar HTML koda. Primer izlistavanja svih linkova unutar HTML stranice se može naći ovde.

Pri startovanju File scanner posla se navodi direktorijum koji sadrži skup tekstualnih datoteka. Ovaj posao treba podeliti na manje poslove gde jedna nit treba da obrađuje jednu ili više datoteka, pritom taj skup mora da ima zbirnu veličinu (u bajtovima) veću od limita koji se navede pomoću konfiguracione datoteke. Treba težiti da imamo što veći broj niti koje obrađuju korpus, ali nije nužno neophodno pronaći raspored posla za koji je broj niti maksimalan. Dozvoljeno je pretpostaviti da sve datoteke imaju približno sličnu veličinu, tako da posao može da se deli “pohlepno”. Kao rezultat rada ovog posla se vraća broj pojavljivanja svih ključnih reči unutar korpusa, i taj rezultat se smešta u Result retriever komponenti.



2.5 Result retriever thread pool

Ova komponenta skladišti rezultate, i ima operacije za dohvatanje rezultata prebrojavanja, kao i obavljanje nekih jednostavnih operacija nad rezultatima. Rezultati se dohvataju postavljanjem upita (koji će dolaziti od Main / CLI komponente), i upiti mogu da imaju dva oblika:

2.5.1 Broj ključnih reči u korpusu

Najjednostavniji upit vraća broj pojavljivanja svih ključnih reči u nekom korpusu, i ima oblik:

`file|ime_direktorijuma`

`web|ime_domena`

U prvoj varijanti se vraća direktan rezultat rada File scanner komponente za navedeni direktorijum koji predstavlja korpus.

U drugoj varijanti se vraća zbirno pojavljivanje svih ključnih reči na stranicama koje pripadaju istom domenu. Web scanner komponenta daje broj pojavljivanja svih ključnih reči za jednu stranicu, a kada se postavi ovaj upit, Result retriever komponenta treba da u okviru svog thread pool-a započne posao sabiranja svih vrednosti za dati domen. Kada se ova vrednost jednom sračuna, treba je uskladištiti tako da se pri narednim upitima dobija bez ponovnog računanja.

2.5.2 Sumarno

Drugi oblik upita omogućava dohvatanje rezultata za sve korpuse koji su trenutno skladišteni (ako brojanje za neki korpus nije završeno, ovaj posao čeka da se završi), i uzima oblik:

`file|summary`

`web|summary`

Konstruisanje ovih rezultata se obavlja unutar thread pool-a. Kao i kod običnih upita, rezultat rada ovog upita treba skladištiti radi bržeg ponovnog dohvatanja.

2.5.3 Napomene

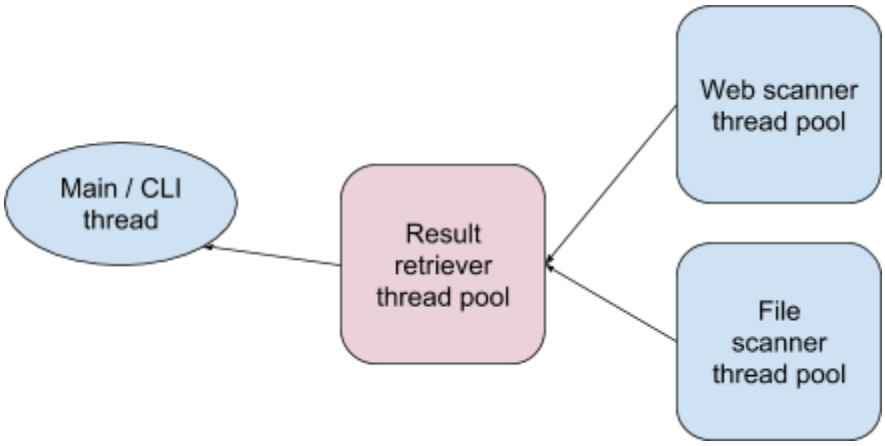
Result retriever komponenta treba da omogući dohvatanje rezultata koje je blokirajuće (get) i koje nije blokirajuće (query). U drugoj varijanti treba ili vratiti rezultat ako je dostpuan, ili vratiti informaciju da računanje tog korpusa nije u toku ako on uopšte ne postoji, ili informaciju da je računanje tog korpusa u toku ako računanje još uvek nije završeno.

Result retriever treba da obezbedi i operacije za brisanje izvesnih skladištenih rezultata, konkretno:

- Brisanje summary rezultata (biće vršeno preko komande na CLI).
- Prepisivanje podataka za tekstualni korpus (vrši se automatski kada se neka datoteka na disku izmeni).
- Brisanje podataka za pojedinačni web domen. Dozvoljeno je samo obeležiti web domenski rezultat kao “zastareo” pri osvežavanju podataka za stranicu koja pripada tom domenu. Nakon toga se pri traženju podataka za taj domen ponovo vrši račun.

Result retriever komponenta bi mogla (ali ne mora) da se opiše sledećim interfejsom:

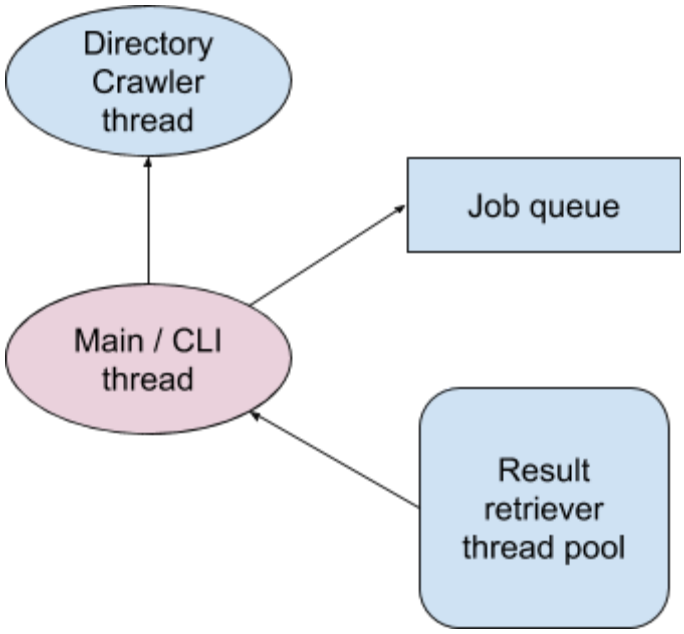
```
public interface ResultRetriever {

	public Map<String, Integer> getResult(String query);

	public Map<String, Integer> queryResult(String query);

	public void clearSummary(ScanType summaryType);

	public Map<String, Map<String, Integer>> getSummary(ScanType summaryType);

	public Map<String, Map<String, Integer>> querySummary(ScanType summaryType);

	public void addCorpusResult(String corpusName, Future<Map<String, Integer>> corpusResult);

}
```



2.6 Main / CLI thread

Main komponenta startuje izvršavanje našeg sistema. Pri startovanju se učitavaju podešavanja iz konfiguracione datoteke, i potom se očekuje komanda od korisnika. Detaljan opis ove datoteke, kao i komandi su dati u odeljku 4. Za sada samo taksativno navodimo kako je ova komponenta povezana sa ostatkom sistema:

- Directory crawler
  - Komanda **`ad`** služi da navede novi direktorijum za obilaženje. U okviru ove komande će biti neophodno dodati stavku u listu direktorijuma koje directory crawler obilazi.
- Job queue
  - Komanda **`aw`** služi da navede novu polaznu stranicu za obilaženje preko web-a. Ova komanda će kreirati novi posao za web i staviti ga direktno u job queue.
- Result retriever
  - Postoji nekoliko komandi za dohvatanje rezultata rada sistema. Sve ove komande će se zapravo obraćati result retriever komponenti i sinhrono ili asinhrono dohvatati rezultate.



3 Razrešavanje grešaka

Neophodno je imati podršku za sledeće situacije, sa navedenim rešenjima. Probleme koji nisu navedeni u tabeli nema potrebe rešavati.

R. Br.	Problem	Rešenje
1.	Korisnik unese nepostojeću komandu ili unese pogrešan broj argumenata.	Prijaviti grešku na konzoli i nastaviti normalno sa radom.
2.	Korisnik navede nepostojeći direktorijum ili URL kojem ne može da se pristupi pri zadavanju posla.	Prijaviti grešku na konzoli i nastaviti normalno sa radom.
3.	File scanner naiđe na datoteku koju ne može da pročita.	Ispisati ime problematične datoteke i nastaviti skeniranje korpusa, bez te datoteke. Dozvoljeno je da se greška prijavi tek pri dohvatanju rezultata.
4.	Web scanner naiđe na URL koji ne može da otvori / dobije kod o grešci.	Ispisati problematičan URL i nastaviti skeniranje bez tog URLa. Dozvoljeno je da se greška prijavi tek pri dohvatanju rezultata.
5.	Query za korpus koji ne postoji.	Prijaviti da korpus ne postoji.
6.	Query za korpus koji nije završen.	Prijaviti da rezultat nije spreman.

Svi navedeni problemi treba da se reše graciozno, tj. da sistem nastavi normalno da funkcioniše. Main nit, job dispatcher i directory crawler, kao i svi pool-ovi treba da nastave da funkcionišu u slučaju da se javi neki od navedenih problema.

4 Podešavanje sistema i komande

Korisnik interaguje sa sistemom tako što unosi komande na komandnoj liniji. Pored toga, rad sistema se dodatno podešava pomoću konfiguracione datoteke.

4.1 Konfiguraciona datoteka

Sistem se podešava pomoću konfiguracione datoteke `app.properties`, koja ima sledeće parametre:

```
#spisak ključnih reči koje se traže u korpusima, odvojene zarezom.
#ključne reči se u korpusima ne broje ako su deo neke druge reči,
#već samo kada stoje samostalno u tekstu
keywords=one,two,three
#prefiks za direktorijume koji sadrže tekstualne korpuse
file_corpus_prefix=corpus_
#period pauziranja za directory crawler, dat u ms
dir_crawler_sleep_time=1000
#ograničenje za file scanner komponentu, dato u bajtovima
file_scanning_size_limit=1048576
#broj skokova koje će web scanner komponenta napraviti pri pretrazi
hop_count=1
#vreme nakon kojeg se briše skup posećenih url-ova, dato u ms
url_refresh_time=86400000
```

Svi ovi parametri se čitaju jednom pri startovanju aplikacije, i neće se menjati tokom rada. Jedini način da se vrednosti ovih parametara izmene je da se aplikacija potpuno zaustavi i ponovo pokrene.

4.2 Komande

Sistem podržava sledeće komande:

Naziv komande: `ad`

Parametar: `String`

Opis: Add directory. Dodaje novi direktorijum za skeniranje, koji se prosleđuje Directory scanner komponenti. Direktorijum se nalazi unutar projekta, i zadaje se kao relativna putanja. Unutar ovog direktorijuma može da bude proizvoljno mnogo poddirektorijuma, ali neki od njih treba da imaju ime koje počinje sa prefiksom navedenim u konfiguracionoj datoteci (`file_corpus_prefix`), i oni predstavljaju tekstualne korpuse za naš sistem.

Naziv komande: `aw`

Parametar: `String`

Opis: Add web. Dodaje stranicu od koje treba da krene novo obilaženje. Broj skokova za ovaj posao se uzima iz konfiguracione datoteke (hop_count). Za svaki URL koji je pronađen na stranici će se napraviti novi posao koji će imati broj skokova za jedan manji od trenutne. Za svaku stranicu koja se ovako obiđe se broje pojavljivanja ključnih reči, i skladište se u result retriever komponenti.

Naziv komande: `get`

Parametar: `String`

Opis: Dohvata rezultat iz Result retriever komponente i ispisuje ga na konzoli. Kao argument se navodi upit (opisan u odeljku 2.5). Ova komanda blokira dalji rad dok ne dobije rezultat.

Naziv komande: `query`

Parametar: `String`

Opis: Dohvata rezultat iz Result retriever komponente i ispisuje ga na konzoli. Kao argument se navodi upit (opisan u odeljku 2.5). Ova komanda ne blokira dalji rad, već samo ispisuje poruku ako rezultati nisu dostupni ili ako posao za taj upit još uvek nije započet.

Naziv komande: `cws / cfs`

Parametar: -

Opis: Clear web summary / Clear file summary. Ove dve komande se navode bez argumenta i kada se one navedu, treba javiti Result retriever komponenti da obriše odgovarajući summary rezultat, ako ga ima sačuvanog.

Naziv komande: `stop`

Parametar: -

Opis: Gasi aplikaciju. Zaustavlja sve thread pool-ove i javlja svim nitima da uredno završe sa radom. Ne koristiti nasilna prekidanja / interrupt / daemon za potrebe ove komande.

4.3 Primeri korišćenja sistema

Primeri pretpostavljaju da se koristi konfiguraciona datoteka navedena u odeljku 4.1. Boldovan tekst predstavlja upisane komande, dok je običan tekst ispis programa.

Primer 1: dodavanje direktorijuma za skeniranje i dohvatanje rezultata. Direktorijum **`data`** unutar projekta sadrži poddirektorijume **`corpus_a`** i **`corpus_a2`** negde unutar svoje strukture i ti direktorijumi sadrže tekstualne datoteke za skeniranje.

`ad data`

```
Adding dir /home/pengu/Apps/eclipse-se/workspace/KiDSDomaci1/data
Starting file scan for file|corpus_a2
Starting file scan for file|corpus_a
```

`get file|corpus_a`

```
{one=56, two=41, three=8}
```

`stop`

```
Stopping...
```

Primer 2: dodavanje web stranice za obilazak i čitanje web summary rezultata. (testirati na drugim URL-ovima da gospodin Gates ne bi pomislio da neko iz Srbije pokušava da mu obori sajt)

aw https://www.gatesnotes.com/2019-Annual-Letter

```
Starting web scan for
web|https://www.gatesnotes.com/2019-Annual-Letter

Starting web scan for web|https://www.gatesnotes.com/

Starting web scan for web|https://www.gatesnotes.com/Books

...

Starting web scan for
web|https://blog.23andme.com/23andme-research/new-study-finds-genetic-
links-risk-premature-births/

Starting web scan for
web|https://www.wired.com/story/wired25-bill-gates-stephen-quake-blood
-tests/
```

query web|summary

```
Summary is not ready yet
```

get web|summary

```
wired.com: {one=0, two=1, three=0}

twitter.com: {one=0, two=0, three=0}

blog.23andme.com: {one=0, two=1, three=1}

windows.microsoft.com: {one=0, two=0, three=0}

gatesnotes.com: {one=30, two=2, three=25}

reddit.com: {one=0, two=0, three=0}
```

5 Predaja zadatka

5.1 Način predaje zadatka

Zadatak se predaje putem mail-a na bmilojkovic@raf.rs. Java projekat imenovati na sledeći način: “kids_d1_ime_prezime_ind”. Npr. “kids_d1_student_studentic_rn0101”.

Arhivirati ovaj direktorijum (.zip) i arhivu poslati kao attachment uz mail.

U tekstu mail-a obavezno navesti:

- Ime i prezime
- Broj indeksa
- Grupa, po zvaničnom spisku

Subject mail-a mora da bude u obliku: “[KiDS] D1 ime_prezime_ind”.

Npr. “[KiDS] D1 student_studentic_rn0101”

Naziv arhive mora da bude u obliku: “kids_d1_ime_prezime_ind.zip”

Npr. “kids_d1_student_studentic_rn0101.zip”

Rok za predaju je:

- Utorak, 16. april 23:59:59 za grupe koje slušaju KiDS utorkom
- Sreda, 17. april 23:59:59 za grupe koje slušaju KiDS sredom
- Petak, 19. april 23:59:59 za grupe koje slušaju KiDS petkom

Rok je definisan po grupi kojoj student zvanično pripada, tj. na čijem spisku se nalazi. Za ponovce / studente bez grupe se primenjuje najkasniji termin - Petak, 19. april 23:59:59.

Neće se pregledati zadaci (tj. biće dodeljeno 0 poena) ako se desi bilo koje od:

- Sadržaj mail-a nije po navedenom obliku.
- Subject mail-a nije po navedenom obliku.
- Naziv arhive nije po navedenom obliku.
- Predaja se desi nakon navedenog roka.

5.2 Odbrana i bodovanje

Odbrana domaćih zadataka je obavezna. Termin za odbranu prvog domaćeg zadatka će biti naknadno objavljen. Odbrane će se vršiti po grupama. Ako ste iz bilo kog razloga sprečeni da prisustvujete odbrani, obavezno to najavite unapred, kako bismo mogli da zakažemo vanredni termin za odbranu.

Svrha odbrane je da se pokaže autentičnost zadatka. Ovo podrazumeva odgovaranje na pitanja u vezi načina izrade zadatka, ili izvršavanje neke izmene nad zadatkom na licu mesta. U slučaju da odbrana nije uspešna, dodeljuje se -15 poena na domaćem zadatku.

Zadatak se boduje na sledeći način:

- Web scanning komponenta - 4 poena
- File scanning komponenta - 3 poena
- Result retriever komponenta - 4 poena
- Directory crawler i job dispatcher - 2 poena
- Main / CLI i konfiguracija - 2 poena
- Greške nisu pravilno razrešene - -2 poena

Zadatak je moguće raditi parcijalno, ali obavezno je da se kompajluje i da može da se pokrene, kao i da je moguće pokazati da implementirana komponenta radi po zahtevima iz odeljka 2 ovog dokumenta. Stavka “Main / CLI i konfiguracija” se prihvata samo uz implementiranu još makar jednu stavku.