Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Laboratorij profila PIIS iz kolegija
Raspodjeljeni sustavi

Filtriranje toka poruka s Twittera

Ivan Srdić (voditelj),

David Geček,

Ivan Golik,

Petar Ilijašić,

Goran Rumin

Zagreb, prosinac 2015.

Sadržaj

1. Zadatak

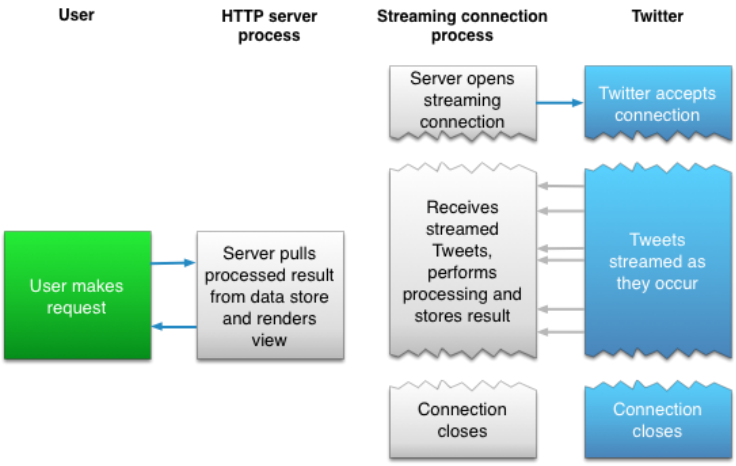
Na društvenoj mreži Twitter se dnevno objavljuje preko 340 milijuna kratkih poruka zvanih tweets, a za to vrijeme poslužitelji ove društvene mreže obrade preko milijardu i pol jednokratnih korisničkih upita. Kako korisnici mogu slijediti stotine drugih osoba na Twitteru, broj objava kojima imaju pristup može biti ogroman. Ideja ovog projekta je olakšati potragu za zanimljivim objavama putem trajnih upita koji će kontinuirano filtrirati tokove objava te korisnicima isporučivati samo one najzanimljivije. Razviti web aplikaciju koja će putem Twitter Streaming API-ja u stvarnom vremenu dohvaćati objave te ih filtrirati na osnovu trajnih upita korisnika. Za indeksiranje upita i rangiranje objava koristiti biblioteke funkcija razvijene na projektu Apache Lucene. Korisničko sučelje treba omogućiti unošenje jednog ili više trajnih upita u obliku ključnih riječi i potrebnih parametara te prikazivati filtrirane objave u stvarnom vremenu.

2. Korištene tehnologije

Aplikacija je nazvana *TweetYouAreLookingFor* te njena izrada zahtjeva korištenje nekoliko tehnologija, ovisno o domeni rada. Domena rada je od dizajna aplikacije do implementacije procesa koji se događa u pozadini korisničkog sučelja. U nastavku će biti definirano koje su se tehnologije koristile pri implementaciji navedenog sustava.

2.1 Twitter Stream API

Twitter Stream API se koristio kao izvor poruka koje treba filtrirati.
Stvaranjem Twitter korisničkog računa dobiva se pristup svim razvojnim alatima koje Twitter nudi. Potrebno je registrirati aplikaciju kako bismo dobili ključeve za autentifikaciju.



Slika 2.1 Mehanizam Twitter Stream API-a

Od tokova podataka koje Twitter nudi najbolji izbor je bio User Streams pošto nudi sve što je potrebno za implementaciju sustava. Za korisnika koji doda aplikaciju u svoj korisnički račun moguće je dobiti većinu informacija među kojima su i tweetovi korisnika koje prati.

Prilikom spajanja na Twitter Stream API potrebno je proslijediti ključeve za autentifikaciju nakon čega počinje tok podataka koji je potrebno filtrirati nakon primitka. Za svakog korisnika potrebno je otvoriti novu konekciju na Twitterov API jer se prilikom autentifikacije koriste ključevi kako od aplikacije tako i od samog korisnika za kojeg želimo dobivati podatke.

2.2 Twitter HBC

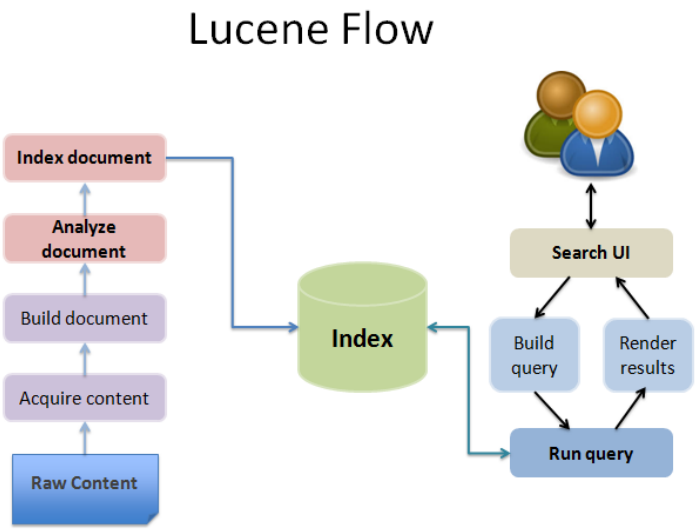
Twitter HBC (Hosebird Client) je Java HTTP klijent koji služi za *konzumiranje* Twitter Streaming API-a. Pomoću Twitter HBC klijenta moguće je na jednostavan način spojiti se na Twitter API koristeći OAuth te primati poruke koje API šalje. Dodatno, Twitter HBC omogućuje apstrahirano specificiranje filtra za tip podataka, vrste točke spajanja, parsera poruka i dodatnih opcija koje Twitter konkretni API podržava.

Nakon specificiranja svih potrebnih podataka moguće je spojiti se na Twitter API. Za primanje poruka preporučuje se otvaranje nove dretve. Poruke se dohvaćaju iz reda koji je definiran prilikom inicijalizacije klijenta. Komunikaciju je moguće prekinuti u bilo kojem trenutku pozivanjem odgovarajuće metode. Zbog razine apstrakcije Twitter HBC klijent znatno olakšava korištenje Twitter Stream API-a jer su stvari poput error handlinga, ponovnog spajanja i ostalih detalja već implementirani.

2.3 Apache Lucene

Apache Lucene je besplatan softver otvorenog koda, originalno pisan u Javi. Koristi se za pretraživanje teksta i ima nekoliko pod-projekata od kojih je za ovaj projekt najbitniji *Apache Lucene Core*. *Lucene Core* omogućava pretraživanje i rangiranje teksta, pretraživanje fraza, indeksiranje. Osim njega, Apache razvija i *Apache Solr*, server i web aplikaciju, koja koristi *Lucene Core* za pretraživanje i nudi dodatne funkcionalnosti. Međutim, pošto je u ovom projektu važna komponenta pretraživanje teksta s dosta preciznim zahtjevima, koristile su se *Lucene* biblioteke bez *Solra*.

Lucene se koristio za indeksiranje teksta (i autora) tweetova prema korisničkim zahtjevima, pretraživanje i rangiranje teksta, a prema tome će se izvoditi filtriranje tweetova prema korisniku (u Searcher.search(query, hitsPerPage) se ubacuje koliko najboljih rezultata sustav treba korisniku prikazati).



Slika 2.2 Kako radi *Apache Lucene*

2.4 Java Spark

Spark radni okvir je besplatan radni okvir otvorenog kôda namijenjen pisanju web-aplikacija u Javi. Svaka web-aplikacija se piše kao Java konzolna aplikacija te time ima pristup svim prednostima Jave uz korištenje Spark-ovog sustava za mapiranje URL-ova i raznih pogona za renderiranje HTML pogleda (eng. *Template engine*) koji omogućavaju *Model-View-Controller* pristup. Osnovna ideja radnog okvira je jednostavnost konfiguracije i pisanja Java kôda pa se samim time najčešće koristi za pisanje REST servisa. Radni okvir već ima integriran web-server te se aplikacija napisana koristeći Spark može pokrenuti kao Java konzolna aplikacija ili se mogu koristiti neki drugi vanjski Java serveri.

Slika 2.3 Java Spark

U ovom projektu Spark radni okvir se koristio za izradu web-aplikacije koja će omogućavati registraciju i prijavu korisnika, definiranje pretplata i slanje filtriranih poruka s Twittera prijavljenim korisnicima u stvarnom vremenu putem Web-socket protokola.

2.5 HTML

HTML ili HyperText Markup Language je jezik za izradu web-stranica. Budući da HTML opisuje strukturu izgleda web-stranice koja se prikazuje, on je prezentacijski (markup) jezik, a ne programski jezik tj. ne omogućava pisanje proceduralne logike. HTML definira elemente stranice, njihov raspored i ulogu, dok im izgled definira CSS.

2.6 CSS

CSS, punim nazivom Cascading Style Sheets, je jezik koji definira izgled i format dokumenta napisanog prezentacijskim jezikom. CSS prvenstveno služi razdvajanju sadržaja dokumenta od njegove prezentacije, odnosno izgleda, što pomaže u preglednijem pristupu sadržaju i omogućava veću kontrolu i fleksibilnost kod specificiranja karakteristika određenih elemenata.

Slika 2.4 HTML, CSS i JavaScript upotpunjuju jedan drugog

2.7 JavaScript

Web-stranici funkcionalnost daju skripte koje su pisane u JavaScriptu. U projektu se uz uobičajene funkcionalnosti navedenih jezika koristito i JavaScript WebSocket API koji je aplikaciji omogućio primanje podataka sa servera bez slanja zahtjeva.

2.8 Bootstrap

Bootstrap je bespatna kolekcija alata otvorenog koda za kreiranje web stranica i web aplikacija. Sadrži niz radnih okvira za različite forme i komponente web aplikacije u području rada sa HTML i CSS jezicima. Također sadrži JavaScript dodatke, čime olakšava izradu dinamičnih web aplikacija.

3. Korisnički slučajevi

Jedan od karakteristika aplikacije *TweetYouAreLookingFor* je da bude intitivan za korištenje i jednostavan, stoga se bazira na nekoliko korisničkih slučajeva:

1) Prijava u sustav *TweetYouAreLookingFor*
2) Postavljanje novih upita za specifične tweetove
3) Odjava iz sustava

3.1 Prijava u sustav *TweetYouAreLookingFor*

Razina: Administracijski cilj
Primarni aktor: Korisnik
Stakeholderi: Korisnik – u svrhu provjere autentičnosti podataka na *Twitter* društvenoj mreži
Preduvjeti: Korisnik je registriran na društvenu mrežu *Twitter*
Uvjeti pri završetku: osoba je prijavljena u sustav *TweetYouAreLookingFor*

Glavni scenarij:

1. Korisnik pokreće sustav *TweetYouAreLookingFor*
2. Korisnik odabire tab *Pokreni*
3. Ukoliko korisnik nije prijavljen u sustav sa podacima svog *Twitter* accounta, sustav preusmjeruje na *Twitter* stranicu za prijavu
4. Korisnik unosi svoje korisničke podatke
5. Uneseni podaci se uspoređuju sa podacima u sustavu radi provjere autentičnosti
6. Korisnik je prijavljen u sustav i usmjeren na stranicu za postavljanje upita

Cijeli scenarij se može neograničeno izvoditi, odnosno nema ograničenja na pokušaj prijave u sustav.

Alternativni scenariji:

a) Prijava je neuspješna, što rezultira porukom o neispravnim autorizacijskim podacima

3.2 Postavljanje novih upita za specifične tweetove

Razina: Korisnički cilj
Primarni aktor: Korisnik
Stakeholderi: Korisnik – postavlja upite za željene tweetove
Preduvjeti: Korisnik je prijavljen u sustav *TweetYouAreLookingFor*
Uvjeti pri završetku: Sustav je uspješno prikazao traženi rezultat

Glavni scenarij:

1. Korisnik pokreće sustav *TweetYouAreLookingFor*
2. Korisnik se prijavljuje u sustav - INCLUDE use case: *Prijava u sustav TweetYouAreLookingFor*
3. Korisnik unosi upit u kojem izjašnjava koje tweetove želi pratiti
4. Sustav obrađuje upit
5. Sustav prikazuje korisniku tražene tweetove

Cijeli scenarij se može neograničeno izvoditi, može slati različite upite sustavu.

Alternativni scenariji:

b) Upit je rezultirao neuspješnom pretragom, čime dobiva prazan rezultat ili poruku kako nema takvih tweetova.

3.3 Odjava iz sustava *TweetYouAreLookingFor*

Razina: Administracijski cilj
Primarni aktor: Korisnik
Stakeholderi: Korisnik – u svrhu odjave korisnika iz sustava i prestanka rada
Preduvjeti: Korisnik je prijavljen u sustav
Uvjeti pri završetku: osoba je odjavljena iz sustava i prestala sa radom

Glavni scenarij:

1. Korisnik je prijavljen u sustav *TweetYouAreLookingFor*
2. Korisnik odabire tab *Log Out*
3. Sustav obrađuje zahtjev
4. Korisnik je uspješno odjavljen iz sustava *TweetYouAreLookingFor*

4. Opis razvijene programske potpore

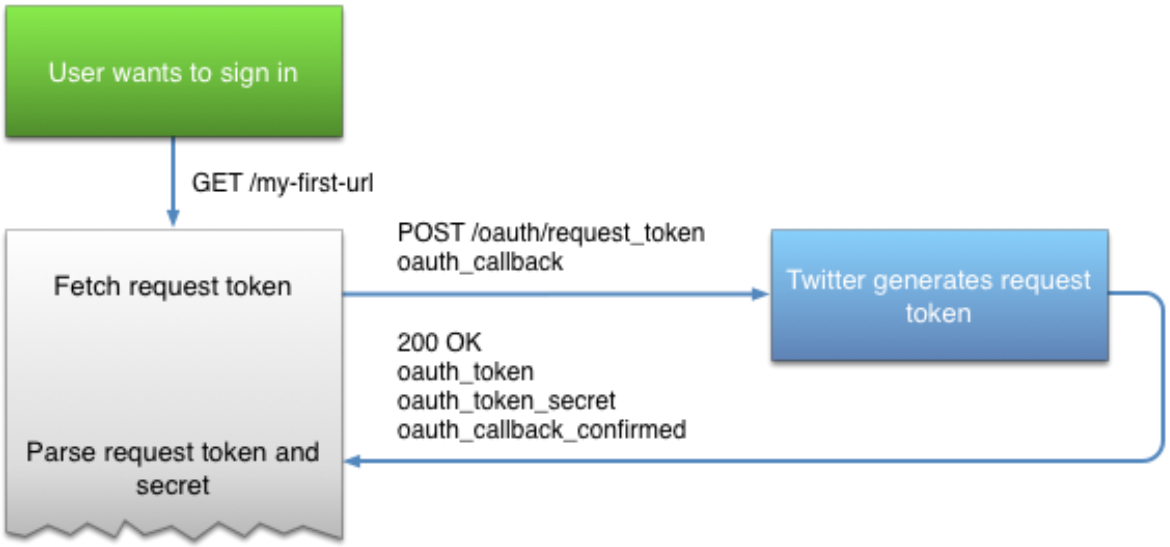
4.1 Komunikacija sa Twitterom

Autorizacija

Za autorizaciju aplikacije se danas koristi standardni OAuth protokol za autentifikaciju. Prednost OAuth protokola je u tome što aplikacija niti u jednom trenutku ne dobije lozinku korisnika, već se korisnik preusmjerava na Twitter gdje se treba prijaviti.

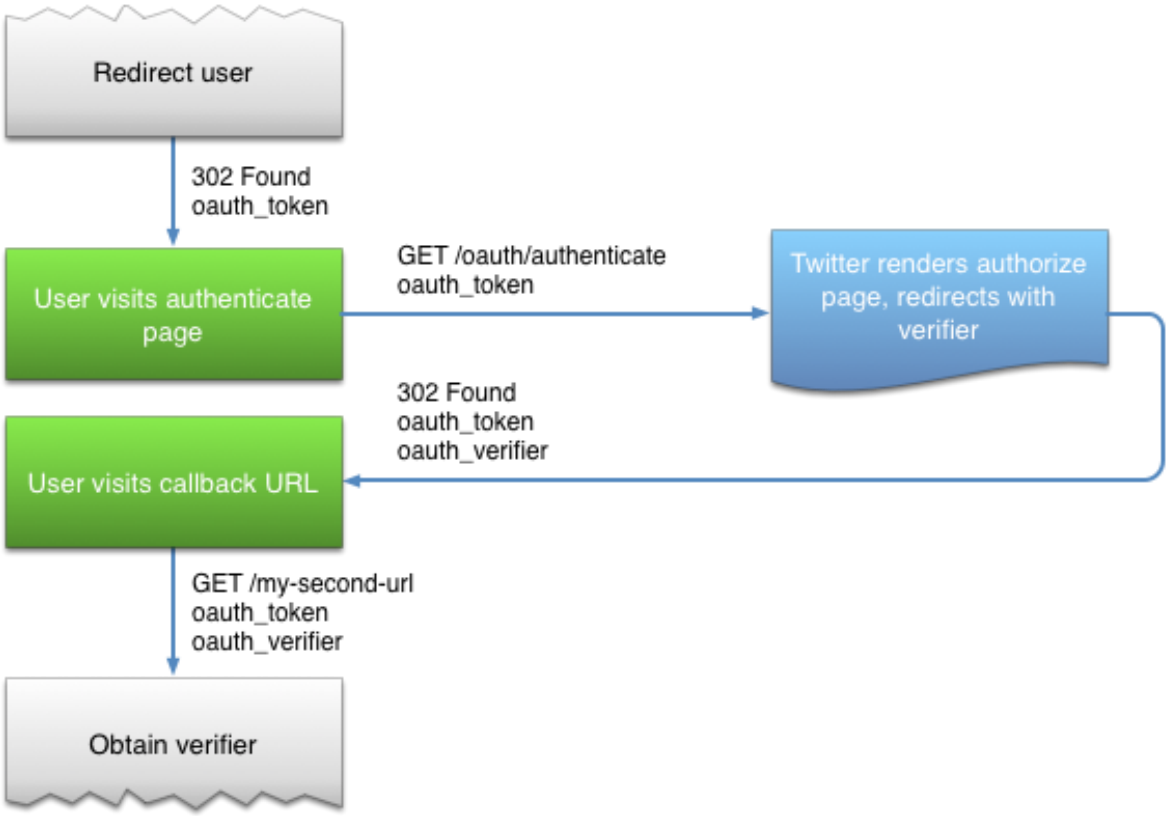
Postupak autorizacije aplikacije je sljedeći:

1. Korisnik posjeti posjeti aplikaciju i želi koristiti dio za koji je potrebno prijavljivanje. Aplikacija dohvaća *request* token i *secret* te ih parsira.



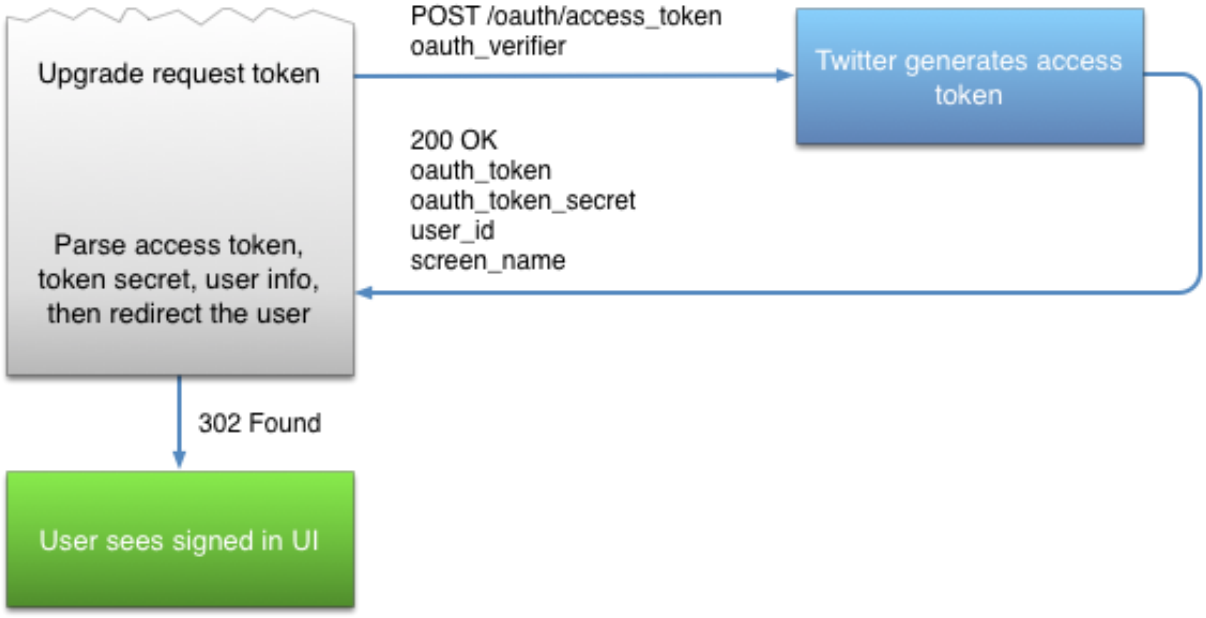
4.1 Dohvat *request* tokena

2. Korisnika se zajedno sa *request* tokenom preusmjerava na stranicu za autorizaciju koju *Twitter* prikazuje. Nakon prijave i autorizacije aplikacije korisnika se zajedno sa ovjeriteljem preusmjerava na *callback* URL.



Slika 4.2 Verifikacija podataka i povratak na *callback* URL

3. Zadnji korak je generiranje access tokena uz pomoć request tokena i verifiera. Access token se koristi za pristup Twitter Stream API-u.



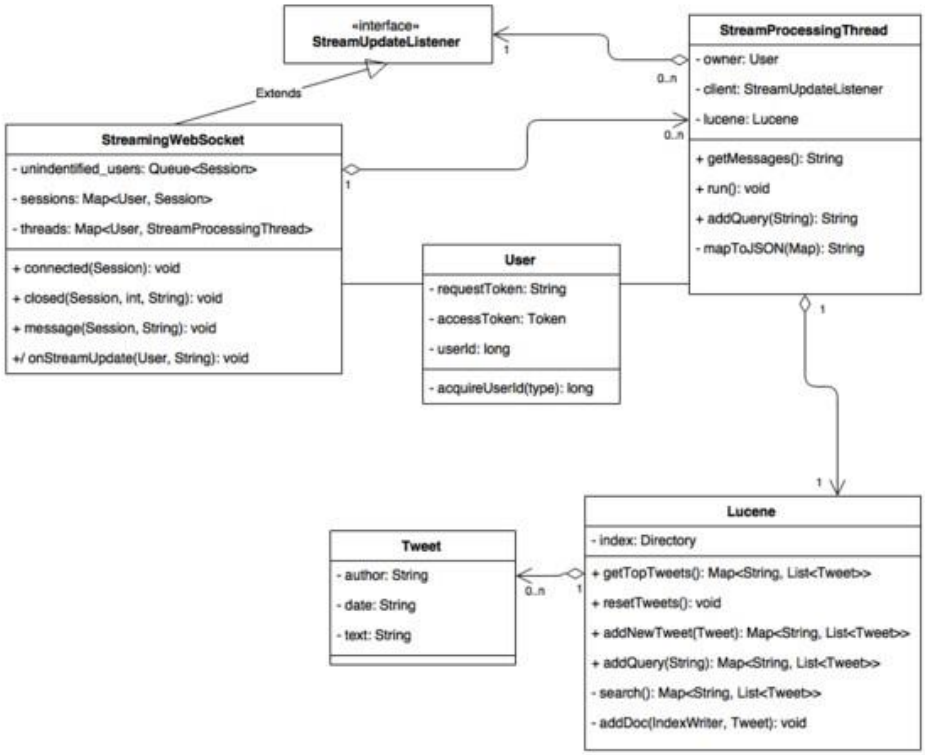
4.3 Generiranje *access* tokena

Za izvršavanje opisanog postupka autorizacije aplikacije se koristi ScribeJava, jednostavnu Java OAuth biblioteku. Koraci su potpuno isti, ali korištenjem ScribeJava nije potrebno razmišljati o formi zahtjeva već samo potrebno proslijediti odgovarajuće podatke. Twitter Stream API ima nekoliko verzija, a za ovaj projekt najkorisnija je *User Streams* pomoću koje se može pristupiti svim podacima prijavljenog korisnika.

Kako bi korištenje *Twitter Stream* API-a bila što jednostavnija, *Twitter* je implementirao *Twitter Hosebird Client*, java klijent koji služi za konzumaciju *Stream* API-a. Pomoću *Twitter Hosebird Client-a* moguće je u nekoliko linija koda inicijalizirati konekciju sa *Stream* API-em nakon čega je moguće koristiti ugrađenu metodu za dohvat sljedeće poruke.

4.2 Web-Socket komunikacija

Nakon prijave korisnika putem Twitterovog servisa, korisnik se preusmjerava na stranicu za zadavanje upita i pregled tweetova. Nakon učitavanja, stranica se spaja na server putem web-socket protokola. Prva poruka koju klijent šalje na server je korisnikov *request token* koji mu se dodjeljuje prilikom prijave na Twitterov servis. *Request token* se koristi za dohvat autorizacijskih podataka za korisnikov račun koji su pohranjeni na serveru prilikom prijave. Iz dohvaćenih podataka stvara se instanca razreda User u kojem se dohvaća Twitterov korisnički identifikator, te se taj identifikator koristi u sustavu za raspoznavanje korisnika. Ako se korisnik prvi put prijavljuje na naš servis za filtriranje tweetova, pokreće se dretva za dohvat tweetova s njegovog računa i stvara se nova instanca Lucene razreda za filtriranje teksta. Daljnje poruke koje klijent šalje serveru putem web-socketa se tretiraju kao novi upiti te se proslijeđuju instanci Lucene razreda. Ako nakon primanja novih tweetova s Twittera dođe do ažuriranja liste tweetova za neki upit, cijela lista se klijentu dostavlja u JSON formatu također putem web-socket protokola. U slučaju da se klijent odspoji, dretva i dalje prikuplja i rangira tweetove, te će pri ponovnom spajanju klijenta dočekati ažurni podaci.



Slika 4.2 Dijagram razreda djela softvera za obradu *tweet*-ova

4.3 Apache Lucene

Svaka Lucene instanca povezana je s određenim korisnikom i sadrži index svih tweetova koje taj korisnik prati. Prilikom objavljivanja novog tweeta poziva se Luceneova metoda:

```
public HashMap<String, List<Tweet>> addNewTweet(Tweet tweet);
```

Ta metoda prima objekt tipa Tweet. Tweet je jednostavni razred koji se sastoji od nama bitnih parametara za jedan tweet: autor, datum i tekst tweeta. Sve varijable su privatne, ali imaju svoje *gettere* i *settere.* Metoda dodaje tweet u index pomoću privatne metode *addDoc()* i vraća *HashMap<String, List<Tweet>>* pomoću privatne metode *search()* gdje je String upit kojeg korisnik prati, a lista je lista top 10 tweetova (ili manje ako ih nema 10) za taj upit. U HashMap se dodaju upiti samo ako je dodani tweet promjenio ljestvicu za taj upit, tako da je moguće da HashMap bude prazan ako nije utjecao niti na jedan upit, a moguće je da ima nekoliko članova ako je utjecao na više upita. Time se minimizira količina podataka koji se šalju korisniku i koje treba osvježiti u pregledniku.

Za dodavanje novih upita koje Lucene treba pratiti koristi se javna metoda:

```
public HashMap<String, List<Tweet>> addQuery(String query);
```

Ova metoda parsira upit koji joj je dan i stavlja ga na listu praćenja. Lista tweetova za novi upit je prazna, pa ova metoda popunjava tu listu iz trenutnog indeksa tweetova i vraća HashMap s jednim članom: novi upit i lista tweetova povezana s novim upitom.

Kad se korisnik ponovno logira i potrebno mu je dostaviti sve liste tweetova, a ne samo promjene, razred Lucene nudi metodu:

```
public HashMap<String, List<Tweet>> getTopTweets();
```

Metoda vraća HashMap sa svim upitima koji su dodani na praćenje i pripadnim listama tweetova.

StreamProcessingThread se brine o tome da indeks svakog korisnika nebi postao prevelik, pa svakih 24 sata poziva Lucenovu metodu *resetTweets()*.

4.4 Bootstrap i JavaScript

Za izradu korisničkog sučelja koristio se *Bootstrap* radni okvir poz nazivom *Creative.* Pomoću radnog okvira i JavaScript metoda je omogućeno pokretanje glavne funkcionalnosti sustava te sustav za prijavu i autentifikaciju korisnika. Za pozadinu je kreirana video petlja koja se beskonačno vrti, a video je preuzet sa *open source* izvora.

```
var webSocket = new WebSocket("ws://" + location.hostname +
":4567/stream/");
webSocket.onopen = function () {
          webSocket.send(getCookie("requestToken"));
};


function getCookie(name) {
     var value = "; " + document.cookie;
     var parts = value.split("; " + name + "=");
     if (parts.length == 2) return
     parts.pop().split(";").shift();
}
```

Navedeni kod omogućuje rad sa kolačićima kako bi se omogućilo da sustav radi sa odgovarajućim korisnikom koji je prijavljen u sustav. Za svakog korisnika se spremaju na server njegovi upiti. Za svaki upit se dinamički kreira blok na stranici koji sadrži tweetove u odnosu na upit koji je postavljen. Dinamičnost aplikacije je izvedena brojnim JavaScript metodama.

5. Budućnost aplikacije

Aplikacija je trenutno na osnovnoj razini funkcionalnosti te ima potencijal proširiti se na druge društvene mreže kao što su *Facebook* ili *LinkedIn*. Svaka društvena mreža koja ima veliki broj objava koje se generiraju na dnevnoj bazi zahtjeva filtriranje kako bi korisnici lakše pristupili sadržajima koje žele. Upiti bi se također mogli izvoditi glasovnim naredbama ili bi se mogla dodati funkcionalnost da tweetovi budu dinamički pročitani koristeći sintetizator glasova za osobe koje imaju neku vrstu invaliditeta. Mogućnosti su u programiranju neograničene, što vrijedi i za ovu aplikaciju.

6. Literatura

https://dev.twitter.com/streaming/overview
https://dev.twitter.com/oauth
https://github.com/scribejava/scribejava
https://github.com/twitter/hbc
https://lucene.apache.org/core/
http://www.lucenetutorial.com/lucene-in-5-minutes.html
http://www.lucenetutorial.com/lucene-vs-solr.htm
http://startbootstrap.com/template-overviews/creative/
http://stackoverflow.com/