

## Arhitektura sistema:

- Aplikacija će se razvijati u Javi uz pomoć Spring Boot-a.
- Aplikacija će biti implementirana pomoću uobičajene troslojne arhitekture (kontroler, servis, repozitorijum).
- Aplikacija će omogućiti eksterni pristup putem odgovarajućeg API-a i biti dostupna preko REST-a. Ovo je namenjeno za pristup aplikacije koja će biti razvijana u okviru predmeta "Upravljanje poslovnim procesima" (UPP).
- Prilikom novog apliciranja za posao (preko UPP aplikacije), prvo će biti kontaktiran Geolocation API kako bi se dobavile koordinate na osnovu IP adrese. Nakon toga će se logovati informacija o pristizanju novog zahteva. Potom će se parsirati pdf dokument koji predstavlja CV aplikanta. Nakon toga će se sve potrebne informacije proslediti Elastic search-u.
- Komunikacija između aplikacije i Elastic search-a će biti omogućena putem REST-a.
- Filebeat će nadgledati .log fajl koji generiše ova aplikacija i slati podatke o toj promeni Logstash-u.
- Logstash će obraditi primljeni .log fajl i proslediti podatke na indeksiranje Elastic search-u.

## Geoprostorna pretraga:

Prilikom apliciranja za posao, korisnik će biti u obavezi da u formi unese adresu prebivališta. Ta adresa će biti konvertovana u geografske koordinate pomoću **Geocoding API**-a dostupnog na <https://positionstack.com/>. Te koordinate će biti sačuvane u bazi podataka u okviru podataka o aplikantu.

Takođe, na osnovu IP adrese sa koje je aplikant aplicirao biće izvučene geografske koordinate pomoću **Geolocation API**-a dostupnog na <https://ip-api.com/>. Ti podaci će biti smešteni u okviru .log fajlova koji će biti generisani u aplikaciji.

Što se tiče direktno geoprostorne pretrage, korisnik unosi ime grada i radijus udaljenosti. Koordinate na osnovu imena grada će biti dobavljene pomoću Geocoding API-a, a za računanje udaljenosti koristiće se **Haversine** metoda.

## Skladište podataka (baza):

### Indexing unit:

#### Aplikanti:

- ime (unos u formi)
- prezime (unos u formi)
- stepen školske spreme (unos u formi)
- longitude (izvedeno iz adrese pomoću geocoding API-a)
- latitude (izvedeno iz adrese pomoću geocoding API-a)
- text (parsirano iz CV dokumenta)

#### Logovi:

- longitude (na osnovu IP adrese prilikom prijave)
- latitude (na osnovu IP adrese prilikom prijave)
- timestamp (trenutak slanja prijave)

#### Ostali podaci:

- email (unos u formi)
- adresa (unos u formi)
- P.S. eventualno podela na adresu, grad, ulicu, itd...

#### CV dokumenti:

Planirano je da svaki pdf sa CV dokumentom koji je stigao na endpoint za apliciranje bude sačuvan na lokalnoj mašini u okviru `/cv` foldera. Za takav pristup će biti potrebno mapiranje foldera u docker volume. S obzirom da nakon parsiranja CV-a sâm pdf neće biti potreban pošto će svi isparsirani podaci biti sačuvani u okviru baze, pretpostavljam da će postojati trenutno/povremeno brisanje "iskorišćenih" dokumenata.

## Servisi:

**Serbian analyzer plugin** za elastic search (<https://github.com/chenejac/udd06>) je kloniran na lokalnoj mašini. U okviru analizera se nalazi i README.md fajl sa komandom za kreiranje .zip fajla potrebnog za izvršavanje unutar Elastic search-a. Taj .zip je ubačen u root folder mog projekta kako bi se kasnije mogao izvršavati u Elastic search kontejneru.

Celokupna ELK stack arhitektura servisa (**Elastic search**, **Logstash**, **Kibana**), kao i **Filebeat**, podignuta je korišćenjem **Docker** kontejnera. Zbog mogućnosti razlikovanja u verzijama, za svaki servis je odabrana ista verzija (7.4.0) sa kojom je kompatibilan Serbian analyzer. Pošto se svi pomenuti servisi nalaze u istoj mreži (**network**), mogu videti jedni druge i međusobno komunicirati. Pored toga, dodate su i odgovarajuće environment promenljive za svaki servis.

Prethodno kreirani .zip plugin-a se instalira prilikom podizanja elastic search kontejnera preko **elasticsearch-plugin install** komande. Celokupni docker-compose fajl je prikazan na slici 1.

```

1  services:
2    elasticsearch:
3      image: docker.elastic.co/elasticsearch/elasticsearch:7.4.0
4      container_name: elasticsearch
5      user: elasticsearch
6      command: >
7        /bin/sh -c "./bin/elasticsearch-plugin install
8        file:///serbian-analyzer/serbian-analyzer.zip;
9        /usr/local/bin/docker-entrypoint.sh"
10     environment:
11       - discovery.type=single-node
12       - node.name=udd
13     ports:
14       - 9200:9200
15       - 9300:9300
16     networks:
17       - elastic
18     volumes:
19       - ./serbian-analyzer:/serbian-analyzer
20
21   kibana:
22     image: docker.elastic.co/kibana/kibana:7.4.0
23     environment:
24       - ELASTICSEARCH_HOSTS=http://elasticsearch:9200
25     ports:
26       - 5601:5601
27     networks:
28       - elastic
29
30   logstash:
31     image: docker.elastic.co/logstash/logstash:7.4.0
32     networks:
33       - elastic
34
35   filebeat:
36     image: docker.elastic.co/beats/filebeat:7.4.0
37     environment:
38       - setup.kibana.host=kibana:5601
39       - output.elasticsearch.hosts=["elasticsearch:9200"]
40     networks:
41       - elastic
42
43   networks:
44     elastic: null

```

Slika 1: docker-compose.yml