



RAZVOJ SKLADIŠTENJA I PRETRAGE DIGITALNE BIBLIOTEKE DIPLOMSKIH  
RADOVA UPOTREBOM MongoDB, Elasticsearch I Blockchain TEHNOLOGIJA

DEVELOPMENT OF DATA STORAGE AND RETRIEVAL OF DIGITAL LIBRARY  
APPLICATION USING MongoDB, Elasticsearch AND Blockchain TECHNOLOGIES

Zlatan Prečanica, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

**Kratak sadržaj:** *U radu je implementirana aplikacija digitalne biblioteke studentskih diplomskih radova. Svaki diplomski rad predstavlja digitalni dokument, te se projekat bavi metodama skladištenja i pretrage diplomskih dokumenata. U te svrhe su korišćene MongoDB, Elasticsearch i Blockchain tehnologije.*

**Keywords:** *Java, SpringBoot, MongoDB, ElasticSearch, Blockchain*

**Abstract:** *This thesis deals with the development of a digital library application. The digital library is used to store and retrieve student's thesis. Each thesis has been presented as a digital document. For storage and search MongoDB, Elasticsearch and Blockchain technologies are used.*

**Ključne reči:** *Java, SpringBoot, MongoDB, ElasticSearch, Blockchain*

## 1. UVOD

Biblioteka je sredena zbirka knjiga po autorskom, predmetnom ili nekom drugom katalogu, koja pripada privatnim licima, državnim institucijama ili ustanovama u kojima se čuvaju zbirke knjiga.

Prve biblioteke nastale su pre više hiljada godina, a prve knjige u tim bibliotekama su bile glinene pločice ispisane klinastim pismom. Nove tehnologije su uticale na promenu paradigme bibliotekara pri čemu su bibliotekari uočili dve važne činjenice: da je današnje digitalno okruženje, kao što su internet i razne društvene mreže, postalo konkurencija bibliotekama i da je bibliotekama u 21. veku osigurana budućnost samo ukoliko uspeju da se prilagode novim tehnologijama.

Digitalna biblioteka je biblioteka u kojoj se čuvaju knjige u digitalnom formatu, a svaka knjiga koja se nalazi u digitalnoj biblioteci dostupna je putem računara i interneta.

Tema ovog rada jeste implementacija jedne takve digitalne biblioteke koja sadrži bazu studentskih završnih (diplomskih) radova.

## 2. KORIŠĆENE TEHNOLOGIJE I TERMINOLOGIJE

**JAVA** [1] - objektno-orijentisani programski jezik koji je razvila kompanija Sun Microsystems početkom 1990-ih godina. Dizajnirana je tako da što manje zavisi od specifičnih karakteristika konkretnog računarskog sistema i njegove arhitekture. Ono što Javu izdvaja od ostalih programskih jezika jeste to što se jednom kompajliran kod može pokrenuti na svim platformama koje podržavaju Javu. Java se izvršava korišćenjem Java virtualne mašine (JVM eng. *Java Virtual Machine*), apstraktne mašine koja postoji samo u memoriji i koja izvorni kod izvršava na računaru.

**SpringBoot** [2] - je Java radni okvir koji se koristi za kreiranje *mikro servisa*. Mikroservisi su tehnički razvoja softvera koja predstavlja varijantu servisno-orijentisanog arhitekturnog (SOA) stila koji strukturira aplikaciju kao kolekciju labavo povezanih servisa. U mikroservis arhitekturi, servisi su sitno granulirani, a protokoli su jednostavni.

**MongoDB** [3] - je vodeća *NoSQL* baza podataka. U skladu sa rastućim interesovanjem za alternativne sisteme za upravljanje bazama podataka, koji se razlikuju od tradicionalnih relacionih baza podataka, pojavio se koncept takozvanih *NoSQL* baza podataka, koje ne koriste *SQL* za povezivanje, nerelacione su, distribuirane, otvorenog koda i horizontalno skalabilne. *MongoDB* je napisan u C++ programskom jeziku i otvorenog je koda, izdata pod kombinacijom *GNU Affero General Public Licence* i *Apache Licence*. *MongoDB* čuva podatke kao *JSON* dokumente sa dinamičkim šemama. *JSON* (*JavaScript Object Notation*) je otvoreni standard zasnovan na tekstu, osmišljen za razmenu odataka koji su pogodni za čitanje ljudima.

**ElasticSearch** [4] - je server koji služi za pretragu i analizu podataka. Napisan je u Java programskom jeziku što omogućava pokretanje na svim platformama. Baziran je na *Lucene* indeksima i omogućava korisnicima da pretraže veliku količinu podataka vrlo brzo. *Lucene* indeksi imaju dodatne strukture podataka koje sadrže jednu ili više kolona tabele i kod kojih su podaci sortirani. Sortirani podaci se mogu mnogo brže pretražiti pomoću struktura kao što je B stablo. Indeksi koji se koriste u relacionim bazama podataka uglavnom koriste liste i stabla za pretragu, dok se *Lucene* indeks razlikuje po tome što koristi nizove.

## NAPOMENA:

Ovaj rad je proistekao iz master rada čiji mentor je bio dr Milan Vidaković.

**Blockchain** [8] - je baza podataka koja se ne nalazi na jednom mestu, već je čine manje baze (blokovi) koji su međusobno digitalno povezani, a koji sadrže informacije o digitalnim transakcijama bilo koje vrste (od vlasničkih listova, preko podataka iz knjige rođenih, do ugovora kojim se regulišu autorska prava). Za razliku od klasične *online* baze podataka, *blockchain* tehnologija omogućava komunikaciju sa nekoliko računara (servera) između kojih se transakcija obavlja. Svako obavljenom transakciji pripisuje se vremenska odrednica, dok se sama transakcija po hronološkom redu dodaje već postojećem lancu blokova. Taj lanac predstavlja niz blokova koji sadrže podatke o svakoj transakciji koja se u okviru nje izvršila. Kako je svaki blok logički povezan sa drugim, *blockchain* je prevazišao ono što ljudski rod nije uspeo od svog nastanka – postao je imun na prevaru.

### Frontend tehnologije

Za implementaciju *frontend* dela aplikacije digitalne biblioteke korišćene su *JavaScript* [5], *jQuery* [6] i *Ajax* [7] tehnologije. Zahvaljujući *jQuery* i *Ajax* tehnologijama postignuta je “*single page*” arhitektura aplikacije. To znači da prilikom prelaska sa jedne funkcionalnosti na drugu, ili prilikom pritiska na bilo koje dugme na stranici, korisnik ostaje na istoj stranici na kojoj je i bio, ne postoji druga stranica koja se učitava, već se sve dešava na jednoj. Zahvaljujući *jQuery* biblioteci omogućeno je skrivanje i prikazivanje elemenata koji u datom kontekstu treba da budu prikazani ili sakriveni dok je glavna prednost *Ajax* biblioteke to što omogućava asinhronu pozive ka *backend-u* odnosno ka *REST API-ju* aplikacije.

### 3. SPECIFIKACIJA APLIKACIJE

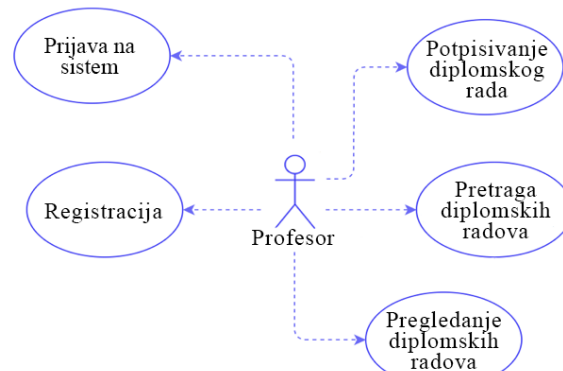
Zadatak obuhvata izradu digitalne biblioteke diplomskih radova. Ideja projekta je da se prikaže funkcionisanje jednog takvog sistema. Postoje dva tipa korisnika, studenti i profesori. Oba tipa korisnika imaju mogućnost registrovanja i prijave na sistem kao i pretraživanja i pregledanja diplomskih radova koji se nalaze u sistemu. Studenti pored ovih mogućnosti imaju i mogućnost dodavanja novih diplomskih radova, dok profesori imaju mogućnost digitalnog potpisivanja radova (Slika 1: Dijagram slučajeva korišćenja profesora). Sistem je zasnovan na tehnologiji mikroservisa i sastoji se od dve aplikacije i tri velike celine.

Prva celina obuhvata *backend* koji je zasnovan na *Java* programskom jeziku i *SpringBoot* radnom okviru za kreiranje web servisa koji ima mogućnost rada sa bazom podataka, indeksiranju dokumenata. Dokumenti (diplomski radovi) se čuvaju u *MongoDB* bazi podataka. Pored čuvanja, dokumenti se i indeksiraju zbog bržeg i efikasnijeg pretraživanja.

U ove svrhe se koristi *Elasticsearch* servis. Druga celina obuhvata *frontend* deo koji ima za zadatak da obezbedi korisnicima lako i efikasno pristupanje sadržaju. Ovaj deo aplikacije je zasnovan na *JavaScript*, *jQuery* i *Ajax* tehnologijama.

Treća i poslednja celina je zasebna aplikacija (mikro servis), koja predstavlja *Blockchain* servis. Ovaj servis je takođe zasnovan na *Java* programskom jeziku i *SpringBoot* radnom okviru. Glavna uloga ovog servisa

jeste da vodi računa o svakoj transakciji (dodavanju novog dokumenta, potpisivanju dokumenta...) i sprečava bilo koju promenu podataka o obavljenim transakcijama, uključujući njihovo brisanje, dodavanje i bilo koji drugi vid falsifikovanja. Ovaj servis takođe ima podršku za rad sa *MongoDB* bazom podataka. Oba mikro servisa imaju podršku za *REST API endpoint-e* preko kojih ove dve aplikacije i komuniciraju.



Slika 1: Dijagram slučajeva korišćenja profesora

### 4. IMPLEMENTACIJA

Projekat se sastoji od dva mikroservisa koji su u osnovi dve *SpringBoot* aplikacije. To su *library* i *blockchain* servisi. Prvi servis predstavlja samu digitalnu biblioteku i sadrži operacije za rad sa korisnicima i digitalnim dokumentima, njihovim čuvanjem, pretraživanjem kao i potpisivanjem. Drugi servis predstavlja, kao što mu i sam naziv kaže, *blockchain* deo koji se koristi za perzistenciju informacija i koji garantuje nepromenljivost digitalnih dokumenata. Oba servisa su zasnovana na *Java* programskom jeziku i imaju podršku za *REST API* servis.

#### SpringBoot projekat

Prilikom kreiranja novog *SpringBoot* projekta koristi se *online* koji se nalazi direktno na oficijalnom sajtu *SpringBoot* radnog okruženja (<https://start.spring.io/>) i pruža veoma brzo i efikasno kreiranje novog projekta zasnovanog na ovom radnom okviru.

#### Konfiguracija projekta

Početna konfiguracija projekta vrši se u *application.properties* fajlu. U njemu se podešavaju parametri aplikacije kao što su port na kome će se aplikacija pokretati, kredencijali za bazu podataka, naziv *elastic-search* klastera kao i host i port *elasticsearch* servisa na kome se nalazi.

#### Indeksiranje dokumenata

Indeksiranje dokumenata je odrađeno kao dio procesa čuvanja novih dokumenata sistemu. Pošto *SpringBoot* radni okvir pruža odličnu podršku za rad sa *elasticsearch* servisom, procedura je slična kao i za rad sa običnom bazom podataka. Prvo je potrebno kreirati model podataka koji će se čuvati, a zatim i repozitorijum koji će vršiti indeksiranje prosleđenog modela. Pošto se *elasticsearch* repozitorijumu prosleđuje model dokumenta koji će se indeksirati, prilikom prvog pokretanja aplikacije u *elastic-search* servisu će se kreirati indeks koji će imati polja

ekvivalentna poljima modela koji će se u njih upisivati. Primer indeksa koji će kreirati *elasticsearch* servis je prikazana na slici 2.

```
1 {
2   "article" : {
3     "aliases" : { },
4     "mappings" : {
5       "default" : {
6         "properties" : {
7           "abstract" : {
8             "type" : "text",
9             "fields" : {
10              "keyword" : {
11                "type" : "keyword",
12                "ignore_above" : 256
13              }
14            }
15          },
16          "areaCode" : {=
38          "author" : {=
105          "committeeMembers" : {=
199          "filename" : {=
208          "keywords" : {=
217          "status" : {=
226          "title" : {=
235        }
236      }
237    },
238    "settings" : {
239      "index" : {
240        "creation_date" : "1552997028653",
241        "number_of_shards" : "5",
242        "number_of_replicas" : "1",
243        "uuid" : "rDUChEvRmGax23auFiORw",
244        "version" : {
245          "created" : "5060499"
246        },
247        "provided_name" : "article"
248      }
249    }
250  }
251 }
```

Slika 2: ElasticArticle indeks

Za indeks "article" se definišu polja po kojima će se vršiti indeksiranje, a kasnije pretraga. Za svako polje se definiše njegov tip. Pored polja indeks sadrži i polje "settings" u kome se nalaze neki osnovni podaci o indeksu kao što je datum kreiranja, naziv, uuid (*Universally Unique Identifier*) itd.

### Rad sa bazom

Za rad sa *MongoDB* bazom se koristi *Hibernate* okvir koji je sastavni deo *SpringBoot* okruženja. Nakon što su kreirani modeli podataka koji će biti čuvani u bazi podataka, potrebno je kreirati repozitorijume i servise koji će da vrše osnovne *CRUD* (*Create Read Update Delete*) operacije nad podacima. Pošto su osnovne *CRUD* operacije već definisane u *Hibernate* okviru, njih nije potrebno ponovo definisati, već se definišu samo one operacije koje su specifične za dati model.

Na primer, ukoliko je potrebno kreirati servis koji vraća samo jednog korisnika sa specifičnim email-om ili nekim drugim atributom iz baze podataka, prvo je neophodno kreirati repozitorijum koji će u bazi pronalaziti korisnika sa specifičnim atributom.

Nakon toga je potrebno kreirati servis koji preko prethodno kreiranog repozitorijuma komunicira sa bazom. Servis predstavlja glavnu vezu između kontrolera web servisa i samog *MongoDB API*-ja.

### Komunikacija sa Blockchain aplikacijom

Pri dodavanju novog dokumenta u biblioteku, pored čuvanja u *MongoDB* bazu i indeksiranja u *Elasticsearch* servis, vrši se i čuvanje u *Blockchain* bloku. Za to je zadužen zaseban blockchain mikroservis. Ovaj servis vodi

računa o svim blokovima i čuva ih u bazi podataka. Aplikacija digitalne biblioteke sa ovom aplikacijom komunicira pomoću *RestTemplate*-a koji je ugrađen u *SpringBoot* okvir i omogućuje komunikaciju sa *REST API endpoint*-ima drugih aplikacija.

### Implementacija pretrage

Metoda prima svih šest parametara po kojima se vrši indeksiranje dokumenata kao i tip pretrage koji predstavlja logičku operaciju. Ti parametri stižu sa *frontend*-a i šalje ih korisnik koji vrši pretraživanje. Logička operacija može da bude logičko I (*AND*) ili logičko ILI (*OR*).

Ukoliko je između polja pretrage operacija logičko I, zahteva se da dokumenti budu filtrirani po svim unesenim parametrima i da samo dokumenti koji zadovoljavaju sve parametre budu vraćeni korisniku kao rezultat pretrage. Ukoliko je operacija logičko ILI, u tom slučaju rezultat su svi dokumenti koji zadovoljavaju uslov bar jednog polja.

### Digitalno potpisivanje dokumenata

Ukoliko profesor odluči da potpiše određene radove, potrebno je da prethodno selektuje one koje želi da potpiše. Nakon toga potreban je unos PIN koda. PIN kod je izgenerisan prilikom registracije na sistem upotrebom posebnog algoritma od privatnog ključa korisnika. Ukoliko je PIN kod dobar, izabrani radovi se potpisuju. Za svrhe digitalnog potpisivanja se koriste klase iz *java.security* paketa.

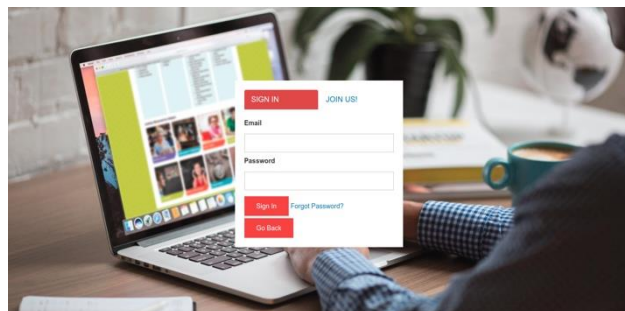
Prvi korak je učitavanje privatnog i javnog ključa korisnika. Nakon toga se učitava PDF dokument. Treći korak je potpisivanje PDF fajla koristeći metode *JAVA API*-ja i privatni ključ korisnika.

Četvrti i poslednji korak je čuvanje tako potpisanog fajla u bazi podataka. Provera da li je fajl uspešno potpisan se vrši pomoću javnog ključa.

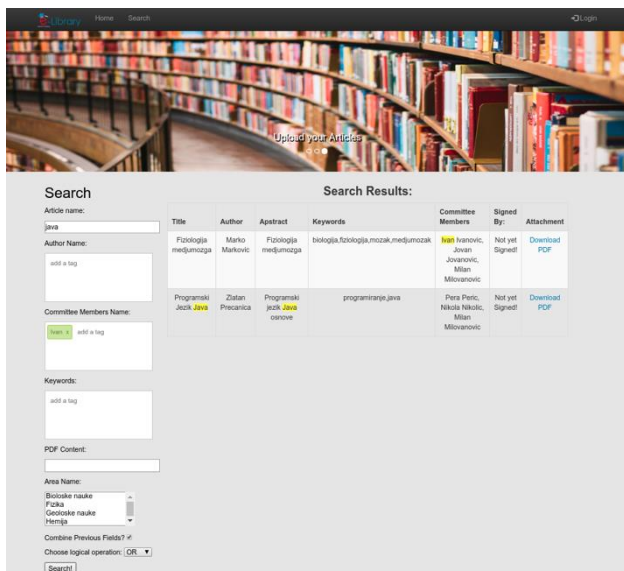
## 5. DEMONSTRACIJA

Stranica za registraciju i prijavu na sistem (*Slika 3: Početna stranica*) je prva stranica koju korisnik vidi. Nakon uspešnog prijavljivanja na sistem korisniku je prikazan spisak svih dokumenata u digitalnoj biblioteci. Ukoliko je prijavljen korisnik tipa student, njemu se nudi opcija dodavanja novog dokumenta.

Ukoliko je na sistem prijavljen korisnik tipa profesor i ukoliko ima nepotpisanih radova njemu se šalje obaveštenje o pristiglim radovima koji čekaju da budu potpisani. I student i profesor imaju mogućnost pretrage diplomskih radova. Rezultat kombinovane pretrage je prikazan na slici 4.



Slika 3: Početna stranica



Slika 4: Rezultat pretrage

## 6. ZAKLJUČAK

Ovaj rad nastoji da ukaže na bogatstvo elektronskih servisa koje nude savremene biblioteke, kao i na potrebu da se pri kreiranju digitalne biblioteke rukovodimo merilima efikasnosti. Zadatak ovog rada je bio kreiranje digitalne biblioteke u kojoj se čuvaju studentski završni (diplomski) radovi. Implementacija se u najvećoj meri bavi pitanjima čuvanja i pretrage digitalnih dokumenata, te su u te svrhe korišćeni *MongoDB* i *Elasticsearch* mehanizmi.

Pored implementacije digitalne biblioteke zadatak je bio i istraživanje blockchain tehnologija kao i pronalaženje načina upotrebe blockchain servisa u unapređenju jednog bibliotekačkog informacionog sistema. Što se tiče daljeg istraživanja na ovu temu, prvenstveno je bitno pratiti razvoj informacionih sistema digitalnih biblioteka i mehanizama za njihovu implementaciju.

## 7. LITERATURA

- [1] Java, <https://www.java.com/en/download/>
- [2] SpringBoot, <https://spring.io/projects/spring-boot>
- [3] MongoDB, <https://www.mongodb.com/>
- [4] Elasticsearch, <https://www.elastic.co/>
- [5] JavaScript, <https://www.javascript.com/>
- [6] jQuery, <https://jquery.com/>
- [7] Ajax, [https://en.wikipedia.org/wiki/Ajax\\_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))
- [8] Blockchain, <https://www.blockchain.com/>
- [9] Wikipedia, <https://www.wikipedia.org/>

### Kratka biografija:

**Zlatan Prečanica** rođen je 7.5.1994, Glina, Hrvatska. Završio osnovnu školu “Boško Palkovljević Pinki” u Sremskoj Mitrovici, srednju elektrotehničku školu “Nikola Tesla”, takođe u Sremskoj Mitrovici. Nakon toga upisao se na Fakultet Tehničkih Nauka u Novom Sadu, odsek računarstvo i automatika.

**Milan Vidaković** rođen je u Novom Sadu 1971. godine. Na Fakultetu tehničkih nauka u Novom Sadu završio je doktorske studije 2003. godine. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti *Primenjene računarske nauke i informatika*.