

Razvoj polustrukturirane baze podataka za potrebe internetskog foruma kao web-aplikacije

Dario, Bogović

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:220033>

Rights / Prava: [Attribution 3.0 Unported](#)/[Imenovanje 3.0](#)

Download date / Datum preuzimanja: **2024-04-19**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Dario Bogović

**RAZVOJ POLUSTRUKTURIRANE BAZE
PODATAKA ZA POTREBE INTERNETSKOG
FORUMA KAO WEB-APLIKACIJE**

DIPLOMSKI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N

Dario Bogović

Matični broj: 47165

Studij: Informacijsko i programsko inženjerstvo

**RAZVOJ POLUSTRUKTURIRANE BAZE PODATAKA ZA POTREBE
INTERNETSKOG FORUMA KAO WEB-APLIKACIJE**

DIPLOMSKI RAD

Mentor :

Dr. sc. Bogdan Okreša Đurić

Varaždin, rujan 2020.

Dario Bogović

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvatanjem odredbi u sustavu FOI-radovi

Sažetak

Rad se sastoji od teorijskog i praktičnog dijela. U teorijskom dijelu opisan je polustrukturirani model podataka, definirane razlike u odnosu na strukturirane i nestrukturirane podatke te predstavljeni tipovi polustrukturiranih podataka, s naglaskom na XML. Slijedi opis i definiranje XML baze podataka te pregled sustava za upravljanje XML bazom podataka. Praktični dio rada sadrži prikaz rada s noSQL dokumentnom bazom podataka i programskom platformom eXist-db, uz izradu konkretne XML baze podataka. Načini pristupa XML bazi podataka, slanje upita i dohvaćanje podataka prikazani su na primjeru Java web-aplikacije za internetski forum.

Ključne riječi: xml; polustrukturirani podaci; XML baza podataka; eXist-db

Sadržaj

1. Uvod	1
2. Podaci	2
2.1. Vrste podataka	3
2.1.1. Strukturirani podaci	3
2.1.2. Nestrukturirani podaci	4
2.1.3. Polustrukturirani podaci	4
3. XML	6
3.1. XPath	9
3.2. XQuery	11
4. XML baza podataka	13
4.1. eXist-db	14
5. Opis aplikacijske domene	18
5.1. Internetski forum	18
5.2. Diskuto	20
6. Model baze podataka	22
6.1. Korisnici (users.xml)	22
6.2. Forumi (forums.xml)	23
6.3. Objave (posts.xml)	24
6.4. Poruke (messages.xml)	26

7. Implementacija	27
7.1. Kreiranje i rad s bazom podataka	28
7.2. Java web-aplikacija	33
8. Primjeri korištenja	39
8.1. Početna stranica	39
8.1.1. Početna stranica prijavljenog korisnika	40
8.2. Registracija	41
8.2.1. Potvrda registracije	42
8.3. Prijava	42
8.4. Pronalaženje Diskuta	44
8.5. Kreiranje novog Diskuta	45
8.6. Pregled Diskuta	46
8.7. Kreiranje nove objave	46
8.8. Ovlasti moderatora nad pojedinim Diskutom	48
8.9. Profil korisnika	49
8.10. Poruke	50
8.11. Pretraga	51
8.12. Postavke	52
9. Zaključak	54
Popis literature	56
Popis slika	58
Popis programskih isječaka	59

1. Uvod

Razvojem Interneta i povećanjem podataka koji su se počeli razmjenjivati na njemu javlja se potreba za efikasnijim zapisom i pohranom podataka. Dolazi do pojave novih modela podataka kao što su polustrukturirani podaci preko kojih razmjena podataka preko Interneta postaje jednostavnija i efikasnija.

Ovaj rad se sastoji od teorijskog i praktičnog dijela. U teorijskom dijelu će se definirati značenje podatka i objasniti razlike u različitim vrstama podataka prema obliku u kojima ih pohranjujemo. Nad svakom vrstom podataka obrazložit će se prednosti i nedostaci. Fokus ovog rada prelazi na polustrukturirane podatke gdje će se pritom predstaviti njen najslavniji predstavnik: XML (Extensible Markup Language). On rješava sve zahtjeve koje polustrukturirani model podataka zahtjeva: omogućava laku čitljivost podataka za čovjeka i računalo i naglašava jednostavnost, općenitost i upotrebljivost u razmjeni podataka. U ovom radu objasnit će se pravila sintakse XML-a i dati teorijska podloga u radu sa standardima XPath i XQuery za odabir, dohvaćanje i transformaciju XML podataka. Definirat će se i njihova sintaksa, te opisati mnoge funkcije koje oni podržavaju. Polustrukturirani podaci se mogu pohraniti u razne NoSQL baze podataka, a ovaj rad detaljnije donosi pregled XML baza podataka. Opisat će se njezine vrste i objasniti način kako treba pristupati u odabiru određene vrste XML baze podataka. Dati će se pregled na neke od najpoznatijih sustava za upravljanje XML bazom podataka. Jedan od najznačajnijih predstavnika je eXist-db koji je besplatan i otvoren sustav za upravljanje XML bazom podataka. U radu će se opisati instalacija i načini njegovog korištenja.

Kao praktični dio rada će se razviti polustrukturirana baza podataka na programskoj platformi eXist-db u kojoj će se podaci pohranjivati u XML formatu koje će koristiti mrežna aplikacija izrađena u programskom jeziku Java. Praktični dio aplikacije se zove "Diskuto", a namjena aplikacije je internetski forum. Ona omogućava komunikaciju između više ljudi (korisnika) oko neke određene teme. U radu će se definirati pojam internetskog foruma, te će se opisati funkcionalnosti izrađene aplikacije. Predstavit će se glavni koncepti prepoznati u domeni aplikacije i detaljizirati njihov sastav preko XML Scheme. Slijedi detaljan opis izrade polustrukturirane baze podataka u eXist-db, te opis načina preko kojeg Java web-aplikacija može ostvariti komunikaciju prema njoj. Opisat će se metode za dohvaćanje čistih podataka pohranjenih u XML formatu. Na kraju slijedi opis strukture implementirane Java web-aplikacije uz prikaz izrađenih funkcionalnosti uz odgovarajuće slike zaslona i njihov opis.

2. Podaci

Podatak [1] je skup prepoznatljivih znakova zapisanih na mediju koji može biti analogni (papira, ploče, fotografije i sl.) ili digitalni (magnetski disk, SSD, USB disk, CD-ROM i sl.). Sa njime zapisujemo određenu činjenicu ili dio činjenice o stanju u realnom sustavu (stvarnom svijetu). Podatak se može promatrati kao simbol sa kojim pokušavamo "uhvatiti" pravu sliku o stvarnom događaju. Oni se bilježe i zapisuju na onaj način koji im je primjeren i koji im odgovara jer ga moramo znati pročitati i intepretirati. Možemo ga zabilježiti tekстом, brojevima, bojom ili zvukom. Osnovna obilježja podatka koja ga čine korisnim su [2]: relevantnost za određenu svrhu, potpunost, točnost, pravovremenost, odgovarajući oblik (format) i dostupnost za prihvatljivu cijenu.

Nakon što podatak radi neke svrhe interpretiramo tada dobivamo informaciju, odnosno obavijest. Informacija [1] je skup obrađenih podataka koji ljudima predstavlja obavijest, novost i uvećava njihovo znanje. Iako se pojmovi informacija i podatak često smatraju sinonimima, njihovo značenje se bitno razlikuje. Za razliku od podatka, informacija donosi neku smislenu novost ili obavijest. Može odgovoriti na pitanja poput "Tko? Što? Gdje? Kada? Koliko? Zašto?". Informacija se zbog toga smatra temeljnim elementom za donošenje odluka. Obradom, organiziranjem i primjenom informacija se stvara znanje. Sa njime se može odgovoriti na pitanje "Kako?". Kontekstualizirano je, što znači da osim informacija potrebno je imati i kontekst kako bi se proizvele određene akcije. Znanje je nužno pri odlučivanju jer ono "zna" kako koristiti informaciju. [2] Na slici 1 je prikazan dijagram koji prikazuje odnos između podatka, informacija i znanja.



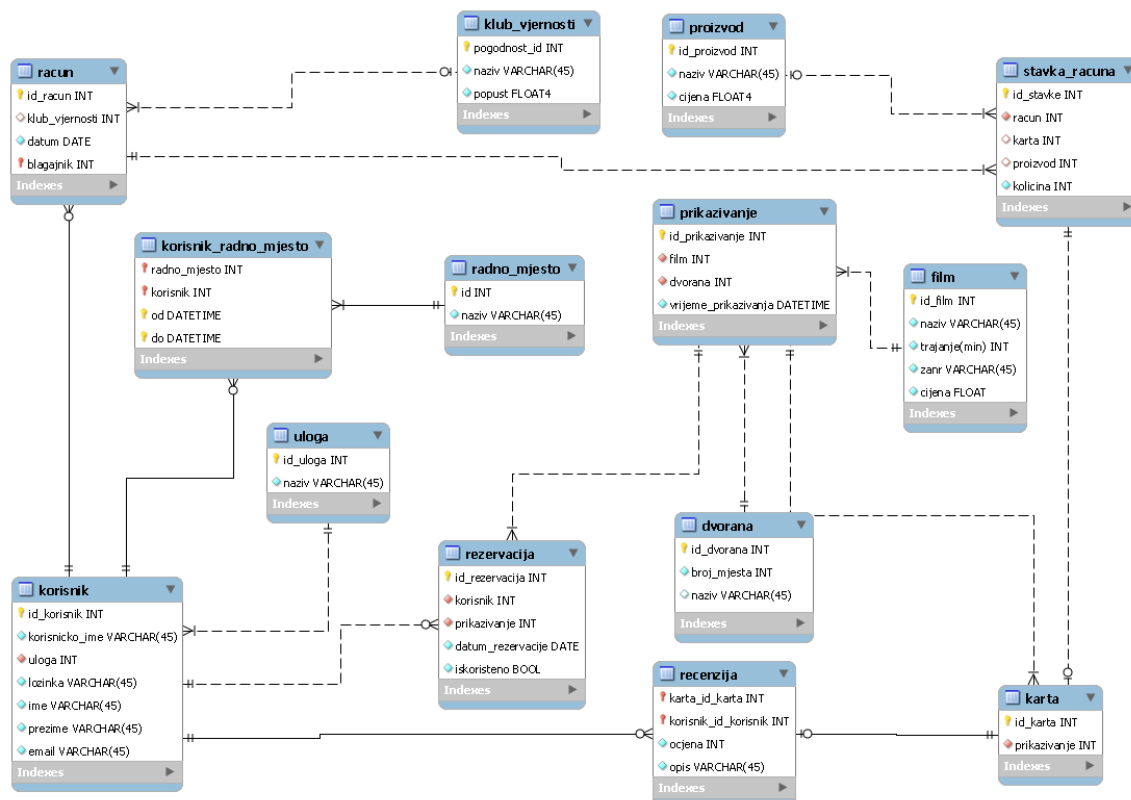
Slika 1: Odnos podatka, informacije i znanja [3]

2.1. Vrste podataka

Prema obliku u kojem pohranjujemo podatke, možemo ih podijeliti u tri kategorije [4]: strukturirani, polustrukturirani i nestrukturirani podaci. Važno je znati oblik podatka kojeg pohranjujemo kako bi znali koja baza podataka je potrebna za pohranu podataka. U nastavku ovoga poglavlja objašnjena je svaka od nabrojanih kategorija.

2.1.1. Strukturirani podaci

Strukturirani podaci su podaci koji imaju određeni strukturu, odnosno podaci se spremaju prema unaprijed definiranoj shemi. Prije nego se podatak kreira, potrebno je dizajnirati bazu podataka i odrediti shemu kako bi se prema njoj mogli zapisivati podaci. Glavni nedostatak ovakvog tipa podataka je loša fleksibilnost u zapisu informacija. Primjerice, kada bi htjeli jednom redu u relaciji pridružiti još jedan atribut morali bi promijeniti shemu i dodati novi stupac. To može biti izrazito neefikasno za tablice koje sadrže više tisuće redova kojima uopće ni ne treba taj atribut. Glavna prednost ovakvog tipa podataka je što, sa vrlo dobrim performansama, lako je moguće raditi strukturirane upite i vršiti spajanja nad podacima. Najlakši je za pretragu i organizaciju jer se sastoji od redova i stupaca u kojima su pohranjeni podaci. Poslovni korisnici ga lako koriste jer ne treba detaljno razumijeti različite vrste podataka i odnose među njima. Ranjivosti su vrlo rijetke zbog toga što su uslijed niza godina istraživanja, razvoja i poboljšanja one eliminirane. Zbog toga je ovakav tip pohranjivanja podataka danas široko korišten. [5]



Slika 2: Primjer modela baze koja koristi strukturirane podatke

Klasičan primjer sustava koji zapisuje strukturirane podatke je relacijska baza podataka. Podaci se nalaze strukturirani u relacijama, odnosno tablicama od kojih se baza podataka sastoji. Relacijska shema baze podataka se sastoji od naziva relacija i popisa atributa. Relacija ili tablica se sastoji od redaka i stupaca (atributa). Atribut, odnosno skup atributa čijim se podacima može jednoznačno identificirati svaki redak naziva se primarnim ključem. [6] Na slici 2 je prikazan primjer modela relacijske baze podataka koja koristi strukturirane podatke. Na slici je prikazan ERA model baze podataka za aplikaciju koja je podrška u poslovanju kina. Baza se sastoji od 14 relacija u kojima se podaci spremaju strukturirano prema zadanim atributima.

2.1.2. Nestrukturirani podaci

Nestrukturirani podaci su podaci koji nemaju određenu strukturu, odnosno nije ih moguće spremati u obliku redova i stupaca u relacijskim bazama. Čine najveći udio od svih podataka u svijetu. Primjer nestrukturiranih podataka su: binarni podaci, tekstualne datoteke, video datoteke, slikovne datoteke i slično. Njihova glavna prednost je što ih nije potrebno klasificirati, ali to ponekada može biti problem u upravljanju sa njima. Kako bi se iskoristili, potrebni su specijalizirani alati i stručnosti u radu sa podacima. Korištenje nestrukturiranih podataka zahtijeva njihovo razumijevanje i razumijevanje kako se podaci mogu povezati kako bi bili korisni. [5]

Treba napomenuti da se podaci smatraju nestrukturiranim ako nemaju shemu koja utječe na to kako se podaci pohranjuju ili im pristupaju. Nestrukturirani podaci mogu imati strukturu koja nije relevantna za način pohrane. Na primjer, prirodni jezik je visoko strukturiran prema sintaksičkim pravilima jezika. Audio i video datoteke mogu imati unutarnji format koji uključuje metapodatke kao i sadržaj. Ovdje postoji struktura unutar datoteke, ali sustavi za pohranu ne koriste tu strukturu i to je razlog zašto ova vrsta podataka se klasificira kao nestrukturirana. [4]

2.1.3. Polustrukturirani podaci

Polustrukturirani podaci ne zahtijevaju strukturu i predefiniranu shemu što ne znači da ona nije moguća, već je samo opcionalna. Drugi podaci će i dalje neometano postojati ako se shema promijeni ili ako se unese podatak sa novim atributima ili tipom podatka. Podaci se ne mogu pohraniti u obliku redaka i stupaca kao kod strukturiranih podataka. Polustrukturirani podaci sadrže oznake i elemente (metapodatke) koji se mogu koristiti za grupiranje podataka i opisivanje kako se pohranjuju. Na taj način se slični entiteti grupiraju i organiziraju u hijerarhiji. Entiteti u istoj grupi mogu ili ne moraju imati iste attribute i svojstva. Veličina i vrsta istih atributa u grupi se može razlikovati.

Najveća prednost polustrukturiranih podataka je laka prilagodba promjenama u strukturi i fleksibilnost u pogledu razmjene podataka. Podaci su prenosivi i nisu ograničeni fiksnom shemom. Nedostaci su to što je relativno nova tehnologija i zbog toga ovakav tip pohranjivanja podataka nije pretjerano raširen [5]. Podaci zbog nepravilne i djelomične strukture mogu otežavati interpretaciju odnosa između podataka. Razlika između sheme i podataka je vrlo nejasna što komplicira oblikovanje strukture podataka. Postoji i izrazita neefikasnost u izvršavanju upita

u usporedbi sa strukturiranim modelima podataka zbog specifične strukture zapisa podataka. Još jedan nedostatak je što je i cijena skladištenja viša u odnosu na strukturirane podatke. [7]

Najpoznatiji modeli podataka koji koriste polustrukturirane podatke su Extensible Markup Language (XML) i JavaScript Object Notation (JSON). XML omogućava definiranje XML sheme koju mogu obraditi programi bez smanjenja njihove upotrebljivosti za ljudske čitatelje. Omogućuje korisniku definiranje oznaka i atributa za pohranjivanje u hijerarhijskom obliku. JSON je standardni format koji koristi ljudski čitljiv tekst za prijenos podatkovnih objekata koji se sastoje od parova atribut i vrijednost. Koristi se prvenstveno za prijenos podataka između poslužitelja i mrežnih aplikacija kao alternativa XML-u. JSON i XML upravo najviše koriste web servisi. [8] U nastavku slijedi primjer JSON podatka:

```
{
  "rezervacija": {
    "broj": 43,
    "timestamp": "2020-07-21T14:26:59.125",
    "potvrđeno": true
  },
  "vlak": {
    "datum": "25/07/2020",
    "vrijeme": "09:30",
    "od": "Rijeka",
    "do": "Zagreb",
    "sjedalo": "54A"
  }
  "putnik": {
    "name": "Ivan Horvat",
    "status": "Umirovljenik",
  },
  "cijena": 60
}
```

Može se vidjeti kako u ovom tipu podatka postoji određena struktura, ali ona nije toliko striktna kao kod strukturiranih podataka. Konkretno kod JSON-a je važno da svaka vrijednost ima svoj naziv, da se podaci razdvajaju zarezom i da vitičaste zagrade moraju sadržavati jedan objekt. Tipovi podataka koje može poprimiti neka vrijednost su: tekst, broj, logička vrijednost, null, polje ili drugi objekt.

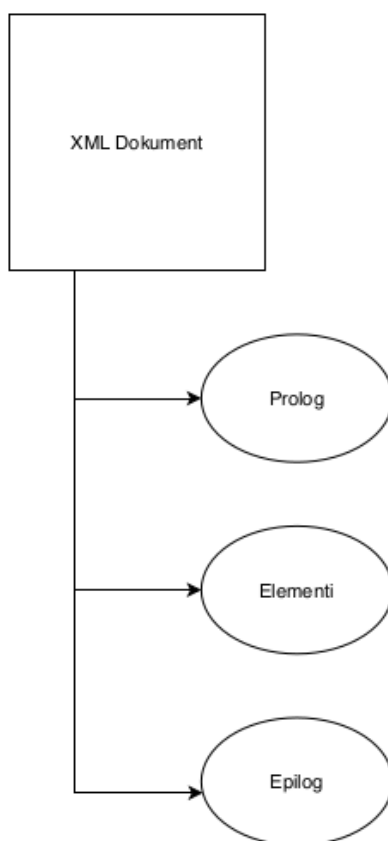
Za ovaj primjer jedne rezervacije karte za vlak se jasno vidi da ovaj objekt ima četiri vrijednosti:

- "rezervacija" - tip je objekt koji u sebi sadrži podatke o rezervaciji: broj rezervacije, datum i vrijeme obavljene rezervacije u tekstualnom formatu i logička vrijednost potvrda rezervacije.
- "vlak" - tip je objekt koji u sebi sadrži podatke o vožnji vlaka: datum polaska, vrijeme polaska, polazište, ishodište i naziv sjedala. Svi podaci su u tekstualnom formatu.
- "putnik" - tip je objekt koji u sebi sadrži podatke o putniku: ime i prezime i njegov status.
- "cijena" - cijena karte u brojčanom formatu

3. XML

XML [9] je primjer zapisa polustrukturiranih podataka koji se koristi za prikaz i razmjenu podataka preko Interneta. Najčešće se koristi još i kao baza podataka, zapisivanje metapodataka, upravljanje informacija i e-commerce.

XML dokument ima tri komponente: opcionalni prolog, obavezni korijenski element i neobavezni epilog koji dolaze poslije korijenskog dokumenta. Prolog sadrži XML deklaracije i reference na vanjski dokument za strukturiranje. Korijenski element je tijelo XML dokumenta i sadrži elemente i attribute. Ispod njega se može nalaziti epilog koji govore o tome kako treba postupati sa dokumentom.



Slika 3: Komponente XML dokumenta

Sintaksa XML-a je prilično jednostavna i sadrži samo nekoliko jednostavnih pravila koje treba slijediti. XML dokument može biti dobro oblikovan ili valjan. Dobro oblikovan dokument pridržava se brojnih ograničenja dobrog oblikovanja koja su detaljno opisana u XML specifikaciji. XML dokument je valjan ako prema XML specifikaciji ima pridruženu deklaraciju vrste dokumenta i ako je dokument u skladu s ograničenjima izraženima u njoj. Dobro oblikovan XML dokument započinje XML deklaracijom koja mora biti na početku dokumenta i može izgledati ovako:

```
<?xml version="1.0" encoding="EBCDIC" standalone="no" ?>
```

Postoje 2 verzije XML-a: 1.0 i 1.1 [10]. Prva XML verzija je kreirana 1998. godine., te i dalje je prihvaćena i preporučuje se za korištenje. Bazirana je na filozofiji da je zabranjeno sve što nije dozvoljeno. Druga verzija XML 1.1 je objavljena 2004. godine i sadrži svojstva koja olakšavaju rad sa programima na jačim računalima. Njezin pristup je da je dozvoljeno sve što nije zabranjeno. Potreba za novom verzijom XML-a se javila zbog razvoja drugih standarda na koje se XML oslanja. Prvenstveno je bio problem sa novim Unicode znakovima koje do tada XML 1.0 nije podržavao. Sa novom verzijom se omogućuje korištenje svih budućih Unicode znakova koji će se bilo kada u budućnosti definirati. U XML deklaraciji je potrebno u navodnicima navesti verziju (1.0 ili 1.1) u kojoj je dokument napisan. Nakon deklaracije verzije slijedi deklaracija kodiranja koja označava skup znakova koji se koristi za stvaranje dokumenta. Kao zadano, XML dokumenti kodirani su u Unicode-u, bilo 8-bitnom (UTF-8) ili 16-bitnom (UTF-16). Atribut kodiranja je neobavezan ako se koristi bilo koji od ovih kodiranja, ali se smatra dobrom praksom navesti je. Atribut "standalone" obavještava XML parser o tome oslanja li se dokument na podatke iz vanjskog izvora kao što je npr. DTD (document type definition). Zadana vrijednost je "no", a ako se postavi na "yes" XML parseru se poručuje da nema dokumenata sa vanjskih izvora.

XML dokument se sastoji od teksta (samih podataka) i oznaka (koje označavaju strukturu dokumenta u aplikaciji koja dokument obrađuje). Oznaka je konstrukcija koja sadržaje dokumenta odvaja lijevim i desnim kutnim zagradama (<>). Parovi oznaka koriste se za ograničavanje podataka. Element je podatkovni objekt kojeg je potrebno identificirati. Naziv elementa je obično opisno ime za podatkovni objekt kojeg element predstavlja. Svaki element u XML dokumentu mora početi s početnom oznakom i završiti sa komplementarnom završnom oznakom. Završna oznaka izgleda kao početna oznaka, osim što završna oznaka uključuje kosu crtu "/" prije naziva elementa. Podaci između početne i završne oznake su sadržaj elementa. Sadržaj može biti tekst ili čak i drugi element sa svojom vlastitom oznakom. U nastavku je prikazan primjer zadavanja elementa:

```
<element> information </element>
```

Početna oznaka je <element>. Sadržaj je "information" Završna oznaka je </element>. Sva tri dijela zajedno čine element. Nazivi elemenata moraju početi slovom ili podvlakom. Nakon prvog znaka slijedi nula ili više slova, brojeva, podvlaka ili crtica. [11] U nastavku je prikazan primjer kako bismo zapisali informacije o nekoj osobi:

```
<label>
<name> Jane Smith </name>
<address>
  <street> 270 Burlington Road </street>
  <city> Bedford </city>
  <state> MA 01730 </state>
</address>
</label>
```

U ovom primjeru može se vidjeti kako je informacija o nekoj osobi zapisana u elementu "label" koji sadrži oznaku "name" gdje je opisano ime osobe, te element "address" koji sadrži tri ugniježđena elementa: "street" koji opisuje ulicu, "city" koji opisuje grad i "state" koji opisuje

državu. Kod ugniježđenih elemenata je važno da dobro formirani dokumenti ima pravilnu strukturu stabla. Ona se postiže na način da završna oznaka elementa koja se nalazi unutar drugog elementa mora doći prije završne oznake roditelja.

U primjeru su korištene oznake za prikaz informacija. Umjesto oznaka i atributi mogu prikazivati informacije. Ponekad samo imenovanje elemenata nije dovoljno, pa atributi mogu pomoći sa dodatnim opisivanjem. Znak jednakosti razdvaja atribut od njegove vrijednosti i može i ne mora sadržavati razmak s jedne ili obje strane vrijednosti. Vrijednost atributa se obavezno mora navesti unutar navodnika, bilo jednostrukih ili dvostrukih. Ako vrijednost atributa mora sadržavati navodnike, moguće je vrijednost atributa navesti sa jednom vrstom navodnika, a podatak unutar atributa sa drugom vrstom. Iznimno, ako je potrebno imati istu vrstu navodnika u oba slučaja to se može postići sa specijalnim znakovima. Atribut se ne smije staviti na završnu oznaku elementa. Stoga prema objašnjenim pravilima, element "address" iz prošlog primjera može se prikazati kao element koji sadrži attribute "street", "city" i "state": [9]

```
<address  
street = "270 Burlington Road"  
city = "Bedford"  
state = "MA 01730"  
>
```

XML dokument mora imati jedinstveni korijenski element: element koji ima početnu oznaku na vrhu dokumenta, a krajnju oznaku na dnu dokumenta. Prva početna oznaka koja se napiše u dokumentu smatra se korijenskim elementom. Čim se nađe krajnja oznaka za element, dokument je završio. Nakon završetka korijenskog elementa ne smije se započeti sa pisanjem nekog drugog elementa jer je samo jedan element dopušten na razini korijena.

Svi XML elementi i nazivi atributa su osjetljivi na veliko i malo slovo, što znači da ako se element započne sa oznakom <element>, on se mora završiti sa oznakom </element>. Ispravan završetak elementa neće biti oznaka </ELEMENT>, </Element> ili neki drugi oblik jer je važno poštivati veliko/malo slovo. U XML-u je moguće zadati i prazne elemente koji se razlikuju od običnih elemenata u tome što nemaju nikakav sadržaj. Oznaka praznog elementa počinje sa nazivom oznake, a završava sa znakovima ">" koji moraju biti zajedno napisani bez razmaka. Prazan element može imati i attribute kao i običan element. U XML-u je moguće pisati i komentare koji mogu davati neke korisne informacije o dokumentu. Početak komentara se označava sa znakovima "<!--", a komentar završava sa znakovima "-->". Unutar komentara može se proizvoljno pisati jer će XML parser ignorirati sve unutar navedenih znakova. [11]

Ako netko želi stvoriti svoj jezik temeljen na XML-u može to učiniti na način da sam kreira sve oznake ili da kreira jedan dio oznaka, a ostale uzme iz nečijeg XML dokumenta. Kako bi se razlikovali elementi uzeti od tuđeg XML dokumenta stvoren je mehanizam dodijele jedinstvenih imena strukturama dokumenta, a naziva se XML Namespaces. Oni omogućavaju pridjeljivanje konteksta elementima, što omogućuje elementima da budu jedinstveni. Imenik je obavezno deklarirati u dokumentu prije nego se koristi.

3.1. XPath

XPath (XML Path Language) [12] je upitni jezik za odabir čvorova iz XML dokumenta. Zasnovan je na stablastoj strukturi XML dokumenta i omogućuje kretanje po stablu odabirom čvorova prema raznim kriterijima. Stvoren je od strane World Wide Web Consortium (W3C), iste grupacije koja je kreirala XML i standardizirani HTML. U uporabi je nekoliko verzija XPath-a: XPath 1.0 (objavljen 1999. godine i još uvijek najkorištenija verzija), XPath 2.0 (objavljen 2007. godine i omogućuje veći izbor tipova sustava), XPath 3.0 (objavljen 2014. godine i omogućuje implementaciju funkcija) i XPath 3.1. (objavljen 2017. godine i omogućuje nove tipove podataka i bolju podršku za JSON).

Svaki XPath se piše u obliku: "axis :: node-test [predicate]*" gdje axis predstavlja smjer navigacije unutar stabla XML dokumenta, node-test predstavlja putanju čvorova, a predikati (koji se nalaze u uglatim zagrada) služe za filtriranje čvorova prema određenom uvjetu. Kako bi se moglo raditi sa podacima u XML dokumentu, XPath omogućuje nekoliko tipova podataka koje se mogu proslijediti u XPath izraz:

- Brojčani tip podatka - u XPath izraz se može proslijediti broj, npr. izraz `"//planet[position()=7]"` će vratiti sedmi element `<planet>` u XML dokumentu. Postoje i brojčani operatori preko kojih se može dobiti brojčana vrijednost: `+`, `-`, `*`, `div` i `mod`. Postoji i nekoliko funkcija koje izraz vraćaju u brojčanu vrijednost: funkcija `"ceiling"` zaokružuje broj na prvi veći cijeli broj, funkcija `"floor"` zaokružuje broj na prvi manji cijeli broj, funkcija `"round"` zaokružuje broj na najbliži cijeli broj i funkcija `"sum"` vraća zbroj proslijeđenih brojeva.
- Tekstualni izraz - u XPath izraz se može proslijediti niz od nula ili više znakova gdje su znakovi prema zadanom u Unicode formatu, npr. izraz `"//planet[name='Mars']"` će vratiti sve elemente `<planet>` koji imaju dijete `<name>` čija je tekstualna vrijednost jednaka `"Mars"`. Postoji i nekoliko funkcija preko kojih se može manipulirati sa tekstualnim izrazom: funkcija `"concat"` spaja sve proslijeđene tekstualne izraze, funkcija `"contains"` vraća istinitu logičku vrijednost ako je tekstualni izraz jednak onome proslijeđenom, funkcija `"normalize-space"` briše razmake na početku i kraju izraza i pretvara više razmaka unutar izraza u jedan razmak, funkcija `"starts-with"` vraća istinitu logičku vrijednost ako tekstualni izraz započinje sa proslijeđenim izrazom, funkcija `"string-length"` vraća broj znakova u izrazu, funkcija `"substring"` vraća podniz nekog izraza, funkcija `"substring-after"` vraća podniz nakon proslijeđenog izraza, funkcija `"substring-before"` vraća podniz prije proslijeđenog izraza i funkcija `"translate"` vraća izraz sa svim pojavama znakova u drugom izrazu zamijenjenom odgovarajućim znakovima u trećem tekstualnom izrazu.
- Logička vrijednost - XPath izraz može biti logička vrijednost `true/false`, npr. izraz `"position()=3"` će vratiti `true` ako je pozicija jednaka trećem čvoru u elementu ili će vratiti `false` ako tvrdnja ne vrijedi. Logičkih operatori preko kojih se može dobiti logička vrijednost su: `!=`, `<`, `>` i `>=`. Postoji i nekoliko funkcija koje izraz vraćaju u logičku vrijednost: funkcija `"boolean"` pretvara argument u logičku vrijednost, funkcija `"false"` vraća lažnu vrijednost, funkcija `"true"` vraća istinitu vrijednost, funkcija `"lang"` ispituje je li jezik čvora isti jeziku proslijeđenom u funkciju, funkcija `"not"` mijenja logičku vrijednost njenog argumenta.

- Set čvorova - XPath izraz može filtrirati čvorove na način da vrati set od nula ili više čvorova prema zadanom izrazu, npr. izraz `"//planet[position() > 3]"` će vratiti sve elemente `<planet>` nakon prva tri elementa. Postoje i funkcije koje mogu raditi nad setom čvorova. Funkcija `"position"` vraća broj koji predstavlja poziciju čvoru u sekvenci čvorova koje se obrađuju i funkcija `"count"` vraća broj čvorova u danom setu čvorova.

Axis ili os se piše na početku XPath izraza i predstavlja smjer navigacije unutar stabla XML dokumenta. Predstavlja vezu između trenutnog čvora sa drugim čvorovima u dokumentu i koristi se za lociranje čvorova u odnosu na taj čvor u stablu. Postoji 13 osi koji se mogu zadavati u XPath izrazu:

- `ancestor` : odabire sve pretke (roditelje, roditelje roditelja, itd.) od trenutnog čvora
- `ancestor-or-self` : odabire trenutni čvor i sve pretke (roditelje, roditelje roditelja, itd.) od trenutnog čvora
- `attribute` : odabire se atribut od trenutnog čvora
- `child` : odabire svu djecu od trenutnog čvora
- `descendant` : odabire sve potomke (djecu, unuke, itd.) od trenutnog čvora
- `descendant-or-self` : odabire trenutni čvor i sve potomke (djecu, unuke, itd.) od trenutnog čvora
- `following` : odabire sve u XML dokumentu nakon završne oznake od trenutnog čvora
- `following-sibling` : odabire svu braću od trenutnog čvora
- `namespace` : odabire sve čvorove prostora imena trenutnog čvora
- `parent` : odabire roditelja trenutnog čvora
- `preceding` : odabire sve čvorove koji se pojavljuju prije trenutnog čvora u dokumentu osim predaka i čvorova prostora imena
- `preceding-sibling` : odabire svu braću koji se pojavljuju prije trenutnog čvora
- `self` : odabire trenutni čvor

XPath koristi izraze puta za odabir čvorova u XML dokumentu. Čvor se bira slijedeći putanju ili korake. Sve čvorove sa određenim imenom može se odabrati tako da se upiše naziv tog čvora. Odabir korijenskog čvora se postiže sa izrazom `"/"`. Odabir svih čvorova koji odgovaraju selekciji gdje god u dokumentu oni bili se postiže sa izrazom `"//"`. Trenutni čvor se može odabrati sa izrazom `"."`, a roditelja trenutnog čvora se odabire sa `".."`. Atributi u čvoru se mogu odabrati sa izrazom `"@"`. Bilo koji element čvora se označava sa oznakom `"*"`, bilo koji atribut čvora se označava sa oznakom `"@*"`, a bilo koji čvor bilo koje vrste se dobije sa izrazom `"node()"`. Korištenjem `"|"` operatora u izrazu XPath može se odabrati nekoliko različitih staza.

[13]

3.2. XQuery

XQuery [14] je funkcionalan upitni jezik koji se koristi za pronalaženje podataka pohranjenih u XML formatu. Smatra se da je XQuery za XML isto ono što je SQL za relacijsku bazu podataka. Dizajniran je za pronalaženje i vraćanje elemenata i atributa iz XML dokumenta, a izgrađen je na XPath izrazima koje koristi za kretanje kroz XML dokument. Može se koristiti još za izdvajanje podataka web servisu, generiranje sažetka izvještaja, pretvorbu XML podataka u XHTML dokument i pretragu Web dokumenta. Općenito se može koristiti za dohvaćanje hijerarhijskih i tabličnih podataka, te za ispitivanje stabla i grafičkih struktura. Najbolji je za baze podataka temeljene na XML-u i objektne baze podataka. Stvoren je od strane World Wide Web Consortium (W3C) 2007. godine, a najnovija verzija XQuery 3.1 je W3C preporuka od 21. ožujka 2017. godine.

Elementi, atributi i varijable koje se koriste u XQuery moraju biti valjani XML nazivi u dokumentu. Važno je i obratiti pozornost na velika i mala slova jer XQuery je osjetljiv na tu razliku. Varijabla je definirana sa oznakom "\$" praćena nazivom varijable, npr. "\$bookstore". Tekstualni izraz mora biti unutar jednostrukih ili dvostrukih navodnika. Komentari se navode tako da se početak označi sa "(:", a kraj sa ":)". Upiti se tvore pomoću izraza FLWOR nastalih po prvim slovima klauzula koje se koriste:

- **For** - koristi se za označavanje čvora
- **Let** - koristi se za postavljanje vrijednosti varijable
- **Where** - koristi se za filtriranje čvorova
- **Order by** - koristi se za sortiranje čvorova
- **Return** - koristi se za određivanje odgovora koji će se vratiti

U jednom XQuery izrazu moguće je koristiti više upita sa ključnim riječima **for** i **let**, a obavezan je barem jedan izraz sa klauzulom **for** ili **let**. Klauzule **where** i **order by** su opcionalne prilikom korištenja klauzule **for**. Na kraju je obavezna klauzula **return** koja određuje izraz koji se vraća kao odgovor. Slično kao i kod XPath-a, ovdje se prema XML dokumentu odnosi kao prema stablu sa čvorovima gdje je korijen stabla korijenski čvor i njemu se pristupa preko funkcije `doc()`.

XQuery, XPath i XSLT dijele istu biblioteku funkcija. Postoji još nekoliko vrsta funkcija koje nudi XQuery za rad sa podacima [15]:

- Funkcije pristupa - `node-name` (vraća naziv čvora), `nilled` (vraća `true` ako je čvor u argumentu `nil`, odnosno ne sadrži ništa), `data` (vraća atomarnu vrijednost čvora), `base-uri` (vraća svojstvo `base-uri` od trenutnog ili određenog čvora), `document-uri` (vraća svojstvo `document-uri` od određenog čvora)
- Funkcije pogrešaka i praćenja - `error` (ispisuje grešku), `trace` (za otklanjanje neispravnosti upita)

- Numeričke funkcije - `abs` (vraća apsolutnu vrijednost broja), `ceiling` (zaokružuje broj na prvi veći cijeli broj), `floor` (zaokružuje broj na prvi manji cijeli broj), `round` (zaokružuje broj na najbliži cijeli broj)
- Tekstualne funkcije - `string-length` (vraća duljinu teksta), `concat` (spaja tekstove zadane u argumentu u jedinstveni tekstualni izraz), `string-join` (spaja tekstove zadane u argumentu sa zadnim delimiterom u jedinstveni tekstualni izraz)
- Logičke funkcije - `boolean` (vraća logičku vrijednost argumenta), `not` (negacija logičke vrijednosti), `true` (vraća istinitu logičku vrijednost), `false` (vraća lažnu logičku vrijednost)
- Vremenske funkcije - `current-date` (vraća trenutni datum), `current-time` (vraća trenutno vrijeme), `current-datetime` (vraća trenutni datum i vrijeme)
- Funkcije čvora - `node-kind` (provjerava je li argument proslijeđen u funkciju čvor), `node-name` (vraća prošireni QName naziv čvora proslijeđenog kao argument), `local-name` (vraća lokalni naziv proslijeđenog čvora)
- Funkcije sekvence - `count` (daje broj elemenata u sekvenci), `sum` (zbraja elemente iz sekvenci), `avg` (računa prosječnu vrijednost elemenata u sekvenci), `min` (vraća najmanju vrijednost iz sekvence), `max` (vraća najveću vrijednost iz sekvence), `distinct-values` (vraća sekvencu bez dupliciranih elemenata), `subsequence` (vraća podniz iz dobivenog niza), `insert-before` (unos element u sekvencu), `remove` (briše element iz sekvence), `reverse` (vraća preokrenutu sekvencu), `index-of` (vraća indeks nekog elementa iz sekvence), `last` (vraća zadnji element iz sekvence), `position` (vraća broj koji predstavlja poziciju čvora u sekvenci čvora koji se obrađuje)
- Kontekstualne funkcije - omogućuju kreiranje novog konteksta upita, upravljati sa trenutnim kontekstom ili dohvatiti informacije o trenutačnom kontekstu

Od verzije 3.0, XQuery nudi dodatne korisne značajke [16]. Kako bi se omogućio rad sa najnovijom verzijom XQuery, potrebno je upit započeti sa naredbom "xquery version 3.1;". Nova verzija donosi try/catch mehanizam kako bi se mogle uhvatiti eventualne greške prilikom izvršavanja upita, switch izraz koji omogućuje izvršavanje samo jednog dijela koda, funkcije, jednostavniji operatori mapiranja, jednostavnije spajanja tekstualnih izraza pomoću operatora "||", anotacije za funkcije i varijable koje se mogu odrediti kao privatne ili javne. Možda najzanimljivija novost koja je došla od verzije 3.0 je dodavanje Group by klauzule u FLWOR izraz koji time omogućuje grupiranje podataka dobivenih prilikom upita.

4. XML baza podataka

XML baza podataka je sustav za održavanje podataka koji omogućuje njihovu uporabu i pohranjivanje u XML formatu. Postoje dvije vrste XML baza podataka [17] :

- XML-enabled baze podataka (XED) - ovaj tip baze podataka radi slično kao relacijska baza podataka, odnosno podaci se pohranjuju u tablici u obliku redova i stupaca. Obično su implementirane kao dodatak u drugim sustavima za upravljanje bazom podataka gdje je moguće preslikavanje XML dokumenta u određenu bazu. Podaci koji se pritom preslikavaju mogu izgubiti metapodatke i strukturu zadanu u dokumentu. Najpoznatije realizacije XML-enabled baze podataka su: Oracle 9i, Oracle 10g, Microsoft SQL Server, IBM DB2 itd.
- Native XML baze podataka (NXD) - koristi se za pohranu velike količine podataka i umjesto tabličnog formata on slijedi XML model za pohranjivanje podataka. Ova vrsta baze podataka ima potpunu kontrolu nad XML-om i time je poželjniji jer omogućuje visoku sposobnost pohrane, održavanja i pregledavanja XML dokumenta, te dohvat podataka pomoću XQuery ili XPath izraza.

Odlučivanje o vrsti baze podataka koja će se koristiti ovisi o specifičnim zahtjevima aplikacije i formatu XML dokumenta. Potrebno je odrediti je li XML dokument usmjeren na podatke ili na dokument kao cjelinu. XML dokument usmjeren na podatke se koristi prilikom interoperabilnosti, te ima čvrstu strukturu, sitnozrnate podatke i ne sadrži miješani sadržaj. Za rad sa njima preporučuje se XML-enabled baze podataka. XML dokumenti usmjereni na dokument kao cjelinu su uglavnom izrađeni za ljude, te nemaju čvrstu strukturu i imaju grubozrnate podatke do te mjere da najmanja neovisna podatkovna jedinica može biti cijeli dokument. Ključni elementi ovog tipa XML dokumenta su polustrukturirani podaci, te za rad sa njima se preporučuju Native XML baze podataka. Zbog specifičnosti aplikacijske domene objašnjene u idućem poglavlju, za potrebe ovoga rada koristit će se Native XML baza podataka, sustav eXist-db. Ostale popularne realizacije NXD baze podataka su: BaseX, Sedna, 28.io, MarkLogic Server.

BaseX [16] je jednostavan (eng. lightweight) sustav za NXD bazu podataka sa dodatnim aplikacijskim poslužiteljima napisanim na Javi. Objavljen je kao open-source aplikacija 2007. godine. BaseX omogućava jednostavno GUI sučelje za korištenje napisano u programskom jeziku Java. U odnosu na eXist, BaseX postiže 99,9% usklađenosti s W3C XQuery 1.0 specifikacijom (eXist ima 99,4%) i implementira specifikacije za XQuery Update 1.0 i XQuery Full-Text. eXist ima stariju verziju implementacije XQuery Update, ali je dostupan dulje od BaseX pa podržava mnoge druge značajke, poput XSLT-a.

Sedna [16] je jednostavan sustav za NXD bazu podataka napisan u programskim jezicima C i C++. Podrijetlo Sedne nije dobro dokumentirano, ali čini se da je započelo oko 2003. kao projekt Instituta za sistemsko programiranje Ruske akademije znanosti. U usporedbi s eXistom, Sedna izravno podržava više API-ja za različite programske jezike. Iako nema vlastiti REST poslužitelj, može se konfigurirati da radi kao modul unutar Apache HTTP poslužitelja.

Sedna postiže 98,8% usklađenosti s W3C XQuery 1.0 specifikacijom što je manje od onoga što postiže eXist.

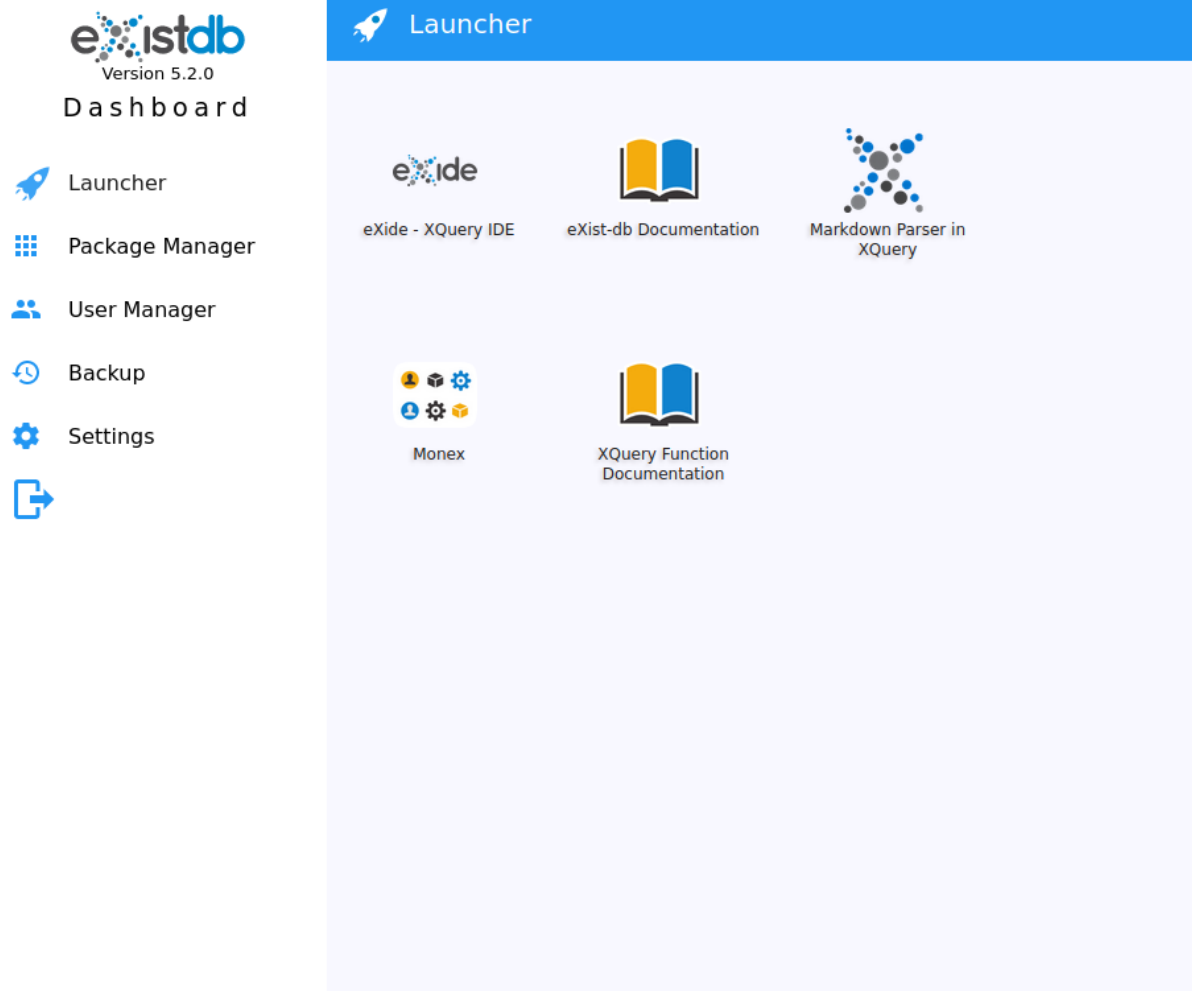
28.io [16] je PaaS (platforma kao usluga) za Zorba open source i integrira Zorbu s MongoDB, podupirući pohranu i indeksiranje XML-a u MongoDB kao glavnu bazu podataka. U usporedbi s eXistom, 28.io nudi uslugu na oblaku i upravlja svojom bazom podataka u potpunosti koristeći XQuery, dok eXist nudi administrativne GUI alate i dodatne API-je. 28.io, slično kao eXist, omogućava aplikacijsku platformu poslužitelja koja omogućuje izgradnju čitavih aplikacija pomoću XQueryja. 28.io (preko Zorba) ima sličnu kompatibilnost s W3C XQuery 1.0 specifikacijom kao eXist, te također podržava XQuery Update, XQuery Full-Text i XQuery Scripting. 28.io je razvila 28msec, tvrtka sa sjedištem u Zürichu u Švicarskoj.

MarkLogic Server [16] je sustav za NXD bazu podataka napisan u C++ koji omogućuje rad sa XQuery upitima i XSLT transformacijom podataka. Razvila ga je MarkLogic Corporation iz SAD, Kalifornije. MarkLogic nudi mogućnost upravljanja velikim količinama podataka (više od petabajt XML podataka), dok eXist može raditi puno skromnije (nekoliko stotina gigabajta, ali to uvelike ovisi o skupu podataka i upitima). Najveća prednost MarkLogica što u odnosu na druge slične sustave ima bolje skaliranje velikog skupa podataka.

4.1. eXist-db

eXist-db (ili skraćeno eXist) je besplatan i otvoreni sustav za native XML bazu podataka (NXD) napisan u programskom jeziku Java. Kreiran je 2000. godine, a trenutna verzija (5.0) izašla je u rujnu 2019. godine. Dobitnik je nagrade za najboljeg pružatelja XML baze podataka 2006. godine od strane InfoWorld-a. Baza podataka je sastavljena od XML dokumenata povezanih u direktorije koje se zovu kolekcije. Glavna prednost eXist-a u odnosu na druge NoSQL sustave je mogućnost korištenja standardiziranog XQuery upitnog jezika za rad sa dokumentima. eXist-db osim XQuery-a omogućava i još neke dodatne standarde i tehnologije: XPath, XSLT, XSL-FO, WebDAV, REST, RESTXQ, XInclude, XML-RPC, XProc. [18]

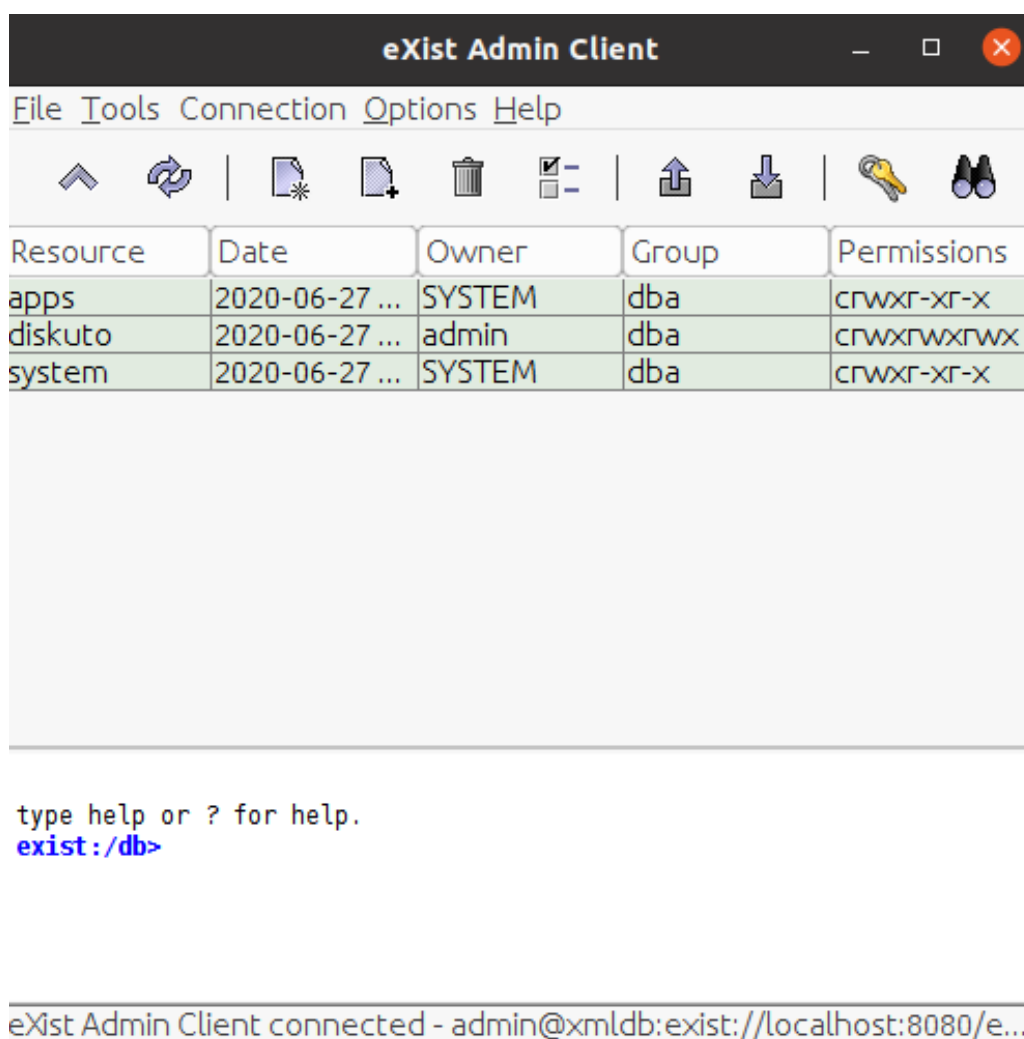
eXist se može instalirati na gotovo svim verzijama operacijskih sustava Linux, Windows i Mac OS X. Najvažniji faktor je da operacijski sustav ima predinstaliran JRE (Java Runtime Environment) ili JDK (Java Development Kit) verzije 1.6 ili novije. Proces instalacije je jednostavan: potrebno je sa službene stranice (<http://www.exist-db.org>) preuzeti željenu verziju i pratiti korake instalacije. eXist prema zadanim postavkama koristi dva TCP porta: port 8080 koji služi za običnu HTTP komunikaciju i port 8443 koji služi za sigurnu HTTPS komunikaciju. Nužno je osloboditi ova dva porta kako bi eXist mogao pravilno raditi, a eventualno se u postavkama mogu promijeniti brojevi portova. Poslužitelj se pokreće desnim klikom na ikonu u traci sustava i odabirom opcije "Start server". Moguće je i pokrenuti poslužitelj preko upravljačke konzole tako da se pozicionira u direktorij gdje je eXist instaliran i pozove naredba "java -jar start.jar". Nakon uspješnog pokretanja, preko Web preglednika se može pristupiti upravljačkoj ploči eXista da se kao adresa upiše "<http://localhost:8080>". Kako bi se bez ograničenja mogle koristiti sve opcije potrebno se prijaviti sa korisničkim imenom i lozinkom zadanom prilikom instalacije. Na slici 4 je prikazan izgled upravljačke ploče programa eXist.



Slika 4: eXist upravljačka ploča

U upravljačkoj ploči moguće je odabrati nekoliko alata i dodatnih paketa koje će u nastavku biti objašnjene. eXide je razvojno okruženje koje može raditi sa XQuery, XML i ostalim resursima pohranjenim u eXide. Sa njim je moguće kreirati nove dokumente ili održavati postojeće. Podržava nekoliko mogućnosti koje postoje u popularnim i modernim razvojnim okruženjima poput refaktoriranja, automatskog nadopunjavanja i slično. eXist-db Documentation omogućava pristup dokumentaciji koja opisuje i olakšava rad sa programom. Pomoć u radu mogu pružiti i opcije Markdown Parser in XQuery i XQuery Function Documentation koji mogu olakšati shvaćanje rada sa XQuery-em. U upravljačkoj ploči je moguće odabrati opciju Monex koja prikazuje statistiku rada sa bazom podataka. U realnom vremenu je moguće promatrati zauzeće resursa na bazi, pratiti rad upita i indeksa, povezati bazu na udaljeni poslužitelj, omogućiti rad sa bazom na tom poslužitelju i slično. S lijeve strane upravljače ploče pod izbornikom Package Manager moguće je upravljati, instalirati i deinstalirati dodatne pakete. Izbornik User Manager omogućava dodavanje i brisanje korisnika i grupa korisnika. Pod izbornikom Backups moguće je napraviti sigurnosnu pohranu ili odrediti vremenske intervale kada bi se ta sigurnosna pohrana događala. Pod izbornikom Settings moguće je izmijeniti postavke programa. Zadnja mogućnost u izborniku s lijeve strane odjavljuje trenutno aktivnog korisnika iz upravljačke ploče.

Java Admin Client je jednostavna kontrolna aplikacija eXist-a kojoj se može pristupiti odabirom opcije na ikoni u traci sustava. Aplikacija omogućava rad sa bazom podataka, kreiranje novih kolekcija, kreiranje i vraćanje sigurnosnih pohrana, izvoz i uvoz podataka, postavljanje baze podataka i slično. Na slici 5 prikazan je izgled navedene aplikacije. Nakon uspješne autorizacije korisniku se prikazuje popis kolekcija u bazi podataka. Korijenska kolekcija u bazi podataka je "/db" i njen naziv nije moguće promijeniti. U njoj je moguće dodavati vlastite kolekcije po želji kao što je za potrebe ovog diplomskog rada izrađena kolekcija "diskuto". Prema zadanim u korijenskoj kolekciji postoje još neke važne kolekcije poput "/db/system" gdje eXist sprema konfiguracijske informacije i "/db/apps/" gdje se nalaze svi paketi instalirani tijekom instalacije ili ručno poslije.



Slika 5: Java Admin Client

Mnogo je načina kako se može pristupiti i upravljati sa podacima u eXist-u. Već je objašnjeno kako se to može preko aplikacije Java Admin Client ili razvojnog okruženja eXide, ali osim toga bazi je moguće pristupiti preko eXist's WebDAV sučelja preko adrese "http://localhost:8080/exist/webdav/" preko vanjskih razvojnih okruženja koje imaju opciju za rad sa eXist-om (poput oXygen) ili preko ugrađenih biblioteka za rad sa Ant poslužiteljom na kojem će biti izrađen i projektni dio ovog diplomskog rada.

eXist-db nudi brojne prednosti u odnosu na sustave za upravljanje relacijskim bazama podataka (RDBMS) ili druge NoSQL baze podataka od kojih su neke opisane u nastavku. Zbog korištenja XML-a i orijentacije na dokument, eXist rješava probleme sa kojima se suočavaju drugi sustavi. RDBMS, ali i neke NoSQL baze podataka zahtijevaju da se prema definiranoj shemi zapisuju podaci, dok za eXist ona nije nužna što donosi fleksibilnost u radu sa podacima. eXist koristi XML dokumente za pohranu podataka, dok mnoge druge NoSQL baze (poput MongoDB i Apache Cassandra) umjesto XML, spremaju dokumente u JSON formatu. XML ima znatno više pozitivnih značajki u odnosu na JSON, što je već i objašnjeno u ovom radu, ali jedna od ključnih prednosti XML-a u odnosu na JSON je što omogućuje rad sa kompleksnijim strukturama dokumenta. eXist omogućuje W3C standardizirani upitni i transformacijski jezik XQuery i XSLT, što znači da na svim drugim sustavima koji ih podržavaju oni će raditi identično kao i na eXistu. RDBMS obično koriste također standardizirani SQL upitni jezik, no u praksi neki složeniji upiti neće raditi na različitim proizvodima RDBMS. Sa druge strane, većina NoSQL baza podataka nemaju standardiziran upitni jezik već je on za svaki zasebni proizvod jedinstven. Kao i mnogi drugi sustavi za upravljanje bazom podataka, eXist omogućuje definiranje indeksa za pretragu što, uz orijentiranje na dokument kao cjelinu, dovodi do toga da su rezultati pretrage preciznije od nekih drugih sustava poput TEI, DocBook i DITA. eXist podržava i XForms koji je još jedan W3C standard. Sa njime je moguće korisnički unos jednostavno zabilježiti u XML dokument u bazi podataka. Većina RDBMS i NoSQL baza podataka omogućuju kontroliranje transakcija prema bazi podataka. Nažalost, eXist trenutno ne podržava transakcije na razini baze podataka, ali korištenje baze podataka se sprema u dnevnik baze podataka što osigurava trajnost i konzistenciju podataka. [16]

5. Opis aplikacijske domene

Kao praktični dio ovog rada izrađena je XML baza podataka na programskoj platformi eXist-db za potrebe internetskog foruma. Aplikacija koristi izrađenu bazu podataka, te je realizirana kao Java web-aplikacija. U ovom poglavlju je definiran pojam internetskog foruma koji je aplikacijska domena praktičnog dijela rada. Zatim slijedi opis funkcionalnosti izrađene web-aplikacije pod nazivom "Diskuto".

5.1. Internetski forum

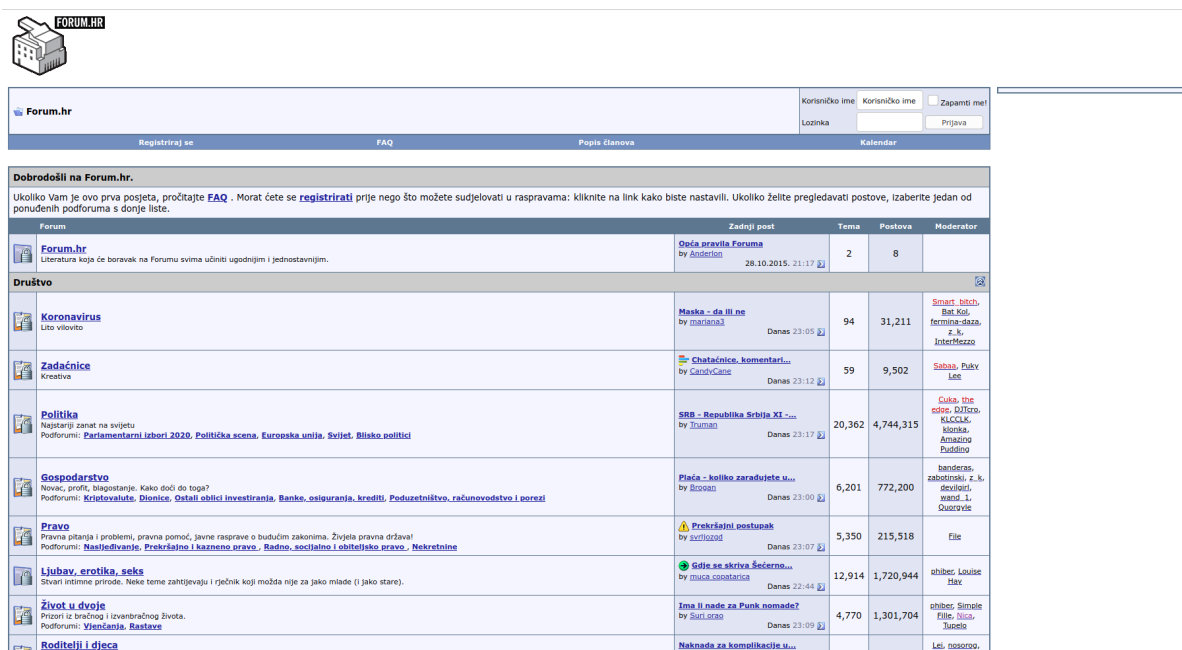
Internetski forum je mrežna aplikacija koja omogućava komunikaciju između više ljudi o nekoj određenoj temi. Sudionici na forumu su najčešće anonimni, a eventualno za sudjelovanje u njemu se može zahtijevati registracija korisničkim imenom i lozinkom. Sve poruke koje korisnik napiše i pošalje su vidljive svim ostalim sudionicima. [19] Zbog toga se internetski forum ponekad naziva oglasna ploča (eng. discussion board). Poruke se mogu ostavljati i čitati bez ograničenja. Radi lakšeg snalaženja forum je podijeljen u nekoliko kategorija prema temama razgovora. Unutar njih se nalaze objave koje otvaraju i započinju korisnici. Postoji više vrsta korisnika:

- Neregistrirani korisnici, zovu se još i gosti, nemaju pravo sudjelovati u raspravama, ali zato mogu čitati i pregledavati sadržaj. Poneki forumi striktno zahtijevaju registraciju korisnika kako bi mogao i čitati forum. Postoje korisnici koji su vrlo česti posjetioci foruma, ali ne sudjeluju u njemu već samo pregledavaju objave. U Internet slangu takav korisnik se naziva "lurker", što dolazi od engleske riječi "to lurk" koja znači skrivanje ili vrebanje nekoga ili nečega u zasjedi.[20]
- Registrirani korisnici su obični korisnici koji sudjeluju u raspravama, otvaraju objave, komentiraju i tako pridonose internetskom forumu.
- Moderatori (kratica "mod") su registrirani korisnici koji imaju prava poput: brisanja, spajanja, premještanja, razdvajanja, zaključavanja ili uređivanja objava. Mogu blokirati, odblokirati, suspendirati ili upozoravati obične korisnike foruma, sve u svrhu boljeg održavanja foruma.
- Administratori (kratica "admin") su registrirani korisnici koji vode internetski forum. Imaju iste ovlasti kao i moderatori, a uz to mogu promovirati obične korisnike na moderatore, upravljati pravilima, kreirati sekcije, uređivati izgled foruma, te raditi razne operacije nad bazom podataka.[21]

Osim gore opisanog, postoje još neke funkcionalnosti koje su sadržane na većini internetskih foruma. Privatne poruke (kratica "PM" od engleske riječi "private message" za privatnu poruku) služe za slanje privatne poruke od jednog korisnika prema drugom korisniku. Obično se takve poruke koriste za osobnu komunikaciju. Korisnik može prilikom slanja uz poruku učiti i datoteku kao prilog. Datoteka se pritom sprema na poslužitelj od foruma, stoga se moraju

definirati ograničenja nad datotekama (tip datoteke i veličina datoteke). Na nekim internet-skim forumima korisnik može oblikovati sadržaj svoje poruke koristeći HTML. Kroz internetske forume su popularizirani i emotikoni ili smajlići, odnosno slikovni prikaz korisnikovog raspoloženja ili osjećanja. Korisnik može drugog korisnika staviti na listu ignoriranja čime neće više vidjeti njegove objave. Mnogi forumi omogućuju korisnicima da postave svoj potpis (kratica "sig" od engleske riječi "signature" za potpis) koji postaje vidljiv na kraju svake korisnikove objave. Korisnik se može i pretplatiti na pojedine dijelove foruma, da ne bi propustio sadržaj objavljen sa tog dijela foruma.[22]

Svaki forum sadrži i popis pravila kojih se korisnici moraju pridržavati. Kršenje pravila sankcioniraju administratori i moderatori, a nepridržavanje pravila mogu prijaviti i obični korisnici. Ako netko prekrši pravilo foruma, prvo ga se obično upozori preko privatne poruke. Ako korisnik nastavlja kršiti pravila zabranjuje mu se pristup forumu na određeni broj dana. Ako i dalje nastavlja kršiti pravila produžuje mu se zabrana pristupa forumu, a može dobiti i trajnu zabranu. Sadržaj koji krši pravila se obično obriše, a ponekad i tema koja je kontroverzna se zaključa i onemogućuje se korisnicima daljnja rasprava. [23] Upravo zbog učestalog kršenja pravila Reddit je nedavno trajno uklonio "r/The_Donald", jednu od najvećih zajednica pristaša američkog predsjednika Donalda Trumpa sa oko 790.000 korisnika. Uklonjeno je stotine tisuća objava i milijuni komentara napisanih od početka djelovanja, 2015. godine. Kao razlog uklanjanja Reddit je naveo poticanje na nasilje i uznemiravanje drugih članova i zajednica Reddit-a. [24]



The screenshot shows the Forum.hr website. At the top, there's a navigation bar with links like 'Registriraj se', 'FAQ', 'Popis članova', and 'Kalender'. Below this is a section titled 'Dobrodošli na Forum.hr.' followed by a welcome message and instructions. The main part of the page is a table listing various forums. Each row includes the forum name, a description, the latest post title, the post date, the number of posts, and the number of members. The forums listed include 'Forum.hr', 'Društvo', 'Koronavirus', 'Zadaci', 'Politika', 'Gospodarstvo', 'Pravo', 'Ljubav, erotika, seks', 'Život u dvoje', and 'Roditelji i djeca'.

Forum	Opis	Zadnji post	Tema	Postova	Moderator
Forum.hr	Literatura koja će boravak na Forumu svima učiniti ugodnijim i jednostavnijim.	Opća pravila Forum by Anderson	28.10.2015. 21:17	2	8
Društvo					
Koronavirus	Lito vilo vito	Naska - da li ne by macka	Danas 23:05	94	31,211
Zadaci	Kreativ	Četvrtina komentari... by GatoCane	Danas 23:12	59	9,502
Politika	Najstariji znanstveni svijet Podforumi: Parlamentarni izbori 2020, Politička scena, Europska unija, Srijet, Blisko politiki	SRS - Republika Srbija XI... by Truman	Danas 23:17	20,362	4,744,315
Gospodarstvo	Novac, profit, blagostanje, kako doći do toga? Podforumi: Kriptovalute, Bitcoine, Ostali otlici, Investiranje, Banke, osiguranje, krediti, Poduzetništvo, računarstvo i porezi	Plača - koliko zaradujete u... by Rocco	Danas 23:00	6,201	772,200
Pravo	Pravna pitanja i problemi, pravna pomoć, javne rasprave o budućim zakonima, Života pravna država! Podforumi: Nastavljanje, Prekršajno i kazneno pravo, Radno, socijalno i obiteljno pravo, Rekreativne	Prekršajni postupak by sylvia	Danas 23:07	5,350	215,518
Ljubav, erotika, seks	Svaki intimni priroda. Nije tema zahtjevu i rječnik koji možda nije za jako mlade (i jako stare).	Štiti se skrive ščerno... by micka.copaticca	Danas 22:44	12,914	1,720,944
Život u dvoje	Prirozi iz bratstva i ljubavnog života. Podforumi: Vjenčanje, Basmare	Ima li nade za Pank noma... by Suci	Danas 23:09	4,770	1,301,704
Roditelji i djeca		Nakonada za kompilekacije u... by Suci	Danas 23:09	4,770	1,301,704

Slika 6: Slika zaslona foruma Forum.hr

Internetski forumi postoje još od samih početaka Interneta. Prve online zajednice i forumi su bili razvijani na BBS (eng. Bulletin Board System) 70-tih godina prošlog stoljeća. Najraniji takvi forumi su bili EIES (eng. Electronic Information Exchange System, nastao 1976. godine) i KOM (šved. KOMpatibila system, nastao 1977. godine). The WELL je najstariji aktivni forum. Razvijen je 1985. godine, ali nije pretjerano popularan jer ima oko 2700 članova.

[25] Neki najpopularniji svjetski forumi su: Reddit, Stack Overflow, Quora, India-Forums, Askubuntu i TripAdvisor. [26] U Hrvatskoj je najstariji i najpoznatiji Forum.hr (slika 6). Razvijen je 1999. godine i sadrži preko 500.000 korisnika. [27] Od ostalih domaćih internetskih foruma najpopularniji su: CROMETEO forum, indexforum, Znatko, OMF, Cafe Mobil. [28]

5.2. Diskuto

Diskuto je naziv internetskog foruma koji je razvijen kao praktični dio ovog rada. Naziv "Diskuto" dolazi od esperanto riječi koja u prijevodu znači "rasprava". Aplikacija objedinjuje funkcionalnosti najpoznatijeg svjetskog i najpoznatijeg domaćeg internetskog foruma: Reddit i Forum.hr.

Diskuto je aplikacija koja omogućava korisnicima raspravu o nekoj određenoj temi. Aplikacija se sastoji od više različitih foruma (Diskuta) u kojima korisnici mogu pokretati različite objave i raspravljati o njima. Forum predstavlja lokaciju na kojoj korisnici pokreću objave sukladno pravilima svakog pojedinog foruma. Kako bi bio pregledniji, svaki forum se može dodatno razgranati po pojedinim kategorijama. Korisnici kada žele otvoriti novu objavu moraju joj zadati naslov, opis (tekstualni ili slikovni) i definirati na kojem forumu i u kojoj kategoriji tog foruma žele otvoriti objavu. Drugi korisnici se zatim mogu javljati, komentirati objavu i tako sudjelovati u raspravi o nekoj temi. Ako neka objava ili komentar krši pravila foruma, moderator može reagirati i obrisati ili zaključati objavu ili komentar. Svaki korisnik može svaku objavu ili komentar označiti sa "+" ako ih smatra korisnima ili zanimljivima ili sa "-" ako ih smatra lošima ili krivima. Nad svakom takvom objavom se računa "karma", odnosno omjer pluseva i minusa. Na taj način može se vidjeti kako drugi korisnici reagiraju na objavu ili komentar. Važno je napomenuti da visoka karma ne znači nužno da tvrdnja koja se šalje kroz objavu/komentar ispravna, već je ona samo rezultat pozitivnih reagiranja drugih korisnika. Karma se preslikava i na korisnika koji je kreirao objavu ili komentar, te je vidljiva javno na njegovom profilu. Tako se može vidjeti koliko korisnik doprinosi svojim sudjelovanjem. Na Diskutu postoje sljedeće vrste korisnika:

- Neregistrirani korisnik - nema pravo sudjelovati u raspravama, ali zato može čitati i pregledavati sadržaj. Može otvarati forume (Diskuta), pregledavati kategorije i pregledavati raspravu na pojedinim objavama.
- Registrirani korisnik - može čitati forume, pregledavati kategorije i pregledavati raspravu na pojedinim objavama, kreirati svoje forume, pretplatiti se na tuđe forume, kreirati nove objave na svojim ili tuđim forumima, sudjelovati u raspravama na svojim ili tuđim objavama, brisati svoje objave ili komentare, prijavljivati tuđe objave ili komentare moderatorima pojedinog foruma, dati "+" ili "-" objavi ili komentaru, pregledavati tuđe profile, slati i primati privatne poruke od drugih korisnika, uređivati svoj korisnički profil, mijenjati jezik aplikacije, pretraživati forume, objave ili profile.
- Vlasnik - registrirani korisnik koji kada kreira svoj forum, on postaje vlasnik na tom forumu. Ima pune ovlasti nad forumom što uključuje: uređivanje opisa i pravila, dodavanje i brisanje kategorija, davanje ili oduzimanje moderatorskih ovlasti drugim korisnicima, pregled

prijavljenih objava i komentara na svom forumu, brisanje tuđih objava i komentara sa svog foruma, zaključavanje pojedinih objava.

- Moderator - registrirani korisnik koji nije kreirao forum, ali su mu dodijeljene sve ovlasti nad forumom koje ima i vlasnik. Razlika u odnosu na vlasnika je i to što vlasnik ima trajne i neopozive ovlasti nad forumom, dok se moderatoru u svakom trenutku može uskratit ovlasti čime na tom forumu onda postaje obični registrirani korisnik.



Slika 7: Logo aplikacije Diskuto

Registracija se obavlja na način da korisnik unosi željeno korisničko ime i lozinku, te e-mail na koji će mu stići kod sa kojim će potvrditi registraciju. U aplikaciju se prijavljuje koristeći se korisničkim imenom i lozinkom, a ako je zaboravio lozinku može ju i poništiti. Osim javnog raspravljanja na objavama, korisnici mogu i međusobno razmijenjivati privatne poruke. Diskuto je dostupan na dva jezika: engleskom i hrvatskom. Promjena jezika se izvršava u postavkama, gdje korisnik može urediti svoje korisničke podatke ili obrisati svoj račun.

6. Model baze podataka

Baza podataka se sastoji od kolekcije u kojoj se nalaze 4 XML dokumenta: forumi (forums.xml), poruke (messages.xml), objave (posts.xml) i korisnici (users.xml). To su ujedno i osnovni koncepti prepoznati iz aplikacijske domene. Dva su načina definiranja strukture XML dokumenta: stariji i stroži način je DTD (Document Type Definitions), a noviji s većim mogućnostima je XML Schema. U nastavku je prikazana XML Schema svih XML dokumenata u kolekciji "db/diskuto", te njihov opis.

6.1. Korisnici (users.xml)

XML dokument "users.xml" sadrži popis svih korisnika u aplikaciji. Kao programski isječak 1 je prikazana XML schema navedenog dokumenta. Registracijom se dodaje novi korisnik (element <user/>, linije 5-37). Elemente "email" (elektronička pošta korisnika, linija 8), "name" (korisničko ime, linija 9) i "password" (lozinka, linija 10) definira korisnik prilikom registracije. Prilikom registracije aplikacija u pozadini dodjeljuje vrijednost elementu "created" (linija 12) koji predstavlja UNIX vrijeme u koje je korisnik kreirao račun i vrijednost elementu "code" (linija 11) koji predstavlja kod za potvrdu registracije. Element "subscriptions" (linije 13-19) sadrži popis svih foruma na koje se korisnik pretplatio, a element "ignore" (linije 20-26) listu svih korisnika koje je ignorirao. Element "disabled" (linija 27) može poprimiti vrijednosti 0 ili 1, a on označava je li korisnik obrisao svoj račun. Element "language" (linija 28) može poprimiti vrijednosti 'en' i 'hr', a on označava jezik na kojem se aplikacija izvršava za tog korisnika.

```
1  <?xml version="1.0"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3  <xs:element name="users">
4    <xs:complexType>
5      <xs:element name="user">
6        <xs:complexType>
7          <xs:sequence>
8            <xs:element name="email" type="xs:string"/>
9            <xs:element name="name" type="xs:string"/>
10           <xs:element name="password" type="xs:string"/>
11           <xs:element name="code" type="xs:integer"/>
12           <xs:element name="created" type="xs:integer"/>
13           <xs:element name="subscriptions">
14             <xs:complexType>
15               <xs:sequence>
16                 <xs:element name="forum" type="xs:string"/>
17               </xs:sequence>
18             </xs:complexType>
19           </xs:element>
20           <xs:element name="ignore">
21             <xs:complexType>
22               <xs:sequence>
23                 <xs:element name="user" type="xs:string"/>
24               </xs:sequence>
25             </xs:complexType>
```

```

26         </xs:element>
27         <xs:element name="disabled" type="xs:boolean"/>
28         <xs:element name="language" type="languageType"/>
29         <xs:simpleType name="languageType">
30             <xs:restriction base="xs:string">
31                 <xs:enumeration value="en"/>
32                 <xs:enumeration value="hr"/>
33             </xs:restriction>
34         </xs:simpleType>
35     </xs:sequence>
36 </xs:complexType>
37 </xs:element>
38 </xs:complexType>
39 </xs:element>
40 </xs:schema>

```

Programski isječak 1: XML Schema za dokument "users.xml"

6.2. Forumi (forums.xml)

XML dokument "forums.xml" sadrži popis svih foruma u aplikaciji. Kao programski isječak 2 je prikazana XML schema navedenog dokumenta. Nakon što korisnik kreira u aplikaciji novi forum, podaci o njemu se spremaju u ovom dokumentu. Prilikom kreiranja foruma dodjeljuju mu se vrijednosti elemenata: "name" (naziv foruma, linija 8), "description" (opis foruma, linija 9), "categories" (sadrži popis svih kategorija, linije 10-16), "moderators" (popis svih moderatora, linije 18-24) i "rules" (pravila foruma, linija 25). Nad forumom postoje još elementi "owner" (označava korisnika koji je vlasnik foruma, linija 17), "created" (UNIX vrijeme u koje je forum kreiran, linija 26) i "subscribers" (broj pretplaćenih korisnika na forumu, linija 27).

```

1  <?xml version="1.0"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3  <xs:element name="forums">
4      <xs:complexType>
5          <xs:element name="forum">
6              <xs:complexType>
7                  <xs:sequence>
8                      <xs:element name="name" type="xs:string"/>
9                      <xs:element name="description" type="xs:string"/>
10                     <xs:element name="categories">
11                         <xs:complexType>
12                             <xs:sequence>
13                                 <xs:element name="category" type="xs:string"/>
14                             </xs:sequence>
15                         </xs:complexType>
16                     </xs:element>
17                     <xs:element name="owner" type="xs:string"/>
18                     <xs:element name="moderators">
19                         <xs:complexType>
20                             <xs:sequence>
21                                 <xs:element name="moderator" type="xs:string"/>

```

```

22         </xs:sequence>
23     </xs:complexType>
24 </xs:element>
25     <xs:element name="rules" type="xs:string"/>
26     <xs:element name="created" type="xs:integer"/>
27     <xs:element name="subscribers" type="xs:integer"/>
28 </xs:sequence>
29 </xs:complexType>
30 </xs:element>
31 </xs:complexType>
32 </xs:element>
33 </xs:schema>

```

Programski isječak 2: XML Schema za dokument "forums.xml"

6.3. Objave (posts.xml)

XML dokument "posts.xml" sadrži popis svih objava na svim forumima u aplikaciji. Kao programski isječak 3 je prikazana XML schema navedenog dokumenta. Prilikom kreiranja objave, u ovom dokumentu se spremaju sve informacije o njoj. Elementi koje sadrži svaka objava su: "id" (jedinstveni identifikator objave, linija 8), "headline" (naslov objave, linija 9), "description" (tekstualni opis objave, linija 10), "attachment" (slikovni opis objave, linija 11), "created" (UNIX vrijeme u koje je objava kreirana, linija 12), "owner" (korisnik koji je kreirao objavu, linija 13), "diskuto" (forum na kojem je objavljena, linija 14), "category" (kategorija na forumu na kojem je objavljena, linija 15), "upvote" (svi korisnici koji su označili objavu sa '+', linije 16-22), "downvote" (svi korisnici koji su označili objavu sa '-', linija 23-29), "reported" (može poprimiti vrijednost 0 ili 1 i označava je li objava prijavljena moderatorima, linija 62), "deleted" (može poprimiti vrijednost 0 ili 1 i označava je li objava obrisana, linija 63), "locked" (može poprimiti vrijednost 0 ili 1 i označava je li objava zaključana, linija 64), element "comments" (linije 30-61) koji predstavlja sve komentare koje sadrži pojedina objava. Svaki komentar ("comment") u sebi sadrži attribute: "id" (identifikator komentara na toj objavi, linija 36), "text" (sami tekst objave, linija 37), "created" (UNIX vrijeme u kojem je objava kreirana, linija 38), "owner" (označava korisničko ime korisnika koji je vlasnik komentara, linija 39), "upvote" (svi korisnici koji su označili komentar sa '+', linije 40-46), "downvote" (svi korisnici koji su označili komentar sa '-', linije 47-53), "reported" (može poprimiti vrijednosti 0 ili 1 i označava je li komentar prijavljen moderatorima, linija 54) i "deleted" (može poprimiti vrijednosti 0 ili 1 i označava je li komentar obrisana, linija 55).

```

1  <?xml version="1.0"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3  <xs:element name="posts">
4      <xs:complexType>
5          <xs:sequence>
6              <xs:element name="post">
7                  <xs:complexType>
8                      <xs:sequence>
9                          <xs:element name="id" type="xs:integer"/>
10                         <xs:element name="headline" type="xs:string"/>

```

```

11      <xs:element name="attachment" type="xs:string"/>
12      <xs:element name="created" type="xs:integer"/>
13      <xs:element name="owner" type="xs:string"/>
14      <xs:element name="diskuto" type="xs:string"/>
15      <xs:element name="category" type="xs:string"/>
16      <xs:element name="upvote">
17          <xs:complexType>
18              <xs:sequence>
19                  <xs:element name="user" type="xs:string"/>
20              </xs:sequence>
21          </xs:complexType>
22      </xs:element>
23      <xs:element name="downvote">
24          <xs:complexType>
25              <xs:sequence>
26                  <xs:element name="user" type="xs:string"/>
27              </xs:sequence>
28          </xs:complexType>
29      </xs:element>
30      <xs:element name="comments">
31          <xs:complexType>
32              <xs:sequence>
33                  <xs:element name="comment">
34                      <xs:complexType>
35                          <xs:sequence>
36                              <xs:element name="id" type="xs:integer"/>
37                              <xs:element name="text" type="xs:string"/>
38                              <xs:element name="created" type="xs:integer"/>
39                              <xs:element name="owner" type="xs:string"/>
40                              <xs:element name="upvote">
41                                  <xs:complexType>
42                                      <xs:sequence>
43                                          <xs:element name="user" type="xs:string"/>
44                                      </xs:sequence>
45                                  </xs:complexType>
46                              </xs:element>
47                              <xs:element name="downvote">
48                                  <xs:complexType>
49                                      <xs:sequence>
50                                          <xs:element name="user" type="xs:string"/>
51                                      </xs:sequence>
52                                  </xs:complexType>
53                              </xs:element>
54                              <xs:element name="reported" type="xs:boolean"/>
55                              <xs:element name="deleted" type="xs:boolean"/>
56                          </xs:sequence>
57                      </xs:complexType>
58                  </xs:element>
59              </xs:sequence>
60          </xs:complexType>
61      </xs:element>
62      <xs:element name="reported" type="xs:boolean"/>
63      <xs:element name="deleted" type="xs:boolean"/>

```

```

64         <xs:element name="locked" type="xs:boolean"/>
65     </xs:sequence>
66 </xs:complexType>
67 </xs:element>
68 </xs:complexType>
69 </xs:element>
70 </xs:schema>

```

Programski isječak 3: XML Schema za dokument "posts.xml"

6.4. Poruke (messages.xml)

Sve poruke koje si korisnici međusobno šalju su sadržane u XML dokumentu "messages.xml". Kao programski isječak 4 je prikazana XML schema navedenog dokumenta. Kada korisnik pošalje privatnu poruku drugom korisniku, podaci se spremaju u ovom dokumentu. Vrijednosti elemenata koji se zapisuju su: "sender" (korisničko ime korisnika koji je pošiljalac poruke, linija 8), "recipient" (korisničko ime korisnika koji je primatelj poruke, linija 9), "time" (UNIX vrijeme u koje je poruka poslana, linija 10), "text" (sadržaj poruke, linija 11) i "seen" (može poprimiti vrijednosti 0 ili 1 i označava je li pošiljalac vidio primljenu poruku, linija 12).

```

1  <?xml version="1.0"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3  <xs:element name="messages">
4      <xs:complexType>
5          <xs:element name="message">
6              <xs:complexType>
7                  <xs:sequence>
8                      <xs:element name="sender" type="xs:string"/>
9                      <xs:element name="recipient" type="xs:string"/>
10                     <xs:element name="time" type="xs:integer"/>
11                     <xs:element name="text" type="xs:string"/>
12                     <xs:element name="seen" type="xs:boolean"/>
13                 </xs:sequence>
14             </xs:complexType>
15         </xs:element>
16     </xs:complexType>
17 </xs:element>
18 </xs:schema>

```

Programski isječak 4: XML Schema za dokument "messages.xml"

7. Implementacija

U ovom poglavlju je opisan način izrade projektnog dijela diplomskog rada. Cilj je izraditi mrežnu aplikaciju pod nazivom "Diskuto" koja bi radila kao internetski forum.

Zamišljeno je da podaci koji se pohranjuju budu u polustrukturiranom obliku zbog specifičnosti aplikacijske domene. Predviđeno je da postoji shema po kojoj će se podaci zapisivati, što isključuje korištenje nestrukturirane vrste podataka. Predviđeno je i da se podaci grade u stablastoj strukturi u kojoj neki elementi nisu obavezni, što isključuje strukturiranu vrstu podataka. Primjerice, u aplikaciji može biti više foruma koji mogu i ne moraju imati više objava koje mogu i ne moraju imati više komentara nad kojima se mogu dodavati kompleksniji objekti. Slično se u obliku stabla granaju elementi kod korisnika. Stoga se zbog fleksibilnosti u zapisu, dohvaćanju i prikazu podataka koriste polustrukturirani podaci. Kao model u kojem će biti zapisani podaci koristit će se Extensible Markup Language (XML) zbog jednostavnog definiranja sheme, mogućnosti rada sa kompleksnijim strukturama i lakoćom dohvaćanja i transformacije podataka.

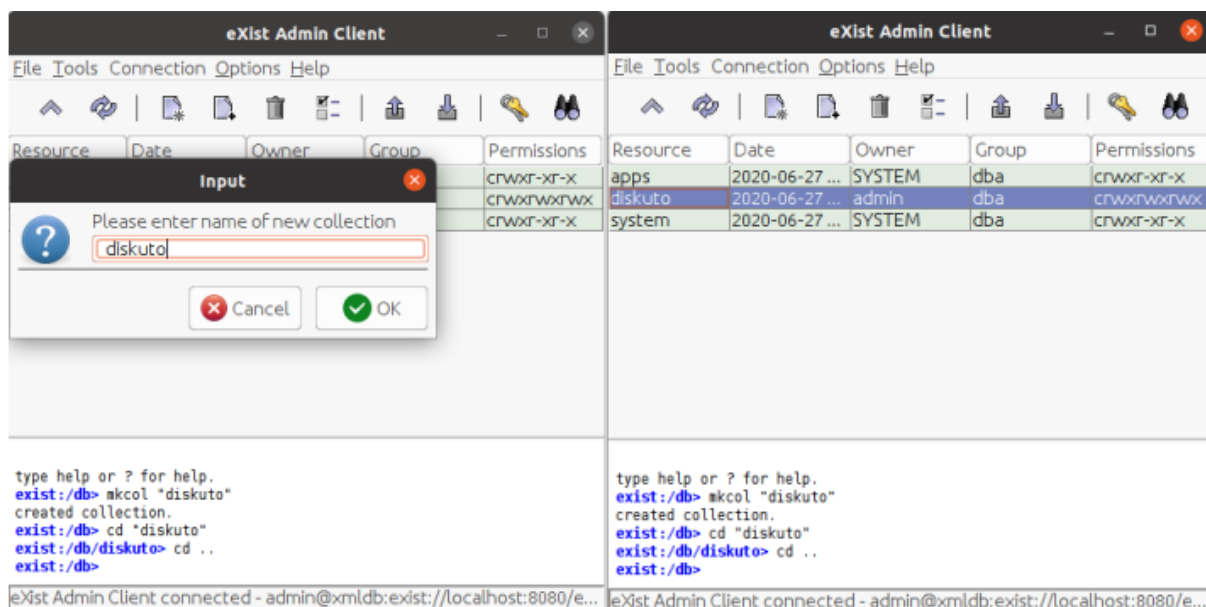
Predviđeno je da se koristi NXD (kratica od Native XML database) baza podataka zbog usmjeravanja na dokument kao cjelinu i izvrsnih performansi u radu sa velikim količinama podataka. Kao sustav za upravljanje NXD bazom podataka koristit će eXist-db zbog podržavanja W3C standarda XQuery i XPath za manipulaciju s podacima, jednostavnosti korištenja i širokog spektra načina pristupa bazi podataka.

Web aplikacija je implementirana u programskom jeziku Java iz razloga što eXist-db pruža biblioteke za jednostavan pristup bazi podataka, a sa mnogim razvojnim okvirima koje nudi Java moguće je vrlo jednostavno implementirati funkcionalnosti aplikacije. Stoga, kao razvojni okvir će se koristiti JSF (eng. Jakarta Server Faces) verzije 2.3. koji će olakšati izradu korisničkog sučelja i omogućiti ponovno korištenje komponenata na različitim dijelovima aplikacije. JSF olakšava realizaciju aplikacije prema MVC (eng. Model-View-Controller) uzorku dizajna prema kojem se aplikacije razvijaju brže i jednostavnije, lakše nadograđuju i lakše otkrivaju eventualne greške u radu. JSF se zajedno koristi sa AJAX-om (eng. Asynchronous JavaScript And XML) koji omogućava dinamičko mijenjanje sadržaja web stranice bez potrebe za njeno osvježavanje što doprinosi ugađenom radu. Dizajn pojedinih web stranica izrađen je pomoću CSS (eng. Cascading Style Sheets). Poslužitelj na kojem se odvija aplikacija je GlasFish 5.1. zbog podrške u radu sa JSF razvojnim okvirom i integracijom sa NetBeans razvojnim okruženjem u kojem se projektni dio rješava. Od dodatnih biblioteka koristi se "javax-mail" koja omogućava slanje elektroničke pošte.

U nastavku je opisano kreiranje baze podataka i služenje sa eXist-db, kreiranje i razvijanje Java web-aplikacije, rad sa bazom podataka, pohrana i pristup podacima u bazi i implementacija pojedinih komponenti aplikacije od početka do konačnog izgleda.

7.1. Kreiranje i rad s bazom podataka

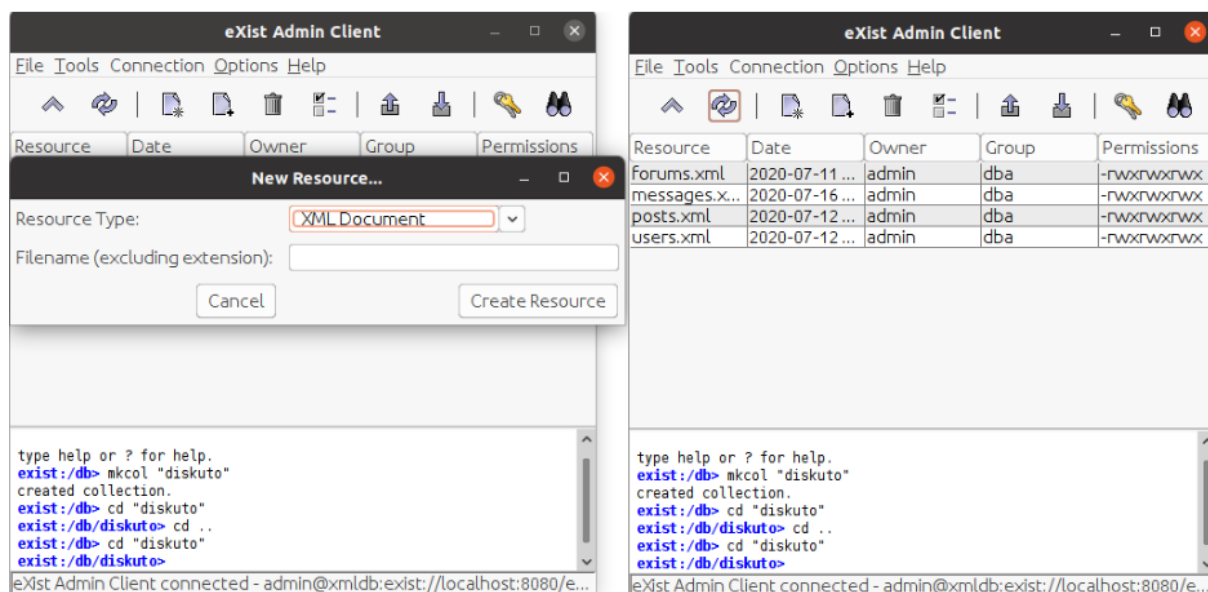
Nakon pokretanja eXist-db i uspješne prijave potrebno je, bilo preko eXide ili Java Admin Clienta, pristupiti bazi podataka. U nastavku je objašnjeno kreiranje baze podataka i potrebnih XML dokumenata preko Java Admin Clienta. Prvo je potrebno kreirati kolekciju u kojoj će biti pohranjeni svi definirani XML dokumenti (slika 8). Nova kolekcija se kreira odabirom u izborniku "File -> Create collection" ili pritiskom na "Ctrl + N". Potrebno je zadati naziv kolekcije pri čemu treba pripaziti da naziv ne smije biti već postojeća kolekcija u bazi podataka. Pritiskom gumba "OK" kolekcija se sprema u bazu podataka.



Slika 8: Kreiranje nove kolekcije u bazi podataka

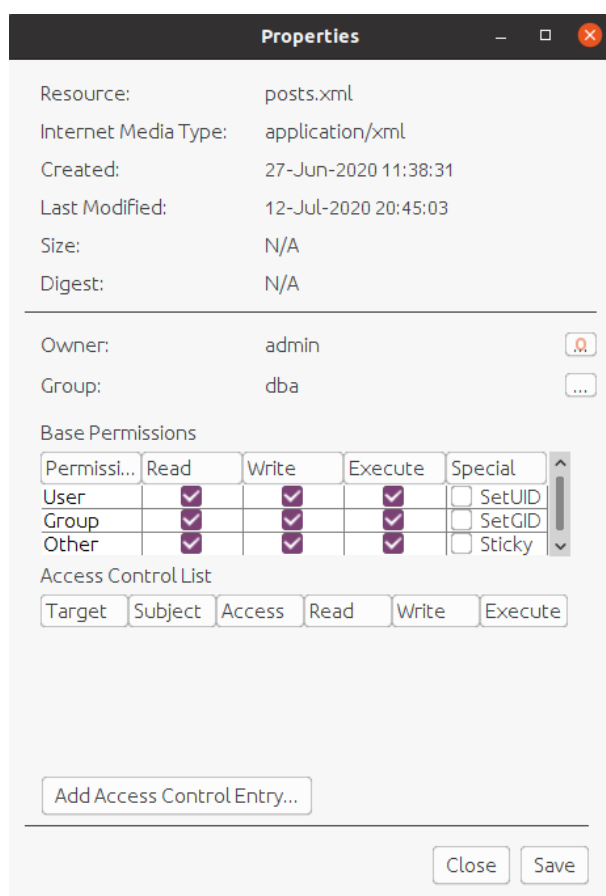
Dvostrukim klikom na kolekciju otvara se njen sadržaj. Inicijalno nema ništa u njoj, pa je potrebno definirati XML dokumente kojima će aplikacija pristupati. U kolekciji je moguće, osim dokumenata i novih kolekcija, dodavati bilo koje druge direktorije ili datoteke različitih formata koji su pohranjeni na računalu na kojem se nalazi baza podataka. Novi direktoriji ili datoteke se uvoze odabirom u izborniku "File -> Store files/directories" ili pritiskom na "Ctrl + S". Za potrebe projektnog dijela diplomskog rada u bazu podataka će biti samo pohranjeni XML dokumenti. Novi XML dokument se može uvesti na prethodno objašnjeni način ili se izravno kreirati u Java Admin Clientu odabirom u izborniku "File -> Create blank document" ili pritiskom na "Ctrl + B". Kao novi resurs može se kreirati XML dokument ili XQuery Module. Nakon što se zada naziv dokumenta (bez ekstenzije .xml), potrebno je pritisnuti gumb "Create Resource" i dokument se sprema u bazu. Na ovaj način kreirani su svi potrebni XML dokumenti za rad aplikacije: forums.xml, messages.xml, posts.xml i users.xml. Prikaz kreiranja novog dokumenta u bazi podataka je prikazan na slici 9.

Da bi vanjske aplikacije mogle pristupati pojedinim kolekcijama, dokumentima i drugim datotekama u bazi podataka, potrebno je postaviti dozvole za čitanje, pisanje i izvršavanje sadržaja. Promjena dozvola se vrši na način da se u alatnoj traci eXist-db odabere u glavnom izborniku "File -> Resource properties" ili pritiskom na "Ctrl + P", pri čemu se otvara prozor



Slika 9: Kreiranje novog dokumenta u bazi podataka

prikazan kao na slici 10. Moguće je vidjeti detalje o odabranom sadržaju: naziv resursa, tip podatka, datum i vrijeme kreiranja, datum i vrijeme zadnje promjene, veličinu i sažetak sadržaja. Moguće je promijeniti vlasnika i grupu sadržaja, te mijenjati prava pristupa za vlasnika, grupu i sve druge korisnike.



Slika 10: Promjena postavki XML dokumenta

Dvostrukim klikom na dokument otvara se uređivač XML dokumenta. Nad svakim dokumentom je potrebno zadati korijenski element; u dokumentu "forums.xml" će to biti "<forums/>", u "messages.xml" će to biti "<messages/>", u "posts.xml" će to biti "<posts/>" i u "users.xml" će to biti "<users/>". Od ostalih opcija koje se nalaze u Java Admin Clientu moguće je brisati dokumente ili kolekcije (preko glavnog izbornika "File -> Remove" ili "Ctrl + D"), kopirati dokumente ili kolekcije (preko glavnog izbornika "File -> Copy" ili "Ctrl + C"), premještati dokumente ili kolekcije na drugu lokaciju unutar baze podataka (preko glavnog izbornika "File -> Move" ili "Ctrl + M"), preimenovati dokumente ili kolekcije (preko glavnog izbornika "File -> Rename" ili "Ctrl + R"), izvesti resurs iz baze na memoriju računala (preko glavnog izbornika "File -> Export a resource to file..." ili "Ctrl + E"), dodavati okidače u bazu (preko glavnog izbornika "Tools -> Edit Triggers" ili "Ctrl + T"), uređivati popis korisnika baze podataka (preko glavnog izbornika "Tools -> Edit Users" ili "Ctrl + U"), te gasiti ili pokretati vezu prema bazi podataka (preko glavnog izbornika "Connection"). [16]

Nakon kreiranja baze podataka, potrebno je kreirati projekt kao Java web-aplikacija na kojem će se izraditi programsko rješenje koje će koristiti kreiranu bazu podataka. Potrebno je kreirati i pomoćne klase preko kojih će ostali dijelovi aplikacije pristupati bazi podataka i preko kojih će dohvaćeni podaci iz XML formata moći biti vraćeni kao čisti podaci bez XML obilježja. Kao programski isječak 5 prikazan je sadržaj pomoćne klase "Database.java" za rad sa bazom podataka, a kao programski isječak 6 prikazan je sadržaj pomoćne klase "XmlHelper.java" za rad sa podacima u XML formatu.

```

1 package org.diskuto.helpers;
2
3 import org.exist.xmlldb.EXistXQueryService;
4 import org.xmlldb.api.DatabaseManager;
5 import org.xmlldb.api.base.Collection;
6 import org.xmlldb.api.base.ResourceSet;
7 import org.xmlldb.api.base.XMLDBException;
8
9 public class Database {
10
11     private final String URI = "xmlldb:exist://localhost:8080/exist/xmlrpc";
12     private final String driver = "org.exist.xmlldb.DatabaseImpl";
13     private final String collection = "/db/diskuto";
14     private final org.xmlldb.api.base.Database database;
15     private final Collection col;
16
17     public Database() throws ClassNotFoundException, InstantiationException,
18         IllegalAccessException, XMLDBException {
19         Class c = Class.forName(driver);
20         database = (org.xmlldb.api.base.Database) c.newInstance();
21         DatabaseManager.registerDatabase(database);
22         col = DatabaseManager.getCollection(URI + collection, "admin", "");
23     }
24
25     public void close() throws XMLDBException {
26         col.close();
27         DatabaseManager.deregisterDatabase(database);
28     }
29 }

```

```

28
29     public ResourceSet xquery(String upit) throws XMLDBException {
30         EXistXQueryService service = (EXistXQueryService) col.getService("
            XQueryService", "1.0");
31         service.setProperty("indent", "yes");
32         return service.query(upit);
33     }
34
35 }

```

Programski isječak 5: Sadržaj pomoćne klase "Database.java"

Preporučeni način za rad sa eXist-db prilikom razvijanja Java aplikacija je korištenje XML:DB API koji pruža zajedničko sučelje za rad s NXD i XED bazama podataka i podržava razvoj aplikacije za višekratnu upotrebu. Na početku je potrebno uvesti biblioteke za rad sa bazom (linije 3-7) koje se mogu naći u direktoriju sa bibliotekama eXist-db. Klasa sadrži nekoliko finalnih varijabli koje su definirane linijama 11-15. U liniji 11 je definiran URI za pristup bazi podataka koji općenito mora biti u formatu "xmldb:exist//[HOST-ADDRESS]/db/kolekcija". Budući da više od jednog upravljačkog programa baze podataka se može registrirati u upravitelj baze podataka, potreban je dio URI-ja "xmldb:exist" da bi se utvrdilo koju klasu upravljačkih programa treba koristiti. Pod [HOST-ADDRESS] upisuje se adresa i port na kojem se nalazi baza podataka. Završni dio URI-ja definira kolekciju kojoj se pristupa. [29] U liniji 12 je definiran naziv upravljača pomoću kojeg se može raditi sa bazom podataka. U liniji 13 definirana je kolekcija u kojoj se nalaze XML dokumenti sa kojima se radi u aplikaciji. Varijabla se kasnije povezuje sa URI-jem kako bi se dobila konačna putanja prema kolekciji u bazi podataka. U linijama 14 i 15 se definiraju objekti baze podataka i kolekcije.

Klasa se sastoji od konstruktora i dvije metode: "close" i "xquery". Prilikom poziva konstruktora kada se inicijalizira novi objekt klase "Database", registrira se nova instanca klase upravljača (linije 18-20) i od upravitelja baze podataka se dohvaća kolekcija pozivanjem statičke metode "DatabaseManager.getCollection" (linija 21). Metoda kao argument prima URI do kolekcije, te korisničko ime i lozinku za prijavu u bazu podataka. Nakon uspješne inicijalizacije nad bazom podataka se mogu vršiti upiti pomoću metode "xquery" koja kao argument prima XQuery upit. Za rad sa upitima koristi se XQueryService klasa eXista (linija 30) koja se može pozvati preko metode "getService" klase Collection. Metoda prima naziv usluge kao prvi argument i njegovu verziju kao drugi argument. Za izvršenje upita poziva se metoda "query" nad inicijaliziranim servisom (linija 32). Metoda prima upit u formatu stringa, a vraća objekt ResourceSet. Nakon što se završi sa radom, poželjno je osloboditi zauzete resurse pozivom metode "close" koja zatvara kolekciju (linija 25) i deregistrira bazu podataka sa upravitelja baze podataka (linija 26).

Nakon dohvaćanja XML-a preko upita iz klase "Database", potrebno je izvući dobivene podatke iz XML elemenata što se može učiniti pomoću metoda iz klase "XmlHelper". Klasa sadrži četiri metode za dohvaćanje podataka iz XML resursa koji se pri instanciranju klase proslijeđuje u konstruktor. U klasi se nalaze dvije varijable: "xml" u kojoj se pohranjuje XML zapis dobivenog resursa i "xp" pomoću kojeg se prolazi i dohvaćaju čvorovi XML resursa. U nastavku slijedi prikaz i pojašnjenje klase.

```

1  package org.diskuto.helpers;
2
3  import java.io.StringReader;
4  import java.util.ArrayList;
5  import java.util.List;
6  import javax.xml.xpath.XPath;
7  import javax.xml.xpath.XPathConstants;
8  import javax.xml.xpath.XPathExpressionException;
9  import javax.xml.xpath.XPathFactory;
10 import org.w3c.dom.NodeList;
11 import org.xml.sax.InputSource;
12 import org.xmldb.api.base.Resource;
13
14 public class XmlHelper {
15
16     XPath xp;
17     String xml;
18
19     public XmlHelper(Resource resource) throws Exception {
20         this.xml = (String) resource.getContent();
21         this.xp = XPathFactory.newInstance().newXPath();
22     }
23
24     public Object makeObject(String object) throws XPathExpressionException {
25         return xp.evaluate("/") + object, new InputSource(new StringReader(xml)),
26             XPathConstants.NODE);
27     }
28
29     public String makeValue(String attribute, Object object) throws
30         XPathExpressionException {
31         return xp.evaluate(attribute, object);
32     }
33
34     public List<String> makeListValue(String expression) throws
35         XPathExpressionException {
36         List<String> list = new ArrayList();
37         NodeList nodes = (NodeList) xp.evaluate(expression, new InputSource(new
38             StringReader(xml)), XPathConstants.NODESET);
39
40         for (int i = 0; i < nodes.getLength(); i++) {
41             list.add(nodes.item(i).getTextContent());
42         }
43
44         return list;
45     }
46
47     public String rawValue() {
48         return xml;
49     }
50 }

```

Programski isječak 6: Sadržaj pomoćne klase "XmlHelper.java"

Na početku je potrebno uvesti nekoliko biblioteka koje su potrebne za pravilan rad klase (linije 3-12). Klasa `StringReader` (linija 3) sadrži metode i varijable za čitanje tekstualnog niza sinkrono ili asinkrono. Klase `ArrayList` (linija 4) i `List` (linija 5) sadrže metode i varijable za pohranu podataka u listu. Klase iz paketa `"javax.xml.xpath.*"` (linije 6-9) sadrže metode i varijable za čitanje i navigaciju prema dijelovima XML dokumenta. Sučelje `NodeList` (linija 10) pruža apstrakciju kolekcije čvorova bez definiranja ili ograničavanja načina na koji se ova zbirka provodi. Klasa `InputSource` (linija 11) omogućuje aplikaciji da kapsulira informacije o ulaznom izvoru objekta, što može sadržavati razne identifikatore i znakovne tokove. Klasa `Resource` (linija 12) omogućuje pohranjivanje jednog resursa iz `ResourceSet` skupa resursa.

Klasa `XmlHelper` ima konstruktor koji prima objekt tipa `Resource` iz kojeg se dohvaća sadržaj i sprema u varijablu `"xml"` (linija 20). Nakon toga se u konstruktoru vrši stvaranje nove instance objekta `XPath` (linija 21). Metoda `"makeObject"` prima naziv elementa ili putanju prema elementu kojeg se želi dohvatiti, te vraća taj element kao objekt (linija 25). Pomoću metode `"makeValue"` se iz objekta dobivenog preko prošle metode mogu dohvatiti vrijednost elementa kojeg taj objekt sadrži (linija 29). Metodi se kao argument proslijeđuje naziv elementa i objekt. Metodom `"makeListValue"`, prema putanji elementa zadanom kao argument, se u listu spremaju sve vrijednosti tog elementa (linije 32-41). Posljednja metoda je `"rawValue"` koja vraća u tekstualnom formatu zapis XML resursa proslijeđenom konstruktoru u istom obliku kojem postoji u bazi podataka (linija 44).

7.2. Java web-aplikacija

Web aplikacija je implementirana u programskom jeziku Java, verzije Java EE 8 Web i izvršava se na poslužitelju `GlassFish 5.1.`, a razvojni okvir u kojem se radi aplikacija je `JSF 2.3`. Razvojno okruženje u kojem je aplikacija razvijana je `Apache NetBeans 12.0`. Projekt je podijeljen u pet izvornih paketa u kojima se nalaze klase koje služe kao poslovna logika aplikacije:

- **org.diskuto.beans** - sadrži komponente na strani poslužitelja zadužene za učajunjenje dijela poslovne logike aplikacije.
 - `BeanHelper` - pomoćno zрно čije metode mogu `Facelets` koristiti. Metode koje sadrži zрно su: `"getActiveUser"` (dohvaća objekt trenutno prijavljenog korisnika u aplikaciji), `"date"` (pretvara UNIX vrijeme u format datuma), `"fullDate"` (pretvara UNIX vrijeme u format datum i vrijeme), `"retrieveDiskuto"` (vraća objekt foruma sa zadanim nazivom u argumentu), `"subscribe"` (pretplaćuje se ili ukida pretplatu na forum zadan kao argument).
 - `ConfirmRegistration` - zрно koje se koristi prilikom potvrde registracije. Sadrži varijable `"insertedCode"` u kojem se zapisuje kod za potvrdu registracije kojeg korisnik unese i `"errorText"` u kojem se zapisuje tekst pogreške koji se korisniku prikazuje. Metode koje sadrži zрно su: `"validate"` (provjerava ispravnost unesenog koda i potvrđuje registraciju ili ispisuje grešku u slučaju pogreške) i `"sendNewMail"` (šalje elektroničku poštu sa kodom za potvrdu registracije).

- Discover - zrno koje se koristi prilikom pretraživanja foruma u aplikaciji. Sadrži varijablu "diskutos" u kojoj u konstruktoru pohranjuje popis foruma iz baze podataka.
- EditDiskuto - zrno za kreiranje i uređivanje foruma. Postoje varijable koje sadrže sve elemente iz XML Scheme za dokument "forums.xml". U konstruktoru se provjerava je li forum zadan kao parametar postoji već u bazi podataka, te ako postoji tada će varijablama pridružiti odgovarajuće vrijednosti iz baze podataka. Zrno sadrži metode "addCategory" koja dodaje kategoriju u listu, "dropCategory" koja briše kategoriju iz liste, "addModerator" koja dodaje moderatora u listu, "dropModerator" koja briše moderatora iz liste, "check" koja provjerava ispravnost unesenih podataka i "save" koja sprema podatke u bazu podataka.
- Forum - zrno koje se koristi za ispis stranice foruma. Sadrži varijable "diskuto" i "category" koje u konstruktoru se provjeravaju postoje li kao parametri, te se dohvaćaju. Na kraju se provjerava ukupan broj objava u forumu i poziva metoda "loadItems" koja učitava objave. U zrnu još postoji metoda "freshPost" koja u argumentu proslijeđenu kategoriju vraća najnoviju objavu.
- Home - zrno koje se koristi pri ispisu početne stranice aplikacije. Sadrži varijablu "messageUsers" koja ispisuje listu korisnika sa kojim korisnik ima nepročitane poruke, "unread" koja ispisuje broj nepročitanih poruka, te ostale varijable koje se koriste pri učitavanju objava pozivom metode "loadItems".
- Language - pomoćno zrno koje se koristi pri dohvaćanju i promjeni jezika aplikacije.
- Login - zrno koje se koristi prilikom korištenja stranice za prijavu korisnika u aplikaciju. Sadrži varijable "username" i "password" u koje se pohranjuju uneseni korisnički podaci prilikom prijave. U metodi "doLogin" provodi se provjera ispravnosti prijave i eventualno se ispisuju greške kroz varijablu "errorText". Metodom "logout" odjavljuje se trenutno prijavljeni korisnik.
- Message - zrno koje se koristi prilikom korištenja stranice za slanje i primanje privatnih poruka od drugih korisnika. Sadrži atribut "me" koji poprima istinitu vrijednost ako je prijavljeni korisnik odabrani korisnik, "ignored" koji poprima istinitu vrijednost ako je odabrani korisnik na listi blokiranih korisnika, "messages" u kojoj se nalazi popis svih poruka sa odabranim korisnikom dohvaćenih u konstruktoru zrna, "chatting" u kojoj se nalazi popis svih korisnika sa kojima je prijavljeni korisnik u prošlosti komunicirao (puni se preko metode "fillIn"), "reply" u koji se pohranjuje poruka koja se šalje metodom "send".
- MyDiskuto - zrno koje se koristi pri ispisu popisa vlastitih foruma. Sadrži varijable "owner" u kojoj se nalazi popis foruma kojih je korisnik vlasnik, te "moderator"u kojem se nalazi popis foruma kojih je korisnik moderator. Navedene varijable se pune pozivom u konstruktoru metode "fillIn".
- NewPassword - zrno koje se koristi prilikom poništavanja korisničke lozinke. Sadrži varijable "state" koja definira u kojoj fazi se nalazi korisnik u poništavanju lozinke, "email" u kojoj se definira elektronička pošta korisnika kojeg se metodom "abolishPassword" poništava lozinka, "code" koji služi za autentikaciju korisnika, "password"

- i "rpassword" u kojima se pohranjuje nova lozinka korisnika metodom "addNewPassword" i "errorText" koji služi za ispis eventualnih grešaka. U zrnju se nalazi još metoda "check" koja provjerava ispravnost unesene nove lozinke.
- NewPost - zrno koje se koristi kod kreiranja nove objave na određenom forumu. Postoje varijable koje sadrže sve elemente iz XML Scheme za dokument "posts.xml". U konstruktoru se dohvaća forum proslijeđen kao parametar, te ako on ne postoji korisnika se preusmjerava na stranicu koja obavještava o pogreški. Metodom "save" se provjerava ispravnost unešenih podataka, te ako je sve u redu objava se sprema u bazu podataka.
 - Post - zrno koje prikazuje objavu i komentare objave. Sadrži varijable "main" koja u konstruktoru dohvaća forum na kojem je objava objavljena, "thing" koja u konstruktoru dohvaća objavu prema zadanom parametru, "comments" u kojem se spremaju komentari objave i "myComment" u kojem se pohranjuje komentar korisnika koji se šalje metodom "sendComment". U zrnju postoje još metode "reportPost" pomoću koje se može prijaviti objava, "reportComment" pomoću koje se može prijaviti komentar, "deletePost" pomoću koje se može obrisati objava, "deleteComment" pomoću koje se može obrisati komentar, "lock" pomoću koje se može zaključati objava, "unlock" pomoću koje se može otključati objava, "upvotePost" pomoću koje se može objava označiti sa +, "downvotePost" pomoću koje se objava može označiti sa -, "upvoteComment" pomoću koje se komentar može označiti sa + i "downvoteComment" pomoću koje se komentar može označiti sa -.
 - Profile - zrno koje se koristi za ispis profila određenog korisnika. Sadrži varijable "user" u kojoj su zapisani svi podaci korisnika koji se dohvaćaju u konstruktoru, "me" koji poprima istinitu vrijednost ako je prijavljeni korisnik odabrani korisnik, "upvotes" broj + oznaka korisnika, "downvotes" broj - oznaka korisnika, "totalPosts" ukupan broj objava, "totalComments" ukupan broj komentara, "post" u kojem se spremaju objave, "comments" u kojem se spremaju komentari, "postsIteratorId" i "commentsIteratorId" koje služe za pomoć pri učitavanju objava i komentara preko metoda "loadPosts" i "loadComments". U zrnju postoji metoda "ignore" koja stavlja odabranog korisnika na listu za ignoriranje.
 - Registration - zrno koje se koristi prilikom registracije novog korisnika. Sadrži varijable "email", "username", "password" i "rpassword" koje korisnik unosi prilikom registracije. Varijablom "errorText" se ispisuje eventualna pogreška nastala provjerom unosa preko metode "check". Metodom "doRegistration" vrši se registracija i sprema se novi korisnik u bazu.
 - Reports - zrno koje se prikazuju prijavljene objave i komentare na određenom forumu. Sadrži varijable "diskuto" u kojem se pohranjuju podaci o odabranom forumu, "items" koja sadrži popis prijavljenih objava i "comments" koja sadrži popis prijavljenih komentara. Metodom "deletePost" se briše odabrana objava, "deleteComment" se briše odabrani komentar, "okPost" se označava da je objava primjerena i briše se sa popisa prijavljenih objava i metodom "okComment" se označava da je komentar primjeren i briše se sa popisa prijavljenih komentara.

- Search - zrno koje se koristi prilikom pretrage. Sadrži varijable "term" u kojoj se zapisuje ključna riječ za pretragu, "userResults" u kojoj se zapisuju korisnici čije korisničko ime ima u sebi ključnu riječ, "diskutoResults" u kojoj se zapisuju forumi čiji nazivi imaju u sebi ključnu riječ, "postResults" u kojoj se zapisuju objave čiji nazivi imaju u sebi ključnu riječ. Metodom "showResults" se prikazuje pretraga po ključnoj riječi.
- Settings - zrno koje se koristi kod promjene postavki korisnika. Sadrži varijable "username" u kojoj se pohranjuje korisničko ime prijavljenog korisnika, "email" u kojem se pohranjuje elektronička pošta korisnika, "wantEmail" koji se postavlja na istinitu vrijednost kada korisnik metodom "changeEmail" želi promijeniti svoj e-mail, "wantPassword" koji se postavlja na istinitu vrijednost kada korisnik metodom "changePassword" želi promijeniti svoju lozinku, "wantDisable" koji se postavlja na istinitu vrijednost kada korisnik metodom "disableAccount" želi obrisati svoj korisnički račun.
- **org.diskuto.helpers** - sadrži pomoćne klase za rad koje koriste zrna i ostali dijelovi aplikacije. Osim već objašnjenih klasa "Database" i "XmlHelper", u paketu se nalaze sljedeće klase:
 - AppHelper - sadrži metode "getActiveUser" koja vraća trenutno prijavljenog korisnika u aplikaciji, "param" koja vraća vrijednost parametra iz Faces konteksta prema zadanom ključu, "getOutput" koja za zadani ključ vraća pripadajući tekstualni niz kod datoteka internacionalizacije, "checkLogged" koja provjerava postoji li prijavljeni korisnik u aplikaciji, "getAttachmentsPath" koja vraća putanju prema direktoriju "attachments" u aplikaciji, "generateId" koja generira jedinstveni identifikator u bilo kojem XML dokumentu, "date" koja pretvara UNIX vrijeme u format običnog datuma, "fullDate" koja pretvara UNIX vrijeme u format datum i vrijeme, "getResourceSet" koja poziva bazu podataka i vraća skup resursa za definirani upit, "getResource" koja poziva bazu podataka i vraća prvi resurs za definirani upit, "login" koja poziva bazu podataka i provjerava jesu li proslijeđeni podaci za prijavu ispravni, "usernameExist" koja poziva bazu podataka i provjerava postoji li korisnik sa proslijeđenim korisničkim imenom u bazi, "forumExist" koja poziva bazu podataka i provjerava postoji li forum sa zadanim nazivom u bazi, "regex" koji vrši provjeru regularnim izrazom.
 - MailHelper - pomoćna klasa za slanje e-mail poruka. Sadrži varijable "to" koja označava e-mail korisnika kojem se šalje mail, "subject" koja označava predmet i "text" koja označava sadržaj elektroničke pošte. Vrijednosti varijabli se postavljaju pri instanciranju novog objekta klase. Klasa sadrži metodu "sendMail" koja na navedeni mail šalje poruku sa definiranim predmetom i tijelom poruke.
 - Retriever - pomoćna klasa koja služi za dohvaćanje objekata iz baze podataka. Sadrži varijablu "key" koja predstavlja ključ za dohvaćanje pojedinog podatka, a vrijednost varijable se postavlja pri instanciranju novog objekta klase. U klasi se nalaze metode: "user" koja dohvaća korisnika iz baze podataka sa određenim korisničkim imenom, "forum" koja dohvaća forum iz baze podataka sa određenim nazivom foruma, "post" koja dohvaća objavu iz baze podataka sa određenim nazivom objave,

"messages" koja dohvaća listu poruka koje trenutno prijavljeni korisnik ima sa određenim korisnikom, "searchUsers" koja dohvaća sve korisnike koji u svom korisničkom imenu sadrže određenu ključnu riječ, "searchForum" koja dohvaća sve forume koji u svom nazivu sadrže određenu ključnu riječ, "searchPosts" koja dohvaća sve objave koje u svom nazivu sadrže određenu ključnu riječ.

- **org.diskuto.listeners** - paket koji sadrži klasu "Listener" u kojoj se sluša životni ciklus web aplikacije. Sadrži jednu javnu statičku varijablu "sc" u kojoj se prilikom inicijalizacije prema kontekst servleta. Pomoću konteksta se može komunicirati sa servlet containerom. Klasa sadrži metode za zapisivanje, brisanje i dohvaćanje varijabli iz sesije: "addToSession" upisuje u sesiju objekt pod zadanim atributom, "deleteFromSession" briše iz sesije objekt pod zadanim atributom, "getFromSession" dohvaća objekt iz sesije pod zadanim atributom.
- **org.diskuto.models** - paket koji preslikava modele dokumenata iz baze podataka. Svaka klasa sadrži attribute definirane u XML Schemi kojima se dodjeljuju vrijednosti pozivom metode "retrieve" prosljeđivanjem objekta klase XmlHelper u kojem se nalazi XML zapis dohvaćen iz baze podataka. Svaka klasa ima metode za pridruživanje i postavljanje vrijednosti varijablama klase. Od ostalih metoda, klasa Comment sadrži: "save" pomoću kojeg se sprema komentar u bazu podataka, "addVote" pomoću kojeg se postavlja + ili - za komentar, "dropVote" pomoću kojeg se briše + ili - sa komentara, "post" pomoću kojeg se dohvaća objekt Post za komentar, "report" pomoću kojeg se prijavljuje komentar, "itsOk" pomoću kojeg se briše komentar sa liste prijava, "delete" pomoću kojeg se briše komentar. Klasa Forum još sadrži metode "save" kojom se sprema novi forum u bazu podataka i "update" kojom se ažurira forum u bazi podataka. Klasa Message sadrži metodu "send" pomoću koje se šalje privatna poruka od određenog pošiljatelja prema određenom primaatelju uz određeni tekst poruke. Klasa Post sadrži metode "save" kojom se objava sprema u bazu podataka, "addVote" pomoću kojeg se postavlja + ili - za objavu, "dropVote" pomoću kojeg se briše + ili - sa objave, "report" pomoću kojeg se prijavljuje objava, "itsOk" pomoću kojeg se briše objava sa liste prijava, "delete" pomoću kojeg se briše objava, "lock" kojom se zaključava objava i "unlock" kojom se otključava objava. Klasa User još sadrži metode "register" kojom se sprema novi korisnik u bazu podataka, "login" kojom se prijavljuje korisnik u aplikaciju, "logout" kojom se odjavljuje korisnik iz aplikacije, "disable" kojom se briše korisnički račun, "sendConfirmMail" kojom se šalje e-mail za potvrdu korisničkom računa, "confirmUser" kojom se potvrđuje korisnički račun, "changeLanguage" kojom se zapisuje promjena jezika u bazi podataka, "subscribe" kojom se korisnik pretplaćuje na određeni forum, "unsubscribe" kojom korisnik ukida pretplatu sa određenog foruma, "ignore" kojom se stavlja određeni korisnik na listu za ignoriranje i "unignore" kojom se briše korisnik sa liste za ignoriranje.
- **org.diskuto.translations** - u paketu se nalaze datoteke tipa Properties u kojima se za svaki jezik koji podržava aplikacija (engleski kao zadani i hrvatski) nalaze atributi kojima se pridružuju sadržaji za ispis na pojedinim web stranicama. Datoteke koje se nalaze u paketu su: "language.properties" (za zadani jezik), "language_en.properties" (za engleski jezik) i "language_hr.properties" (za hrvatski jezik).

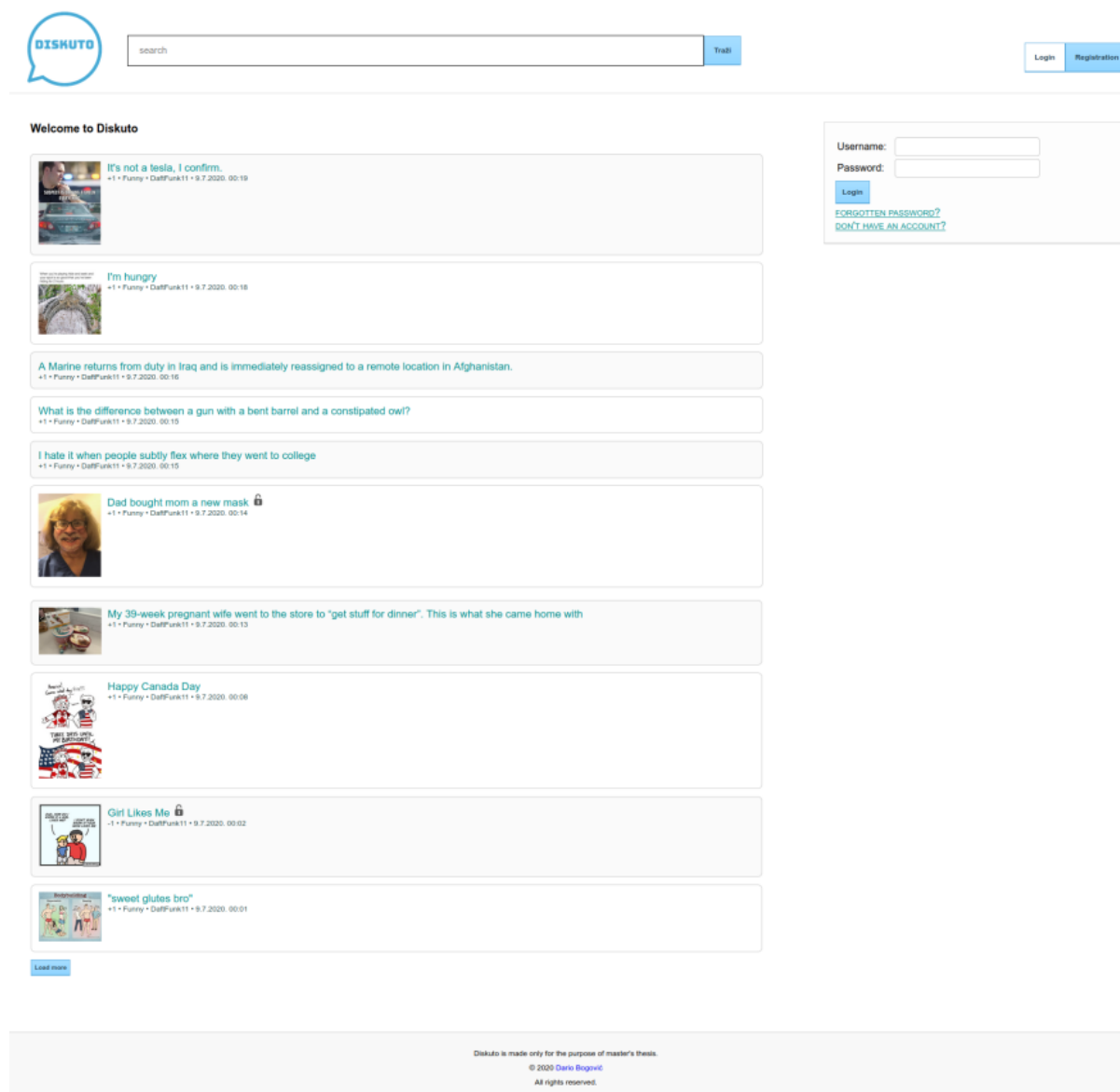
Od ostalih datoteka u Java web-aplikaciji nalaze se .xhtml dokumenti koji su detaljnije objašnjeni u poglavlju 8 uz prikaz snimke zaslona i detaljan opis funkcionalnosti. U aplikaciji postoje sljedeće konfiguracijske datoteke: "web.xml" (definira slušače, Faces Servlet, trajanje sesije, početnu stranicu, stranicu koja se prikazuje prilikom pogrešaka i sl.), "glassfish-web.xml" (konfiguracijska datoteka na razini poslužitelja koji izvršava aplikaciju), "faces-config.xml" (konfiguracijska datoteka na razini JSF razvojnog okvira u kojoj se deklariraju jezici i paketi za lokalizaciju). U aplikaciji postoji direktorij "resources" koji je podijeljen na nekoliko sljedećih poddirektorija:

- **attachments** - u direktoriju se pohranjuju slike koje se mogu učitati kao prilog prilikom kreiranja nove objave. Svaka slika ima naziv u obliku "res-<id objave>".
- **css** - u direktoriju se nalaze .css datoteke kojim se uređuje izgled i raspored stranica.
- **js** - u direktoriju se nalaze .js datoteke koje se izvršavaju kada aplikacija pozove Ajax zahtjev. Pomoću njih se definira pokretanje animacije na određenim mjestima prilikom čekanja na dovršavanje zadatka kojeg aplikacija izvršava.
- **pages** - u direktoriju se nalaze razne dodatne vanjske datoteke i slike poput ikone aplikacije, animacije koja se pokreće prilikom čekanja na dovršavanje zadatka kojeg aplikacija izvršava, te razne druge ikone i logotipi.

8. Primjeri korištenja

U ovom poglavlju su opisani primjeri korištenja izrađene web aplikacije. Svaka izrađena funkcionalnost je prikazana uz odgovarajuće slike zaslona i njihov opis.

8.1. Početna stranica

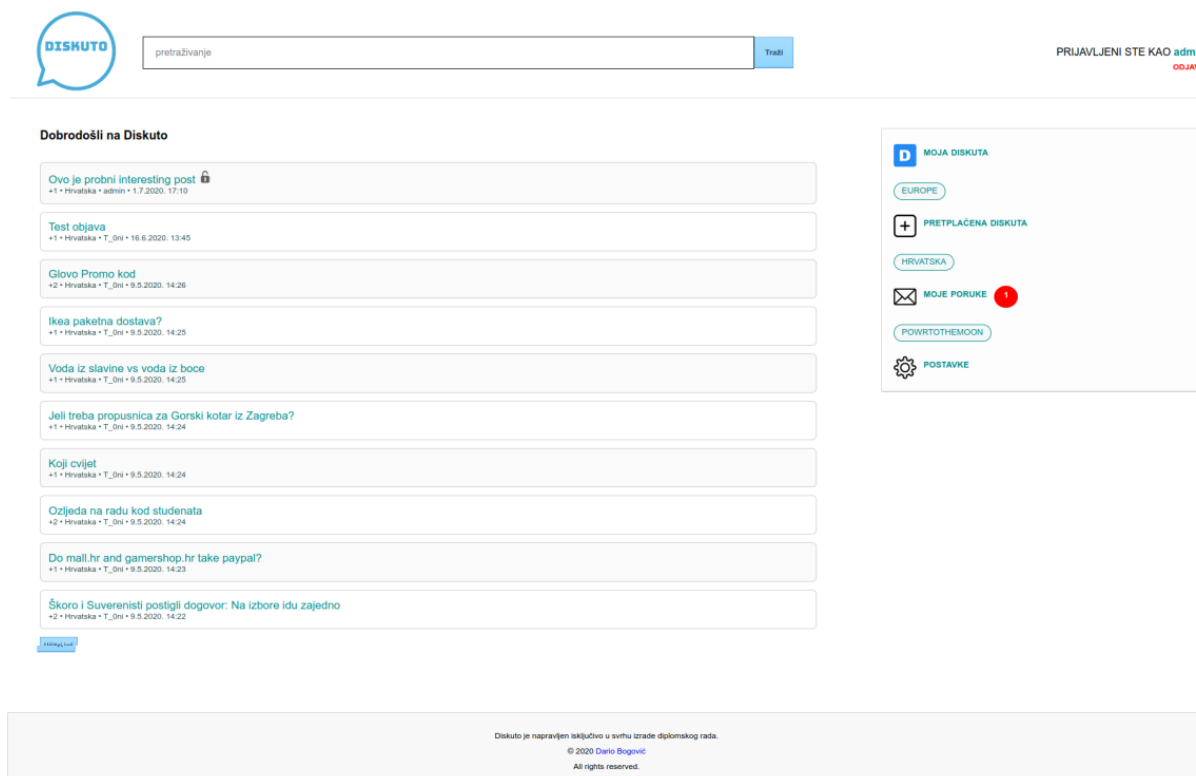


Slika 11: Prikaz početne stranice (neprijavljen korisnik)

Na slici 11 je prikazana početna stranica web aplikacije. Sastoji se od tri dijela: zaglavlja stranice, podnožja stranice i glavnog dijela stranice. Zaglavlje stranice sadrži s lijeva na desno: logo, pretraživač web aplikacije i dvije poveznice koje vode na prijavu korisnika u aplikaciju ili na registraciju korisnika. Glavni dio početne stranice je prikazan u sredini i sadrži s desne strane blok element za brzu prijavu korisnika s poveznicama koje vode do forme za obnavljanje zabo-

ravljene lozinke ili za registraciju novog korisnika. S lijeve strane glavnog dijela web aplikacije nalazi se poruka dobrodošlice i ispod koje poveznice na 10 najnovijih objava u aplikaciji. Na kraju se nalazi gumb za učitavanje dodatnih 10 objava. Podnožje stranice sadrži detalje o web aplikaciji i poveznicu na e-mail adresu autora.

8.1.1. Početna stranica prijavljenog korisnika



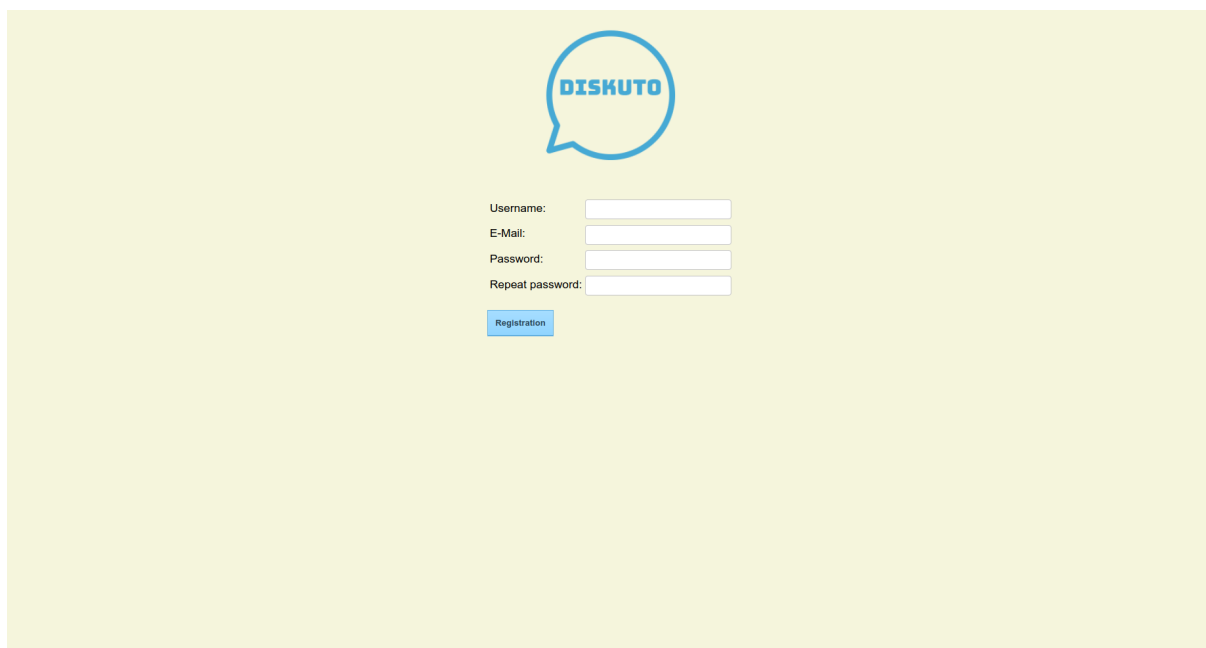
Slika 12: Prikaz početne stranice (prijavljen korisnik)

Prijavom korisnika, bilo preko stranice namijenjene prijavi ili preko brze prijave, korisnika se preusmjerava na njegovu početnu stranicu. Za razliku od početne stranice neprijavljenog korisnika, kod prijavljenog korisnika ona sadrži u glavnom dijelu s lijeve strane popis najnovijih objava sa foruma na koje se korisnik pretplatio (za razliku od popisa najnovijih objava sa svih foruma koji se prikazuje neprijavljenom korisniku), a sa desne strane glavni izbornik sa mogućnostima. U glavnom izborniku može se odabrati opcija "Moja Diskuta" koja korisnika vodi na stranicu sa svojim kreiranim forumima na kojem može kreirati forume ili im mijenjati postavke. Ispod poveznice na opciju se nalazi popis svih vlasititih foruma, te se klikom na njih može brzo pristupiti. Sljedeća mogućnost je "Pretplaćena Diskuta" koja korisnika vodi na stranicu za otkrivanje novih foruma na koje bi se korisnik mogao pretplatiti. Ispod poveznice se nalazi popis svih foruma na koje se korisnik pretplatio, te se njima može brzo pristupiti. Mogućnost "Moje poruke" korisnika vodi do stranice sa privatnim porukama. S desne strane se može vidjeti broj nepročitanih poruka koje korisnik ima, a ispod poveznice se može brzo pristupiti razgovoru s korisnicima s kojima ima nepročitanih poruka. Zadnja opcija u glavnom izborniku su "Postavke" koje korisnika vode do stranice sa postavkama korisničkog računa. U zaglavlju stranice,

s desne strane, je prikazan naziv trenutno prijavljenog korisnika. Pritiskom na naziv otvara se javni profil korisnika. Ispod naziva, nalazi se poveznica "Odjava" koja korisnika odjavljuje iz aplikacije.

8.2. Registracija

Na početnoj stranici kod neprijavljenog korisnika, klikom na poveznicu "Registration" u zaglavlju stranice ili "Don't have an account?" u glavnom dijelu stranice, otvara se web stranica sa formom za registraciju korisnika u aplikaciji prikazana na slici 13.



Slika 13: Forma za registraciju novog korisnika

Prilikom registracije, korisnik mora unijeti željeno korisničko ime, e-mail, lozinku i ponoviti lozinku. Postoje nekoliko ograničenja prilikom registracije koja se moraju poštovati:

- Svi podaci moraju biti obavezno uneseni
- Korisničko ime i e-mail ne smiju postojati u bazi; svako korisničko ime se veže za isključivo jednog korisnika, te svaki e-mail se veže isključivo za jednog korisnika
- Korisničko ime ne smije biti kraće od 3 znaka, lozinka ne smije biti kraća od 8 znakova, a e-mail mora biti pravilnog formata
- Unesena lozinka i ponovljena lozinka moraju biti jednake

Ukoliko neko ograničenje nije zadovoljeno, korisniku će se na zaslon ispisati poruka sa greškom. Tek kada je sve u redu sa unešenim podacima, u bazu se spremaju podaci i korisnika se preusmjerava na formu za potvrdu registracije. U međuvremenu, korisniku se dodijeljuju još neki meta-podaci: generira se kod za potvrdu koji će korisniku biti poslan na njegov e-mail,

dodijeljuje mu se datum i vrijeme kada mu je kreiran račun, te se kao jezik aplikacije postavlja se engleski. Na kraju se šalje e-mail za potvrdu registracije.

8.2.1. Potvrda registracije

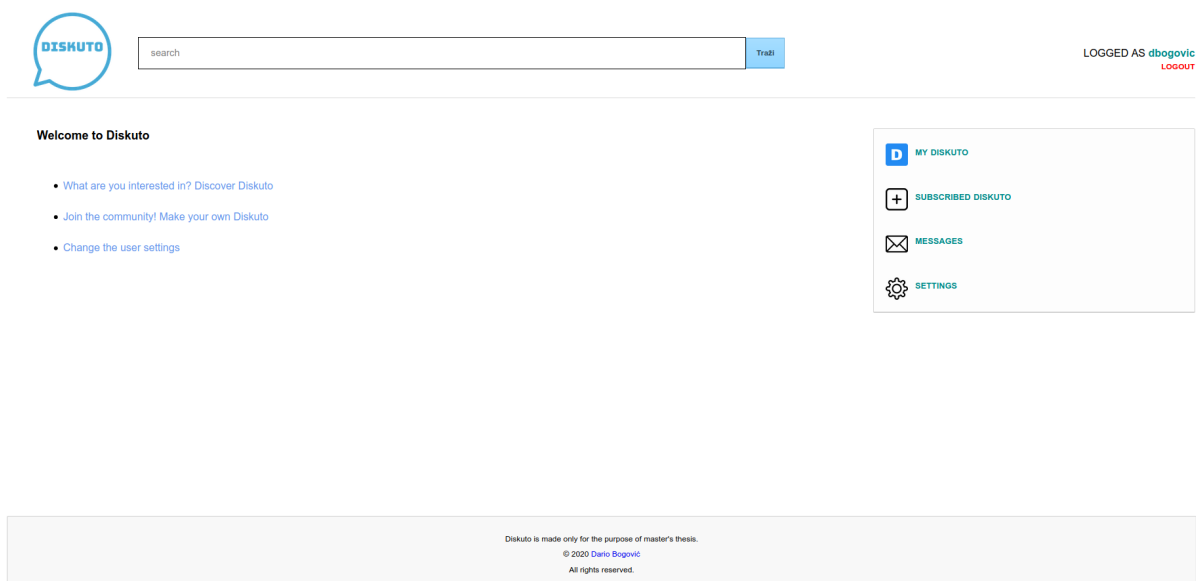
Slika 14: Forma za potvrdu registracije novog korisnika

Na slici 14 je prikazana forma za potvrdu registracije novog korisnika. Kako bi potvrdio registraciju, korisnik mora unijeti šesteroznamenasti kod kojeg je primio na mail. Ukoliko krivo unese kod, prikazuje mu se poruka greške. Ako je bilo problema oko primanja koda, korisnik može zatražiti ponovno slanje na njegov e-mail klikom na poveznicu "Send again". Tek kada unese ispravan kod, korisniku se potvrđuje registracija i postaje član Diskuta. Nepotvrđeni korisnici ne mogu se služiti aplikacijom, odnosno imaju iste ovlasti kao i drugi neregistrirani korisnici.

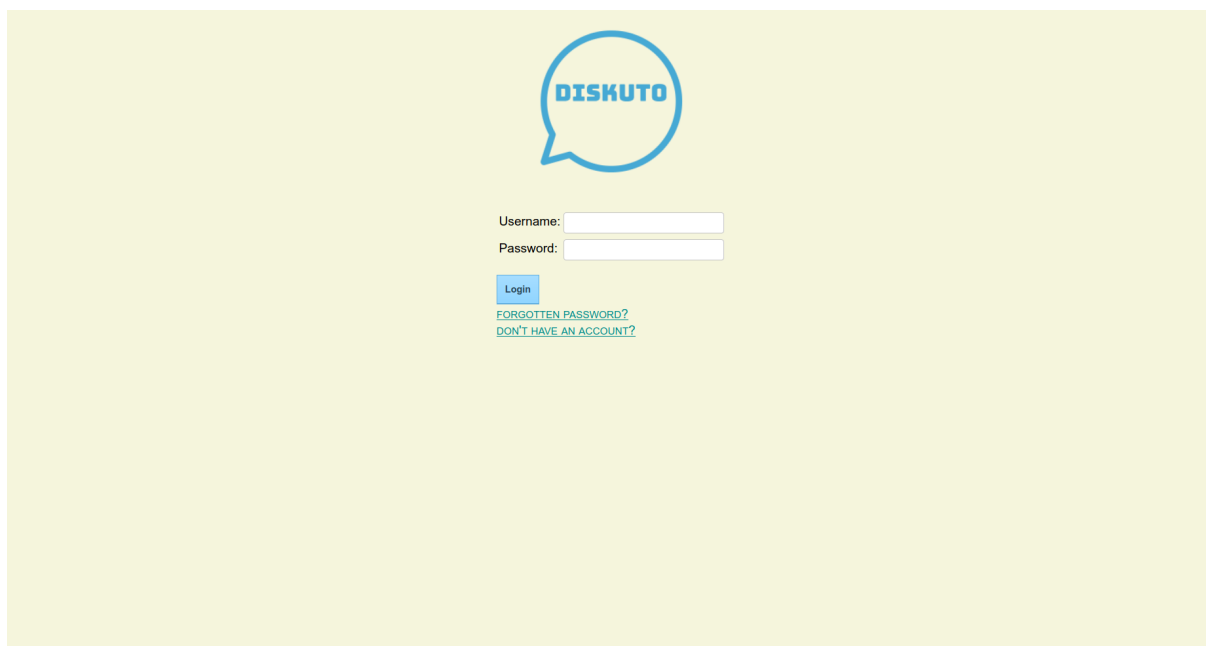
Potvrdom registracije, korisnika se preusmjerava na početnu stranicu koja je prikazana na slici 15. Ona sadrži sve kao i početna stranica za druge korisnike, osim što na mjestu gdje bi trebali biti prikazane objave prikazuje se neke smjernice za novog korisnika. Savjetuje mu se da otkrije forume i pretplati se na njih, da ima mogućnost kreirati svoje forume i uređivati ih, te da pogleda stranicu sa postavkama gdje može urediti svoje korisničke podatke ili promijeniti jezik aplikacije.

8.3. Prijava

U zaglavlju početne stranice, odabirom na poveznicu "Login", korisnika se preusmjerava na stranicu za prijavu prikazanu na slici 16. Kako bi se prijavio u aplikaciju, korisnik je dužan unijeti svoje korisničko ime i lozinku. Jedino sa ispravnim korisničkim imenom i lozinkom korisnik se može prijaviti u aplikaciju, inače mu se javlja poruka o neispravno unesenim po-



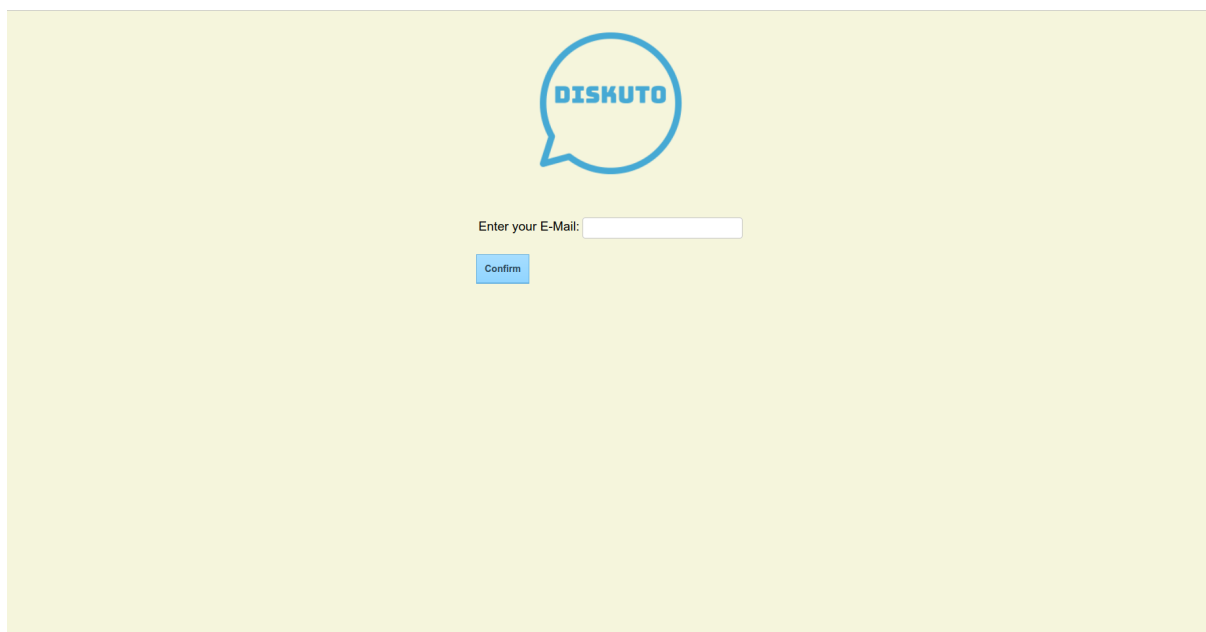
Slika 15: Prikaz početne stranice za novoregistriranog korisnika



Slika 16: Forma za prijavu korisnika

dacima. Ako korisnik nema korisnički račun, pritiskom na poveznicu "Don't have an account?" preusmjerava ga se na stranicu za registraciju. Ako je korisnik zaboravio lozinku, klikom na poveznicu "Forgotten password?" otvara mu se stranica za poništavanje lozinke.

Korisnik mora unijeti svoj e-mail na kojem će se poslati poveznica za kreiranje nove lozinke. Ako je e-mail nepostojeći u bazi, javit će se poruka greške. Nakon unosa pravilnog e-maila, korisniku se poništava stara lozinka i šalje mu se poruka sa uputama za kreiranje nove lozinke. U poruci će dobiti poveznicu koja ga vodi do forme za kreiranje nove lozinke. Moraju se poštivati ista pravila pri unosu lozinke kao i kod registracije (ne smije imati manje od 8 znakova i lozinka i ponovljena lozinka moraju biti jednake), u suprotnosti javit će mu se greška i korisnik

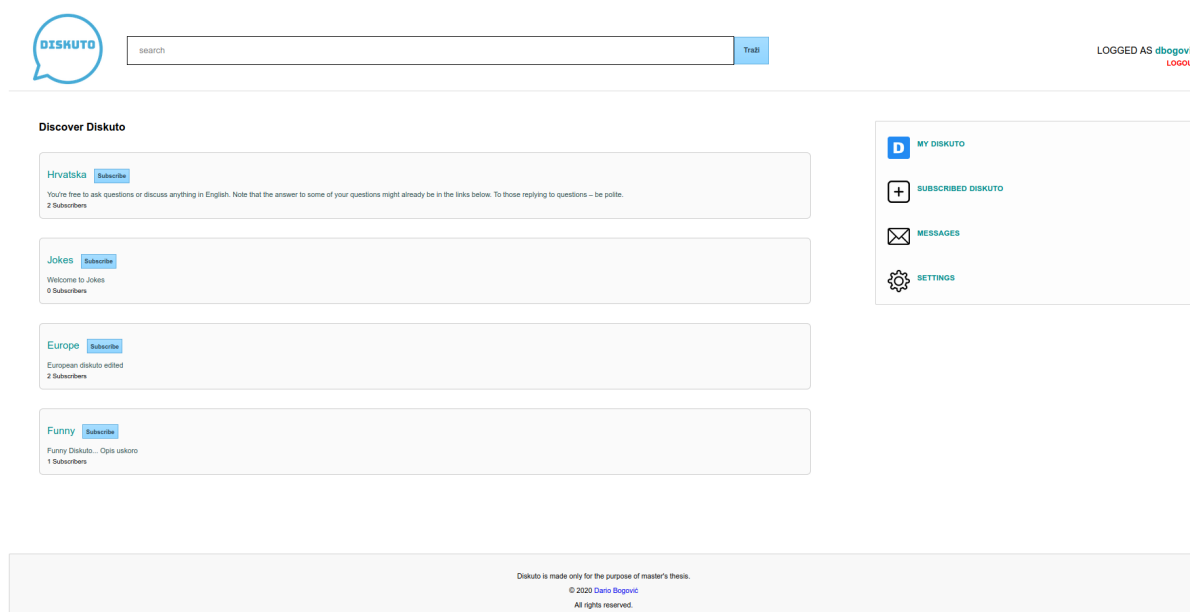


Slika 17: Stranica za poništavanje lozinke

neće moći obnoviti lozinku. Sve dok korisnik ne zada novu lozinku neće se moći prijaviti u aplikaciju. Obnovom lozinke korisnik se može prijaviti u aplikaciju i služiti se njome.

8.4. Pronalaženje Diskuta

Na početnoj stranici u glavnom izborniku, odabirom opcije "Pretplaćena Diskuta" (eng. "Subscribed Diskuto") korisniku se otvara stranica za pronalaženje zanimljivih foruma na koje bi se mogao pretplatiti i pratiti ih. Izgled ove stranice je prikazan na slici 18.



Slika 18: Stranica za otkrivanje Diskuta

Korisniku se prikazuje popis svih foruma koji se nalaze u bazi podataka. Za svaki pojedinačni forum je prikazan naziv, opis i broj pretplaćenih korisnika. Klikom na naziv, korisnika se preusmjerava na odabrani forum. Kraj naziva postoji jedan gumb preko kojeg je moguće pretplatiti se, odnosno ukinuti pretplatu ako već postoji pretplata na njega. Pretplatom na forum, korisnik postaje njegov član i na početnoj stranici prima najnovije objave koje su objavljene na njemu. Pretplaćeni forumi se prikazuju u glavnom izborniku odmah ispod opcije za pretragu foruma, čime korisnik može na brz i jednostavan način svojim forumima pristupiti.

8.5. Kreiranje novog Diskuta

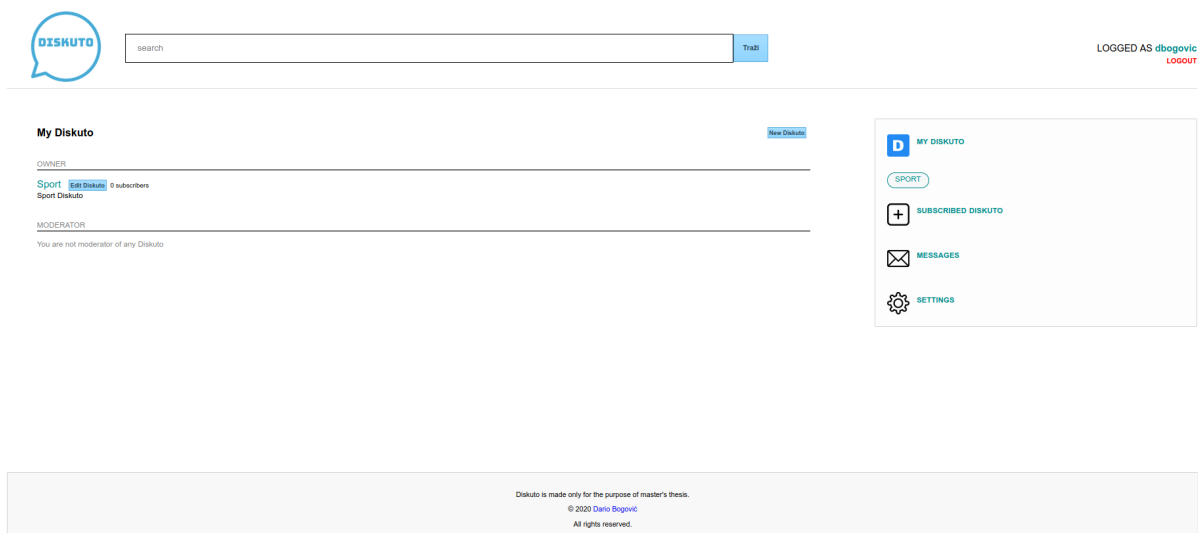
Kako bi kreirao novi forum, korisnik se mora prebaciti na stranicu "Moja Diskuta" (eng. "My Diskuto") tako da odabere opciju u glavnom izborniku. Korisniku se tada prikazuje popis svih foruma kojih je vlasnik i na kojima je moderator. Pritiskom na gumb "Novi Diskuto" (eng. "New Diskuto"), korisniku se otvara forma za kreiranje novog foruma.

The screenshot shows the 'New Diskuto' form. It has a header with the 'DISHUTO' logo, a search bar, and a 'Traži' button. The user is logged in as 'dbogovic'. The form itself is titled 'New Diskuto' and contains several input fields: 'Name:*' (with 'Sport' entered), 'Description:*' (with 'Sport Diskuto' entered), 'Rules:*' (with 'Some rules...' entered), 'Categories:*' (with a dropdown arrow), and 'Moderators:' (with a dropdown arrow). Below these fields are two lists: 'LIST OF CATEGORIES' showing 'Football', 'Basketball', 'Tennis', and 'Formula 1' with checkboxes, and 'NO MODERATORS' which is currently empty. A 'Spremi' button is at the bottom left. On the right side, there is a sidebar with a 'D' icon and links for 'MY DISKUTO', 'SUBSCRIBED DISKUTO', 'MESSAGES', and 'SETTINGS'.

Slika 19: Forma za kreiranje novog Diskuta

Obavezno je unijeti naziv foruma, njegov opis, pravila, te barem jednu kategoriju koju će forum sadržavati. Opcionalno je moguće dodati moderatore koji će moderirati rasprave i sankcionirati eventualna kršenja pravila. Naziv foruma je konačan i njega se nakon kreiranja ne može promijeniti, dok svi drugi atributi mogu kasnije biti modificirani. Postoji ograničenje da ne smije postojati forum sa istim nazivom u bazi. U slučaju greške korisniku će se ispisati poruka, a tek kada svi podaci budu pravilno uneseni podaci se spremaju u bazu i forum postaje kreiran.

Korisnika se preusmjerava na stranicu sa popisom foruma kojima je vlasnik ili moderator. Novokreirani forum se prikazuje u tablici, kao i u glavnom izborniku sa desne strane. Klikom na gumb "Uredi Diskuto" (eng. "Edit Diskuto"), korisnik može promijeniti opis i pravila foruma, brisati i dodavati kategorije, te korisnicima davati ili uklanjati ovlasti za moderatora foruma.



Slika 20: Popis mojih Diskuta

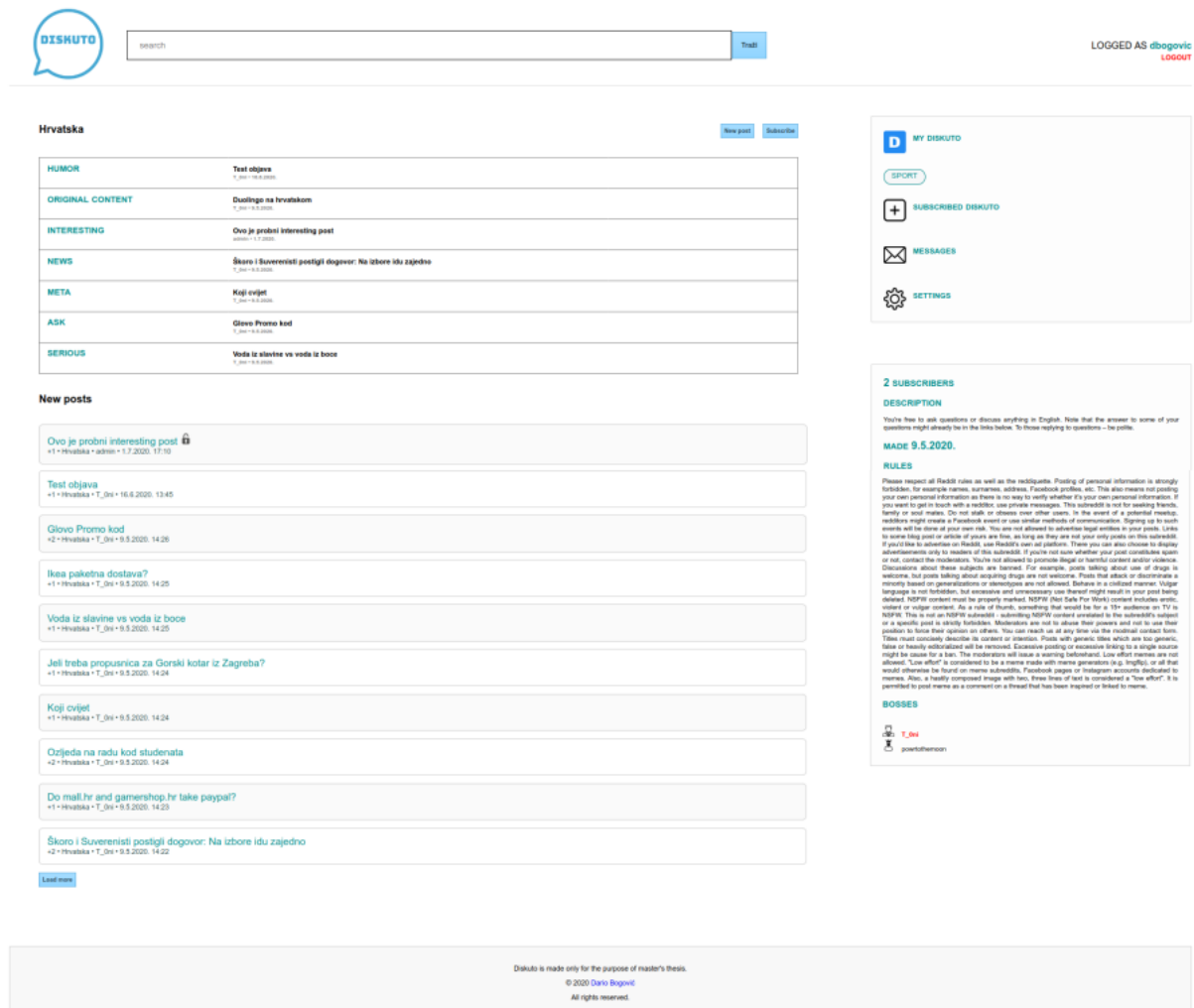
8.6. Pregled Diskuta

Na slici 21 je prikazan izgled stranice foruma za korisnika koji nije vlasnik ili moderator. Ispod glavnog izbornika aplikacije prikazan je blok sa detaljima određenog foruma: broj pretplatnika, opis, datum kreiranja, pravila foruma i poveznice na profile vlasnika i moderatora foruma. Korisnik se može pretplatiti na forum klikom na gumb "Pretplati se" (eng. "Subscribe"), a postaviti novu objavu klikom na gumb "Nova objava" (eng. "New Post"). U glavnom dijelu aplikacije se nalazi popis svih kategorija koje forum sadrži, te sa desne strane poveznicu na najnoviju objavu na pojedinoj kategoriji. Klikom na pojedinu kategoriju otvara se popis objava sa te kategorije. Ispod elementa sa kategorijama, nalazi se popis 10 najnovijih objava na forumu i gumb za učitavanje dodatnih 10 objava. Nad svakom objavom je prikazan naslov objave odnosno poveznica nad svaku objavu, karma (odnos pluseva i minusa) koju ima, forum na kojem je objavljena, poveznica na profil vlasnika objave i datum i vrijeme kreiranja objave.

8.7. Kreiranje nove objave

Klikom na gumb za kreiranje nove objave nad pojedinim forumom, korisnika se preusmjerava na formu za kreiranje objave (slika 22). Prikazan je forum nad kojem se kreira nova objava, a korisnik mora unijeti naslov i dodatno opisati svoju objavu ili učitati sliku. Obavezno je odabrati i kategoriju na koju želi da se objava spremi. Slika koja se unosi mora biti u slikovnom formatu, a ako nije ili ako korisnik je nešto zaboravio unijeti prikazuje mu se poruka greške. Tek kada je sve u redu objava se sprema u bazu i postaje vidljiva na forumu.

Izgled objave je vidljiv na slici 23. Prikazan je naslov objave, poveznica do stranice kategorija foruma na kojem je ona objavljena, karma (odnos pluseva i minusa) koju ima, fo-



Slika 21: Izgled stranice foruma

rum na kojem je objavljena, autor i vrijeme kreiranja objave. Sa lijeve strane se nalazi gumb za označavanje objave sa "+" čime se povećava karma objave i karma autora, odnosno sa "-" čime se smanjuje karma objave i karma autora. Prema zadanome autor objave automatski sam sebi zadaje "+", ali naknadno može si maknuti "+" ili si dodijeliti "-". U svakom trenutku autor objave može obrisati svoju objavu. Ako drugi korisnici foruma smatraju da objava krši pravila foruma, mogu ju prijaviti moderatorima. Ispod naslova objave slijedi tekstualni opis ili slika objave, ovisno o tome što je autor stavio kao opis. Ispod opisa se nalazi tekstualno polje za unos komentara koji se sprema u bazu pritiskom na gumb ispod njega. Zatim se nalazi element sa komentarima koji ispisuje broj ukupnih komentara i sadržaj komentara. Nad svakim komentarom se ispisuje autor objave i poveznica do njegovog profila, karma (odnos pluseva/minusa), datum i vrijeme kada je komentar postavljen, tekst komentara i gumbi sa označavanjem komentara kao "+" čime se povećava karma komentara i njegovog autora, odnosno kao "-" čime se smanjuje karma komentara i njegovog autora. Prema zadanome autor komentara automatski sam sebi zadaje "+", ali naknadno može si maknuti "+" ili si dodijeliti "-". Ispod se nalazi sami tekst komentara. U svakom trenutku autor komentara može obrisati svoj komentar. Ako drugi korisnici foruma smatraju da komentar krši pravila foruma, mogu ga prijaviti moderatorima.

DISKUTO

search Trazi

LOGGED AS dbogovic LOGOUT

New post

Diskuto:
Sport

Headline:

Description:

or upload an image:
Browse... No file selected.

Category:
Football

Save

MY DISKUTO

SPORT

SUBSCRIBED DISKUTO

HRVATSKA

MESSAGES

SETTINGS

Diskuto is made only for the purpose of master's thesis.
© 2020 Damo Bogovic
All rights reserved.

Slika 22: Forma za kreiranje nove objave

DISKUTO

search Trazi

LOGGED AS dbogovic LOGOUT

Tko će uz Dinamo biti drugi klub koji ide u kvalifikacije za ligu prvaka?

Kraj sezone je izrazito zanimljivo, jako blizu su Lokomotiva, Rijeka, Hajduk i Osijek.

Type the comment

Send

Comments (4)

admin
9 • 11.7.2020. 18:12 • Delete
Lokomotiva!!!

powerthemoon
+1 • 11.7.2020. 18:13 • Delete
Jako dobra tema... mislim da Hajduk nema šanse, ali Rijeka je u jakoj dobroj formi

T_0ni
+2 • 11.7.2020. 18:13 • Delete
joj admin, kakva lokomotiva

T_0ni
+1 • 11.7.2020. 18:14 • Delete
sigurno Hajduk!!

MY DISKUTO

SPORT

SUBSCRIBED DISKUTO

HRVATSKA

MESSAGES

SETTINGS

0 SUBSCRIBERS

DESCRIPTION

Sport Diskuto

MADE 11.7.2020.

RULES

Some rules...

BOSSSES

dbogovic

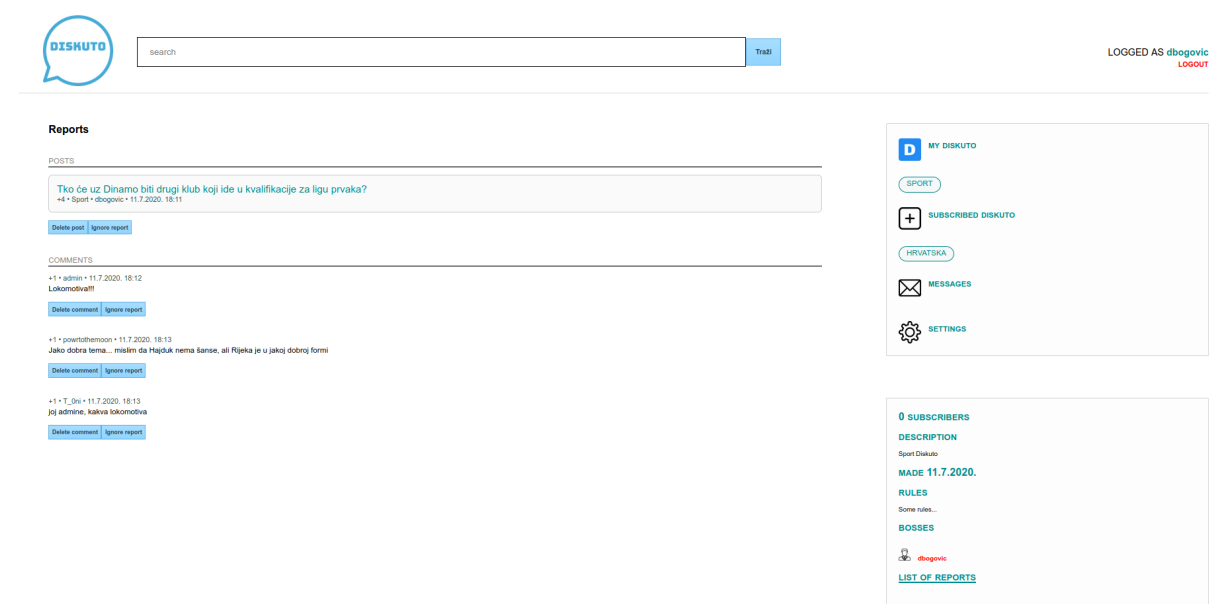
LIST OF REPORTS

Slika 23: Izgled objave

8.8. Ovlasti moderatora nad pojedinim Diskutom

Kao što je već opisano u poglavlju kod kreiranja foruma, moderatori imaju naprednije ovlasti nad forumom od običnih korisnika. Imaju iste ovlasti kao i vlasnik, a oni se dodaju i brišu na stranici za uređivanje i brisanje foruma. Razlika u odnosu na vlasnika je jedino ta što je vlasnik "doživotni moderator", odnosno on ima trajne i neopozive ovlasti nad forumom. Moderatoru mogu obrisati bilo koju objavu ili komentar za koji smatraju da je neprimjeren, za razliku od običnih korisnika koji jedino mogu obrisati svoju objavu/komentar. Moderatoru mogu

zaključati objavu i tako onemogućiti komentiranje na njoj. U svakom trenutku objava se može otključati i tako omogućiti korisnicima da nastave komentirati. Zaključane objave imaju kraj svog naslova ikonu lokota, čime se razlikuju od drugih otključanih objava. Drugi korisnici mogu moderatorima prijaviti objave ili komentare za koje smatraju da su neprimjerene. U elementu opisa foruma, moderatori klikom na poveznicu "Popis prijava" (eng. "List of reports") mogu otvoriti stranicu sa popisom svih prijavljenih objava/komentara na njihovom forumu.

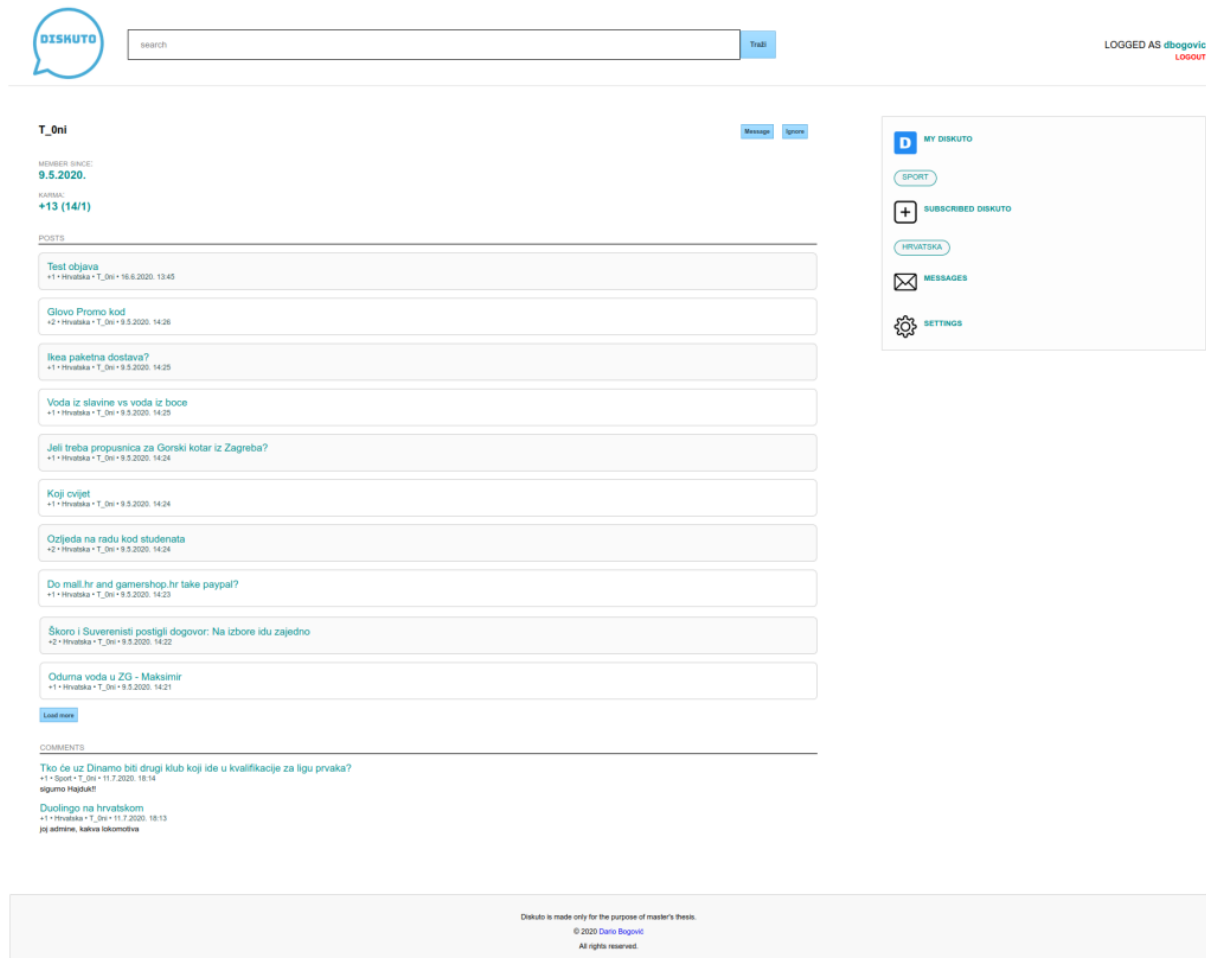


Slika 24: Popis prijavljenih objava/komentara

Moderatori mogu odlučiti obrisati objavu/komentar ako smatraju da krši pravila foruma ili zanemariti prijavu ako smatraju da je sve u redu.

8.9. Profil korisnika

Bilo gdje na stranici kada se klikne na korisničko ime pojedinog korisnika, preusmjerava se na stranicu njegovog profila kao na slici 25. U glavnom dijelu stranice vidljivo je korisničko ime i gumbi za slanje poruke korisniku i za ignoriranje korisnika. Kada se klikne gumb za slanje poruke, korisnika se preusmjerava na stranicu sa porukama na kojoj može vidjeti prethodnu komunikaciju sa odabranim korisnikom (ako je bilo) i ima mogućnost poslati mu poruku. Naravno to je moguće ako korisnik nije drugog korisnika blokirao (ili obratno). Klikom na gumb za ignoriranje korisnik više neće vidjeti objave i komentare odabranog korisnika, kao ni primati i slati privatne poruke od njega. Deblokirati ga se može u postavkama korisničkog računa ili ako korisnik ponovno ode na njegov profil i klikne gumb za deblokiranje. Ispod naziva korisničkog imena i gumbova, mogu se vidjeti detalji o odabranom korisniku poput datuma kada je kreiran korisnički račun i ukupne karme (razlika između pluseva i minusa na svim objavama i komentarima). Ispod ovog elementa, navedene su sve objave i svi komentari pojedinog korisnika. Radi boljih performansi stranice navedeno je samo 10 njih, a klikom na gumb za učitavanje može se učitati dodatnih 10 objava/komentara. Kod objava vidljiv je naslov objave čijim odabirom se



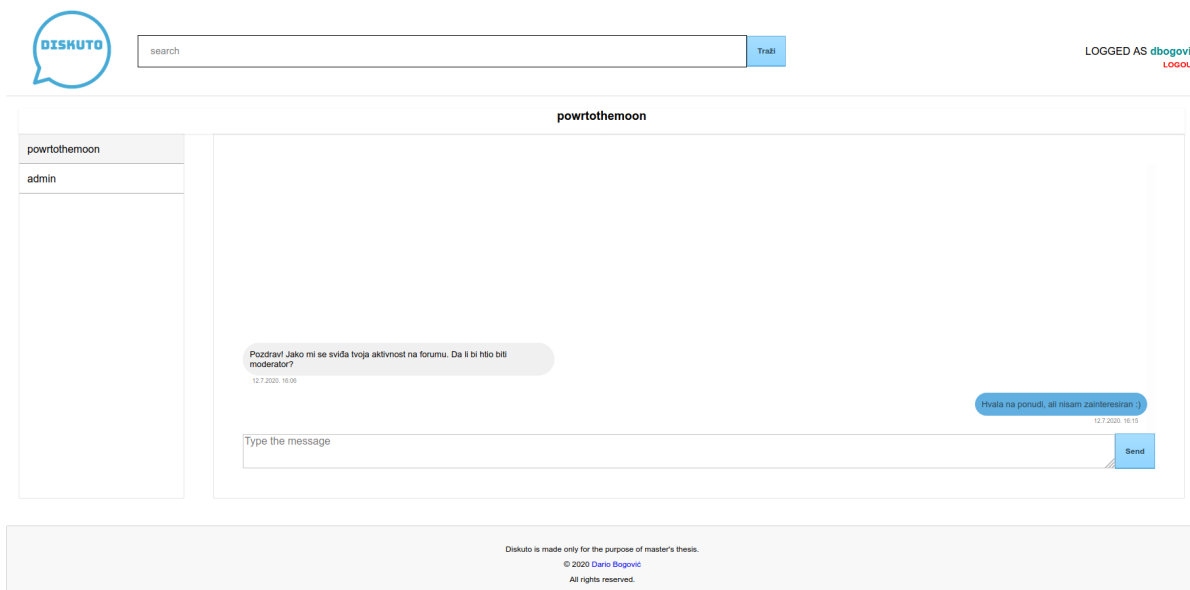
Slika 25: Profil korisnika

preusmjerava na samu objavu, te detalji o objavi: karma, datum kreiranja i forum na kojem je objavljen. Kod komentara vidljiv je naslov objave na kojoj je komentar objavljen i čijim odabirom se preusmjerava na samu objavu, detalji poput karme, datuma kreiranja, foruma na kojem je objavljen, te na kraju sami tekst komentara.

8.10. Poruke

Osim što korisnici mogu javno komunicirati, oni mogu slati jedni drugima i privatne poruke. Korisnik inicira razgovor sa drugim korisnikom odabirom gumba za slanje poruka na njegovom privatnom profilu. Tada se korisnika preusmjerava na stranicu sa porukama kojoj se može pristupiti i sa glavnog izbornika odabirom opcije "Poruke" (eng. "Messages").

Prikaz stranice za slanje privatnih poruka je vidljiv na slici 26. Između zaglavlja i podnožja stranice nalazi se glavni dio stranice koji se sastoji od: korisničkog imena korisnika sa kojim je trenutno aktivna komunikacija i poveznica na njegov profil, s lijeve strane popis svih korisnika sa kojima je korisnik komunicirao u prošlosti i poveznica na njihovu stranicu komunikacije, te u glavnom dijelu se nalaze same poruke koje su korisnici međusobno razmijenili.



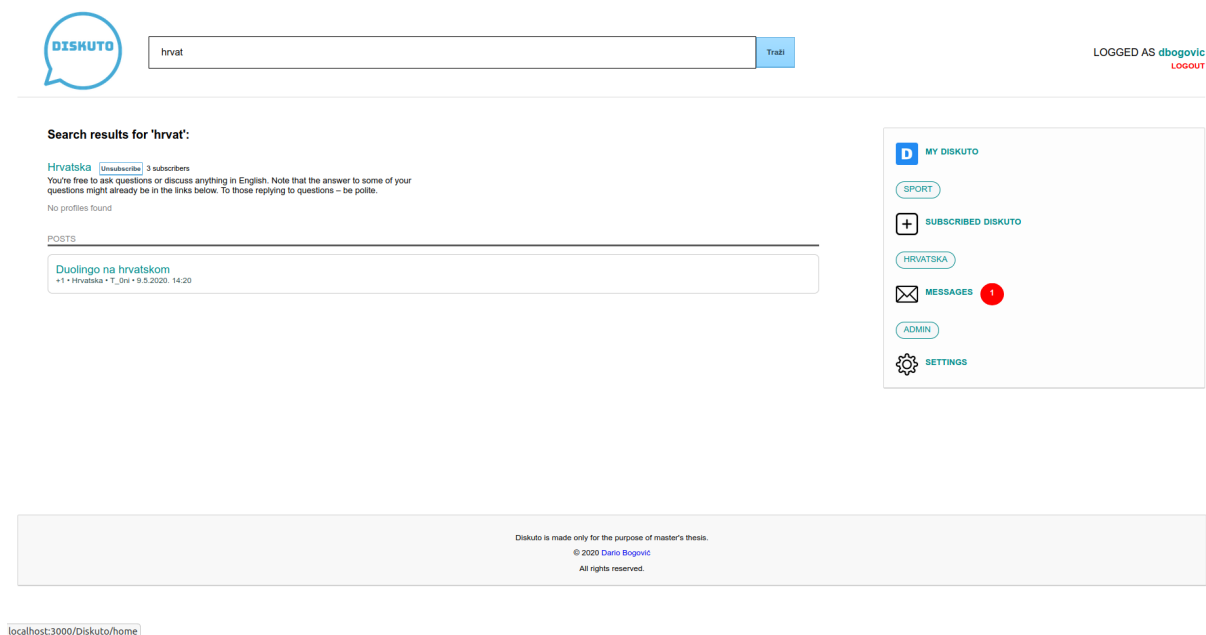
Slika 26: Poruke

Poruke koje je korisnik primio su poravnate sa lijeve strane i imaju crnu boju teksta na sivoj pozadini. Poruke koje je korisnik poslao su poravnate sa desne strane i imaju tamnoplavu boju teksta na plavoj pozadini. Ispod svake poruke nalazi se točan datum i vrijeme kada je poruka poslana ili primljena. Ispod elementa sa porukama nalazi se tekstualni element za slanje poruke. Poruku je jedino moguće poslati ako se korisnici ne ignoriraju, odnosno ako jedan nije stavio drugoga na listu ignoriranja. Kada pošiljalac pošalje poruku, primatelja se obavještava na način da u glavnom izborniku aplikacije mu se prikaže da ima nepročitane poruke i ispod mu se prikaže na poveznici razgovora sa korisnikom od kojeg je primio poruku. Sve dok primatelj poruku ne pročita, ta će mu obavijest biti prikazana u glavnom izborniku.

8.11. Pretraga

Tijekom većine vremena služenja sa aplikacijom se u zaglavlju nalazi traka za pretragu foruma. Registrirani korisnici mogu pretražiti po ključnoj riječi sve forume, korisnike i objave na forumima. Ako je ključna riječ sadržana u nazivu foruma kao rezultat pretrage će se vratiti naziv i poveznica na forum, opis, broj pretplatnika i gumb za pretplatu na navedeni forum. Ako je ključna riječ sadržana u korisničkom imenu korisnika kao rezultat pretrage će se vratiti korisničko ime i poveznica na profil, te gumb za slanje privatne poruke korisniku. Ako je ključna riječ sadržana u naslovu objave na nekom forumu kao rezultat pretrage će se vratiti naslov i poveznica na objavu, te podaci o objavi (karma, forum na kojem je objavljena, autor objave, datum i vrijeme objave).

Na slici 27 je prikazan primjer izgleda stranice za pretragu ako je unesena ključna riječ "hrvat". Vidljivo je kako je aplikacija vratila kao rezultat pretrage jedan forum pod nazivom "Hrvatska", niti jedan profil sa korisničkim imenom koji sadrži tu ključnu riječ, te jednu objavu koja u sebi sadrži ključnu riječ "hrvat". Ako aplikacija vrati forum kao rezultat pretrage, ona će ispi-

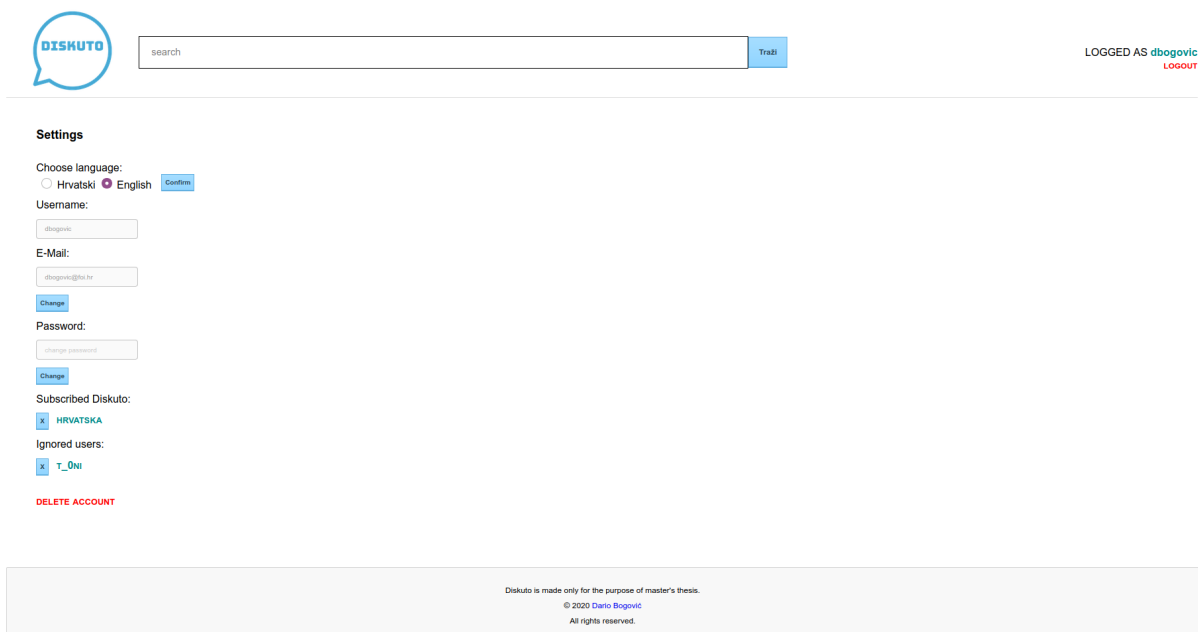


Slika 27: Izgled stranice za pretragu

sati naziv foruma koja je poveznica na stranicu foruma, gumb za pretplatu/poništenje pretplate i detalje o forumu poput broja pretplaćenih korisnika i opis. Ako aplikacija vrati korisnika kao rezultat pretrage, ona će ispisati korisničko ime koje je ujedno i poveznica na profil, te gumb čijim klikom se preusmjerava na slanje privatne poruke. Ako aplikacija vrati objavu kao rezultat pretrage, ispisat će se puni naslov objave koja je i poveznica na samu objavu, te detalji o objavi poput karme (razlika pluseva/minusa), poveznica na forum gdje je objava objavljena, poveznica na profil autora objave i datum i vrijeme kada je objava objavljena. Ako aplikacija ne vrati niti jedan forum, korisnika ili objavu ispisat će se obavijest o neuspjehu.

8.12. Postavke

Korisničkim postavkama se može pristupiti odabirom opcije "Postavke" (eng. "Settings") u glavnom izborniku aplikacije. Korisnik može promijeniti jezik aplikacije, a na raspolaganju su mu dva jezika: engleski i hrvatski. Klikom na gumb za promjenu spremaju se postavke i u bazi, pa kada se korisnik ponovno prijavi prikazati će mu se željeni jezik kojeg je promijenio u postavkama. Korisničko ime je nepromijenjivo, te ga se nakon registracije ne može mijenjati. Korisnik može promijeniti e-mail tako da nakon što klikne gumb za promjenu umjesto sadašnjeg unese novi e-mail. Ako unese krivi format, aplikacija mu neće dozvoliti promjenu i ispisat će se greška. Korisnik može promijeniti i svoju lozinku tako da nakon što klikne gumb za promjenu unese novu lozinku. I dalje stoji pravilo da lozinka ne smije biti preslaba, odnosno ne smije imati manje od osam znakova. Ako je lozinka preslaba, aplikacija mu neće dozvoliti promjenu lozinke i ispisat će se greška. Korisnik također može vidjeti na koje foruma je trenutno pretplaćen, te klikom na gumb "X" koji stoji pokraj poveznice na forum može ukloniti pretplatu. Korisnik može vidjeti i koje je korisnike blokirao, te klikom na gumb "X" koji stoji pokraj poveznice na njegov profil može ga deblokirati. Klikom na opciju "Obriši račun" (eng. "Delete account") korisnik



DISKUTO

search [Traži](#)

LOGGED AS **dbogovic**
[LOGOUT](#)

Settings

Choose language:
☐ Hrvatski ☒ English [Confirm](#)

Username:

E-Mail:

[Change](#)

Password:

[Change](#)

Subscribed Diskuto:
☒ HRVATSKA

Ignored users:
☒ T_0nu

[DELETE ACCOUNT](#)

Diskuto is made only for the purpose of master's thesis.
© 2020 Dario Bogović
All rights reserved.

Slika 28: Korisničke postavke

može trajno obrisati svoj račun. Za potvrdu je potrebno unijeti trenutnu lozinku. Ako je lozinka pogrešna ispisat će se greška, a ako unese dobru lozinku i potvrdi je račun se trajno briše i prikazuje mu se poruka kao na slici:



Slika 29: Poruka o deaktiviranom računu

9. Zaključak

U ovom radu je opisan razvoj polustrukturirane baze podataka za potrebe internetskog foruma. Definiran je podatak kao skup prepoznatljivih znakova zapisanih na mediju koji može biti analogni ili digitalni. Predstavljene su vrste podataka prema obliku u kojem ih pohranjujemo, pa tako razlikujemo strukturirane podatke koji se spremaju prema unaprijed definiranoj shemi, nestrukturirane podatke koji nemaju određenu strukturu i koji čine najveći udio od svih podataka u svijetu, te polustrukturirani podaci koji ne moraju biti pohranjeni prema strukturi i nad njima je u ovom radu stavljen naglasak. Najveća prednost u radu sa polustrukturiranim podacima, što je prepoznato i prilikom izrade praktičnog dijela ovog rada, je fleksibilnost u zapisu podataka i laka prilagodba promjeni strukture. Za razvoj polustrukturirane baze podataka se u ovom radu podaci pohranjuju prema XML standardu zbog jednostavnosti definiranja sheme, mogućnosti rada sa kompleksnijim strukturama i lakoćom dohvaćanja i transformacije podataka pomoću standardiziranih jezika XPath i XQuery. XML podaci se mogu pohraniti u XML bazu podataka koja omogućava njihovu trajnu pohranu. Postoje dvije vrste XML baza podataka: XML-enabled koje omogućuje preslikavanje XML dokumenata u određenu bazu podataka i Native XML koje slijede XML model za pohranjivanje podataka. Najpoznatije programske platforme za rad sa Native XML bazom podataka su: BaseX, Sedna, 28.io, MarkLogic Server i eXist-db.

Kao praktični dio aplikacije izrađena je polustrukturirana baza podataka u kojoj su podaci pohranjeni u XML formatu. Vrsta baze podataka je Native XML (NXD) baza podataka, a izvodi se na eXist-db. Izrađena je Java web-aplikacija koja koristi modeliranu bazu podataka. Aplikacijska domena je internetski forum, odnosno mrežna aplikacija koja omogućava komunikaciju između više ljudi o nekoj određenoj temi. Naziv realizirane aplikacije je "Diskuto". Izvršava se na GlassFish 5.1 poslužitelju, a kao razvojni okvir se koristio JSF 2.3. Dizajn pojedinih web stranica je vlastito izrađen pomoću CSS-a.

Na kraju možemo zaključiti kako polustrukturirane baze podataka mogu riješiti probleme u načinu zapisivanja podataka koje postoje kod relacijskih baza podataka. Ipak, zbog visokog stupnja organiziranosti u relacijskim bazama je lakše moguće dohvaćati, učitavati i analizirati podatke. Potrebno je dobro znati zahtjeve aplikacijske domene kako bi se moglo definirati oblike u kojima će podaci biti pohranjeni, a time i znati odrediti vrstu baze podataka. Polustrukturirane baze podataka mogu biti od koristi kod entiteta čiji atributi su povezani u stablastoj strukturi, što odgovara modelu polustrukturiranih podataka. Jedan od primjera je i internetski forum za čije je potrebe u ovom radu razvijena jedna polustrukturirana baza podataka. Ova vrsta baze podataka je od masovnijeg korištenja Interneta doživjela procvat, no po mom mišljenju zbog načina na kojem se danas još uvijek pohranjuju podaci ona vjerovatno neće dosegnuti popularnost relacijskih baza podataka.

Popis literature

- [1] Ž. Panian i K. Ćurko, *Poslovni informacijski sustavi*. Element, 2010.
- [2] V. Čerić, M. Varga i H. Birolla, *Poslovno računarstvo*. Znak, 1998.
- [3] Y. (KnowledgeMT). (2017.). Data-Information-Knowledge in 3 minutes or less, adresa: <https://www.youtube.com/watch?v=sIjSY05JE9Q>. (pristupljeno: 03.08.2020).
- [4] D. Sullivan, *Professional Data Engineer Study Guide*. Sybex, 2020.
- [5] R. Sint, S. Schaffert, S. Stroka i R. Ferstl, „Combining Unstructured, Fully Structured and Semi-Structured Information in Semantic Wikis,” *Proceedings of the Fourth Workshop on Semantic Wikis – The Semantic Wiki Web*, str. 73–87, 2009.
- [6] V. Bosilj, *Informacijska tehnologija u poslovanju*. Element, 2004.
- [7] M. Maleković i M. Schatten, *Teorija i primjena baza podataka*. Fakultet organizacije i informatike, 2017.
- [8] C. in Research i M. for Official Statistics. (2020.). Semi-structured data, adresa: https://ec.europa.eu/eurostat/cros/content/semi-structured-data_en. (pristupljeno: 22.07.2020).
- [9] B. Thuraisingham, *XML Databases and the Semantic Web*. CRC Press, 2002.
- [10] EDUCBA. (2020.). Introduction to XML Versions, adresa: <https://www.educba.com/xml-versions/>. (pristupljeno: 30.08.2020).
- [11] B. E. Travis, *XML and SOAP Programming for BizTalk Servers*. Microsoft, 2000.
- [12] W. Recommendation. (1999.). XML Path Language (XPath), adresa: <https://www.w3.org/TR/1999/REC-xpath-19991116/>. (pristupljeno: 01.09.2020).
- [13] S. Holzner, *XPath: Navigating XML with XPath 1.0 and 2.0 Kick Start*. Sams Publishing, 2003.
- [14] W. Recommendation. (2010.). XQuery 1.0: An XML Query Language (Second Edition), adresa: <https://www.w3.org/TR/2010/REC-xquery-20101214/>. (pristupljeno: 01.09.2020).
- [15] Javatpoint. (2018.). XQuery Tutorial, adresa: <https://www.javatpoint.com/xquery-tutorial>. (pristupljeno: 26.07.2020).
- [16] E. Siegel i A. Retter, *eXist*. O'Reilly Media, 2015.
- [17] G. Pavlovic-Lazetic, „Native XML databases vs. relational databases in dealing with XML documents,” *Kragujevac Journal of Mathematics*, str. 181–199, 2007.

- [18] P. A. Today. (2018.). eXist-db, adresa: <https://www.predictiveanalyticstoday.com/exist-db>. (pristupljeno: 30.07.2020).
- [19] PCMag. (2020.). Definition of Internet forum, adresa: <https://www.pcmag.com/encyclopedia/term/internet-forum>. (pristupljeno: 14.07.2020).
- [20] U. Dictionary. (2003.). Lurker, adresa: <https://www.urbandictionary.com/define.php?term=lurker>. (pristupljeno: 15.07.2020).
- [21] D. Between. (2017.). Difference between Admin and Moderator, adresa: <http://www.differencebetween.info/difference-between-admin-and-moderator>. (pristupljeno: 15.07.2020).
- [22] vBulletin. (2020.). vBulletin Community Forum help, adresa: <https://forum.vbulletin.com/help>. (pristupljeno: 15.07.2020).
- [23] eukhost. (2010.). Forum Rules And Policies, adresa: <https://www.eukhost.com/blog/webhosting/forum-rules-and-policies/>. (pristupljeno: 15.07.2020).
- [24] T. Conversation. (2020.). <https://theconversation.com/reddit-removes-millions-of-pro-trump-posts-but-advertisers-not-values-rule-the-day-141703>, adresa: <https://www.predictiveanalyticstoday.com/exist-db>. (pristupljeno: 01.08.2020).
- [25] Quora. (2018.). When did the first Internet forum start and which was it? Adresa: <https://www.quora.com/When-did-the-first-Internet-forum-start-and-which-was-it>. (pristupljeno: 15.07.2020).
- [26] Tutorialspoint. (2020.). Top 10 Online Forums, adresa: <https://www.tutorialspoint.com/top-10-online-forums>. (pristupljeno: 01.08.2020).
- [27] P. B. (2016.). Forum.hr još uvijek je živ i zdrav, adresa: <http://planb.hr/forum-hr-jos-uvijek-ziv-zdrav/>. (pristupljeno: 01.08.2020).
- [28] www.hr. (2020.). Forumi, adresa: <https://www.hr/wwwhr/entert/penpal/forum/index.hr.html>. (pristupljeno: 01.08.2020).
- [29] eXist-db Documentation. (2019.). Writing Java Applications with the XML:DB API, adresa: https://exist-db.org/exist/apps/doc/devguide_xmlldb. (pristupljeno: 06.08.2020).

Popis slika

1.	Odnos podatka, informacije i znanja [3]	2
2.	Primjer modela baze koja koristi strukturirane podatke	3
3.	Komponente XML dokumenta	6
4.	eXist upravljačka ploča	15
5.	Java Admin Client	16
6.	Slika zaslona foruma Forum.hr	19
7.	Logo aplikacije Diskuto	21
8.	Kreiranje nove kolekcije u bazi podataka	28
9.	Kreiranje novog dokumenta u bazi podataka	29
10.	Promjena postavki XML dokumenta	29
11.	Prikaz početne stranice (neprijavljen korisnik)	39
12.	Prikaz početne stranice (prijavljen korisnik)	40
13.	Forma za registraciju novog korisnika	41
14.	Forma za potvrdu registracije novog korisnika	42
15.	Prikaz početne stranice za novoregistriranog korisnika	43
16.	Forma za prijavu korisnika	43
17.	Stranica za poništavanje lozinke	44
18.	Stranica za otkrivanje Diskuta	44
19.	Forma za kreiranje novog Diskuta	45
20.	Popis mojih Diskuta	46
21.	Izgled stranice foruma	47
22.	Forma za kreiranje nove objave	48

23.	Izgled objave	48
24.	Popis prijavljenih objava/komentara	49
25.	Profil korisnika	50
26.	Poruke	51
27.	Izgled stranice za pretragu	52
28.	Korisničke postavke	53
29.	Poruka o deaktiviranom računu	53

Popis programskih isječaka

1.	XML Schema za dokument "users.xml"	22
2.	XML Schema za dokument "forums.xml"	23
3.	XML Schema za dokument "posts.xml"	24
4.	XML Schema za dokument "messages.xml"	26
5.	Sadržaj pomoćne klase "Database.java"	30
6.	Sadržaj pomoćne klase "XmlHelper.java"	32