Bookshelf

Veb aplikacija za recenziju knjiga

Detaljni arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 20.11.2022. | 1.0 | Inicijalna verzija | Basila |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc121089969)

[2. Opseg dokumenta 5](#_Toc121089970)

[3. Reference 5](#_Toc121089971)

[4. Predstavljanje arhitekture 5](#_Toc121089972)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc121089973)

[6. Pogled na slučajeve korišćenja 5](#_Toc121089974)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc121089975)

[6.2 Kratak opis slučajeva korišćenja 8](#_Toc121089976)

[6.2.1 Pregled osnovnih podataka o knjizi 8](#_Toc121089977)

[6.2.2 Prikaz prosečne ocene knjige 9](#_Toc121089978)

[6.2.3 Prikaz komentara knjige 9](#_Toc121089979)

[6.2.4 Pretraživanje knjiga po žanru 9](#_Toc121089980)

[6.2.5 Pretraživanje knjiga po nazivu 9](#_Toc121089981)

[6.2.6 Prikaz podataka o korisniku 9](#_Toc121089982)

[6.2.7 Prikaz liste čitalaca 9](#_Toc121089983)

[6.2.8 Recenzija knjige ocenom od 1 do 5 9](#_Toc121089984)

[6.2.9 Komentarisanje knjige 9](#_Toc121089985)

[6.2.10 Prijavljivanje na sistem 9](#_Toc121089986)

[6.2.11 Povratak zaboravljene šifre 9](#_Toc121089987)

[6.2.12 Registacija 9](#_Toc121089988)

[6.2.13 Ažuriranje podataka o korisniku 9](#_Toc121089989)

[6.2.14 Podnošenje zahteva za dodavanje knjiga 9](#_Toc121089990)

[6.2.15 Odobravanje zahteva za dodavanje knjiga 10](#_Toc121089991)

[6.2.16 Dodavanje knjiga 10](#_Toc121089992)

[6.2.17 Kreiranje novog čitaoca 10](#_Toc121089993)

[6.2.18 Brisanje postojećeg čitaoca 10](#_Toc121089994)

[6.2.19 Dodavanje knjige na listu za čitanje 10](#_Toc121089995)

[6.2.20 Uklanjanje knjige sa liste za čitanje 10](#_Toc121089996)

[6.2.21 Prikaz liste za čitanje 10](#_Toc121089997)

[6.2.22 Statistički prikaz aktivnosti 10](#_Toc121089998)

[7. Pogled na logičku arhitekturu sistema 10](#_Toc121089999)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 11](#_Toc121090000)

[7.1.1 Korisnički interfejs 11](#_Toc121090001)

[7.1.2 Aplikaciona logika 11](#_Toc121090002)

[7.1.3 Pristup podacima 11](#_Toc121090003)

[7.1.4 HTML 11](#_Toc121090004)

[7.1.5 Node.js 11](#_Toc121090005)

[7.1.6 MongoDB 11](#_Toc121090006)

[8. Pogled na procese 12](#_Toc121090007)

[8.1 Procesi 12](#_Toc121090008)

[8.1.1 Web čitač 12](#_Toc121090009)

[8.1.2 Veb server 12](#_Toc121090010)

[8.1.3 MongoDB Server 12](#_Toc121090011)

[9. Pogled na raspoređivanje sistema 12](#_Toc121090012)

[9.1 Klijent 13](#_Toc121090013)

[9.2 Veb server 13](#_Toc121090014)

[9.3 DBMS server 13](#_Toc121090015)

[10. Pogled na implementaciju sistema 13](#_Toc121090016)

[10.1 Model domena 13](#_Toc121090017)

[10.2 Šema baze podataka 14](#_Toc121090018)

[10.3 Komponente sistema 14](#_Toc121090019)

[10.3.1 Komponente korisničkog interfejsa 14](#_Toc121090020)

[10.3.2 Komponente aplikacione logike 15](#_Toc121090021)

[10.3.3 Komponente za pristup podacima 15](#_Toc121090022)

[11. Performanse 16](#_Toc121090023)

[12. Kvalitet 16](#_Toc121090024)

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Bookshelf veb aplikacije.

# Opseg dokumenta

Dokument se odnosi na Bookshelf veb aplikaciju koja će biti razvijena od strane Bookworms-a. Bookshelf je veb aplikacija namenjena za recenziju knjiga. Namena sistema je efikasno prezentovanje, kreiranje i održavanje sadržaja potrebnih za rad jedne platforme za recenziju i pretraživanje knjiga.

# Reference

Spisak korišćene literature:

1. Bookshelf – Definicija projekta, BOO\_01, V1.0, 2022, Bookworms.
2. Bookshelf – Planirani raspored aktivnosti na projektu, BOO\_02, V1.0, 2022, Bookworms.
3. Bookshelf – Plan realizacije projekta, BOO\_02, V1.0, 2022, Bookworms.
4. Bookshelf – Vizija sistema, BOO\_03, 2022, Bookworms.
5. Bookshelf – Specifikacija zahteva, BOO\_04, 2022, Bookworms.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Bookshelf veb aplikacija će biti implementiran kao veb aplikacija zasnovana na Node.js-u i MongoDB bazi podataka [4].
2. Klijentski deo Bookshelf veb aplikacije će biti optimizovan za sledeće Veb čitače: Samsung internet 16.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla) [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za nauku.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

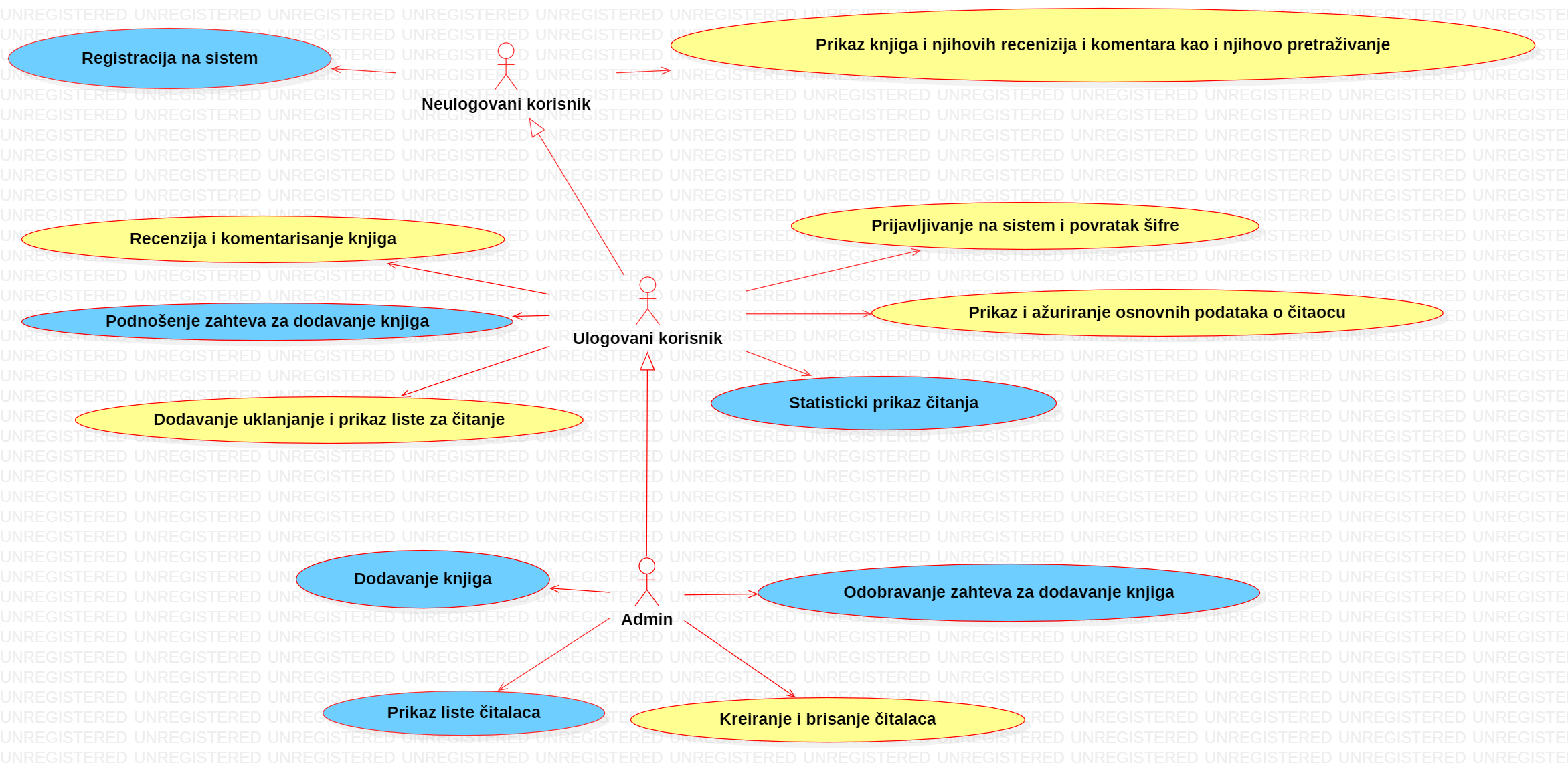
Slučajevi korišćenja Bookshelf veb aplikacije su:

* Prikaz knjiga, njihovih recenzija i komentara kao i njihovo pretraživanje
  + Pretraživanje knjiga
    - Pretraživanje knjiga po nazivu
    - Pretraživanje knjiga po žanru
  + Prikaz podataka o knjizi
    - Prikaz osnovnih podataka o knjizi
    - Prikaz više podataka o knjizi
      * Prikaz recenzija
        + Prikaz prosečne ocene
        + Prikaz komentara
* Prikaz i ažuriranje podataka o korisnicima
  + Ažuriranje podataka o korisniku
  + Prikaz podataka korisnika
* Prikaz liste čitalaca
* Recenzija i komentarisanje knjige
  + Recenzija knjige ocenom od 1 do 5
  + Komentarisanje knjige
* Prijavljivanje na sistem i povratak šifre
  + Prijavljivanje na sistem
  + Povratak zaboravljene sifre
* Registracija
* Podnošenje zahteva za dodavanje knjiga
* Odobravanje zahteva za dodavanje knjiga
* Dodavanje knjiga
* Kreiranje i brisanje čitalaca
  + Kreiranje novog čitaoca
  + Brisanje postojećeg čitaoca
* Dodavanje, uklanjanje i prikaz liste za čitanje
  + Dodavanje knjige na listu za čitanje
  + Uklanjanje knjige sa liste za čitanje
  + Prikaz liste za čitanje
* Statistički prikaz čitanja

Ove slučajevi korišćenja mogu da iniciraju posetilac veb aplikacije, čitalac ili administrator.

## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Bookshlef veb aplikacije prikazan je na sledećoj slici:



Slučajevi korišćenja „*kreiranje i brisanje čitalaca“, „prikaz knjiga,njihovih recenzija i komentara, kao i njihovo pretraživanje“, „prikaz i ažuriranje podataka o čitaocu“, „recenzija i komentarisanje knjiga“, „prijavljivanje na sistem i povratak šifre“ i „dodavanje, uklanjanje i prikaz liste za čitanje“* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *„prikaz knjiga,njihovih recenzija i komentara, kao i njihovo pretraživanje“* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *kreiranje i brisanje čitalaca* je prikazan na sledećoj slici: Detaljni UML dijagram za slučaj korišćenja *prikaz i ažuriranje podataka o čitaocu* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *prijavljivanje na sistem i povratak šifre je* prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *recenzija i komentarisanje knjiga* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *dodavanje, uklanjanje i prikaz liste za čitanje*  je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Pregled osnovnih podataka o knjizi

Kratak opis: Prikaz glavne stranice.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Čitalac, Administrator.

### Prikaz prosečne ocene knjige

Kratak opis: Prikaz stranice sa više informacija o knjizi.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Čitalac, Administrator.

### Prikaz komentara knjige

Kratak opis: Prikaz stranice sa informacijama o određenom članu laboratorije.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Čitalac, Administrator.

### Pretraživanje knjiga po žanru

Kratak opis: Prikaz stranice sa rezultatima pretrage.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Čitalac, Administrator.

### Pretraživanje knjiga po nazivu

Kratak opis: Prikaz stranice sa rezultatima pretrage.

Akteri koji iniciraju slučaj korišćenja: Posetilac, Čitalac, Administrator.

### Prikaz podataka o korisniku

Kratak opis: Prikaz stranice sa informacijama o određenom korisniku.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Prikaz liste čitalaca

Kratak opis: Prikaz stranice sa listom čitalaca.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Recenzija knjige ocenom od 1 do 5

Kratak opis: Recenzija knjige ocenom od 1 do 5.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Komentarisanje knjige

Kratak opis: Dodavanje komentara na stranici sa više informacija o knjizi.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Prijavljivanje na sistem

Kratak opis: Prijavljivanje korisnika na veb aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Povratak zaboravljene šifre

Kratak opis: Povratak zaboravljene šifre već registrovanog korisnika.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Registacija

Kratak opis: Registracija korisnika na veb aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Ažuriranje podataka o korisniku

Kratak opis: Ažuriranje podataka o sebi od strane prijavljenog korisnika.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Podnošenje zahteva za dodavanje knjiga

Kratak opis: Podnošenje zahteva za dodavanje nove knjige na glavnu stranicu.

Akteri koji iniciraju slučaj korišćenja: Čitalac.

### Odobravanje zahteva za dodavanje knjiga

Kratak opis: Odobravanje prethodono podnetog zahteva za dodavanje knjige.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Dodavanje knjiga

Kratak opis: Dodavanje knjige na spisak dostupnih knjiga na glavnoj stranici.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje novog čitaoca

Kratak opis: Kreiranje korisničkog naloga za novog čitaoca.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje postojećeg čitaoca

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg čitaoca.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Dodavanje knjige na listu za čitanje

Kratak opis: Dodavanje dostupnih knjiga na TBR listu.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Uklanjanje knjige sa liste za čitanje

Kratak opis: Uklanjanje knjiga sa TBR liste.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Prikaz liste za čitanje

Kratak opis: Prikaz stranice sa TBR listom.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

### Statistički prikaz aktivnosti

Kratak opis: Pristup statističkom prikazu čitanja u protekloj nedelji.

Akteri koji iniciraju slučaj korišćenja: Čitalac, Administrator.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

Logički pogled na Bookshelf aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Veb stranice, Node.js kontorlere i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži Node.js kontrolere zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži Node.js kontorlere koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs veb aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji i JS fajlovi koji generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i JS.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Bookshelf veb aplikacije. Sadrži Node.js kontrolere koje realizuju funkcionalnost karakterističnu za domen primene veb aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i Node.js paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži Node.js kontrolere zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MongoDB bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa Node.js i MongoDB baza podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Veb čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### Node.js

tehnologija

Tehnologija Node.js-a obezbeđuje mehanizam za upravljanje zahtevima koji stižu od strane klijenta. Kontroleri su ti koji su zaduženi za gerisanje odgovora koji se prosleđuje klijentu.

### MongoDB

DBMS

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Bookshelf veb aplikacije.

# Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Veb aplikacije zasnovane na Node.js-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Veb servera. Sa stanovišta projektanta Node.js Veb aplikacije nije potrebno voditi računa o načinu rada Veb servera i načinu izvršavanja skripti.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Bookshelf-a kao Veb aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Bookshlef veb aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Veb aplikaciju zasnovanu na Node.js-u i MongoDB bazi podataka.



### Web čitač

Veb čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica koristeći informacije dobijene od nekog Veb servera.

Veb čitač zavisi od Veb servera koji generiše i vraća odgovarajuće informacije (JSON objekat) na zahtev.

### Veb server

Veb server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Veb čitača.On odlučuje koji kontroler će biti zadužen za generisanje odgovora.

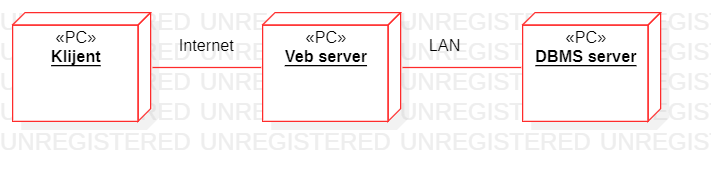
### MongoDB Server

MongoDB Server je proces koji izvršava funkcionalnost MongoDB sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja Bookshelf veb aplikacije.



## Klijent

Pristup Bookshelf veb aplikaciji se obavlja preko klijentskih računara na kojima se izvršava Veb čitač. Za povezivanje između klijenta i Veb servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Veb server

Računar na kome se izvršava Veb server opslužuje više klijenata koji pristupaju preko Interneta.U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Veb serverom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava MongoDB server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

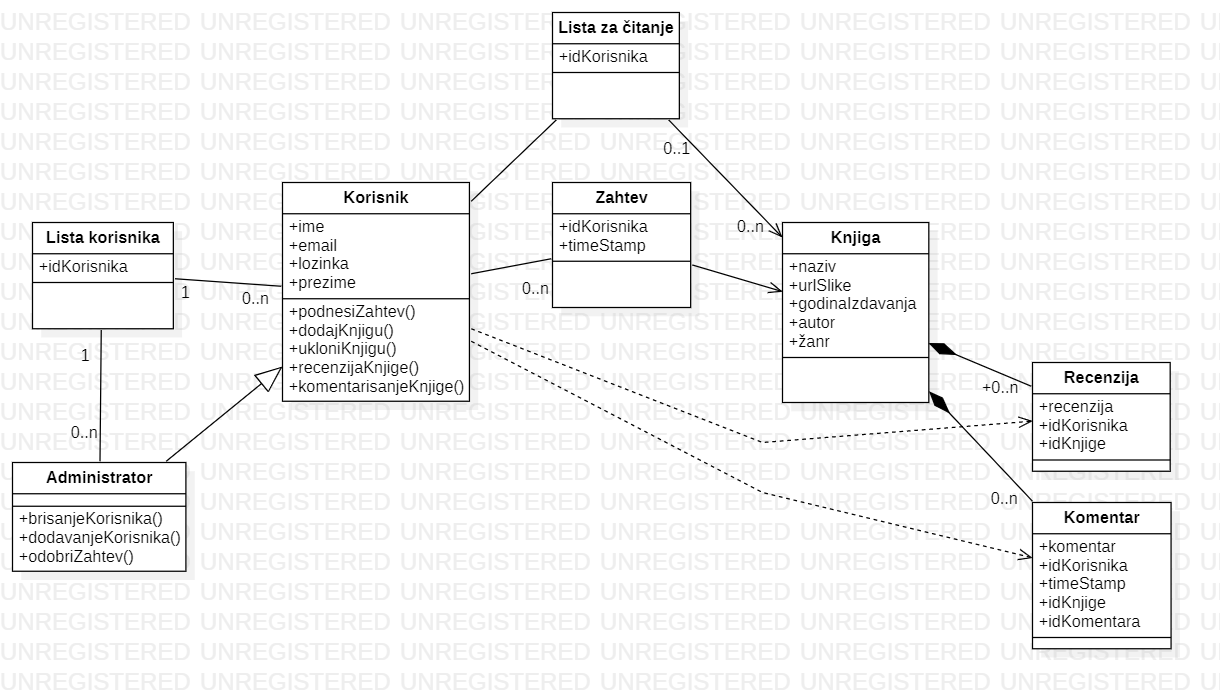
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju Bookshelf veb aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

Model domena za koji se Bookshelf veb aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (Node.js kontrolera) koje će biti implementirane.



## Šema baze podataka

Detaljna šema MongoDB baze podataka je prikazana na sledećem dijagramu. Dijagram je kreiran korišćenjem aplikacije StarUML.



## Komponente sistema

Komponente sistema Bookshelf veb aplikacije su React komponente čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je komponenta prikazana kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **index.jsx** je implementira stranicu aplikacije čiji sadržaj može da varira od parametara koji joj se proslede u zahtevu.

Komponenta **style.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.jsx** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru Node.js skripta **backend.js**. Pomenuti skript se uključuje na početku index.jsx-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 100 simultanih pristupa korisnika veb aplikacije.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. Bookshelf veb aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada veb aplikacija nije dostupna ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.