BookNest

Veb aplikacija za recenziju knjiga

Detaljni arhitekturni projekat

Verzija 1.1

Pregled izmena

| **Datum** | **Verzija** | **Opis** | **Autor** |
| --- | --- | --- | --- |
| 03.12.2024. | 1.0 | Inicijalna verzija | Munevera |
| 11.12.2024. | 1.1 | Dopunjena i prerađena verzija | Munevera |
|  |  |  |  |
|  |  |  |  |

**Sadržaj**

[1.](#_gjdgxs) Cilj dokumenta 5

[2.](#_30j0zll) Opseg dokumenta 5

[3.](#_3znysh7) Reference 5

[4.](#_2et92p0) Predstavljanje arhitekture 5

[5.](#_3dy6vkm) Ciljevi i ograničenja arhitekture 5

[6.](#_1t3h5sf) Pogled na slučajeve korišćenja 5

[6.1](#_4d34og8) Dijagrami slučajeva korišćenja 6

[6.2](#_2s8eyo1) Kratak opis slučajeva korišćenja 8

[6.2.1](#_17dp8vu) Pregled osnovnih podataka o knjizi 8

[6.2.2](#_26in1rg) Prikaz prosečne ocene knjige 9

[6.2.3](#_lnxbz9) Prikaz komentara knjige 9

[6.2.4](#_35nkun2) Pretraživanje knjiga po žanru 9

[6.2.5](#_1ksv4uv) Pretraživanje knjiga po nazivu 9

[6.2.6](#_44sinio) Prikaz podataka o korisniku 9

[6.2.7](#_2jxsxqh) Prikaz liste čitalaca 9

[6.2.8](#_z337ya) Recenzija knjige ocenom od 1 do 5 9

[6.2.9](#_3j2qqm3) Komentarisanje knjige 9

[6.2.10](#_1y810tw) Prijavljivanje na sistem 9

[6.2.11](#_4i7ojhp) Povratak zaboravljene šifre 9

[6.2.12](#_2xcytpi) Registracija 9

[6.2.13](#_1ci93xb) Ažuriranje podataka o korisniku 9

[6.2.14](#_qsh70q) Dodavanje knjiga 10

[6.2.15](#_3as4poj) Kreiranje novog čitaoca 10

[6.2.16](#_1pxezwc) Brisanje postojećeg čitaoca 10

[6.2.17](#_49x2ik5) Dodavanje knjige na listu za čitanje 10

[6.2.18](#_2p2csry) Uklanjanje knjige sa liste za čitanje 10

[6.2.19](#_147n2zr) Prikaz liste za čitanje 10

[7.](#_23ckvvd) Pogled na logičku arhitekturu sistema 10

[7.1](#_ihv636) Pregled arhitekture – organizacija paketa i podsistema u slojeve 11

[7.1.1](#_32hioqz) Korisnički interfejs 11

[7.1.2](#_41mghml) Aplikaciona logika 11

[7.1.3](#_2grqrue) Pristup podacima 11

[7.1.4](#_vx1227) React 11

[7.1.5](#_3fwokq0) Asp.Net 11

[7.1.6](#_1v1yuxt) Microsoft SQL Server Menagement 11

[8.](#_4f1mdlm) Pogled na procese 12

[8.1](#_2u6wntf) Procesi 12

[8.1.1](#_19c6y18) Web čitač 12

[8.1.2](#_28h4qwu) Veb server 12

[8.1.3](#_37m2jsg) .Net proces 12

[9.](#_1mrcu09) Pogled na raspoređivanje sistema 12

[9.1](#_46r0co2) Klijent 13

[9.2](#_2lwamvv) Web server 13

[9.3](#_111kx3o) DBMS server 13

[10.](#_3l18frh) Pogled na implementaciju sistema 13

[10.1](#_206ipza) Model domena 13

[10.2](#_4k668n3) Šema baze podataka 14

[10.3](#_2zbgiuw) Komponente sistema 14

[10.3.1](#_1egqt2p) Komponente korisničkog interfejsa 14

[10.3.2](#_3ygebqi) Komponente aplikacione logike 15

[10.3.3](#_2dlolyb) Komponente za pristup podacima 15

[11.](#_3cqmetx) Performanse 16

[12.](#_1rvwp1q) Kvalitet 16

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture BookNest veb aplikacije.

# Opseg dokumenta

Dokument se odnosi na BookNest veb aplikaciju koja će biti razvijena od strane BookWorms-a. BookNest je Web 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. BookNest – Definicija projekta, BW\_MP01, V1.1, 2022, BookWorms.
2. BookNest – Planirani raspored aktivnosti na projektu, BW\_MP02, V1.1, 2024, BookWorms.
3. BookNest – Plan realizacije projekta, BW\_MP02, V1.1, 2024, BookWorms.
4. BookNest – Vizija sistema, BW\_MP03, V1.2 2024, BookWorms.
5. BookNest – Specifikacija zahteva, BW\_MP04, V1.3 2024, 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. BookNest će biti implementirana kao Web aplikacija koristeći React.js za frontend i Microsoft SQL Server Management za bazu podataka[4].
2. Klijentski deo BookNest veb aplikacije će biti optimizovan za sledeće Veb čitače: Samsung Internet pretraživač 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

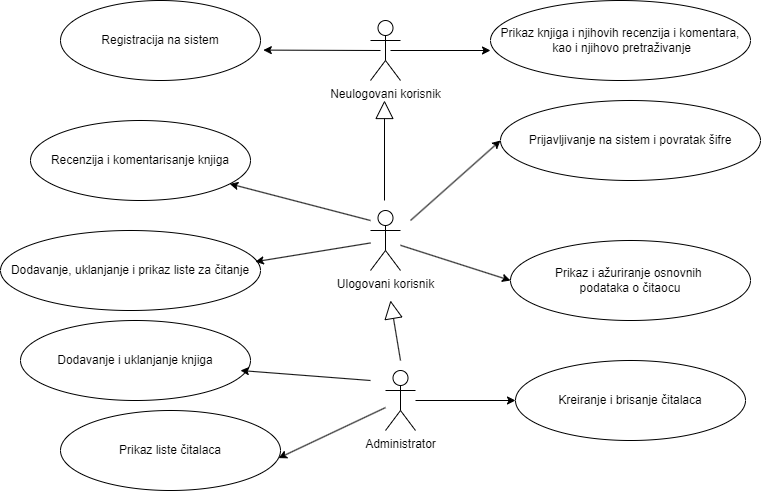
U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

Slučajevi korišćenja BookNest 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 šifre
* Registracija
* 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

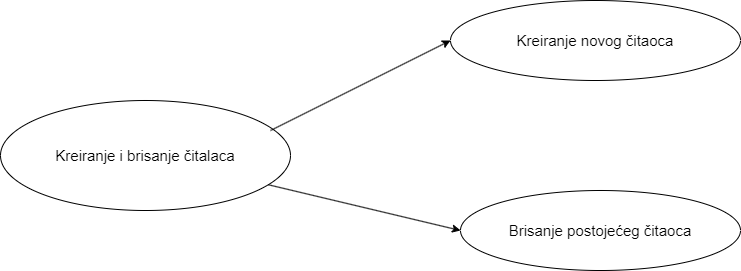
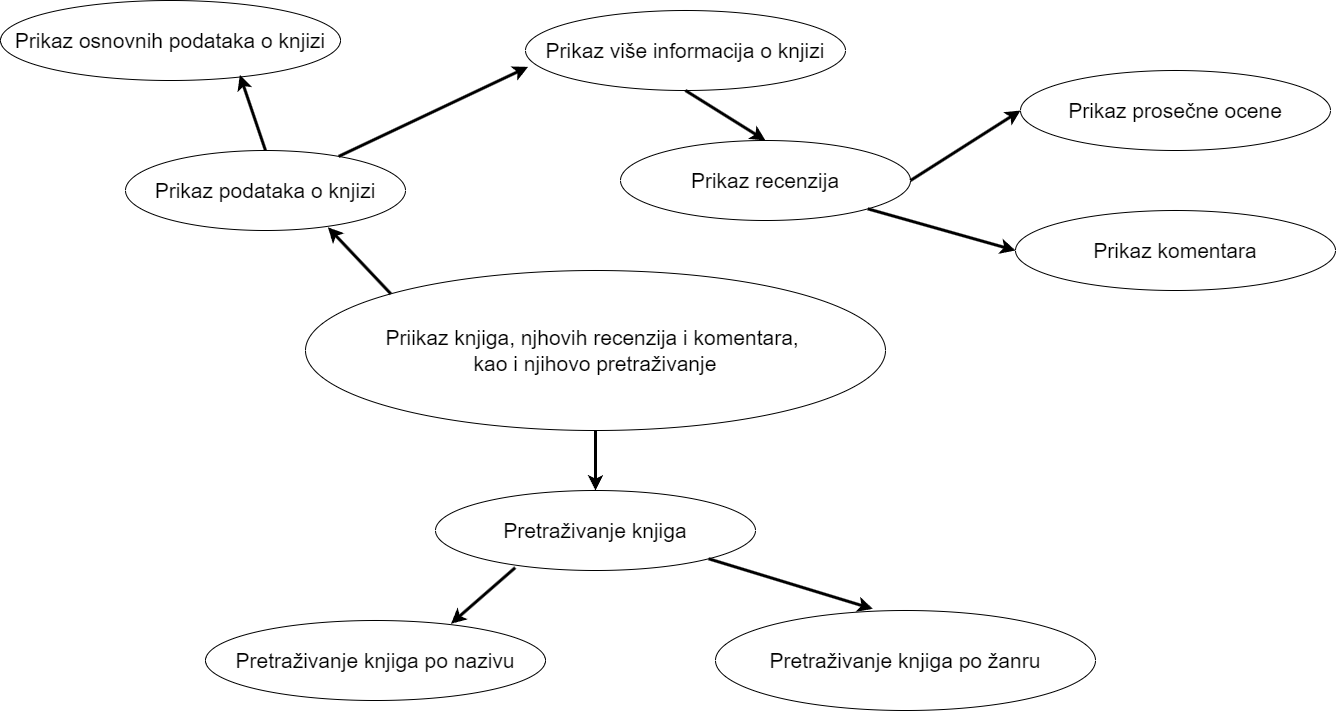
Ove slučajevi korišćenja mogu da iniciraju posetilac veb aplikacije, čitalac ili administrator.

## Dijagrami slučajeva korišćenja

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

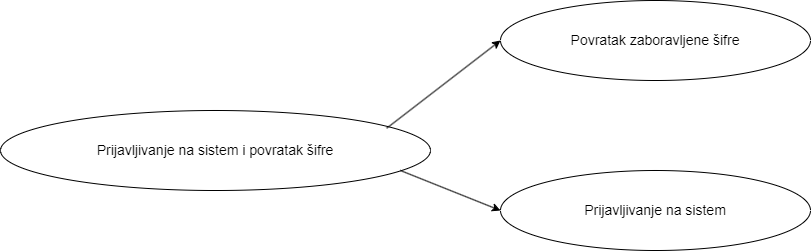
Slučajevi korišćenja „*kreiranje i brisanje čitalaca“, „prikaz knjiga,njihovih recenzija i komentara, kao i njihovo pretraživanje“, „prikaz i ažuriranje podataka o čitaocu“, „recenzija i komentarisanje knjiga“, „prijavljivanje na sistem i povratak šifre“ i „dodavanje, uklanjanje i prikaz liste za čitanje“* 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:

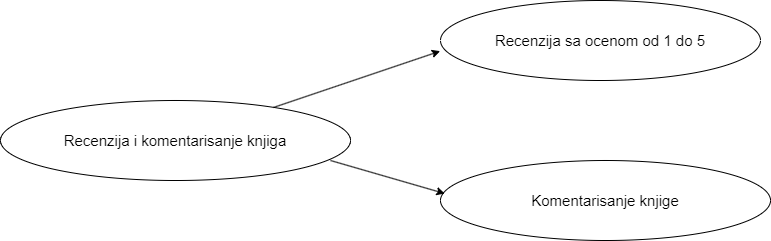
Detaljni UML dijagram za slučaj korišćenja *kreiranje i brisanje čitalaca* je prikazan na sledećoj slici: Detaljni UML dijagram za slučaj korišćenja *prikaz i ažuriranje podataka o čitaocu* je prikazan na sledećoj slici:



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 komentara knjige.

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.

### **Registracija**

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.

### **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 listu knjiga za čitanje (To Be Read – TBR).

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.

# 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 BookNest aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži React komponente 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 .NET API kontrolere zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži .NET servise i metode koji služe kao interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u SQL Server bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs portala koristeći React komponente koje definišu strukturu i funkcionalnost interfejsa. Multimedijalni sadržaji i dinamičke React komponente omogućavaju kreiranje interaktivnih stranica preko kojih korisnici intuitivno komuniciraju sa sistemom. Sloj korisničkog interfejsa direktno zavisi od sloja aplikacione logike, kao i od tehnologija poput React-a i CSS-a za stilizaciju.

### Aplikaciona logika

sloj

Sloj aplikacione logike predstavlja centralni deo troslojne arhitekture sistema. Implementiran pomoću .NET Web API servisa, ovaj sloj realizuje funkcionalnosti specifične za poslovne procese portala. Njegova glavna uloga je posredovanje između korisničkog interfejsa i sloja za pristup podacima, omogućavajući validaciju unosa, obradu zahteva i implementaciju poslovne logike. Ovaj sloj zavisi od sloja za pristup podacima i .NET biblioteka za realizaciju ključnih funkcionalnosti.

### Pristup podacima

sloj

Sloj za pristup podacima je osnovni deo troslojne arhitekture, odgovoran za rukovanje podacima sistema. Sadrži .NET servise i repozitorijume za pribavljanje, ažuriranje i upravljanje podacima koji se čuvaju u SQL Server bazi podataka. Iako nije direktno povezan sa korisničkim interfejsom, ovaj sloj zavisi od SQL Server sistema i Entity Framework Core biblioteke za interakciju sa bazom.

### REACT

tehnologija

React je moćna JavaScript biblioteka namenjena za razvoj komponenti korisničkog interfejsa. Njena fleksibilnost omogućava dinamičko osvežavanje sadržaja i efikasno upravljanje stanjem aplikacije, pružajući korisnicima besprekorno iskustvo prilikom navigacije.

### .NET

tehnologija

.NET je robusna razvojna platforma za kreiranje server-side Web API servisa. Njeni alati omogućavaju implementaciju poslovne logike, interakciju sa bazama podataka i generisanje JSON odgovora koji se koriste za komunikaciju sa frontend-om.

### SQL Server

DBMS

SQL Server je visoko pouzdan sistem za upravljanje bazama podataka, koji obezbeđuje sigurno i efikasno skladištenje, pribavljanje i analizu podataka. Njegova napredna funkcionalnost omogućava održavanje integriteta podataka i visok nivo performansi čak i u složenim aplikacijama.

# 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.

Web aplikacije koje koriste React i .NET tehnologije oslanjaju se na savremenu klijent-server arhitekturu, pružajući fleksibilnost i visok nivo performansi. Na klijentskoj strani, React omogućava kreiranje interaktivnog i dinamičkog korisničkog interfejsa, dok .NET na serverskoj strani obezbeđuje snažno upravljanje poslovnom logikom i obradu zahteva.

Iz perspektive projektanta, detalji niskog nivoa, poput rada web servera, ne zahtevaju posebnu pažnju, jer web server automatski obrađuje HTTP zahteve i prosleđuje ih .NET aplikaciji. React se, s druge strane, brine o prikazu sadržaja i njegovom dinamičkom osvežavanju unutar korisničkog pretraživača, čineći interakciju s aplikacijom glatkom i intuitivnom za krajnje korisnike.

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

## Procesi

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



### Web čitač

**Web čitač** je komponenta koja omogućava prikaz korisničkog interfejsa aplikacije razvijene pomoću React-a. Nakon što primi sadržaj od web servera, web čitač renderuje React komponente i dinamički ažurira stranicu bez potrebe za ponovnim učitavanjem. Web čitač zavisi od web servera za statičke fajlove React-a, kao i od backend API-ja koji pruža podatke potrebne za rad aplikacije.

### Web server

**Web server** funkcioniše kao posrednik između web čitača i backend aplikacije. On obrađuje zahteve od različitih web čitača i prosleđuje ih .NET aplikaciji. Za API zahteve, web server generiše JSON odgovore bazirane na poslovnoj logici implementiranoj u backend-u.

### .NET proces

**.NET proces** obrađuje poslovnu logiku i vrši komunikaciju sa SQL Serverom. Ovaj proces prima zahteve od web servera, kao što su zahtevi za dohvat podataka, ažuriranje ili validaciju, i obrađuje ih putem kontrolera u .NET Web API-ju. Zahtevi za bazu podataka se izvršavaju putem Entity Framework Core-a ili direktnih SQL upita, omogućavajući efikasan rad sa podacima.

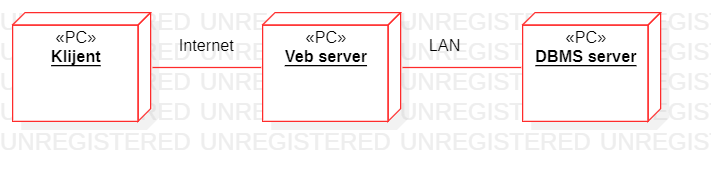
### SQL Server

### **SQL Server** je sistem za upravljanje bazama podataka koji omogućava sigurno i efikasno čuvanje, obradu i vraćanje podataka. Kao proces, SQL Server može istovremeno obraditi više upita, izvršiti ih nad bazom i vratiti rezultate aplikaciji koja je postavila upit.

# 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 BookNest veb aplikacije.



## Klijent

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

## Web server

Računar na kome se izvršava **web server** opslužuje više klijenata koji pristupaju aplikaciji putem interneta. Web server u ovoj arhitekturi (npr. **IIS** ili **Kestrel**) realizuje osnovne funkcionalnosti kao što su serviranje statičkih resursa (HTML, CSS, JavaScript fajlova za React aplikaciju) ili prosleđivanje API zahteva .NET aplikaciji koja se izvršava na istom računaru ili u istom okruženju

## DBMS server

DBMS server je računar na kome se izvršava SQL 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).

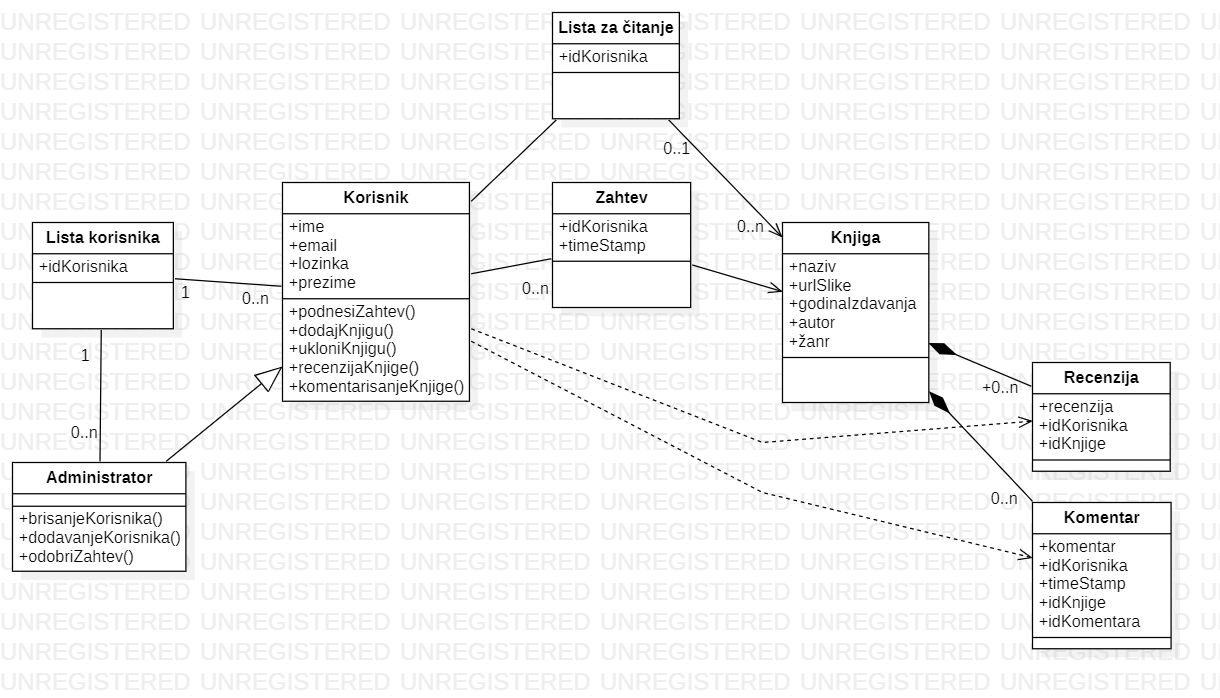
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju BookNest 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 BookNest Web 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 (React.js kontrolera) koje će biti implementirane.



## Šema baze podataka

Detaljna šema SQL Server baze podataka je prikazana na sledećem dijagramu. Dijagram je kreiran korišćenjem aplikacije draw.io Diagrams.



## Komponente sistema

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