BeHuman

Online humanitarna platforma

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 29.04.2022. | 1.0 | Inicijalna verzija | Katarina Maksimovic  David Marinkovic  Milica Miletic |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 8

6.2.1 Izbor jezika 8

6.2.2 Pregled osnovnih podataka o laboratoriji 8

6.2.3 Pregled spiska članova 8

6.2.4 Pregled podataka o određenom članu laboratorije 8

6.2.5 Pregled publikacija po autoru 8

6.2.6 Pregled publikacija po tipu 8

6.2.7 Pregled publikacija po godini 8

6.2.8 Pregled spiska projekata 8

6.2.9 Pregled podataka o određenom projektu 8

6.2.10 Prijavljivanje 9

6.2.11 Ažuriranje podataka o članu 9

6.2.12 Dodavanje nove publikacije 9

6.2.13 Brisanje postojeće publikacije 9

6.2.14 Ažuriranje osnovnih podataka o laboratoriji 9

6.2.15 Kreiranje novog člana 9

6.2.16 Brisanje postojećeg člana 9

6.2.17 Arhiviranje postojećeg člana 9

6.2.18 Kreiranje projekta i postavljanje vođe 9

6.2.19 Ažuriranje podataka o projektu 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 10

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 PHP 11

7.1.6 MySQL 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 PHP-CGI 11

8.1.4 MySQL Server 12

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 14

10.3.3 Komponente za pristup podacima 15

11. Performanse 16

12. Kvalitet 16

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture BeHuman aplikacije.

# Opseg dokumenta

Dokument se odnosi na BeHuman web aplikaciju koja će biti razvijena od strane M3 tima. BeHuman predstavlja skraćenicu za Portal humanitarne organizacije. Namena sistema je efikasno i jednostavno zahtevanje i pruzanje pomoci,kao i mogucnost volontiranja.Pored toga,olaksana komunikacija izmedju donatora i zahtevaoca pomoci.

# Reference

Spisak korišćene literature:

1. BeHuman – Predlog projekta, M3-BeHuman-01, V1.0, 2022, M3.
2. BeHuman – Planirani raspored aktivnosti na projektu, V1.0, 2022, M3.
3. BeHuman – Plan realizacije projekta, V1.0, 2022,M3.
4. BeHuman – Vizija sistema, V1.0, 2022, M3.
5. BeHuman – Specifikacija zahteva, V1.0, 20022, M3.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, 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. BeHuman aplikacija biće implementirana kao Web aplikacija zasnovana na JavaScript jeziku, koristeći Node.js platformu, Express i MongoDB [4].
2. Klijentski deo BeHuman aplikacije biće optimizovan za sledeće Web čitače: Google Chrome 75.0 i noviji, Firefox (Mozilla) 65.0 i noviji, Opera 60.0 i noviji i Microsoft Edge [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

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

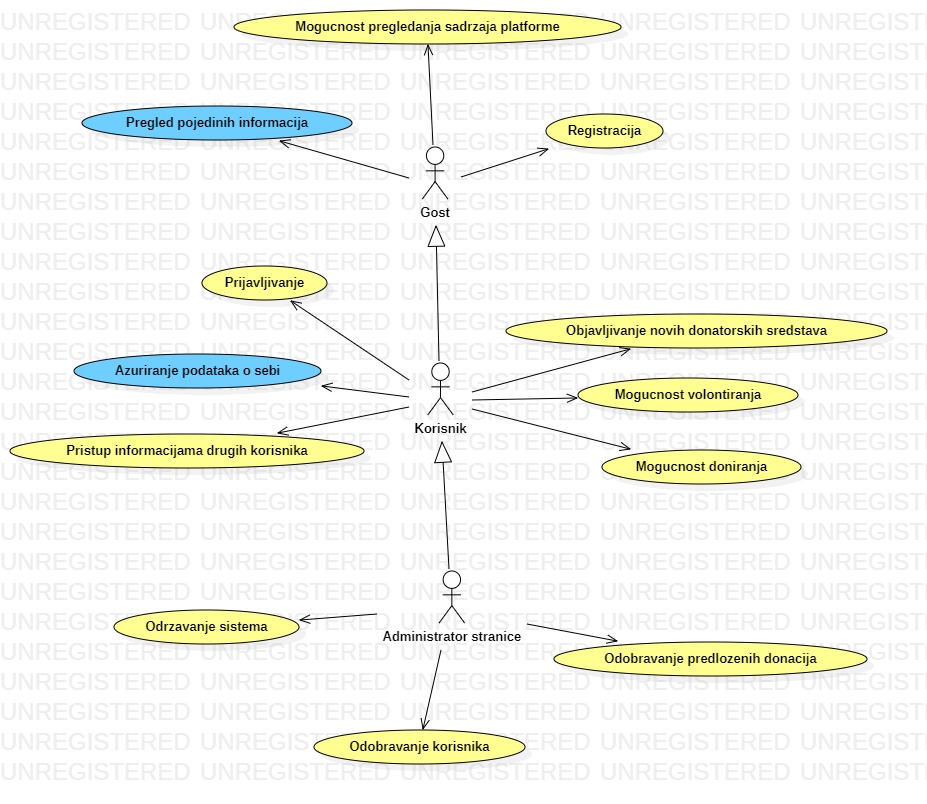
Slučajevi korišćenja BeHuman aplikacije su:

* Mogucnost pregledanja sadrzaja platforme
* *Pregled pojedinih informacija*
  + Pregled postavljenih donacija
  + Pregled podataka o korisnicima
    - Pregled podataka o volonterima
    - Pregled podataka o donatorima
    - Pregled podataka o ljudima koji zahtevaju pomoc
* Registracija
* *Azuriranje podataka o sebi*
  + Azuriranje svojih donacija
  + Azuriranje licnih podataka
* Prijavljivanje
* Pristup informacijama drugih korisnika
* Mogucnost doniranja
* Mogucnost volontiranja
* Objavljivanje novih donatorskih sredstava
* Odrzavanje sistema
* Odobravanja predlozenih donacija
* Odobravanje korisnika

Ove slučajeve korišćenja mogu da iniciraju gost, korisnik ili administrator stranice.

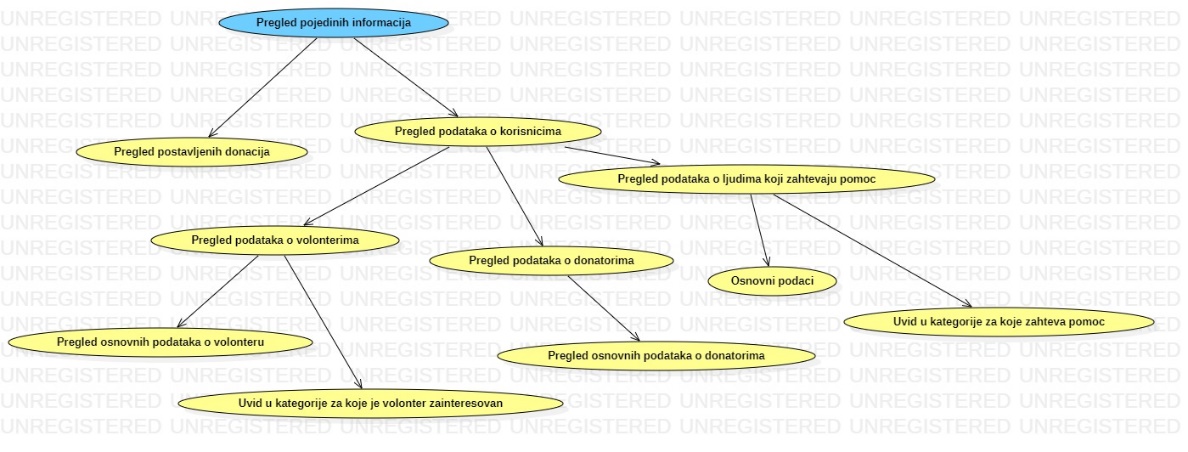
## Dijagrami slučajeva korišćenja

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

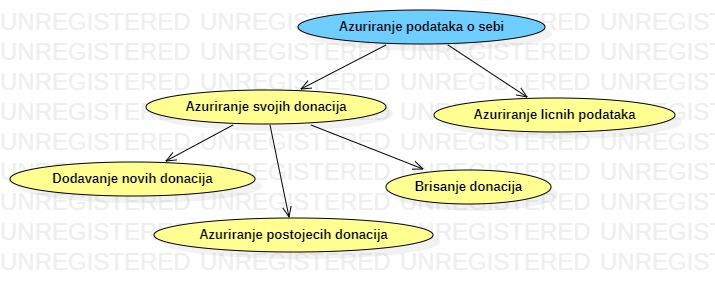


Slučajevi korišćenja *pregled pojedinih informacija* i *azuriranje podataka o sebi*  obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *pregled pojedinih informacija* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *azuriranje podataka o sebi* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Mogucnost pregledanja sadrzaja platforme

Kratak opis: Korisnik ima mogucnost pregleda najnovijih donacija.

Akteri koji iniciraju slučaj korišćenja: Gost, Korisnik, Administrator stranice

### Pregled pojedinih informacija

*6.2.2.1 Pregled postavljenih donacija*

Kratak opis: Korisnik ima mogućnost pregleda postavljenih donacija.  
Akteri koji iniciraju slučaj korišćenja: Gost, Korisnik, Administrator stranice.

*6.2.2.2 Pregled podataka o korisnicima*

*6.2.2.2.1 Pregled podataka o volonterima*

Kratak opis: Korisnik ima mogućnost pregleda podataka o volonterima.  
 Akteri koji iniciraju slučaj korišćenja: Gost, Korisnik, Administrator stranice.

*6.2.2.2.2 Pregled podataka o donatorima*

Kratak opis: Korisnik ima mogućnost pregleda podataka o donatorima.  
 Akteri koji iniciraju slučaj korišćenja: Gost, Korisnik, Administrator stranice.

*6.2.2.2.3 Pregled podataka o ljudima koji zahtevaju pomoc*

Kratak opis: Korisnik ima mogućnost pregleda podataka o ljudima koji zahtevaju pomoc.

Akteri koji iniciraju slučaj korišćenja: Gost, Korisnik, Administrator stranice.

### Registracija

Kratak opis: Registracija korisnika na sajt sa ciljem pristupa specificnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: Gost.

### Azuriranje podataka o sebi

*6.2.4.1 Azuriranje svojih donacija*

Kratak opis: Korisnik ima mogućnost izmene licnih objava.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator stranice

*6.2.4.2 Azuriranje licnih podataka*

Kratak opis Korisnik ima mogućnost ažuriranja licnih podataka.  
Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator stranice.

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika na sajt sa ciljem pristupa specifičnim funkcijama koje zahtevaju autorizaciju.  
Akteri koji iniciraju slučaj korišćenja: Korisnik,Administrator stranice.

### Pristup informacijama drugih korisnika

Kratak opis: Korisnik ima mogucnost pristupa informacijama drugih korisnika.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Administrator stranice.

### Mogucnost doniranja

Kratak opis: Korisnik ima mogucnost doniranja.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Mogusnost volontiranja

Kratak opis: Korisnik ima mogucnost volontiranja.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Objavljivanje novih donatorskih sredstava

Kratak opis:Korisnik ima mogucnost objavljivanja novih donatorskih sredstava.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Odrzavanje sistema

Kratak opis: Administrator stranice obezbeđuje jednostavan pristup i dostupnost BeHuman stranice.  
Akteri koji iniciraju slučaj korišćenja: Administrator stranice.

### Odobravanje predlozenih donacija

Kratak opis:Administrator ima mogucnost odobravanja predlozenih donacija.

Akteri koji iniciraju slučaj korišćenja: Administrator stranice.

### Odobravanje korisnika

Kratak opis: Administrator stranice vrši odobravanje novih korisničkih naloga.

Akteri koji iniciraju slučaj korišćenja: Administrator stranice.

# 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.

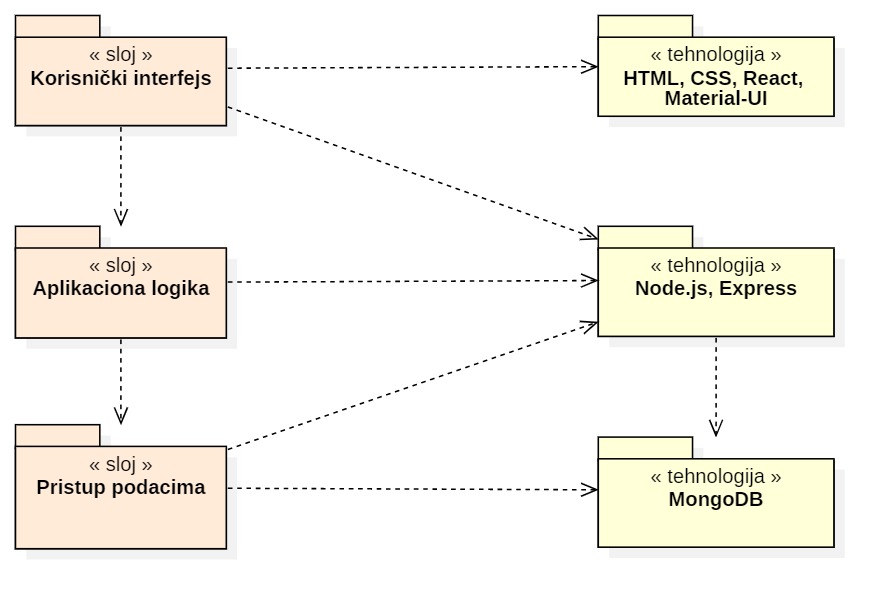
Logički pogled na BeHuman web aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice i Node.js skripte koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

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

Paket *Pristup podacima* sadrži Express komponente i Node.js skripte 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 Web aplikacije. U njemu su sadržane sve HTML stranice, React komponente, CSS stilovi i Node.js skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i Node.js.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi BeHuman Web aplikacije. Sadrži Node.js skripte i Express komponente koje realizuju funkcionalnost karakterističnu za domen primene portala 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 skripte i Express komponente 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, Express i MongoDB baze podataka.

### HTML

tehnologija

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

### CSS

tehnologija

Dok HTML definiše gradivne elemente stranica, CSS je jezik formatiranja pomoću kog se definiše izgled elemenata web stranice.

### React

tehnologija

React predstavlja predstavlja web framework koji se koristi za web aplikacija i web servera.

### JavaScript+Node.js+Express

tehnologija

JavaScript predstavlja programski jezik za programiranje klijentske strane. Node.js je višeplatformsko JavaScript radno okruženje otvorenog koda za izvršavanje JavaScript-a na serverskoj strani. Node.js omogućava da se JavaScript koristi za skripte na serverskoj strani koje omogućavaju da se sadržaj dinamičkih web stranica generiše na serveru pre nego što se pošalje do web pregledača korisnika. Express je okruženje nad Node.js koje obezbeđuje lakše pisanje koda u JavaScript-u na serverskoj strani.

### MongoDB

DBMS

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju BeHuman web aplikacije. MongoDB čuva podatke kao JSON *(JavaScript Object Notation)* dokumente sa dinamičkim šemama. Za razliku od tradicionalnih baza podataka, MongoDB koristi koncept takozvanih NoSQL baza podataka, koje ne koriste SQL za povezivanje.

# Pogled na procese

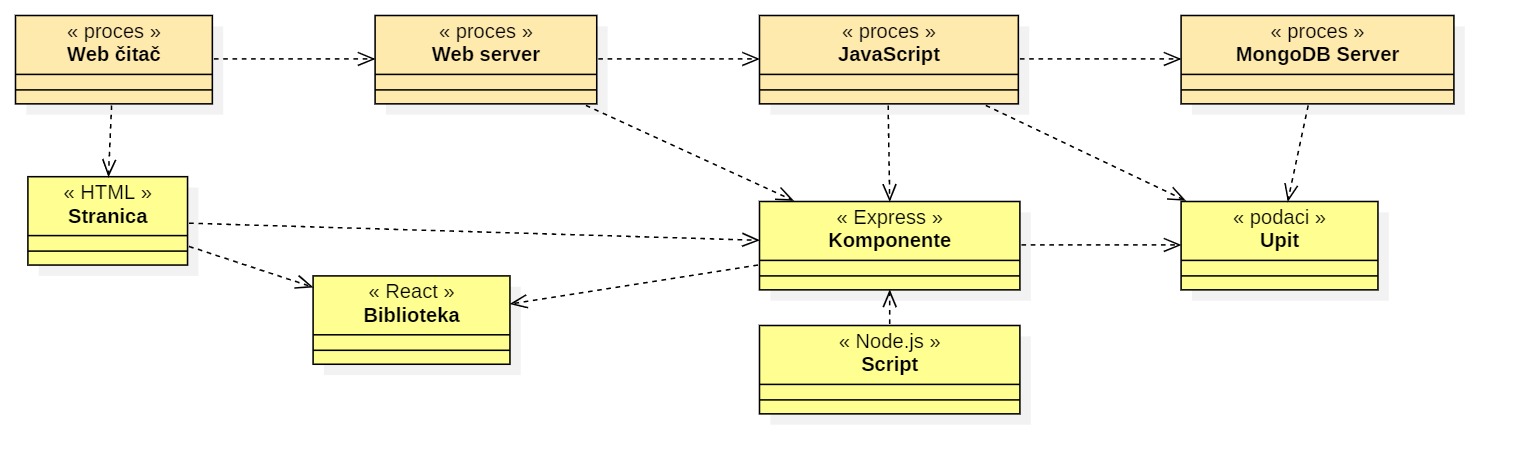
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

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

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje BeHuman portala kao Web aplikacije.

## Procesi

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



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog web servera.

Web čitač zavisi od web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana stranica JavaScript, web server inicira izvršenje Node.js procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj Node.js procesa.

### Node.js

NodeJS proces obavlja posao obrade zadatog JavaScript-a i generiše odgovarajući tekstualni sadržaj koji web server šalje web čitaču. Za izvršenje JavaScripta ovaj proces može da zahteva usluge MongoDB servera-a. Komunikacija između Node.js procesa i MongoDB servera se obavlja preko CRUD funkcija i vraćanja rezultata.

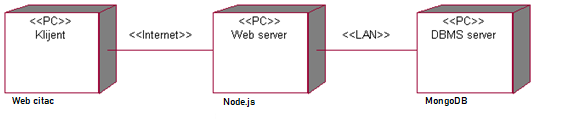
### MongoDB Server

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

# 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 BeHuman web aplikacije.



## Klijent

Pristup BeHuman 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 preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi Node.js-a koji vrše obradu zadatih JavaScript skripti. U najopštijoj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web 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).

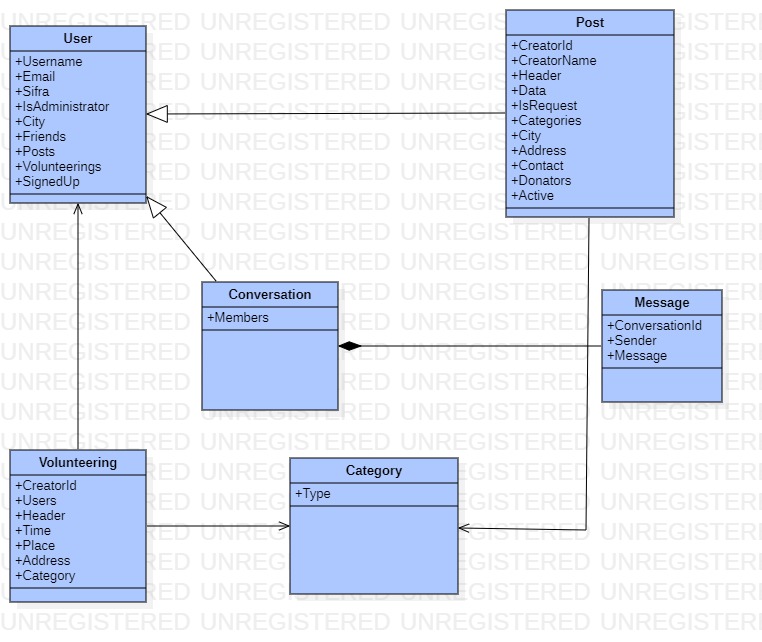
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju BeHuman aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u tri kategorije: komponente korisničkog interfejsa, komponente aplikacione logike i komponente za pristup podacima.

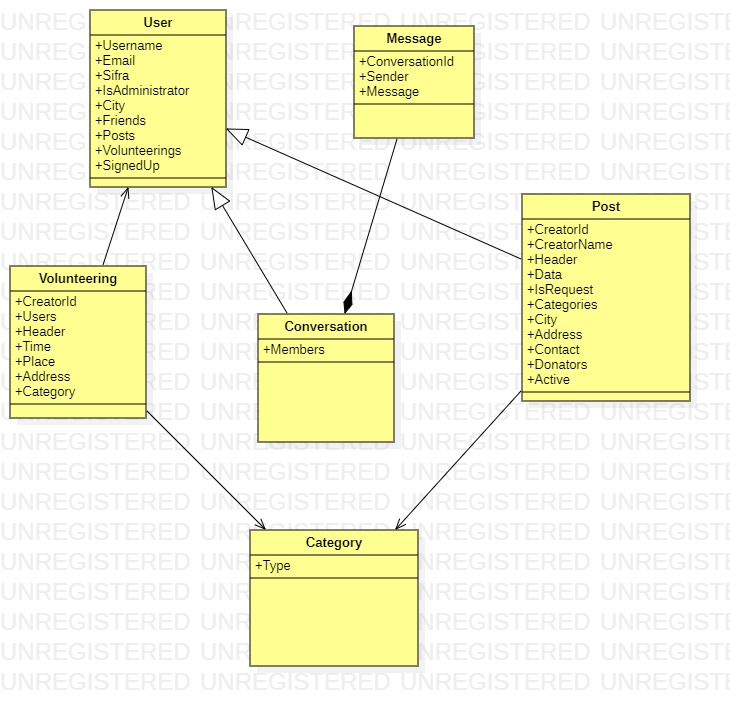
## Model domena

Model domena za koji se BeHuman 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 koje će biti implementirane.



## Šema baze podataka

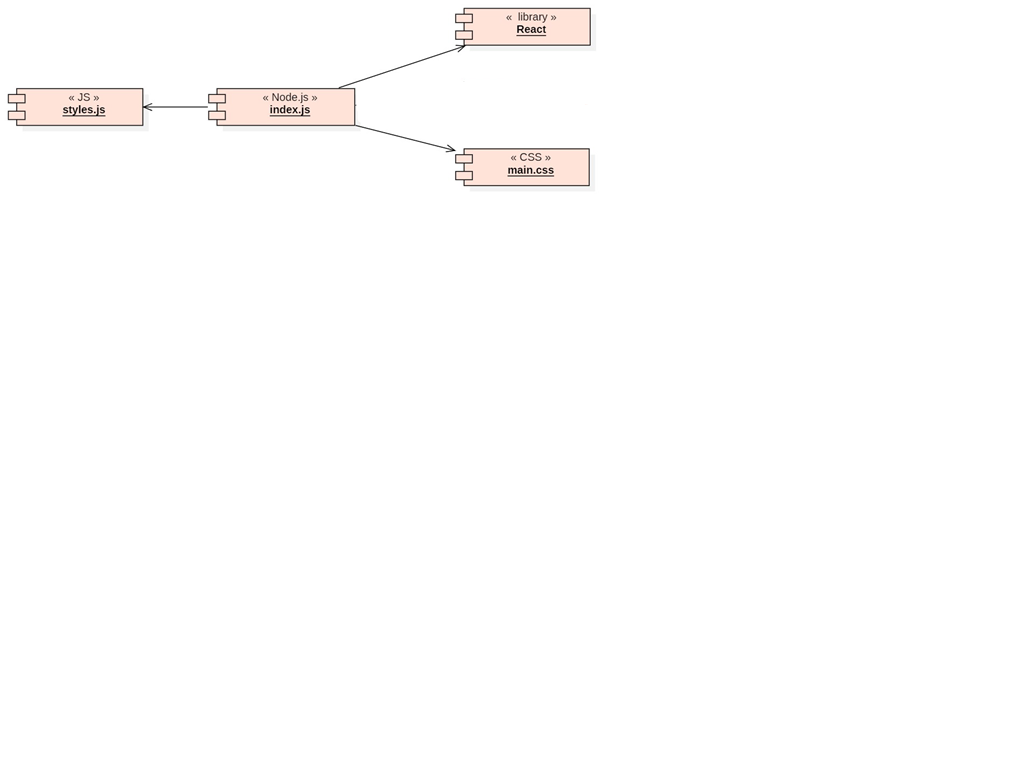
S obzirom da je korišćena NoSql baza podataka i da nema standardizovanog načina za predstavljanje ove baze, za prikaz baze podataka BeHuman aplikacije iskorišćen je UML klasni dijagram. Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka je kreirana korišćenjem MongoDB Compass alata, a dijagram je kreiran korišćenjem StarUML-a.

## Komponente sistema

Komponente sistema BeHuman sajta su Node.js skripte, CSS, HTML i Express fajlovi čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Node.Js skript prikazan kao klasa, atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta i importovanih modula.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je skup više različitih tipova komponenti. Sastoji se od index.js skripte, više styles.js dokumenata i React biblioteke.



Komponenta **index.js** predstavlja centar sistema i koristi se za generisanje dinamičkih HTML stranica čiji sadržaj može da varira u zavisnosti od parametara koji joj se proslede pri renderovanju.

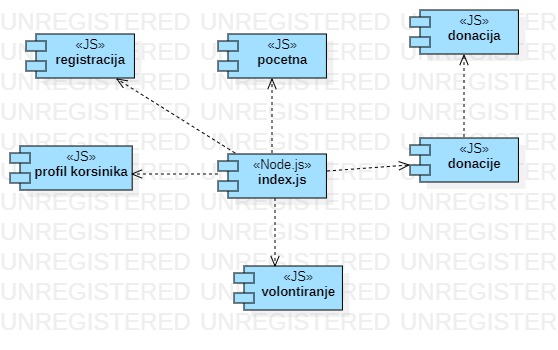
Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

Komponenta **styles.js** predstavlja generalizaciju JS fajlova koji se koriste za dodatno stilizovanje pojedinačnih dinamičkih HTML stranica.

Komponenta **React** predstavlja eksternu biblioteku koja služi za implementaciju frontend i UI.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.js** 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:



Kratak pregled funkcija svake komponente:

· pocetna – početna stranica sajta; prikaz osnovnih informacija o donacijama

· registracija – stranica za kreiranje profila korisnika

· profil korisnika – stranica za prikaz podataka o ulogovanom korisniku

· donacije – stranica sa listom donacija i osnovnim informacijama o njima, a klikom na knjigu otvara se posebna stranica „donacija“

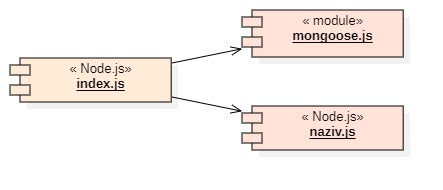
· donacija – stranica koja sadrži detaljne informacije o donaciji

· volontiranje – stranica sa vrstama volontiranja

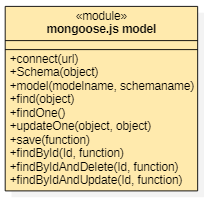
Sve dolazne zahteve procesira index.js i u zavisnosti od zahteva renderuje stranicu koja omogućava određene funkcionalnosti

### Komponente za pristup podacima

Pristup NoSql bazi podataka je omogućen pomoću funkcija koje su definisane u okviru eksternog Node.js modula **mongoose.js**. Povezivanje sa bazom se obavlja u okviru **index.js** fajla. Kreirane su i skripte **<naziv>.js** koje se koristi za čuvanje podataka u bazi, a mongoose.js i **<name>.js** skripte se uključuju na početku index.js, tako da su sve funkcije za pristup podacima i komunikaciju sa bazom dostupne svim kreiranim komponentama.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* connect - povezivanje sa bazom podataka, poziva se u db.js skriptu
* Schema – kreiranje nove šeme za pristup bazi
* model – kreiranje modela za bazu
* find– vraća objekte sa osobinama navedenim u objektu koji se prosleđuje kao parametar funkcije
* findOne– vraća jedan objekat sa osobinama navedenim u objektu koji se prosleđuje kao parametar funkcije
* updateOne – ažuriranje jednog objekta sa navedenim osobinama
* save – čuvanje novog objekta u bazi
* findById – vraća objekat na osnovu Id
* findByIdAndDelete – briše objekat iz baze na osnovu Id
* findByIdAndUpdate– ažuriranje podataka o objektu sa navedenim Id-om

# 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 1000 simultanih pristupa korisnika portalu.
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. BeHuman aplikacija će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.